Informatica® PowerExchange for Cassandra 10.1.1 HotFix 1

User Guide

Informatica PowerExchange for Cassandra User Guide
10.1.1 HotFix 1
June 2017

Publication Date: 2018-09-26

Table of Contents

Preface

The *Informatica PowerExchange® for Cassandra User Guide* describes how to use PowerExchange for Cassandra with Informatica Data Services to extract data from and load data to Cassandra. The guide is written for database administrators and developers who are responsible for developing mappings and workflows. This guide assumes that you have knowledge of Cassandra 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 Cassandra

This chapter includes the following topics:

PowerExchange for Cassandra Overview

Use PowerExchange for Cassandra to connect to a Cassandra database. You can read data from or write data to column families in a Cassandra keyspace through the Data Integration Service.

PowerExchange for Cassandra supports Cassandra Query Language (CQL) to write queries to the Cassandra database. You can create virtual tables to normalize data in Cassandra collections, such as a list, set, or map.

You can use PowerExchange for Cassandra in the following data integration scenarios:

- Your organization needs to extract data from Cassandra and aggregate data into a data warehouse or data mart for interactive data exploration and analysis.
- Your organization needs to load data from flat files, relational database, or message queues to Cassandra.
- Your organization needs to ensure consistency between your SQL data store and Cassandra data store.

Introduction to Cassandra

Cassandra is an open source, NoSQL database that is highly scalable and provides high availability. You can use Cassandra to store large amounts of data spread across data centers or when your applications require high write access speed.

In a Cassandra database, a column family is similar to a table in a relational database and consists of columns and rows. Similar to the relational database, each row is uniquely identified by a row key. The column name uniquely identifies each column in the column family. The number of columns in each row can vary, and client applications can determine the number of columns in each row.

You can read, write, and manipulate a group of data by using collections in Cassandra. The Cassandra database supports the following collection types:

- Set
- List
- Map

To effectively query data, you can use the dynamic column family feature in Cassandra. For example, the following CQL definition creates a column family that can store information about users and their friend lists.

```
CREATE TABLE users_list(
    username text PRIMARY KEY,
    friendlist list<text>);
```

Each row in the users_list column family contains a user name and the corresponding list of friends for each user. The friendlist column is a collection of list type. The rows in the users_list column family can contain different number of friendlist columns, and each row effectively represents a snapshot of a query on a user's friend list.

The following CQL insert statement inserts a row that contains a user name and the corresponding friend list:

```
insert into users_list (username, friendlist) values(user1,{' userxyz', 'userabc','
userqwe'});
```

PowerExchange for Cassandra Implementation

To extract and load Cassandra data, create a Cassandra data object in the Developer tool. You can include the data object as a source or target in a mapping. You can run the mapping or add the mapping to a workflow to process the data.

PowerExchange for Cassandra includes the Informatica Cassandra ODBC driver that connects to the Cassandra database. You can create an ODBC connection to extract data from or load data to a Cassandra database.

Note: If the table names or column names in the Cassandra database contain reserved key words, mappings fail. When you create a Cassandra ODBC connection, select quotes as the SQL identifier character and select the Support mixed-case identifiers option to avoid mapping failure.

You can also configure the multinode Cassandra server setup with the corresponding replication strategy. The Data Integration Service can fail over to secondary servers if the primary server fails.

The Developer tool imports a column based on the schema that you set for the column family. If a column contains a collection such as a set, list, or map, the Developer tool creates a virtual table and imports the collection as columns at the same level as other columns. The Developer tool creates a virtual table for each collection in the column family.

You can identify sets, lists, and maps with the naming convention of the column. The naming convention of a collection is <top level element name>.<collection>.

When you run a mapping, the Data Integration Service uses the Cassandra ODBC data source name in the machine that runs the Data Integration Service to extract data from or load data to a Cassandra database.

Virtual Tables

The Informatica Cassandra ODBC driver creates virtual tables if the column family contains collections such as set, list, or map.

