

cloudera[®]

Cloudera Connector for Teradata

Important Notice

© 2010-2018 Cloudera, Inc. All rights reserved.

Cloudera, the Cloudera logo, and any other product or service names or slogans contained in this document are trademarks of Cloudera and its suppliers or licensors, and may not be copied, imitated or used, in whole or in part, without the prior written permission of Cloudera or the applicable trademark holder. If this documentation includes code, including but not limited to, code examples, Cloudera makes this available to you under the terms of the Apache License, Version 2.0, including any required notices. A copy of the Apache License Version 2.0, including any notices, is included herein. A copy of the Apache License Version 2.0 can also be found here: <https://opensource.org/licenses/Apache-2.0>

Hadoop and the Hadoop elephant logo are trademarks of the Apache Software Foundation. All other trademarks, registered trademarks, product names and company names or logos mentioned in this document are the property of their respective owners. Reference to any products, services, processes or other information, by trade name, trademark, manufacturer, supplier or otherwise does not constitute or imply endorsement, sponsorship or recommendation thereof by us.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Cloudera.

Cloudera may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Cloudera, the furnishing of this document does not give you any license to these patents, trademarks copyrights, or other intellectual property. For information about patents covering Cloudera products, see <http://tiny.cloudera.com/patents>.

The information in this document is subject to change without notice. Cloudera shall not be liable for any damages resulting from technical errors or omissions which may be present in this document, or from use of this document.

Cloudera, Inc.

**395 Page Mill Road
Palo Alto, CA 94306
info@cloudera.com
US: 1-888-789-1488
Intl: 1-650-362-0488
www.cloudera.com**

Release Information

Version: Connectors 1.x
Date: August 28, 2018

Table of Contents

Cloudera Connector Powered by Teradata User Guide.....	4
Version Scheme for Teradata Connector.....	4
Cloudera Connector Powered by Teradata Release Notes.....	4
<i>CDH 6 compatible version of Cloudera Connector Powered by Teradata 1.7c6 Available.....</i>	<i>4</i>
<i>New Features in Cloudera Connector Powered by Teradata.....</i>	<i>4</i>
<i>Limitations for Cloudera Connector Powered by Teradata.....</i>	<i>6</i>
<i>Known Issues and Workarounds.....</i>	<i>6</i>
<i>Getting Support.....</i>	<i>6</i>
Prerequisites for Teradata Connector.....	6
Installing the Teradata Connector.....	7
<i>Installation with CDH 5 and Cloudera Manager 5.....</i>	<i>7</i>
<i>Installation without Cloudera Manager.....</i>	<i>8</i>
Upgrading the Teradata Connector.....	9
<i>Upgrading with CDH 5 and Cloudera Manager 5.....</i>	<i>9</i>
<i>Upgrading without Cloudera Manager.....</i>	<i>9</i>
Using the Cloudera Connector Powered by Teradata.....	10
<i>Input Methods.....</i>	<i>12</i>
<i>Output Methods.....</i>	<i>13</i>
Uninstalling Teradata Connectors.....	13
<i>Uninstallation with CDH 5 and Cloudera Manager 5.....</i>	<i>13</i>
<i>Uninstallation without Cloudera Manager 5.....</i>	<i>14</i>
Appendix: Apache License, Version 2.0.....	15

Cloudera Connector Powered by Teradata User Guide

Cloudera Connector Powered by Teradata provides high-speed data transfer between Teradata and CDH.

This connector allows various Sqoop tools, such as `sqoop-import` and `sqoop-export`, to operate in highly efficient direct modes, and expose options that are specific to Teradata.

This guide describes how to install, configure, and use the connector in a Sqoop 1 installation and provides reference information for connector operation. This guide is intended for:

- System and application programmers
- System administrators
- Database administrators
- Data analysts
- Data engineers

Version Scheme for Teradata Connector

This topic describes the versioning scheme used for Cloudera Connector Powered by Teradata. The version string consists of the following parts:

`<major_Sqoop_version>.<Connector_version>c<major_CDH_version>`

- `<major_Sqoop_version>`: Identifies the major version of Sqoop for which the connector has been compiled. Currently, its only values are 1 and 2.
- `<Connector_version>`: Revision of the connector itself, possible values are 1, 2, 3, 4...
- `<major_CDH_version>`: The major CDH version for which the connector has been compiled and tested.

