Informatica®

Informatica® Cloud Data Integration

Hive Connector

Informatica Cloud Data Integration Hive Connector
November 2024

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2024-11-05

Table of Contents

Preface

Use Hive Connector to learn how to read from or write to Hive by using Cloud Data Integration. Learn to create a connection, develop and run mappings, mapping tasks, and dynamic mapping tasks in Cloud Data Integration.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at http://www.informatica.com/cloud. This site contains information about Informatica Cloud integration services.

Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

https://network.informatica.com/community/informatica-network/products/cloud-integration

Developers can learn more and share tips at the Cloud Developer community:

https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers

Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

https://marketplace.informatica.com/

Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at https://www.informatica.com/trust-center.html.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The Informatica Intelligent Cloud Services Status page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, go to https://status.informatica.com/ and click **SUBSCRIBE TO UPDATES**. You can then choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or online.

For online support, click **Submit Support Request** in Informatica Intelligent Cloud Services. You can also use Online Support to log a case. Online Support requires a login. You can request a login at https://network.informatica.com/welcome.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at https://www.informatica.com/services-and-training/support-services/contact-us.html.

CHAPTER 1

Introduction to Hive Connector

You can use Hive Connector to connect to Hive from Data Integration. You can use Hive Connector on a Linux operating system.

You can read data from and write data to Hive tables. The tables can also be partitioned or bucketed in Hive. You can access Hive from Kerberos or non-Kerberos clusters and on Amazon EMR 6.3 kerberized cluster where the Hive metastore is on AWS Glue or MySQL. You can also connect to Hive on a Cloudera CDP Private Cloud 7.x distribution enabled with Knox. For the other distributions that apply for Hive Connector, see Distributionson page 10.

You can use a Hive object as a source and target in mappings and mapping tasks. You can switch mappings to advanced mode to include transformations and functions that enable advanced functionality.

Hive Connector assets

Create assets in Data Integration to integrate data using Hive Connector.

When you use Hive Connector, you can include the following Data Integration assets:

- Dynamic mapping task
- Mapping
- Mapping task

For more information about configuring assets and transformations, see *Mappings*, *Transformations*, and *Tasks* in the Data Integration documentation.

Hive Connector administration

Before you create a connection and run assets with Hive Connector, complete the prerequisites.

Access a Kerberos or TLS enabled Hadoop cluster

You must have the Hive Connector package, which is part of the Secure Agent installation.

Note: Hortonworks HDP packages is not applicable for mappings that run on the advanced cluster.

1. If the cluster is Kerberos enabled, perform the following tasks:
    a. Make an entry of the Key Distribution Center (KDC) in the `/etc/hosts` file.
    b. Copy the `/etc/krb5.conf` file from the cluster node where the Hive service runs to the following directory on the Secure Agent machine:

    ```
    <Secure Agent installation directory>/apps/jdk/zulu<latest_version>/jre/lib/security
    ```

    c. Import the keytab file to the agent location.
    d. Copy the site.xml files from the Hadoop cluster and add them to a folder in the agent machine.

    After you copy the Hadoop configuration file in the agent machine, specify the path in the **Configurations File Path** field in the Hive connection properties.
2. If the cluster is TLS enabled, perform the following tasks:
    a. Import the certificate alias file to the following location:

    ```
    <Secure Agent installation directory>/apps/jdk/zulu<latest_version>/jre/lib/security/
    cacerts
    ```

    b. Import the truststore file to the Secure Agent location.

    Specify the truststore file and password in the **JDBC URL** field in the Hive connection properties.
3. Restart the Secure Agent.

Running a mapping on Azure HDInsights Kerberos cluster with WASB storage

To read and process data from sources that use a Kerberos-enabled environment, you must configure the Kerberos configuration file, create user authentication artifacts, and configure Kerberos authentication properties for the Informatica domain.

