Informatica® PowerExchange for HBase 10.1.1

User Guide

Informatica PowerExchange for HBase User Guide
10.1.1
December 2016

Publication Date: 2018-09-27

Table of Contents

Preface

The *Informatica PowerExchange® for HBase User Guide* provides information about extracting data from and loading data to HBase. The guide is written for database administrators and developers who are responsible for developing mappings that read data from HBase tables and write data to HBase tables.

This guide assumes that you have knowledge of HBase and Informatica Data Services.

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.

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.
- 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 HBase

This chapter includes the following topic:

PowerExchange for HBase Overview

PowerExchange for HBase provides connectivity to an HBase data store.

Use PowerExchange for HBase to read data from the HBase columns families or write data to the columns families in an HBase table. You can read or write data to a column family or a single binary column.

You can connect to an HBase data store and view all the HBase tables. Select an HBase table and view all the column families. If you know the schema of the data source, you can specify the columns in the column families when you create the data object. If you do not know the schema of the data source, you can search the rows in an HBase table to identify the columns and their occurrence probability.

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

Example

You work for a mobile service provider that needs to load the data in WAP log files to HBase tables and generate multiple reports. WAP log files can contain a number of columns with information about the mobile users, internet usage, and data volume. On any day, the WAP log files could be three to four billion rows of data and can be around two terabytes in size.

You can use PowerExchange for HBase to consolidate data received during the day, filter and transform the data, and load it to HBase tables. Based on the data in the tables, analysts can run queries and perform real-time analysis of daily operations, statistics related of gateway usage, and data volume.

CHAPTER 2

PowerExchange for HBase Configuration

This chapter includes the following topics:

Prerequisites

Before you use PowerExchange for HBase, install and configure Informatica Data Services and Big Data Management™.

Create the following services in the Informatica domain:

- Data Integration Service
- Model Repository Service

Configuration

Configure Informatica Data Services to use PowerExchange for HBase.

To use PowerExchange for HBase to access data in an HBase table, perform the following tasks:

1. Configure the `developerCore.ini` file on the machine that hosts the Developer tool to update the location of the Hadoop cluster binaries. The configuration file is in the following directory: `INFA_HOME/clients/DeveloperClient`.
   Based on the Hadoop distribution that you use, set the Hadoop Distribution Directory property as follows: `-DINFA_HADOOP_DIST_DIR=hadoop\<distribution>_<version>`. By default, the Hadoop Distribution Directory property value is set for a Cloudera distribution.
2. In the Administrator tool, configure the following properties for the Data Integration Service.
   - Data Integration Service Hadoop Distribution Directory
   - Hadoop Distribution Directory

3. If you want to import HBase table from Hadoop cluster where Ranger is enabled, create a user in Ranger with the same name as the Windows NT Login user that launches Informatica Client. Verify that the user in Ranger has Create and Read privileges on the HBase table to import.

CHAPTER 3

HBase Connections

This chapter includes the following topics:

HBase Connections Overview

Create an HBase connection to read data from or write data to an HBase table.

Use the Developer tool, Administrator tool, Analyst tool, or infacmd to create the connections.

HBase Connection Properties

Use an HBase connection to access HBase. The HBase connection is a NoSQL connection. You can create and manage an HBase connection in the Administrator tool or the Developer tool. Hbase connection properties are case sensitive unless otherwise noted.

The following table describes HBase 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.

Property	Description
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 HBase.
ZooKeeper Host(s)	Name of the machine that hosts the ZooKeeper server. The name is case sensitive. When the ZooKeeper runs in the replicated mode, specify a comma-separated list of servers in the ZooKeeper quorum servers. If the TCP connection to the server breaks, the client connects to a different server in the quorum.
ZooKeeper Port	Port number of the machine that hosts the ZooKeeper server.
Enable Kerberos Connection	Enables the Informatica domain to communicate with the HBase master server or region server that uses Kerberos authentication.
HBase Master Principal	Service Principal Name (SPN) of the HBase master server. Enables the ZooKeeper server to communicate with an HBase master server that uses Kerberos authentication. Enter a string in the following format: `hbase/<domain.name>@<YOUR-REALM>` Where: - domain.name is the domain name of the machine that hosts the HBase master server. - YOUR-REALM is the Kerberos realm.
HBase Region Server Principal	Service Principal Name (SPN) of the HBase region server. Enables the ZooKeeper server to communicate with an HBase region server that uses Kerberos authentication. Enter a string in the following format: `hbase_rs/<domain.name>@<YOUR-REALM>` Where: - domain.name is the domain name of the machine that hosts the HBase master server. - YOUR-REALM is the Kerberos realm.