For example:

- 1.6c5 - Sixth revision of a Sqoop 1-based connector that is compatible with CDH 5.

Cloudera Connector Powered by Teradata Release Notes

This section summarizes the high level changes and most important new features in the Cloudera Connectors for Teradata.

CDH 6 compatible version of Cloudera Connector Powered by Teradata 1.7c6 Available

Cloudera Connector Powered by Teradata 1.7c6 is compatible with CDH 6. It does not contain new features or changes.



Note: Cloudera Connector Powered by Teradata 1.7c5 and lower are not compatible with CDH 6.

New Features in Cloudera Connector Powered by Teradata

The following new features are included in Cloudera Connector Powered by Teradata.

New Features in Cloudera Connector Powered by Teradata Version 1.7c5

Cloudera Connector Powered by Teradata now supports Teradata 16.x. This release upgrades the JDBC driver to version 16.10.00.05 and the TDCH library to version 1.5.4.

Cloudera Connector Powered by Teradata now supports importing tables without `split-by` column specified when the number of mappers is set to 1.

Cloudera Connector Powered by Teradata now supports the `internal.fastexport` input method. For table import, the following values for the `--input-method` option are valid:

- `split.by.partition`
- `split.by.hash`
- `split.by.value`
- `split.by.amp`
- `internal.fastexport`

Note that the query import still only supports the `split.by.partition` input method.

The `internal.fastexport` method implements coordination between the mappers and a coordinator process (running on the edge node where the job was submitted). The host name and the port of this process are automatically resolved, but there are new options introduced for manual configuration:

- `--fastexport-socket-hostname`: Configures the host of the coordinator process. It sets the `tdch.input.teradata.fastexport.coordinator.socket.host` Java property exposed by the underlying Teradata Connector for Hadoop (TDCH) library.
- `--fastexport-socket-port`: Configures the port of the coordinator process. It sets the `tdch.input.teradata.fastexport.coordinator.socket.port` Java property exposed by the underlying Teradata Connector for Hadoop (TDCH) library.

For more information on these properties, see the Teradata Connector for Hadoop tutorial provided by Teradata.

New Features in Cloudera Connector Powered by Teradata Version 1.6.1c5

- Adds support for SLES 12.

New Features in Cloudera Connector Powered by Teradata Version 1.6c5

- Upgrades the JDBC driver to version 15.10.00.22 and the TDCH library to version 1.5.0. These libraries contain several bug fixes and improvements.
- Adds the `--schema` argument, used to override the `<td-instance>` value in the connection string of the Sqoop command. For example, if the connection string in the Sqoop command is `jdbc:teradata://<td-host>/DATABASE=database1`, but you specify `--schema database2`, your data is imported from `database2` and not `database1`. If the connection string does not contain the `DATABASE` parameter — for example `jdbc:teradata://<td-host>/CHARSET=UTF8` — you can also use the `--schema database` argument to have Sqoop behave as if you specified the `jdbc:teradata://<td-host>/DATABASE=databasename,CHARSET=UTF8` connection string.

New Features in Cloudera Connector Powered by Teradata Version 1.5c5

New features:

- Fixed compatibility issue with CDH 5.5.0 and higher.

New Features in Cloudera Connector Powered by Teradata Versions 1.4c5

New features:

- Added support for JDK 8.
- Added `--error-database` option.
- Added ability to specify format of date, time, and timestamp types when importing into CSV.
- Import method `split.by.amp` now supports views.
- Upgraded Teradata connector for Hadoop to version 1.3.4.

New Features and Changes in Cloudera Connector Powered by Teradata 1.3c5

New features:

- Upgraded Teradata Connector for Hadoop to version 1.3.3.
- Parcel distribution now contains Teradata JDBC driver; manual download no longer required.

Cloudera Connector Powered by Teradata User Guide

- Added support for query import into Avro file format.

Changes:

- Export method `multiple.fastload` has been removed.