Virtual tables depict the normalized view of a Cassandra collection. You can import virtual tables as an ODBC data object and create mappings. The Informatica Cassandra ODBC driver creates a virtual table for each collection in the column family. The virtual table for a collection uses the following naming convention by default: `<original column family name>_vt_<original collection name>`

Each virtual table has a key column that references back to the primary key column in the original column family. The virtual table has an index column that shows the position of the data within the original array. The index column uses the following naming convention by default: `<original column name>_index`. Other columns in the virtual table represent the elements in the array and are named after the array element. If the array is of scalar type, the data column uses the following naming convention by default: `<original column name>_value`

Example

The following CQL statement defines a Cassandra column family that contains a set:

```
CREATE TABLE users_list(
    username text PRIMARY KEY,
    first_name text,
    last_name text,
    city text,
    last_login timestamp,
    emails set<text>
    );
```

When you use the Informatica Cassandra ODBC driver to import the column family, the driver creates the users_list_vt_emails virtual table for the users_list table.

The following table shows the denormalized view of the users_list_vt_emails virtual table:

username	emails_value
user1	xyz@gmail.com
user1	abc@hotmail.com
user1	qwe@yahoo.com
user2	zxc@gmail.com
user2	jkl@hotmail.com

CHAPTER 2

PowerExchange for Cassandra Configuration

This chapter includes the following topics:

PowerExchange for Cassandra Configuration Overview

You can use PowerExchange for Cassandra on Windows or Linux. You must configure PowerExchange for Cassandra before you can extract data from or load data to a Cassandra database.

The Informatica Cassandra ODBC driver is installed on the machines on which you installed Informatica services and clients. PowerExchange for Cassandra supports the version of the CQL that the Cassandra database supports. Configure the connection and advanced properties for the Informatica Cassandra ODBC driver on those machines.

Prerequisites

PowerExchange for Cassandra installs with the Informatica services and clients.

Before you can use PowerExchange for Cassandra, perform the following tasks:

- Install or upgrade the Informatica services.
- Create a Data Integration Service and a Model Repository Service in the Informatica domain.

- Ensure that you have the PowerExchange for Cassandra license file. You do not require a separate ODBC license to use PowerExchange for Cassandra.

For more information about product requirements and supported platforms, see the Product Availability Matrix on Informatica Network:
https://network.informatica.com/community/informatica-network/product-availability-matrices/overview

Query Mode

You can use PowerExchange for Cassandra to process the queries as SQL statement or as CQL statement. The Cassandra ODBC driver uses SQL with CQL fallback as the default query mode.

You can select one of the following query modes:

SQL

The ODBC driver treats all incoming queries as SQL and does not accept any CQL query syntax that is not standard SQL-92 syntax.

CQL

The ODBC driver treats all incoming queries as CQL and does not accept any non-CQL syntax.

SQL with CQL Fallback

The driver tries to treat the incoming query as SQL. If an error occurs while handling the query as SQL, the driver then passes the original query to Cassandra to execute as CQL. SQL with CQL fallback is the default query mode used by the ODBC driver.

Tunable

You can use PowerExchange for Cassandra to specify the consistency level to use when you read data from or write data to a Cassandra database.

You can select one of the following tunable consistency levels:

ANY

Writes the data to at least one node when you write to a Cassandra database.

ONE

Writes the data to at least one replica node and reads the data from one of the closest replicas.

TWO

Writes the data to at least two replica nodes and reads the most recent data from two of the closest replicas.

Three

Writes the data to at least three replica nodes and reads the most recent data from three of the closest replicas.

QUORUM

Writes the data to the minimum required replica nodes and reads the record after the response from the minimum number of replica nodes from any data center.

LOCAL QUORUM

Writes the data to the minimum required replica nodes in the same data center and reads the record after the response from the current data center.

EACH QUORUM

Writes the data to the minimum required replica nodes in all the data centers.

ALL

Writes the data to all the replica nodes in the cluster and reads the record after the response from all the replicas.

LOCAL ONE

At least one of the replica node from the local data center must successfully acknowledge the data when you write data to a Cassandra database and must read the data from one of the closest replicas in the local data center.

Load Balancing Policy

You can use PowerExchange for Cassandra to specify the host to which Cassandra driver must connect or ignore while establishing the connection.

You can select one of the following load balancing policies:

DC Aware

Indicates that nodes of the local data center that the driver must query before any other node from other data centers in a round-robin fashion.

Round Robin

Indicates that each time a driver executes a query, the driver returns hosts, and shifts each query in a round-robin fashion.