Creating an HBase Connection

Create an HBase connection before you create an HBase data object.

1. In the Developer tool, click **Window** > **Preferences**.
2. Select **Informatica** > **Connections**.
3. Expand the domain in the **Available Connections**.
4. Select the connection type **NoSQL** > **HBase**, and click **Add**.
5. Enter a connection name and an optional description.

6. Select HBase as the connection type.
7. Click **Next**.
8. Configure the connection properties.
9. Click **Test Connection** to verify the connection to HBase.
10. Click **Finish**.

Troubleshooting an HBase Connection Configured for Clusters that Do Not Use Kerberos Authentication

I got a valid Kerberos ticket when I configured a Kerberos connection in the Developer client to connect to the HBase master server that uses Kerberos authentication. Later when I used the same setup and disabled the Kerberos authentication to connect to a cluster that does not use secure authentication, the connection failed.

The error occurs because the mode of authentication in the `hive-site.xml` file does not update and the same ticket that was generated for a Kerberos cluster is used for the cluster that does not use Kerberos authentication.

If you previously enabled Kerberos connection, and you now want to communicate with the HBase master server that does not use Kerberos authentication, restart the Developer client after you disable the Kerberos connection.

CHAPTER 4

HBase Data Objects

This chapter includes the following topics:

HBase Data Objects Overview

An HBase data object is a physical data object that represents data based on an HBase resource. After you create an HBase connection, create an HBase data object and a data object operation.

When you create an HBase Data Object, you can select an HBase table and view all the column families in the table. You can specify the column names in the column family if you know the column name and data type, or you can search the rows in the HBase table and specify the columns.

You can read from and write to a column family or to a single binary column. When you create the data object, specify the column families to read or choose to get all the data as a single stream of binary data. You can also specify the column families to which you can write or choose to write all the data as a single stream of binary data.

Create a data object read operation to read data from the HBase column families and create a data object write operation to insert data to the HBase column families.

Data Object Column Configuration

When you want to read data from or write data to columns in a column family, you can specify the columns when you create the HBase data object.

You can add the columns in the column families or you can search for the columns names in the column family and select the columns. You can also choose to read from or write to a single binary port.

Add Columns

When you create a data object, you can specify the columns in one or more column families in an HBase table.

When you add an HBase table as the resource for a HBase data object, all the column families in the HBase table appear. If you know the details of the columns in the column families, you can select a column family and add the column details. Column details include column name, data type, precision, and scale.

Although data is stored in binary format in HBase tables, you can specify the associated data type of the column to transform the data. To avoid data errors or incorrect data, verify that you specify the correct data type for the columns.

Verify that you specify valid column details when you add columns to avoid unexpected run-time behaviors. If you add a column that does not exist in the column family and create a data object read operation, the Data Integration Service returns a null value for the column at run time. If you do not specify a value for a column when you write data to an HBase table, the Data Integration Service specifies a null value for the column at run time.

If the HBase table has more that one column family, you can add column details for multiple column families when you create the data object. Select one column family at a time and add the columns details. The column family name is the prefix for all the columns in the column family for unique identification.

Search and Add Columns

When you create a data object, you can search the rows in an HBase table to identify the column in the table and select the columns you want to add.

When you do not know the columns in an HBase table, you can search the rows in the table to identify all the columns and the occurrence percentage of the column. You can infer if the column name is valid based on the number of times the column occurs in the table. For example, if column name eName occurs rarely while column name empName occurs in a majority of rows, you can infer the column name as empName.

When you search and add columns, you can specify the maximum number of rows to search and the occurrence percentage value for a column. If you specify the maximum numbers of rows as 100 and the column occurrence percent as 90, all columns that appear at least 90 times in 100 rows appear in the results. You can select the columns in the results to add the columns to the data object.

Get All Columns

Binary data or data that can be converted to a byte array can be stored in an HBase column. You can read from and write to an HBase tables in bytes.

When you create a data object, you can choose to get all the columns in a column family as a single stream of binary data.

Use the HBase data object as a source to read data in all the columns in the column family as a single stream of binary data. Use the HBase data object as a target to write data in all the columns in the source data object as a single column of binary data in the target HBase table.