New Features in Cloudera Connector Powered by Teradata Versions 1.2c5

New features:

- Upgraded Teradata Connector for Hadoop to version 1.2.1.
- Added support for Avro.
- Added support for Incremental import.
- Added support for `--where` argument.
- Added support for Hive import.
- Added support for Importing all tables using `import-all-tables`.
- Added support for Query Bands.
- Added new import method `split.by.amp` (supported only on Teradata 14.10 and higher).

New Features in Cloudera Connector Powered by Teradata Version 1.0.0

This is the first release of this new connector. This connector features:

- Support for secondary indexes.
- Especially fast performance in most cases.

Limitations for Cloudera Connector Powered by Teradata

Limitations for Cloudera Connector Powered by Teradata has the following functional limitations.

- Does not support HCatalog.
- Does not support import into HBase.
- Does not support upsert functionality (parameter `--update-mode allowinsert`).
- Does not support the `--boundary-query` option.
- Does not support Parquet file format.
- Does not support export to Teradata VIEWS.
- Does not support Kerberos authentication.
- By default speculative execution is disabled for the Teradata Connector. This avoids placing redundant load on the Teradata database.

Known Issues and Workarounds

None.

Getting Support

Support for the Cloudera Connector for Teradata is available through Cloudera Enterprise Support. Refer to [Cloudera Support](#) for more details.

Prerequisites for Teradata Connector

To use the Cloudera Connector Powered by Teradata, you must have a functioning CDH installation, including Sqoop components. Depending on how Sqoop is installed, you may need administrative privileges to create or modify configuration files.

The Teradata connector uses catalog tables and views to look up metadata information. Therefore, the user making the connection must have the `SELECT` privilege on the `DBC` database. Check with your Teradata administrators or operators if you are not sure whether you have the required privileges. You must have `SELECT` privileges on at least one of the following DBC database object types:

- DBC.columns
- DBC.databases
- DBC.tables
- DBC.indices

Depending on the input method used, the Teradata connector might need to create temporary tables or temporary views to import data. Check with your Teradata administrators or operators to determine if you have the required privileges.

The Cloudera Connector Powered by Teradata requires the following additional permissions to use `*.fastload` data methods.

- DBC.table_levelconstraints
- DBC.triggers
- DBC.tvn
- DBC.dbase
- DBC.referencingtbls

For more information about how to install, configure, and use Sqoop, see the Sqoop documentation in the [CDH 5 Installation Guide](#).

Cloudera Connector powered by Teradata versions 1.2c5, 1.3c5, 1.4c5, 1.5c5, 1.6c5, 1.7c5

- Teradata 13.00 and higher
- CDH 5.0 and higher
- Sqoop 1.4 and higher, but not compatible with Sqoop2

Cloudera Connector powered by Teradata versions 1.0.0, 1.2c4, 1.3c4, 1.4c4

- Teradata 13.00 and higher
- CDH 4.2 and higher
- Sqoop 1.4 and higher, but not compatible with Sqoop2

Installing the Teradata Connector

Use one of the following methods to install the Sqoop connector for Teradata:

- If you have a CDH 5 cluster managed by Cloudera Manager 5, see [Installation with CDH 5 and Cloudera Manager 5](#) on page 7.
- If your cluster is not managed by Cloudera Manager, see [Installation without Cloudera Manager](#) on page 8.

Installation with CDH 5 and Cloudera Manager 5

Step 1: Adding the Sqoop Client Gateway

The Sqoop1 Client Gateway sets up and manages the connector configuration for the hosts where you execute Sqoop1 commands. If you do not already have the Sqoop1 Client Gateway deployed in your cluster, deploy it before proceeding.



Important: The Sqoop 1 Client Gateway is required for the Teradata Connector to work correctly. Cloudera recommends installing the Sqoop 1 Client Gateway role on any host used to execute the Sqoop CLI. If you do not already have the Sqoop Client service running on your cluster, see [Managing the Sqoop 1 Client](#) for instructions on how to add the service using the Cloudera Manager Admin Console.

Step 2: Download, Distribute, and Activate the Sqoop Parcels

Parcels for Sqoop connectors are prefixed by `SQOOP_`, followed by the name of the connector.

Follow the instructions in [Managing Parcels](#) to download, distribute, and activate Sqoop parcels.