Configuring the Cassandra ODBC Advanced Properties for Data Source Name

You must configure the advanced properties when you create a data source name in the Cassandra ODBC driver.

When you create or configure a connection from a Windows machine, you can select the options available in the **Advanced options** of the ODBC Data Source Administrator. You can also create or configure a connection from a Linux machine by editing the `odbc.ini` file in the following location: `<INFA_HOME>/tools/cassandra/lib`.

Cassandra ODBC Data Source Name Advanced Properties

The following table describes the advanced properties for the Cassandra ODBC data source name:

Property	Description
Query Mode	The query modes that the ODBC driver uses to process queries. You can select one of the following query modes: - SQL - CQL - SQL with CQL Fallback
Tunable	The consistency levels that you can use when you read data from or write data to a Cassandra database. You can select one of the following tunable configurations: - ANY - ONE - TWO - THREE - QUORUM - LOCAL QUORUM - EACH QUORUM - ALL - LOCAL ONE
Load Balancing policy	The policies that the ODBC driver uses to determine which host communicates with the ODBC driver. You can select one of the following load balancing policies: - DC Aware - Round Robin
Binary Column Length	The default length to report when exposing BLOB columns. Default is 4000.
String Column Length	The default length to report when exposing ASCII, TEXT, and VARCHAR columns. Default is 4000.
Virtual Table Name Separator	The separator name for naming a virtual table built from a collection. Default is _vt_.
Blacklist hosts	The host names or IP addresses that you want to ignore while establishing the connection. **Note:** You can enter the host names in quotation marks, separated by comma.
Whitelist hosts	The host names or IP addresses that you want to connect. **Note:** You can enter the host names in quotation marks, separated by comma.
Blacklist datacenter hosts	The datacenter names that you want to ignore while establishing the connection. **Note:** You can enter the host names in quotation marks, separated by comma.
Whitelist datacenter hosts	The datacenter names that you want to connect. **Note:** You can enter the host names in quotation marks, separated by comma.
Enable Token Aware	Specifies how the ODBC driver reduces load on Cassandra nodes. When you enable token aware option, the driver uses the primary key of queries to route requests directly to the Cassandra nodes.
Enable Latency Aware	Specifies how the ODBC driver restricts sending new queries to slow performing Cassandra nodes.

Property	Description
Enable Null Values Insertion	The null values in an INSERT statement.
Enable Case Sensitive	Specifies whether the ODBC driver differentiates between lowercase and uppercase letters in schemas, tables, and columns. **Note:** You can specify the case by using double quotation marks.
Use SQL_WVARCHAR for string data type	Specifies whether to use SQL_WVARCHAR for data types that support Unicode character sets.
Enable Paging	The number of rows that the ODBC driver reads or writes per page from the Cassandra database for large volumes of data.
SSL	Not Applicable.

Configuring the Informatica Cassandra ODBC Driver on Linux

You must configure the Informatica Cassandra ODBC driver with details of the Cassandra database and ODBC driver manager before you can run Cassandra mappings.

Edit the `odbc.ini` file to configure the driver in the following location: `<INFA_HOME>/tools/cassandra/lib`

1. Enter the correct ODBCInstLib for the ODBC Driver Manager in all the .ini files.
2. Replace <INFA_HOME> with the path to the Informatica services installation directory in all the .ini files.
3. Add the following information to the LD_LIBRARY_PATH environment variable:
   - `<INFA_HOME>/tools/cassandra/lib`
   - 64-bit library directory of the ODBC Driver Manager
4. Add the path of the odbc.ini file to the ODBCINI environment variable.
5. Add AuthMech=1 in the `odbc.ini` file, if you want to connect to a Cassandra database using **User Name** and **Password**.

