How to Write Data to HDFS

Abstract

You can offload large volumes of transactional data from a data warehouse source to Hadoop for high performance processing and data transformation. Business intelligence applications can use this processed data to query the transactional data to make decisions. Use PowerExchange for HDFS to configure mappings that can extract data and write to a Hadoop Distributed Filesystem (HDFS) through an HDFS connection.

Supported Versions

- PowerCenter Big Data Edition 9.6.0
- PowerExchange for HDFS 9.6. 0
- PowerCenter Big Data Edition Trial 9.6. 0

Table of Contents

Overview

Business users want to query transactional data to get information to analyze customer preferences and business trends. Business intelligence applications provide reports, dashboards, scorecards, and alerts by querying the data in the data warehouses.

Create mappings that extract data from a data warehouse source and write to HDFS through an HDFS connection. By offloading large volumes of transactional data to HDFS, you can query the entire data set. Configure the mapping to distribute the processing across Hadoop cluster nodes for higher performance.

Business users can query large volumes of transformed and conformed data to get accurate information for business decisions.

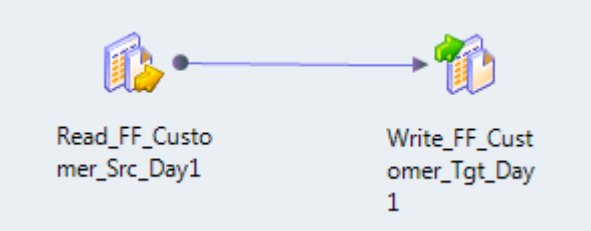
Scenario

A financial institution gathers large volumes of online transactional data through their online banking service. They want to query the entire data set to get information for business decisions.

Instead of relying on small data samples, and aggregations for analytics, the financial institution offloads their data to HDFS for processing and analytics. The data warehouse resources can be refocused to repeatable operational analytics.

A data warehouse can store data in a relational source or a flat file source. Create a mapping that extracts the data from the source and writes to HDFS using an HDFS connection.

The following image shows a mapping that reads transactional data for a day and writes to an HDFS target:



After you load data to HDFS, create mappings to process the data that can run in a Hive environment.

Write Data to HDFS

To write to an HDFS target, you must create an HDFS connection object and configure an HDFS flat file data object.

Complete the following steps to use PowerExchange for HDFS to write to HDFS:

1. Create an HDFS connection.
2. Create an HDFS flat file data object.

   Note: Specify data object properties such as file location on the Hadoop cluster and an HDFS connection name.
3. Use an HDFS flat file object as a target in a mapping.
4. Configure the validation environment and run-time environment for the mapping.
5. Save and run the mapping.

HDFS Connection Properties

Configure HDFS connections properties to create an HDFS connection.

The following table describes the properties for an HDFS connection:

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	The 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 765 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Default is Hadoop File System.

Property	Description
User Name	The user name to access HDFS.
NameNode URI	The URI to access HDFS. Use the following format to specify the NameNode URI in Cloudera and Hortonworks distributions: `hdfs://<namenode>:<port>` Where - <namenode> is the host name or IP address of the NameNode. - <port> is the port that the NameNode listens for remote procedure calls (RPC). Use one of the following formats to specify the NameNode URI in MapR distribution: - `maprfs:///` - `maprfs:///mapr/my.cluster.com/` Where `my.cluster.com` is the cluster name that you specify in the `mapr-clusters.conf` file.

Creating an HDFS Connection

Create an HDFS connection to access an HDFS target.

Create an HDFS connection before you import physical data objects.

1. Click **Window** > **Preferences**.
2. Select **Informatica** > **Connections**.
3. Expand the domain.
4. Select the connection type **File Systems** > **Hadoop File System**, and click **Add**.
5. Enter a connection name.
6. Optionally, enter a connection description.
7. Click **Next**.
8. Configure the connection properties.
9. Click **Test Connection** to verify the connection to HDFS.
10. Click **Finish**.

HDFS Target

You can include an HDFS flat file data object as a target.

Complete the following steps to configure an HDFS target in a mapping:

1. Select the HDFS flat file object to edit the general, column, and format properties.
2. Select the Input to edit run-time properties.
3. Add the HDFS flat file object to the mapping with write access.

HDFS Flat File Data Object Write Properties

The Data Integration Service uses write properties when it writes data to an HDFS flat file.

The following table describes the HDFS connection properties and compression properties that you configure for HDFS flat file targets:

Property	Description
Connection Type	The type of connection. Select the following option: - Hadoop File System. The target file is in HDFS. Default is None.
Connection Name	The name of the connection. Select an HDFS connection or assign a mapping parameter that defines the connection details.
Compression Format	Optional. Specifies the compression format. Select from the following options: - None - Gzip - Bzip2 - Lzo - Custom
Compression Codec	Required for custom compression. Specify the fully qualified class name implementing the Hadoop `CompressionCodec` interface.

For more information about how to access a flat file physical data object, view the following video at:
http://www.youtube.com/watch?v=jZYpCBCMpxQ

Author

Srilakshmi Sitaraman
Senior Technical Writer

Acknowledgements

The author would like to acknowledge Informatica Hadoop team for their technical assistance.