Installation without Cloudera Manager



Note: For the sample file and directory names used in the instructions below, replace `x` with the CDH major version you are using. For example, for a CDH 5 setup, `sqoop-connector-teradata-1.2cX` should be replaced by `sqoop-connector-teradata-1.2c5`.

1. Install the Sqoop connector by opening the distribution archive in a convenient location such as `/usr/lib`. Opening the distribution creates a directory that contains the jar file of the compiled version of the connector. Note the path to this jar file. The directory that is created when the file is expanded varies according to which connector you are using. Examples of typical resulting paths include:
 - Cloudera Connector Powered by Teradata 1.2cX:
`/usr/lib/sqoop-connector-teradata-1.2cX/sqoop-connector-teradata-1.2cX.jar`
2. Copy the Teradata JDBC drivers (`terajdbc4.jar` and `tdgssconfig.jar`) to the `lib` directory of the Sqoop installation. You can obtain these drivers by extracting them from the Teradata Connector JAR that you downloaded from: <https://www.cloudera.com/downloads/connectors/sqoop/teradata/1-2c5.html>. Without these drivers, the connector will not function correctly.
3. Confirm that the `managers.d` directory exists in the Sqoop configuration directory.



Note: Depending on how Sqoop is installed, its configuration directory can be in `/etc/sqoop/conf`, `/usr/lib/sqoop/conf`, or elsewhere if Sqoop was installed using the tarball distribution.

If the `managers.d` directory does not exist, create it and ensure that the directory permissions are set to `755`.

4. Create a text file in the `managers.d` directory with a descriptive name such as `cldra_td_connector`. Ensure the file permissions are set to `644`.
5. The `cldra_td_connector` file must have the connector class name followed by the complete path to the directory where the connector jar is located.

For example, for the Cloudera Connector powered by Teradata 1.2cX:

```
com.cloudera.connector.teradata.TeradataManagerFactory= \
/usr/lib/sqoop-connector-teradata-1.2cX/sqoop-connector-teradata-1.2cX.jar
```



Note: The preceding command is shown on two lines but must be entered on a single line.

`TeradataManagerFactory` acts as a single point of delegation for invoking the connector bundled with this distribution. Alternatively, you can specify `TeradataManagerFactory` inside a `sqoop-site.xml` file, which must be inside a classpath directory.

If you are using Cloudera Connector powered by Teradata, use the following:

```
<configuration>
  <property>
    <name>sqoop.connection.factories</name>
    <value>com.cloudera.connector.teradata.TeradataManagerFactory</value>
  </property>
</configuration>
```

This configures a Sqoop action to use the Teradata connector inside Oozie.

Upgrading the Teradata Connector

Use these instructions if you are upgrading one of the connectors to a newer version (for example, if you need to upgrade Cloudera Connector Powered by Teradata from version 1.2c4 to 1.3c4).

Upgrading with CDH 5 and Cloudera Manager 5

Step 1: Distributing the Sqoop Connector Parcels

1. In the Cloudera Manager Admin Console, click **Hosts** in the top navigation bar and then go to the **Parcels** tab. Parcels for the Sqoop connectors are listed on this page, prefixed by "SQOOP_", followed by the name of the connector.
2. Click **Download** for the connectors you want to install.
3. After the parcels have been downloaded, click **Distribute** to distribute and unpack the connectors on all hosts in your Hadoop cluster.
4. After the parcels have been distributed, click **Activate** to make them available to the cluster. Sqoop connectors are listed as **Activated** on the **Parcels** page. You must redeploy the client configuration (Step 3) for activation to take effect.

Step 2: Deploying Client Configuration Files

1. In the Cloudera Manager Admin Console, go to the Sqoop Client service.
2. From the **Actions** menu at the top right of the service page, select **Deploy Client Configuration**.
3. Click **Deploy Client Configuration** to confirm redeployment of the client configuration.

Upgrading without Cloudera Manager



Note: For the sample file and directory names used in the instructions below, replace *x* with the CDH major version you are using. For example, for a CDH 5 setup, `sqoop-connector-teradata-1.2cX` should be replaced by `sqoop-connector-teradata-1.2c5`. Be sure to update all existing paths to new paths for your updated software.