   When the authentication mechanism is enable, you must specify the username and password in `odbc.ini` file using the following keywords:

   ```
   UID=<USER_NAME>
   ```

   ```
   PWD=<PASSWORD>
   ```

6. Add entries for all the Cassandra data sources in the `odbc.ini` file.

   The following section shows a sample entry in the `odbc.ini` file:

   ```
   [Sample Informatica Cassandra DSN]
   Description=Simba Cassandra ODBC Driver (64-bit) DSN
   Driver=<INFA_HOME>/lib/libinformaticacassandraodbc64.so
   Required: You can also specify these values in the connection string.
   Host=[Host]
   Port=9042
   DefaultKeyspace=[keyspace]
   uid=[user id]
   ```

```
# Advanced Options: You can also specify these values in the connection string.
QueryMode=2
TunableConsistency=1
COLoadBalancingPolicy=0
BinaryColumnLength=4000
StringColumnLength=4000

#Virtual Table Name Separator
VTTableNameSeparator=_vt_

BlacklistFilteringHosts=[Blacklist Filtering Hosts]
WhitelistFilteringHosts=[Whitelist Filtering Hosts]
BlacklistDatacenterFilteringHosts=[Blacklist Datacenter Filtering Hosts]
WhitelistDatacenterFilteringHosts=[Whitelist Datacenter Filtering Hosts]

EnableTokenAware=1
EnableLatencyAware=[Enable Latency Aware]
EnableNullInsert=1
EnableCaseSensitive=[Enable Case Sensitive]
UseSqlWVarchar=[Use Sql WVarchar]

#Paging Settings
EnablePaging=1
RowsPerPage=10000

# Check the hostname in the server certificate with the server hostname
UseSslIdentityCheck=0

#The full path of the PEM file containing the certificate for verifying the server.
SSLTrustedCertsPath=${driver_dir}/${libdir64}/cacerts.pem

# The full path of the PEM file containing the certificate for verifying the client.
SSLUserCertsPath=[SSL User Certificate Path]

# The full path of the file containing the private key used to verify the client.
SSLUserKeyPath=[SSL User Key Path]

The password of the private key file.
SSLUserKeyPWD=[SSL User Key Password]

#Note: You must specify PWD in the connection string if authentication is on.
#UID can be saved as a part of the DSN or specified in the connection string.
```

Configuring Data Source Name on Windows

Configure the connection properties, advanced properties, and keyspace when you create a data source name.

1. Run the 64-bit ODBC Data Source Administrator.
2. Click the **Drivers** tab and verify that Informatica Cassandra ODBC Driver appears in the alphabetical list of driver names.
3. Click the **User DSN** tab to create a data source name that the current Windows user can use.
4. Click **Add**.

   The **Create New Data Source** dialog box appears.
5. Select Informatica Cassandra ODBC Driver and click **Finish**.

   The **Informatica Cassandra ODBC Driver Setup** dialog box appears.

6. Specify the following Cassandra ODBC connection properties:
    a. Enter a name for the data source.
    b. Enter a description for the data source.
    c. Enter the host name or IP address of the server on which the Cassandra instance runs.
    d. Optionally, enter the host names of the secondary Cassandra servers, separated by the comma delimiter.
    e. Enter the port number from which you can access Cassandra.
    f. Enter the name of the Cassandra keyspace to use by default.
    g. Enter the user name and password for the Cassandra database if you select the authentication mechanism as **User Name and Password**.
7. Click **Advanced Options** to configure the Cassandra ODBC advanced properties.
8. To enable logging, click **Logging Options** and then perform the following steps:
    a. Select the **Log Level** value that indicates the level of information to write to the log file.
    b. In **Log Path**, enter the location to save the log file.
    c. In **Log Namespace**, type the name of the component for which to log messages.
    d. Click **OK**.
9. Click **Test** to verify the connection to Cassandra.
10. Click **OK** to close the **Informatica Cassandra ODBC Driver Setup** dialog box.
11. Click **OK** to close the **ODBC Data Source Administrator** dialog box.

Using the Sample Informatica Cassandra DSN

You can configure and use the sample Informatica Cassandra DSN instead of creating a new data source name.

1. Run the 64-bit ODBC Data Source Administrator.
2. Click the **System DSN** tab.
3. Select the **Sample Informatica Cassandra DSN** and then click **Configure**.

    The **Informatica Cassandra ODBC Driver Setup** dialog box appears.
4. Configure the Cassandra ODBC connection properties.
5. Configure the **Advanced Options** and **Logging Options**.
6. Click **Test** to verify the connection to Cassandra.
7. Click **OK** to close the **Informatica Cassandra ODBC Driver DSN Setup** dialog box.
8. Click **OK** to close the **ODBC Data Source Administrator** dialog box.