The Data Integration Service generates the data in the binary column based on the protobuf format. Protobuf format is an open source format to describe the data structure of binary data. The protobuf schema is described as messages.

HBase Data Object Properties

Specify the data object properties when you create the data object.

General Properties

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

Property	Description
Name	Name of the HBase data object.
Location	The project or folder in the Model repository where you want to store the HBase data object.
Connection	Name of the HBase connection.

Add Column Properties

In the **Column Families** dialog box, select the column family to which you want to add the columns. The following table describes the column properties that you configure when you associate columns with column families:

Property	Description
Name	Name of the column in the column family.
Type	Data type of the column.
Precision	Precision of the data.
Scale	Scale of the data.

Search and Add Column Properties

The following table describes the column properties that you configure when you search for columns in column families and add the required columns:

Property	Description
Maximum rows to sample	Maximum number of rows in the HBase table you want to include while searching for columns. Default is 100.
Column occurrence percent	The threshold occurrence percentage of the column. A column appears in the results when the occurrence percentage value of the column meets or exceeds the threshold value. Default is 90.

HBase Data Object Read Operation Properties

HBase data object read operation properties include run-time properties that apply to the HBase data object.

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

The following table describes the Advanced properties for an HBase data object read operation:

Property	Description
Date Time Format	Format of the columns of the date data type. Specify the date and time formats by using any of the Java date and time pattern strings.

HBase Data Object Write Operation Properties

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

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

The following table describes the Advanced properties for an HBase data object write operation:

Property	Description
Date Time Format	Format of the columns of the date data type. Specify the date and time formats by using any of the Java date and time pattern strings.
Auto Flush	Optional. Indicates whether you want to enable Auto Flush. You can set auto flush to the following values: - Enable **Auto Flush** to set the value to true. The Data Integration Service runs each Put operation immediately as it receives them. The service does not buffer or delay the Put operations. Operations are not retried on failure. When you enable auto flush, the operations are slow as you cannot run operations in bulk. However, you do not lose data as the Data Integration Service writes the data immediately. - Disable **Auto Flush** to set the auto flush value to false. When you disable auto flush, the Data Integration Service accepts multiple Put operations before making a remote procedure call to perform the write operations. If the Data integration Service stops working before it flushes any pending data writes to HBase, that data is lost. Disable auto flush if you need to optimize performance. Default is disabled.

Parameterization of HBase Data Objects

You can parameterize the HBase connection and the HBase data object operation properties.

You can parameterize the following data object read operation properties for HBase data objects:

- Connection in the run-time properties
- Filter condition in the query properties
- Date Time Format in the advanced properties

You can parameterize the following data object write operation properties for HBase data objects:

- Connection in the run-time properties.
- Date Time Format in the advanced properties.

Creating an HBase Data Object

Create an HBase data object to specify an HBase resource.

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

   The **New HBase Data Object** dialog box appears.
4. Enter a name for the data object.
5. Click **Browse** next to the **Location** option and select the target project or folder.
6. Click **Browse** next to the **Connection** option and select a connection from which you want to import the HBase resource.
7. To add a resource to the data object, click **Add** next to the **Resource** option.

   The **Add Resource** dialog box appears.
8. Navigate or search for the resources to add to the data object and click **OK**.

   You can add one HBase table to the data object.
9. Click **Next**. The **Column Families** dialog box appears.
10. Select a column family and specify the columns in it. Choose to add columns or get all columns.
    - To manually add, or search and add columns to the column family, select the **Add Columns** option.
    - To read from or write all columns in the column family to a single binary column, select the **Get all columns** option.
11. Add the columns in the column family. Choose to add columns or search the column names in the column family and add the columns.
    - To specify the columns from the column family when you know the column name and datatype, select the column family to which you want to add the columns and click **Add**. Configure the add properties.
    - To search columns in the column family and add them, click **Search and Add**. The **Search and Add** dialog box appears.
12. Specify the following details in the **Search and Add** dialog box:
    a. Specify the maximum rows in the HBase tables you want to include in the search.
    b. Specify the threshold value of the column occurrence percentage.
    c. Click **Go**.

       The column name and the occurrence percentage of the column in the table appears in the results.
    d. Select the columns that you want to specify for the column family. Configure the add properties.
13. Click **Next.**

    The **Create Row** dialog box appears.
14. Select the **Include Row ID** option to generate a row ID for the HBase table.
15. Specify the datatype, precision, and scale for the row ID and click **Next.**

    The **Review Columns** dialog box appears. The column family name is the prefix for all the column names in that column family for unique identification. Default datatype of the row ID is String.
16. Review the columns in the column families and 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 data object after you create it.