1. Install the Sqoop connector by opening the distribution archive in a convenient location such as `/usr/lib`. Opening the distribution creates a directory that contains the jar file of the compiled version of the connector. Note the path to this jar file. The directory that is created when the file is expanded varies according to which connector you are using. Examples of typical resulting paths include:
 - Cloudera Connector Powered by Teradata 1.2cX:
`/usr/lib/sqoop-connector-teradata-1.2cX/sqoop-connector-teradata-1.2cX.jar`
2. Copy the Teradata JDBC drivers (`terajdbc4.jar` and `tdgssconfig.jar`) to the `lib` directory of the Sqoop installation. You can obtain these drivers from the Teradata download website: <http://downloads.teradata.com/download/connectivity/jdbc-driver>. Without these drivers, the connector will not function correctly.
3. Confirm that the `managers.d` directory exists in the Sqoop configuration directory.



Note: Depending on how Sqoop is installed, its configuration directory can be in `/etc/sqoop/conf`, `/usr/lib/sqoop/conf`, or elsewhere if Sqoop was installed using the tarball distribution.

- If the `managers.d` directory does not exist, create it and ensure that the directory permissions are set to 755.
4. Create a text file in the `managers.d` directory with a descriptive name such as `cldra_td_connector`. Ensure the file permissions are set to 644.
 5. The `cldra_td_connector` file must have the connector class name followed by the complete path to the directory where the connector jar is located.

For example, for the Cloudera Connector powered by Teradata 1.2cX:

```
com.cloudera.connector.teradata.TeradataManagerFactory= \  
/usr/lib/sqoop-connector-teradata-1.2cX/sqoop-connector-teradata-1.2cX.jar
```



Note: The preceding command is shown on two lines but must be entered on a single line.

`TeradataManagerFactory` acts as a single point of delegation for invoking the connector bundled with this distribution. Alternatively, you can specify `TeradataManagerFactory` inside a `sqoop-site.xml` file, which must be inside a classpath directory.

If you are using Cloudera Connector Powered by Teradata, use the following:

```
<configuration>  
  <property>  
    <name>sqoop.connection.factories</name>  
    <value>com.cloudera.connector.teradata.TeradataManagerFactory</value>  
  </property>  
</configuration>
```

This configures a Sqoop action to use the Teradata connector inside Oozie.

Using the Cloudera Connector Powered by Teradata

After you have installed the connector and copied the JDBC drivers for Teradata to the `lib` directory of the Sqoop installation, use this connector by invoking Sqoop tools with the appropriate connection string.

The connection string format is `jdbc:teradata://<td-host>/DATABASE=<td-instance>`:

- `<td-host>` is the hostname of the machine on which the Teradata server runs.
- `<td-instance>` is the Teradata database instance name.

For example, the following command invokes the Sqoop import tool with three mappers:

```
$ sqoop import --connect jdbc:teradata://localhost/DATABASE=sqooptest \  
--username sqooptest --password xxxxx --table MY_TABLE --num-mappers 3 \  
--target-dir /user/sqooptest/MY_TABLE
```

The following command invokes the Sqoop export tool with three mappers:

```
$ sqoop export --connect jdbc:teradata://localhost/DATABASE=sqooptest \  
--username sqooptest --password xxxxx --export-dir /user/sqooptest/MY_TABLE \  
--table MY_TABLE_TARGET --num-mappers 3
```

You can control the behavior of the connector by using extra arguments. Extra arguments must appear at the end of the command. Use a double-dash separator (`--`) to indicate the end of the standard arguments and the beginning of the extra arguments. For example, the following command uses the double-dash (`--`) separator (in bold for emphasis) to separate the standard arguments `--table` and `--num-mappers` from the extra arguments `--input-method` and `--query-band`:

```
$ sqoop ... --table MY_TABLE --num-mappers 3 -- --input-method split.by.amp --query-band  
DC=BP\;Location=Europe
```