To run a mapping for Hive Connector using the Azure HDInsights with Windows Azure Storage Blob (WASB) kerberos cluster, perform the following steps:

1. Go to the `/usr/lib/python2.7/dist-packages/hdinsight_common/` directory on the Hadoop cluster node.
2. Run the following command to decrypt the account key:
   /decrypt.sh ENCRYPTED ACCOUNT KEY
3. Edit the `core-site.xml` file, in Agent conf location.
4. Replace the encrypted account key provided in the **fs.azure.account.key.STORAGE_ACCOUNT_NAME.blob.core.windows.net** property with the decrypted key, received as the output of the step #2.
5. Comment out the following properties to disable encryption and decryption of the account key:
    - **fs.azure.account.keyprovider.STORAGE_ACCOUNT_NAME.blob.core.windows.net**
    - **fs.azure.shellkeyprovider.script**
6. Save the `core-site.xml` file.
7. Copy the hdinsight_common folder from `/usr/lib/python2.7/dist-packages/hdinsight_common/` to the Secure Agent location.

8. Open the `core-site.xml` file in a browser to verify if the xml tags appear and ensure that there are no syntax issues.
9. Restart the Secure Agent.

Note: Azure HDInsights Kerberos cluster with WASB storage is not applicable for mappings that run on the advanced cluster.

Prerequisites for mappings in advanced mode

Before you use Hive Connector in a mapping in advanced mode, you must also download the Hive JDBC library and configuration files for the Hadoop cluster that you want to access.

Download the Hive JDBC libraries for mappings in advanced mode

Before you can configure mappings in advanced mode, perform the following tasks:

1. Download the Hive JDBC libraries from the cluster that you want to access.
2. Place the jar in the following Secure Agent location:

   ```
   <Secure Agent installation directory>/ext/connectors/thirdparty/
   informaticallc.hiveadapter/spark/lib
   ```

   Note: If the file path structure `informaticallc.hiveadapter/spark/lib` is not created, you must first create the folder structure, and then place the jars.

Download the configuration files from the Hadoop cluster

Before you configure a Hive connection for a mapping, download the client configuration files from the cluster distribution to the Secure Agent location.

1. Log in to the Hadoop cluster node.
2. Copy the client configuration files from the following directories on the Hadoop cluster node and paste them in a location in the Secure Agent machine:

   `/etc/hive/conf`

   `/etc/hadoop/conf`

   - To use a Hive connection in mappings to access Hive sources on Hadoop clusters, you require the following client configuration files:

     `core-site.xml`

     `hdfs-site.xml`

     `hive-site.xml`

   - To use a Hive connection for mappings in advanced mode, you require the following client configuration files:

```
core-site.xml

hdfs-site.xml

hive-site.xml

mapred-site.xml

yarn-site.xml
```

Later, when you create a Hive connection, specify the path of the configuration files in the **Configuration Files Path** field.

Configure Hive Connector to download the distribution-specific Hive libraries

You must configure Hive Connector to download the distribution specific Hive third-party libraries. The **Informatica Hive third-party script** and the **Informatica Hive third-party properties** files are available as part of the Hive Connector package in the Secure Agent installation.

Distributions applicable for Hive mappings

You can utilize the following distribution versions when you use Hive Connector to run mappings:

- Cloudera CDH 6.1
- Amazon EMR 5.20, 6.3, and 6.4
- Cloudera CDP 7.1 private cloud and Cloudera CDW 7.2 public cloud
- Azure HDInsight 4.0
- Hortonworks HDP 3.1

Distributions applicable for Hive mappings in advanced mode

You can utilize the following distribution versions when Hive Connector runs on the advanced cluster:

- Cloudera CDH 6.1
- Cloudera CDP 7.1 private cloud and Cloudera CDW 7.2 public cloud
- Azure HDInsight 4.0
- Amazon EMR 6.1, 6.2, and 6.3

Perform the following tasks to download distribution specific Hive third-party libraries before you use Hive Connector:

1. Run the script to copy the third-party libraries to the Secure Agent location. Ensure that you have full permissions to the directories where the Hive libraries are copied.
   The script is interactive and you need to specify the job type and the Hadoop cluster you want to use, when prompted.
2. Add the runtime DTM property, **INFA_HADOOP_DISTRO_NAME**, and set its value to the applicable distribution that you want to use.
3. Restart the Secure Agent.