CHAPTER 3

Cassandra Read Operations

This chapter includes the following topics:

Cassandra Read Operations Overview

You can import a Cassandra column family as an ODBC data object in the Developer tool and use it as a source in a mapping.

When you run a Cassandra mapping, the Data Integration Service uses the Informatica Cassandra ODBC data source to extract data from Cassandra.

You can configure the optimization levels at the following locations:

- Mapping run-time configuration in the Developer tool
- Data viewer run-time configuration in the Developer tool
- Mapping deployment in application settings in the Developer tool
- Mapping configuration of an application in the Data Integration Service through the Administrator tool

Example: Analyzing Data Stored in a Cassandra Database

To store real-time data on weather, you need to store data on a database that supports multiple and fast write operations.

Your organization uses a Cassandra database to store real-time data from a weather station. You need to fetch the weather data for a specific period of time and represent the data in a graphical format for analysis.

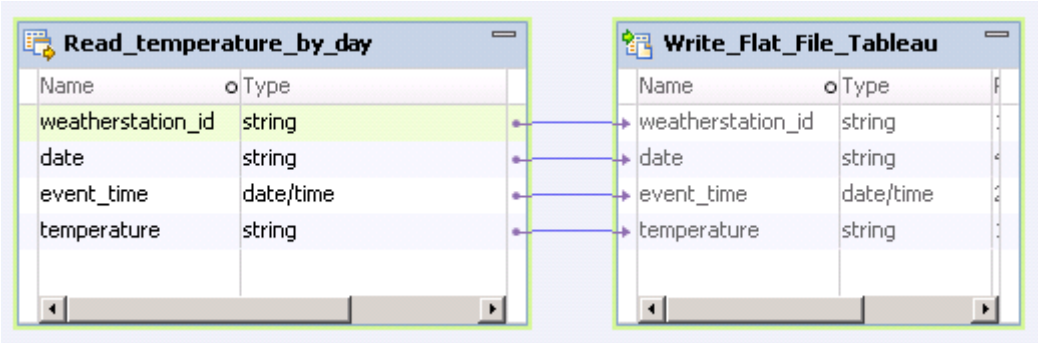
The weather station stores the temperature at a particular time period in a Cassandra column family. The following is the definition of the temperature_by_day column family:

```
CREATE TABLE temperature_by_day (
                weatherstation_id text,
                date text,
                event_time timestamp,
```

```
temperature text,
PRIMARY KEY ((weatherstation_id,date),event_time));
```

Create a mapping to read data from Cassandra and write it to a flat file. You can use Tableau to read the flat file and create a time-series graph for temperature.

The following image shows the mapping:



Mapping Input

The mapping source is a column family in the Cassandra database. The source contains information for your organization, including weatherstation ID, time, and temperature. Source columns include columns such as weatherstation ID, date, event time, and temperature.

The following table describes the contents of the temperature_by_day column family:

Field	Data type
weatherstation_id	String
date	String
event_time	Date/Time
temperature	String

Mapping Output

The mapping output is a flat file data object. The flat file data object contains the same columns as the source. When you run the mapping, the Data Integration Service reads weather information from the Cassandra database and writes the data to a flat file. Analysts can use the flat file in Tableau to create a time-series graph for temperature.

CHAPTER 4

Cassandra Write Operations

This chapter includes the following topics:

Cassandra Write Operations Overview

You can import a Cassandra column family as an ODBC data object and create mappings in the Developer tool to write data to Cassandra.

You must configure the ODBC driver before you import Cassandra column families.

When you run a Cassandra mapping, the Data Integration Service uses the Informatica Cassandra ODBC data source to load data to the Cassandra database.

When you run Cassandra mappings, ensure that the optimization level is set to none. If you do not set the optimization level as none, the mappings might fail.

You can configure the optimization levels at the following locations:

- Mapping run-time configuration in the Developer tool
- Data viewer run-time configuration in the Developer tool
- Mapping deployment in application settings in the Developer tool
- Mapping configuration of an application in the Data Integration Service through the Administrator tool

Example: Read Data from Flat Files Generated from Sensors and Write to Cassandra

A weather channel uses flat files with comma-separated values to store real-time weather details generated from sensors. To provide better performance and scalability, the weather channel needs to migrate data from the flat file to a Cassandra database.

The weather channel stores the real-time temperature details in a flat file.