Creating an HBase Data Object Operation

Create a data object operation from a data object.

Before you create a data object operation, you must create the data object with the resource.

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 read operation or a write operation.
5. Click **Add**.

   The **Select a resource** dialog box appears.
6. Select the resource 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.

CHAPTER 5

HBase Mappings

This chapter includes the following topics:

HBase Mappings Overview

After you create an HBase data object operation, you can develop a mapping.

You can add an HBase data object operation as a source or as a target in a mapping.

When you configure the data object columns, you can get data in all columns in a column family to a single column as binary data. Use the Data Processor transformation to convert the binary data into the required data types.

You can validate and run mappings in the native environment or a Hive environment. When you run a mapping in a Hive environment, the Data Integration Service creates multiple Map jobs to read data or write data in parallel.

HBase Validation and Run-time Environments

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

The Data Integration Service validates whether the mapping can run in the selected environment.

You can deploy the mapping and run it in the selected environment. You can run standalone mappings or mappings that are a part of a workflow.

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 on a Hadoop cluster, you can select a Blaze, Spark, or Hive engine. The Data Integration Service pushes the mappings to the selected engine for processing.

Filtering Source Data

When you configure a mapping that reads data from an HBase source, you can enter a filter expression to filter records read from the source.

You can select the mapping and add the filter expression in the Query tab in the Properties view. You can use any comparison operators in the filter expression.

When you run the mapping, the Data Integration Service filters the source data based on the expressions.

Note: For numeric data types, the Data Integration Service applies the operators for positive values and not for negative values.

When you enter multiple filter expressions, the AND logical operator is applied between the expressions.

Note: If you use the not equal, less than, less than or equal to operators and some columns do not meet the filter condition, the Data Integration Service returns a Null value for these columns. If you use the equal, greater than, greater than or equal to operators and some columns do not meet the filter condition, the Data Integration Service does not return the rows associated with these columns.

Example

The following table lists the columns in the CF column family in an HBase table. There are rows that have c1 and c2 columns, rows that have at least one of the columns, and rows that have neither of the columns.

Row	Column Value
1	column=CF__c1, value=john
1	column=CF__c2, value=jane
2	column=CF__c1, value=jane
3	column=CF__c2, value=jdoe
4	column=CF__c8, value=adam

Create an HBase data object called Name and add it to an HBase mapping. Add the following filter expressions:

Name.CF__C1 = 'john' AND Name.CF__C2 = 'jane'

The Data Integration Service returns the following output because of the equal to operator in the filter expressions. The service does not return rows that contain null values.

ROW: 1

c1: john

c2: jane

However, the output is different when you add the following filter expressions:

Name.CF__C1 != 'john' AND Name.CF__C2 != 'jane'

The Data Integration Service returns rows that contain null values because of the not equal to operator in the filter expression.

Row 2

c1: jane

Row 3:

c2: jdoe

ROW: 4

c1: null

c2: null

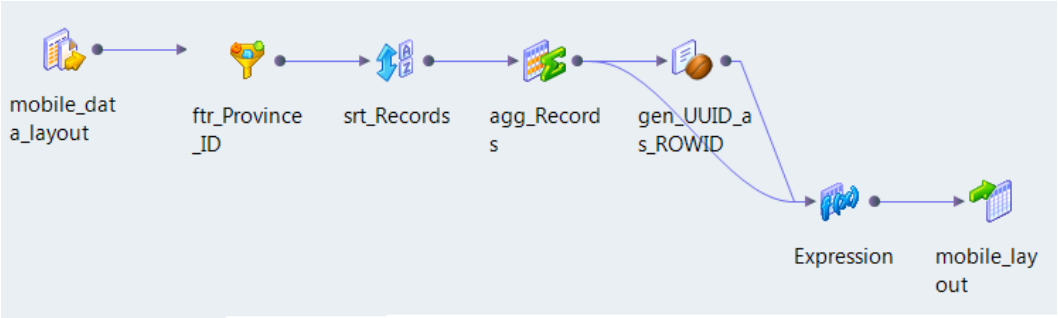
HBase Mapping Example

Your organization is a mobile service provider and it needs to load the data in WAP log files to HBase tables and generate multiple reports.