Step 1. Run the script on a Linux system

The Hive Connector package that contains the Informatica Hive third-party script and the Informatica Hive third-party property files is part of the Secure Agent installation. When you run the Hive third-party script, you can specify the distribution that you want to use.

1. Go to the following Secure Agent installation directory where the Informatica Hive third-party script is located:

   `<Secure Agent installation directory>/downloads/package-hiveadapter.<version>/package/hive/thirdparty/informatica.hiveadapter/scripts/`

2. Copy the `scripts` folder outside the Secure Agent installation directory.

   You can do this if the Secure Agent does not have internet access to download the third-party libraries or due to other network restrictions.

3. If you want to run the script in the same machine where the Secure Agent is installed, perform the following tasks:

   a. From the terminal, run the following command from the `scripts` folder: `sh downloadHiveLibs.sh`

   b. When prompted, select Data Integration or CDI Advanced Mode for which you want to run the script.

      - Enter 1 to select **CDI**.
      - Enter 2 to select **CDI Advanced Mode**.

   c. When prompted, specify the value of the Hadoop distribution that you want to use.

      The third-party libraries are copied to the following directory based on the option you selected in step 3b:

      - For **CDI**:
        `<Secure Agent installation directory>/apps/Data_Integration_Server/ext/deploy_to_main/distros/Parsers/<Hadoop distribution version>/lib`
      - For **CDI Advanced Mode**:
        `<Secure Agent installation directory>/ext/connectors/thirdparty/informaticallc.hiveadapter/spark/lib`

        `<Secure Agent installation directory>/apps/Data_Integration_Server/ext/deploy_to_main/distros/Parsers/<Hadoop distribution version>/lib`

      where the value of the Hadoop distribution version is based on the Hadoop distribution you specified.

4. If you copy the `scripts` folder to a machine where the Secure Agent is not installed, perform the following tasks:

   a. Perform steps 3a and 3b.

      The third-party libraries are copied to the following directories based on the option you selected in step 3b:

      - **For CDI:**
        `<CurrentDirectory>/deploy_to_main/distros/Parsers/<Hadoop distribution version>/lib`

        Manually copy the `deploy_to_main` directory to the following Secure Agent location: `<Secure Agent installation directory>/apps/Data_Integration_Server/ext`, or replace the directory if it is already present.

- For **CDI Advanced Mode**: `<CurrentDirectory>/informaticallc.hiveadapter/spark/lib`
  Manually perform the following tasks:

  Copy the `informaticallc.hiveadapter` directory to the following Secure Agent location: `<Secure Agent installation directory>/ext/connectors/thirdparty/`

  Copy the `deploy_to_main` directory to the following Secure Agent location: `<Secure Agent installation directory>/apps/Data_Integration_Server/ext`, or replace the directory if it is already present.

where the value of the Hadoop distribution version is based on the Hadoop distribution you specified.

Note: CDH_6.1 option is applicable for Cloudera CDH 6.1, Cloudera CDP 7.1 private cloud, and Cloudera CDW 7.2 public cloud in mappings. For mappings in advanced mode, CDH_6.1 is applicable only for Cloudera CDH 6.1. EMR_5.20 is applicable for EMR_6.1, EMR_6.2, and EMR_6.3 for Hive mappings in advanced mode, whereas EMR_5.20 is applicable only for Amazon EMR 5.20, EMR 6.3, and EMR 6.4 in mappings.