Table 1: Teradata Connector Feature Support

Parameter	Tool	Description
<code>--staging-table</code>	import and export	Override the default staging table name. This parameter applies only if staging tables are used during data transfer.
<code>--staging-database</code>	import and export	Override the default staging database name. This parameter applies only if staging tables are used during the data transfer.
<code>--staging-force</code>	import and export	Force the connector to create the staging table if the input/output method supports staging tables.
<code>--input-method</code>	import	Specify the input method used to transfer data from Teradata to Hadoop.
<code>--output-method</code>	export	Specify the output method used to transfer data from Hadoop to Teradata.
<code>--batch-size</code>	import and export	Specify the number of rows processed together in one batch.
<code>--access-lock</code>	import	Improve concurrency. When used, the import job is not blocked by concurrent accesses to the same table.
<code>--query-band</code>	import and export	Allow arbitrary query bands to be set for all queries that are run by the connector. The expected format is a semicolon-separated key=value pair list. A final semicolon is required after the last key=value pair. For example, <code>Data_Center=XO;Location=Europe;</code>
<code>--error-table</code>	export (only for <code>internal.fastload</code>)	Specify a prefix for created error tables.
<code>--error-database</code>	export (only for <code>internal.fastload</code>)	Override the default error database name.
<code>--fastload-socket-hostname</code>	export (only for <code>internal.fastload</code>)	Hostname or IP address of the host on which you are running Sqoop, one that is visible from the Hadoop cluster. The connector can autodetect the interface. This parameter overrides the autodetection routine.
<code>--keep-staging-table</code>	import	By default, the connector drops all automatically created staging tables when export fails. This option leaves the staging tables with partially imported data in the database.
<code>--num-partitions-for-staging-table</code>	import (only for <code>split.by.partition</code>)	Number of partitions to use for the automatically created staging table. The connector automatically generates the value based on the number of mappers used.
<code>--skip-xviews</code>	import and export	By default, the connector uses Teradata system views to obtain metadata. With this parameter, the connector switches to XViews instead.

Parameter	Tool	Description
<code>--date-format</code>	import and export	Use custom format for columns of <code>date</code> type. The parameter uses SimpleDateFormat formatting options.
<code>--time-format</code>	import and export	Use custom format for columns of <code>time</code> type. The parameter uses SimpleDateFormat formatting options.
<code>--timestamp-format</code>	import and export	Use custom format for columns of <code>timestamp</code> type. The parameter uses SimpleDateFormat formatting options.

Input Methods

Cloudera Connector Powered by Teradata supports the following methods for importing data from Teradata to Hadoop:

- `split.by.amp`
- `split.by.value`
- `split.by.partition`
- `split.by.hash`

split.by.amp Method

This optimal method retrieves data from Teradata. The connector creates one mapper per available Teradata AMP, and each mapper subsequently retrieves data from each AMP. As a result, no staging table is required. This method requires Teradata 14.10 or higher.

split.by.value Method

This method creates input splits as ranges on the split by column (usually the table's primary key). Each split is subsequently processed by a single mapper to transfer the data using SELECT queries. All splits can access all AMPs to retrieve data, so you should set the number of mappers between 20 and 30 because there is a limit for all-AMP concurrently running operations on the Teradata appliance. Ensure that users transferring data have sufficient pool space available for the SELECT queries.

split.by.partition Method

This method is preferred for extracting a large amount of data from the Teradata system. Behavior of this method depends whether source table is partitioned or not.

split.by.hash Method

This input method is similar to the `split.by.partition` method. Instead of directly operating on value ranges of one column, this method operates on the hash of the column. You can use importing by hash to extract data in situations where `split.by.value` and `split.by.partition` are not appropriate. Each mapper can access all AMPs available in the system, so set the number of mappers between 20 and 30 because there is a limit for all-AMP concurrent jobs on the Teradata appliance.

The following example shows import using input method `split.by.hash`:

```
$ sqoop import --connect jdbc:teradata://localhost/DATABASE=sqooptest \  
--username sqooptest --password xxxxx --table MY_TABLE --num-mappers 3 \  
--target-dir /user/sqooptest/MY_TABLE --input-method split.by.hash
```

If your input table is not partitioned, the connector creates a partitioned staging table and runs an INSERT into SELECT query to move data from the source table into the staging table. Subsequently, each mapper transfers data from one partition, resulting in a single AMP operation. With a single AMP, you can use a large number of mappers to obtain optimal performance. The amount of available permanent space must be as large as your source table and the amount of pool space required to run the SELECT queries.