WAP log files can contain columns with information about the mobile users, internet usage, and data volume. On a single day, the WAP log files can be three to four billion rows of data and can be around two terabytes in size.

You can consolidate the data in the WAP log files that you receive through the day. You can then perform transformations based on your requirements.

The following figure shows the HBase mapping example:



You can use the following objects in an HBase mapping:

Flat File Data Object

The source for the mapping is a flat file data object that contains the data in a WAP log file.

Create a flat file data object and specify the WAP log file as the resource for the data object. Source columns in the flat file data object include Province ID, data volume, URL, and session duration. Configure the read properties of the data object.

Transformations

Add transformations to get aggregate data about the internet usage of the mobiles users in a particular province.

- The ftr_Province_ID Filter transformation filters the data in the log files based on the value you specify for the province ID column.
  The Data Integration Service returns the rows that meet the filter condition.
- The srt-Records Sorter transformation sorts the data in ascending order based on the province ID.
- The agg_Records Aggregator transformation collects statistics about internet usage and data volume of the mobile users for a particular province.

Use the result of the Sorter transformation as an input to the Aggregator transformation. You can increase Aggregator transformation performance with the sorted input option.

- The gen_UUID_as_ROWID Java transformation generates a unique row key ID before you load the data to HBase tables.
  Each row in an HBase table has a unique row key ID. You can write the generated key value as the row key ID for each row in the HBase table
- The Expression transformation formats the data before you load it to the Hbase table.

HBase Data Object

The target of the mapping is an HBase data object. Specify the columns in the HBase table to which you want to write the data.

Create an HBase data object write operation to write data to the HBase table.

After you run the mapping, the Data Integration Service writes the transformed data to the HBase table. Analysts can run queries and perform real-time analysis of daily operations, statistics about gateway usage, and data volume based on the data in the HBase tables.

APPENDIX A

Data Type Reference

This appendix includes the following topics:

Data Type Reference Overview

Informatica Developer uses the following data types in HBase mappings:

- HBase native data types. HBase data types appear in the physical data object column properties.

  Note: Although data is stored in binary format in HBase tables, you can specify the data type associated with a column when you create the HBase data object.

- 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.

HBase and Transformation Data Types

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

HBase 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 Hive environment. You can use the user-defined functions to transform the binary data.
String	String	1 to 104,857,600 characters

HBase Data Type	Transformation Data Type	Range and Description
Short	Integer	-2,147,483,648 to 2,147,483,647
Integer	Integer	-2,147,483,648 to 2,147,483,647
Float	Decimal	Precision 1 to 28
Long	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Double	Double	Precision 15
DateTime	Date/Time	Date and time formats are specified by using any of the Java date and time pattern strings. For example, The "EEE, d MMM yyyy HH:mm:ss z" pattern string is interpreted as Wed, 14 Dec 2013 12:08:56. The "yyyy.MM.dd HH:mm:ss z" pattern string is interpreted as 2013.12.14 12:08:56 PDT.

APPENDIX B

Glossary

column familiy

A group of multiple columns in an HBase table. HBase users can group columns based on appropriate logic to provide boundaries between the data. An HBase table has at least one column family, which is a collection of all the columns in the table. The column family name is the prefix for all the column names in that column family for unique identification.

HBase column

Basic unit for storing data in an HBase table.

HBase row

A row in an HBase table has one or more columns. Each row is uniquely identified by a row key. All rows are sorted lexicographically by their row key. In lexicographical sorting, each row key is compared on a binary level, byte by byte, from left to right with other row keys.

HBase table

An HBase table is made up of rows and columns similar to any database table.

ZooKeeper

A reliable, highly available, persistent, and distributed coordination service that provides a centralized infrastructure and services that enable synchronization across a cluster. PowerExchange for HBase connects to an HBase data store through the ZooKeeper.

ZooKeeper host

Name of the machine that hosts the ZooKeeper server. When the ZooKeeper is run in the replicated mode, specify a comma-separated list of servers in the ZooKeeper quorum servers.

ZooKeeper port

Port number of the machine that hosts the ZooKeeper server.

ZooKeeper quorum servers

A quorum is a replicated group of ZooKeeper servers in the same application. ZooKeeper can be run in standalone or replicated mode. In a distributed environment, it is generally run in the replicated mode so that all the servers that make up the ZooKeeper service know about each other. If a majority of the servers in the quorum are available, the ZooKeeper service will be available. If the TCP connection to a server breaks, the client connects to a different server in the quorum.

INDEX