The Hadoop distribution directory created under `deploy_to_main/distros/Parsers/` changes based on the distribution you select:

- If you select CDH_6.1, CDP_7.1, or CDW_7.2, the Hadoop distribution directory created is CDH_6.1.
- If you select EMR_5.20, EMR_6.1, EMR_6.2, or EMR_6.3, the Hadoop distribution directory created is EMR_5.20.
- If you select HDInsight_4.0, the Hadoop distribution directory created is HDInsight_4.0.
- If you select HDP_3.1, the Hadoop distribution directory created is HDP_3.1.

Step 2. Set the custom property for the Data Integration Service

Set the INFA_HADOOP_DISTRO_NAME property for the DTM in the Secure Agent properties and set the value of the distribution version that you want to use.

1. Open Administrator and select **Runtime Environments**.
2. Select the Secure Agent for which you want to configure the DTM property.
3. On the upper-right corner of the page, click **Edit**.
4. Add the following DTM properties in the **Custom Configuration** section:
   - Service: Data Integration Service
   - Type: DTM
   - Name: INFA_HADOOP_DISTRO_NAME
   - Value: <distribution_version>
     where the following values are applicable based on the distribution version you want to access:

     For CDH_6.1, CDP_7.1, and CDW_7.2, set the value as CDH_6.1.

     For EMR_5.20, EMR_6.1, EMR_6.2, EMR_6.3, and EMR_6.4, set the value as EMR_5.20.

     For HDInsight_4.0, set the value as HDInsight_4.0.

     For HDP_3.1, set the value as HDP_3.1.

Step 3. Restart the Secure Agent

After you complete the configurations and set the properties, restart the Secure Agent to reflect the changes.

Configure Hive Connector for Cloudera CDP 7.1 private cloud and Cloudera CDW 7.2

If you use the Cloudera CDP 7.1 private cloud and Cloudera CDW 7.2 distributions, you need to comment out the following S3A delegation token property, if it is available, in the following `<Configuration files path>/core-site.xml` file on the Secure Agent machine:

```
<property>
    <name>fs.s3a.delegation.token.binding</name>
   <value>org.apache.knox.gateway.cloud.idbroker.s3a.IDBDelegationTokenBinding</value>
  </property>
```

IAM authentication

To access the file system for staging data on Amazon S3, you can either specify the access key, secret key, and the Amazon S3 property name, each separated by a semicolon in the additional properties in the Hive connection, or you can use IAM authentication.

You can configure IAM authentication for the Secure Agent that runs on an Amazon Elastic Compute Cloud (EC2) system for secure and controlled access to Amazon S3 resources.

When you configure an IAM role, Hive Connector by default uses the IAM role to access the staging directory on Amazon S3.

Configure IAM authentication

Before you connect to Hive using IAM authentication, you must configure IAM authentication on EC2. You can use Hive connections configured for IAM authentication both in mappings and mappings in advanced mode.

1. Create an IAM role. For more information about creating the IAM role, see the AWS documentation.
2. After you create the IAM role, assign the following policy to the IAM role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "John": "VisualEditor2",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetEncryptionConfiguration",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObjectAcl",
        "s3:GetObject",
        "s3:PutObjectAcl",
        "s3:DeleteObject",
        "s3:Delete*",
        "s3:Put*",
        "s3:ListBucketMultipartUploads",
        "s3:AbortMultipartUpload"
      ],
      "Resource": [
        "arn:aws:s3:::<hive-staging-bucket-name>/*",
        "arn:aws:s3:::<hive-staging-bucket-name>"
```