If your table is already partitioned, no extra staging table is created. However, you can force the connector to re-partition your data using the `--staging-force` parameter to achieve better performance. Without forcing repartition of the data, this method opens all-AMP operation, so you should use between 20 and 30 mappers. If your source table is a PI table, and your split by column is the table's primary key, the connector creates a single AMP operation, and you can use high number of mappers.

Output Methods

Cloudera Connector Powered by Teradata supports the following output methods to export data from Hadoop to Teradata:

- `batch.insert`
- `internal.fastload`

batch.insert Method

This method uses JDBC batch jobs to export data to the Teradata appliance. This method should be used only when other methods are not a viable. It creates a partitioned staging table before the export job, and then subsequently each mapper transfers data from Hadoop to one partition. After all mappers end, the INSERT into SELECT statement is called to transfer the data from staging to table to final destination. Ensure that you have sufficient permanent space for two copies of your data. You also need sufficient spool space for running the INSERT into SELECT query. The number of mappers that this method can use is limited only by the number of concurrent sessions allowed on the Teradata appliance.

internal.fastload Method

This method requires a partitioned staging table. Data is first exported by each mapper into a different partition and then moved to the target table, using the INSERT into SELECT statement. Make sure that you have sufficient permanent and spool space to store two copies of your data and to move them from the staging table to the target table. All mappers participate in one FastLoad job coordinated by an internal protocol. This is the fastest method for exporting data from Hadoop to a Teradata appliance. Because all mappers participate in the one FastLoad job, only one Teradata utility slot is used for the export job. The number of mappers is limited only by total number of AMPs available in your system.

The Teradata server is started on the machine where the sqoop command is running, and all mappers started by this sqoop command must connect to it. Because a mapper can be run on any hosts in the cluster, all hosts must have access to the machine running the sqoop command.

Uninstalling Teradata Connectors

You can use one of the following ways to uninstall the Sqoop connectors for Teradata:

- If you have a CDH 5 cluster managed by Cloudera Manager 5, see [Uninstallation with CDH 5 and Cloudera Manager 5](#) on page 13.
- If your cluster is not managed by Cloudera Manager, see [Uninstallation without Cloudera Manager 5](#) on page 14.

Uninstallation with CDH 5 and Cloudera Manager 5

Perform the following steps to uninstall the Sqoop connectors for Teradata using Cloudera Manager 5:

1. Removing the Sqoop Connector Parcels:

- In the Cloudera Manager Admin Console, click **Hosts** in the top navigation bar and then go to the **Parcels** tab. Parcels for the Sqoop connectors are listed on this page, prefixed by "SQOOP_", followed by the name of the connector.
- The Sqoop connectors are listed as **Activated**. To deactivate a parcel, click **Actions** on an activated parcel and select **Deactivate**.
- To remove the parcel, click the down arrow to the right of the **Activate** button and select **Remove from Hosts**.

2. Redeploy client configuration:

- a. In the Cloudera Manager Admin Console, go to the Sqoop Client service.
- b. From the **Actions** menu at the top right of the service page, select **Deploy Client Configuration**.
- c. Click **Deploy Client Configuration** to confirm redeployment of client configuration.

Uninstallation without Cloudera Manager 5

To remove the Cloudera Connector for Teradata:

- Delete the `cldra_td_connector` file from `managers.d` directory located under the Sqoop configuration directory.
- Remove the files from the connectors distribution. These files might be in a location under `/usr/lib/`.
- Remove the JDBC drivers for Teradata that you copied to the `lib` directory of the Sqoop installation. Sqoop no longer needs these drivers. The Teradata JDBC drivers are named `terajdbc4.jar` and `tdgssconfig.jar`.

Appendix: Apache License, Version 2.0

SPDX short identifier: Apache-2.0

Apache License
Version 2.0, January 2004
<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License.

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims

licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution.

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

1. You must give any other recipients of the Work or Derivative Works a copy of this License; and
2. You must cause any modified files to carry prominent notices stating that You changed the files; and
3. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
4. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions.

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.

Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks.

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty.

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability.

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability.

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

```
Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```