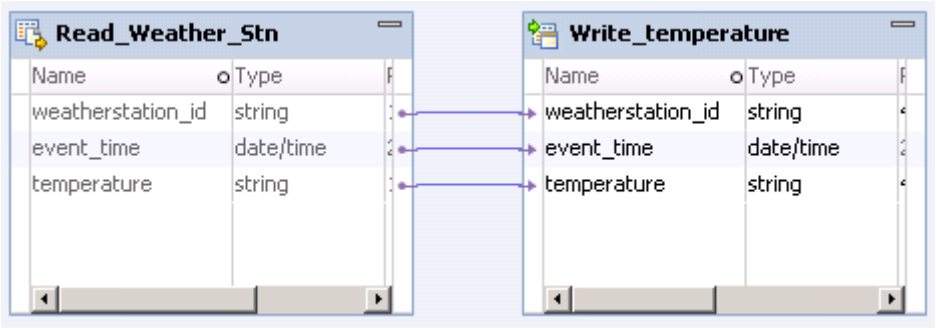
The following table describes the content of a flat file that stores temperature details:

WeatherStation_ID	String
Event_time	Timestamp
Temperature	String

Create a mapping to extract data from the flat file and load it to a Cassandra column family. Create a flat file data object in read mode and create a Cassandra data object in write mode. The following is the definition of the temperature column family:

```
CREATE TABLE temperature (
        weatherstation_id text,
        event_time timestamp,
        temperature text,
        PRIMARY KEY (weatherstation_id,event_time)
);
```

The following image shows the mapping:



Mapping Input

The mapping source is a comma-separated flat file. The source contains information on weather station, time, and temperature. Source columns include columns such as weatherstation_id, date, event_time, and temperature.

Mapping Output

The mapping output is a Cassandra data object to write data to the temperature column family in the Cassandra database. When you run the mapping, the Data Integration Service reads weather information from the flat file and writes the data to the Cassandra database.

CHAPTER 5

Virtual Table Operations

This chapter includes the following topics:

Virtual Table Operations Overview

The Informatica Cassandra ODBC driver creates a separate virtual table for each collection type in the column family. When you use a virtual table in a mapping, you can perform read, write, append, update, or delete operations.

Supported Virtual Table Operations

You can use virtual table operations to access, write, or delete rows from a virtual table.

You can perform the following operations in a virtual table:

Read

Use the read operation to retrieve rows from a virtual table.

Insert

Use the insert operation to add one or more rows to a virtual table. Each row that you insert must correspond to a primary key in the source table. If you insert data into a row key that contains data, Cassandra overwrites the original data in the row with the new data.

Note: Cassandra database treats all insert operations as upsert operations.

Update

Use the update operation to update rows in a virtual table. Use the DD_UPDATE strategy in an update strategy expression to update rows in a virtual table.

Note: If the collection type is list, sort the corresponding source or target virtual table data in descending order before you pass data to the Update Strategy transformation.

Append

Use the append operation to add an element to the end of the list type.

Delete

Use the delete operation to remove one or more rows from a virtual table. You can use the DD_DELETE strategy in an update strategy expression or use a delete command, such as a Pre SQL or Post SQL command, to remove rows.

Note: If the collection type is list, sort the corresponding source or target virtual table data in descending order before you pass data to the Update Strategy transformation.

Virtual Table Operations Examples

The example mappings illustrate the operations that you can perform on a virtual table.

The login details of users are stored in a Cassandra column family. You need to read, insert, update, append, or delete values from the location column.

The following users_list table represents a Cassandra column family that contains a collection of type list:

username	first_name	last_name	city	last_login	location
user1	john	doe	SFO	2014-8-03 01:30:00	0.0.0.0, 123.123.123.2
user2	jane	smith	NYC	2014-8-03 01:40:00	123.88.99.2, 0.0.0.0
user3	brian	lee	SFO	2014-8-04 11:15:00	123.123.2.2, 127.0.0.0
user4	james	adam	NYC	2014-8-05 05:22:00	123.123.1.2, 127.0.0.0

The Informatica Cassandra ODBC driver creates the users_list main table and users_list_vt_location virtual table.