```
                ]
            }
        ]
    }
```

3. Create an EC2 instance. Assign the IAM role that you created in step 2 to the EC2 instance.
4. Install the Secure Agent on the EC2 system.

Configure IAM authentication for a Hive connection to run mappings in advanced mode

Before you connect to Hive using IAM authentication from a mapping in advanced mode, you must have the following IAM roles to manage an advanced cluster:

- kops role
- Secure Agent role
- master role
- worker role

Note: Kops role is the default. Master and worker roles are not mandatory if you want to work only with the Kops role.

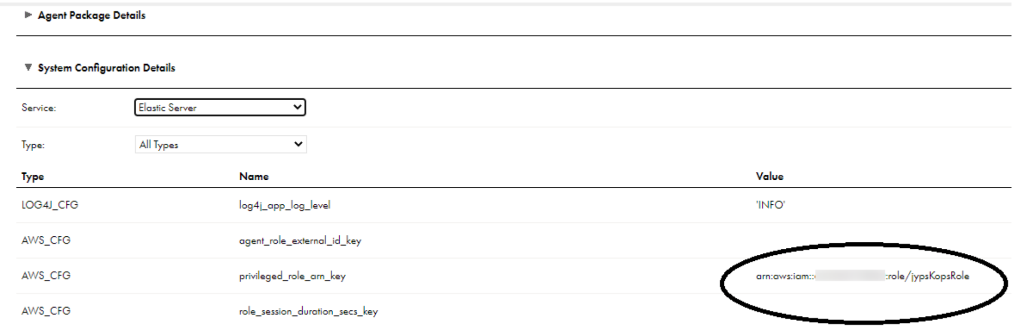
For more information about these roles, see the *Advanced Cluster* help.

To use IAM authentication for a Hive Connector to run mappings in advanced mode, perform the following tasks for the advanced cluster:

1. Create IAM roles. For more information, see the topic "Create IAM roles" in the see the Advanced Cluster help.
2. After you create the IAM role, you must assign the following policy to the IAM role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "John": "VisualEditor2",
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation",
        "s3:GetEncryptionConfiguration",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObjectAcl",
        "s3:GetObject",
        "s3:PutObjectAcl",
        "s3:DeleteObject",
        "s3:Delete*",
        "s3:Put*",
        "s3:ListBucketMultipartUploads",
        "s3:AbortMultipartUpload"
      ],
      "Resource": [
        "arn:aws:s3:::<hive-staging-bucket-name>/*",
        "arn:aws:s3:::<hive-staging-bucket-name>"
      ]
    }
  ]
}
```

3. Log in to Informatica Intelligent Cloud Services and from Administrator, navigate to the **System Configuration Details**, select **Elastic Server** as the service, add the ARN of the kops role created:



▶ Agent Package Details

▼ System Configuration Details

Service: Elastic Server

Type: All Types

Type	Name	Value
LOG4J_CFG	log4j_app_log_level	'INFO'
AWS_CFG	agent_role_external_id_key	
AWS_CFG	privileged_role_arn_key	arn:aws:iam::[redacted]:role/jypsKopsRole
AWS_CFG	role_session_duration_secs_key	

Managed identity

To stage data on Azure when mappings are in advanced mode, you can either specify the required details in the additional properties in the Hive connection or you can configure managed identity for the Secure Agent to access Microsoft Azure storage account

The Secure Agent uses a managed identity to log in to the Microsoft Azure cloud and to create an advanced cluster. When you configure managed identity, the agent, by default, uses the managed identity to access the staging directory on the Azure storage account.

To use managed identity to stage data in Azure, complete the following tasks:

1. Create a managed identity.
   For instructions about creating a managed identity, refer to the Microsoft Azure documentation.
2. Create an agent role to define the permissions for the managed identity.
3. Add role assignments to assign the agent role to the managed identity and to assign the managed identity to the Secure Agent machine.
   For more information, see the "Create a managed identity for the Secure Agent" topic in the *Advanced Clusters* help.

Using Azure Data Lake Storage Gen2 as the staging directory on the Azure HDInsights cluster

To connect to Hive on the Azure HDInsights cluster and read and write data successfully using a mapping in advanced mode, you must use Azure Data Lake Storage Gen2 as the staging location. You can authenticate as a service principal to connect to Hive on the Azure HDInsights cluster.

To use Azure Data Lake Storage Gen2 as storage on the Azure HDInsights cluster, perform the following tasks:

1. In the downloaded `core-site.xml` file on the agent machine, comment out the following property:

```
<!--
<property>
```

```
<name>fs.azure.account.auth.type</name>
<value>Custom</value>
</property>
-->
<!--
<property>
<name>fs.azure.account.oauth.provider.type</name>
<value>com.microsoft.azure.storage.oauth2.TokenServiceBasedAccessTokenProvider</
value>
</property>
-->
<!--
<property>
<name>fs.azure.enable.delegation.token</name>
<value>true</value>
</property>
-->
```

2. Add the following properties:

```
<property>
<name>fs.azure.account.auth.type</name>
<value>OAuth</value>
</property>
<property>
<name>fs.azure.account.oauth.provider.type</name>
<value>org.apache.hadoop.fs.azurebfs.oauth2.ClientCredsTokenProvider</value>
</property>
<property>
<name>fs.azure.account.oauth2.client.id</name>
<value><clientId></value>
</property>
<property>
<name>fs.azure.account.oauth2.client.secret</name>
<value><SecretKey></value>
</property>
<property>
<name>fs.azure.account.oauth2.client.endpoint</name>
<value>https://login.microsoftonline.com/<tenantId>/oauth2/token</value>
</property>
```

3. In the Hive connection properties in Data Integration, specify the following properties in the **Additional Properties** field:

```
fs.azure.account.auth.type=OAuth;
fs.azure.account.oauth.provider.type=org.apache.hadoop.fs.azurebfs.oauth2.ClientCreds
TokenProvider;
fs.azure.account.oauth2.client.id=<clientId>;fs.azure.account.oauth2.client.secret=<S
ecretKey>;
fs.azure.account.oauth2.client.endpoint=https://login.microsoftonline.com/<tenantId>/
oauth2/token
```

4. Perform any one of the following steps to generate the OAuth token to authenticate using a service principal:

   Note: The driver retrieves the OAuth access token from the credential service before making any storage requests.

   - Log in to the Ambari Web application with the Azure Active Directory user credentials.
   - Run the following script on the Hadoop cluster node with the Azure Active Directory user credentials: `/usr/lib/hdinsight-common/scripts/RegisterKerbTicketAndOAuth.sh`
     See the following sample command: `/usr/lib/hdinsight-common/scripts/RegisterKerbTicketAndOAuth.sh <AAD Username@REALM>`

Rules and guidelines for Hadoop distributions

Consider the following rules and guidelines in mappings in advanced mode for Hadoop distributions:

- When you configure a Target transformation to write to Hive on the Cloudera CDH 6.1 distribution, operations such as delete, update, upsert, and data driven are not applicable. You can use only the insert operation.
- Mappings that run on the advanced cluster to access Hive on the Amazon EMR, Azure HDI, or Cloudera CDH 6.1 distributions cannot use the HDFS staging directory.
- To run mappings on the advanced cluster to read from or write data to Hive on the Cloudera 6.1 distribution, you must get the `hive-exec-2.1.1-cdh6.1.0.jar` file from Cloudera.
  For more information about the steps to run the script and use the Cloudera 6.1 distribution, see "Configure Hive Connector to download the distribution-specific Hive libraries" on page 10.

  After you download the required Cloudera CDH 6.1 jars, add `hive-exec-2.1.1-cdh6.1.0.jar` to the same directory as the downloaded files.

  For example, copy the jar to the following directory: `<Secure Agent installation directory>/apps/Data_Integration_Server/ext/deploy_to_main/distros/Parsers/ CDH_6.1/`
- Do not use the Amazon S3 staging directory for mappings that run on an advanced cluster to access Hive on the Azure HDI distribution. The test connection fails with the following error:
  `java.lang.reflect.InvocationTargetException`
  You must instead use Azure Data Lake Storage Gen2 as the staging location for Azure HDI.