The following image displays the users_list main table:

	Name	Native Type	Precisi...	Scale	Primar...	Nullable	Description
1	username	varchar	4000	0	☑	☐	
2	city	varchar	4000	0	☐	☑	
3	first_name	varchar	4000	0	☐	☑	
4	last_login	timestamp	29	9	☐	☑	
5	last_name	varchar	4000	0	☐	☑	

The following image displays the users_list_vt_location virtual table:

Columns

	Name	Native Type	Precisi...	Scale	Primar...	Nullable	Description
1	username	varchar	4000	0	☑	☐	
2	location_index	integer	10	0	☑	☐	
3	location_value	varchar	4000	0	☐	☐	

The following example mappings and data preview illustrate the virtual table operations:

Read Operation

The data preview of the users_list_vt_location virtual table displays the following output:

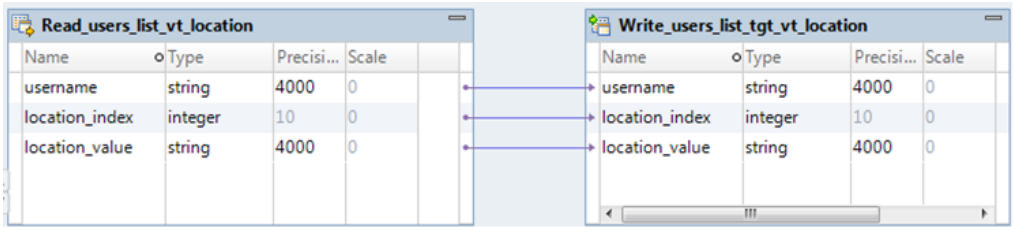
Output

Name: users_list_vt_location

	username	location_index	location_value
1	user2	0	123.88.99.2
2	user2	1	0.0.0.0
3	user4	0	123.123.1.2
4	user4	1	127.0.0.0
5	user1	0	123.123.123.2
6	user1	1	0.0.0.0
7	user3	0	123.123.2.2
8	user3	1	127.0.0.0

Insert Operation

The following pass-through mapping inserts rows into a virtual table:



The mapping source is the users_list_vt_location table. The mapping target is the users_list_tgt_vt_location table.

The following image displays the data preview of the target virtual table:

Output

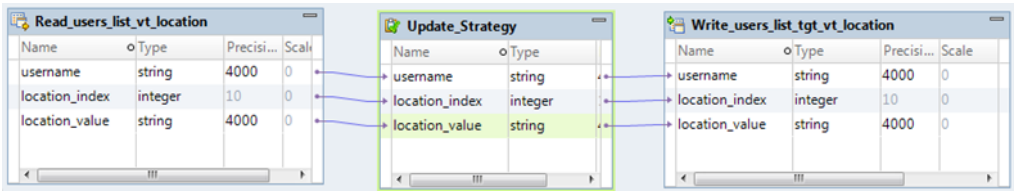
Name: users_list_tgt_vt_location

	username	location_index	location_value
1	user2	0	123.88.99.2
2	user2	1	0.0.0.0
3	user4	0	123.123.1.2
4	user4	1	127.0.0.0
5	user1	0	123.123.123.2
6	user1	1	0.0.0.0
7	user3	0	123.123.2.2
8	user3	1	127.0.0.0

Update Operation

You can use a DD_UPDATE strategy in an Update Strategy transformation to update rows in a virtual table.

The following mapping uses an Update Strategy transformation to update rows in a virtual table:

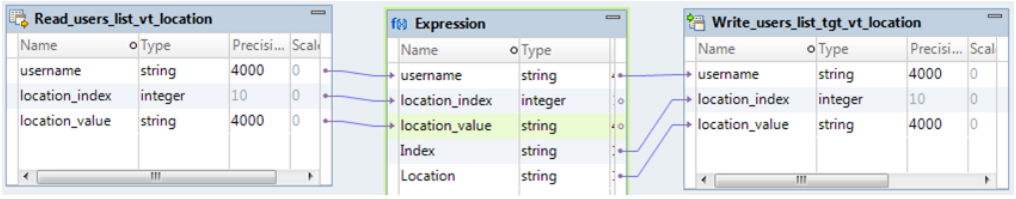


The mapping source is the users_list_vt_location table. The DD_UPDATE strategy in the update strategy expression updates rows in the target virtual table. The mapping target is the users_list_tgt_vt_location table.