CHAPTER 2

Hive connections

Create a Hive connection to access Hive data.

You create a Hive connection on the **Connections** page. When you create the Hive connection, enter the connection attributes that are specific to the Hive product that you want to connect to.

Note: You cannot use the serverless runtime environment for a Hive connection.

Hive connection properties

To use Hive Connector in a mapping task, you must create a connection in Data Integration.

When you set up a Hive connection, you must configure the connection properties.

The following table describes the Hive connection properties:

Connection property	Description
Authentication Type	You can select one of the following authentication types: - Kerberos. Select **Kerberos** for a Kerberos cluster. - LDAP. Select **LDAP** for an LDAP-enabled cluster. **Note:** LDAP is not applicable to mappings in advanced mode. - None. Select **None** for a Hadoop cluster that is not secure or not LDAP-enabled.
JDBC URL *	The JDBC URL to connect to Hive. Specify the following format based on your requirement: - To view and import tables from a single database, use the following format: `jdbc:hive2://<host>:<port>/<database name>` - To view and import tables from multiple databases, do not enter the database name. Use the following JDBC URL format: `jdbc:hive2://<host>:<port>/` **Note:** After the port number, enter a slash. - To access Hive on a Hadoop cluster enabled for TLS, specify the details in the JDBC URL in the following format: `jdbc:hive2://<host>:<port>/<database name>;ssl=true;sslTrustStore=<TrustStore_path>;trustStorePassword=<TrustStore_password>`, where the truststore path is the directory path of the truststore file that contains the TLS certificate on the agent machine.
JDBC Driver *	The JDBC driver class to connect to Hive.
Username	The user name to connect to Hive in LDAP or None mode.

Connection property	**Description**
Password	The password to connect to Hive in LDAP or None mode.
Principal Name	The principal name to connect to Hive through Kerberos authentication.
Impersonation Name	The user name of the user that the Secure Agent impersonates to run mappings on a Hadoop cluster. You can configure user impersonation to enable different users to run mappings or connect to Hive. The impersonation name is required for the Hadoop connection if the Hadoop cluster uses Kerberos authentication.
Keytab Location	The path and file name to the Keytab file for Kerberos login.
Configuration Files Path *	The directory that contains the Hadoop configuration files for the client. Copy the site.xml files from the Hadoop cluster and add them to a folder in the Linux box. Specify the path in this field before you use the connection in a mapping to access Hive on a Hadoop cluster: - For mappings, you require the core-site.xml, hdfs-site.xml, and hive-site.xml files. - For mappings in advanced mode, you require the core-site.xml, hdfs-site.xml, hive-site.xml, mapred-site.xml, and yarn-site.xml files.
DFS URI *	The URI to access the Distributed File System (DFS), such as Amazon S3, Microsoft Azure Data Lake Storage, and HDFS. **Note:** For mappings in advanced mode that run on the advanced cluster, Azure Data Lake Storage Gen2 is supported on the Azure HDinsight cluster. Based on the DFS you want to access, specify the required storage and bucket name. For example, for HDFS, refer to the value of the **fs.defaultFS** property in the **core-site.xml** file of the Hadoop cluster and enter the same value in the **DFS URI** field.
DFS Staging Directory	The staging directory in the Hadoop cluster where the Secure Agent stages the data. You must have full permissions for the DFS staging directory. Specify a transparent encrypted folder as the staging directory.
Hive Staging Database	The Hive database where external or temporary tables are created. You must have full permissions for the Hive staging database.