Append Operation

You can append rows only to virtual tables created for list type.

The following mapping appends the location 99.99.99.0 to all the lists in the users_list_tgt_vt_location table:



The mapping source is the users_list_vt_location table. The Expression transformation sets the list index to -1. The mapping target is the users_list_tgt_vt_location table.

The following image displays a data preview of the rows in the target table before you append rows to the target table:

Output

Name: users_list_tgt_vt_location

	username	location_index	location_value
1	user2	0	0.0.0.0
2	user4	0	0.0.0.0
3	user1	0	0.0.0.0
4	user3	0	0.0.0.0

The following image displays a data preview of the rows appended to the target virtual table:

Output

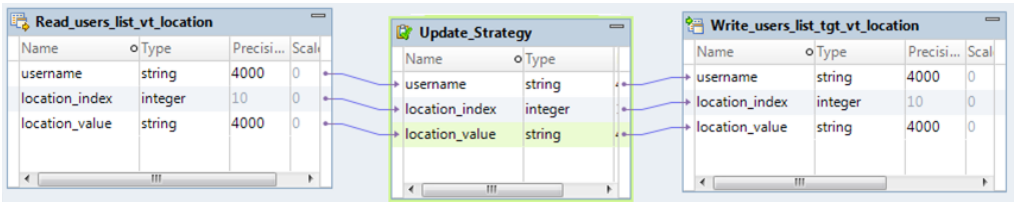
Name: users_list_tgt_vt_location

	username	location_index	location_value
1	user2	0	0.0.0.0
2	user2	1	99.99.99.0
3	user4	0	0.0.0.0
4	user4	1	99.99.99.0
5	user1	0	0.0.0.0
6	user1	1	99.99.99.0
7	user3	0	0.0.0.0
8	user3	1	99.99.99.0

Delete Operation

You can use a DD_DELETE strategy in an Update Strategy transformation to delete rows from a virtual table.

The following mapping uses an Update Strategy transformation to delete rows from a virtual table:



The mapping source is the users_list_vt_location table. The DD_DELETE strategy in the update strategy expression deletes rows from the target virtual table. The mapping target is the users_list_tgt_vt_location table.

APPENDIX A

Data Type Reference

This appendix includes the following topic:

Cassandra, ODBC, and Transformation Data Types

When you import a Cassandra column family as a data object, the transformation data types corresponding to the ODBC data types appear in the Developer tool.

The Informatica Cassandra ODBC driver reads Cassandra data and converts the Cassandra data types to ODBC data types. The Data Integration Service converts the ODBC data types to transformation data types.

The following table lists the Cassandra data types and the corresponding ODBC and transformation data types:

Cassandra Data Types	ODBC Data Types	Transformation Data Types
ASCII	Varchar	Varchar
Bigint	Bigint	Bigint
Blob	Varbinary	Varbinary
Boolean	Bit	Bit
Date	Date	Datetime
Decimal	Decimal	Decimal
Double	Double	Double
Float	Real	Real
Inet	Varchar	Varchar. Cassandra Inet data type supports IP addresses in IPv4 and IPv6 formats. Informatica Cassandra ODBC driver does not support reading or writing IP addresses in IPv6 format.
Int	Integer	Integer

Cassandra Data Types	ODBC Data Types	Transformation Data Types
Smallint	Smallint	Integer
Text	Wvarchar	Varchar
Timestamp	Varchar	Time stamp. The Informatica Cassandra ODBC driver treats the Time stamp data type values in Coordinated Universal Time (UTC) format. Range of Timestamp value is 1601-01-01 00:00:00.000 to 9999-12-31 23:59:59.999
Timeuuid	Varchar	Varchar
Tinyint	Tinyint	Integer
Uuid	Varchar	Varchar. When you use comparison operators in data of Uuid data type, enclose values of Uuid data type within single quotation marks. For example, in the following query, the value for the Uuid data type is enclosed within single quotation marks: `DELETE from pc.map_nesting_tgt_vt_col_map2 where pc.map_nesting_tgt_vt_col_map2.col_uuid='152d2030-14e6-11e4-aef1-fd88d8253a43';`
Varchar	Wvarchar	Varchar
Varint	Numeric	Numeric

Note: You can process custom data types as String with PowerExchange for Cassandra.

INDEX