Connection property	Description

Additional Properties	Applies to mappings in advanced mode.
	The additional properties required to access the DFS.
	Configure the property as follows:
	<DFS property name>=<value>;<DFS property name>=<value>
	For example:
	To access the Amazon S3 file system, specify the access key, secret key, and the Amazon S3 property name, each separated by a semicolon:
	`fs.s3a.<bucket_name>.access.key=<access key value>;`
	`fs.s3a.<bucket_name>.secret.key=<secret key value>;`
	`fs.s3a.impl=org.apache.hadoop.fs.s3a.S3AFileSystem;`
	To access the Azure Data Lake Storage Gen2 file system, specify the authentication type, authentication provider, client ID, client secret, and the client endpoint, each separated with a semicolon:
	`fs.azure.account.auth.type=<Authentication type>;`
	`fs.azure.account.oauth.provider.type=<Authentication_provider>;`
	`fs.azure.account.oauth2.client.id=<Client_ID>;`
	`fs.azure.account.oauth2.client.secret=<Client-secret>;`
	`fs.azure.account.oauth2.client.endpoint=<ADLS Gen2 endpoint>`

* These fields are mandatory parameters.	

Accessing multiple storage systems

Create an Hive connection to read data from or write data to Hive.

Use the DFS URI property in the connection parameters to connect to various storage systems. The following table lists the storage system and the DFS URI format for the storage system:

Storage	DFS URI Format
HDFS	`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). `hdfs://<nameservice>` in case of NameNode high availability.
WASB in HDInsight	`wasb://<container_name>@<account_name>.blob.core.windows.net/<path>` where: - `<container_name>` identifies a specific Azure Storage Blob container. **Note:** `<container_name>` is optional. - `<account_name>` identifies the Azure Storage Blob object. **Note:** Not applicable for a Hive connection in mappings that run on the advanced cluster.

Storage	DFS URI Format
Amazon S3	`s3a://home`
Azure Data Lake Gen2 in HDInsight	`abfss://<container name>@<storage name>.dfs.core.windows.net` where: - `<container_name>` identifies a specific Azure Data Lake Gen2 container. - `<storage name>` identifies the Azure Data Lake Gen2 storage account name. **Note:** Applicable to a Hive connection used in mappings configured in advanced mode.

JDBC URL format

Hive Connector connects to the HiveServer2 component of Hadoop with JDBC.

Hive Connector uses the following JDBC URL format:

```
jdbc:hive2://<host>:<port>/<database name>
```

The following parameters describe the JDBC URL format:

- hive2. Contains the protocol information. The version of the Thrift Server, that is, hive2 for HiveServer2.
- host, port. The host and port information of the Thrift Server.
- database name. The database that the Connector needs to access.

For example, `jdbc:hive2://invrlx63iso7:10000/default` connects to the database of Hive and uses a Hive Thrift server HiveServer2 that runs on the server invrlx63iso7 on port 10000.

Hive Connector uses the Hive thrift server to communicate with Hive.

Access the DFS for Hive on Cloudera Data Platform and Cloudera Data Warehouse using mappings

To access the DFS file system for Hive endpoints on the Cloudera Data Platform and Cloudera Data Warehouse using Hive mappings, you can configure the required DFS properties in the `core-site.xml` file.

To access the Amazon S3 file system, specify the access key, secret key, and the Amazon S3 property name in the `core-site.xml` file:

```
fs.s3a.<bucket_name>.access.key=<access key value>
fs.s3a.<bucket_name>.secret.key=<secret key value>
fs.s3a.impl=org.apache.hadoop.fs.s3a.S3AFileSystem
```

To access the Azure Data Lake Storage Gen2 file system, specify the authentication type, authentication provider, client ID, client secret, and the client endpoint in the `core-site.xml` file:

```
fs.azure.account.auth.type=<Authentication type>
fs.azure.account.oauth.provider.type=<Authentication_provider>
fs.azure.account.oauth2.client.id=<Client_ID>
fs.azure.account.oauth2.client.secret=<Client-secret>
fs.azure.account.oauth2.client.endpoint=<ADLS Gen2 endpoint>
```

Creating a Hive connection

To use Hive Connector in a mapping task, you must create a connection in Data Integration.

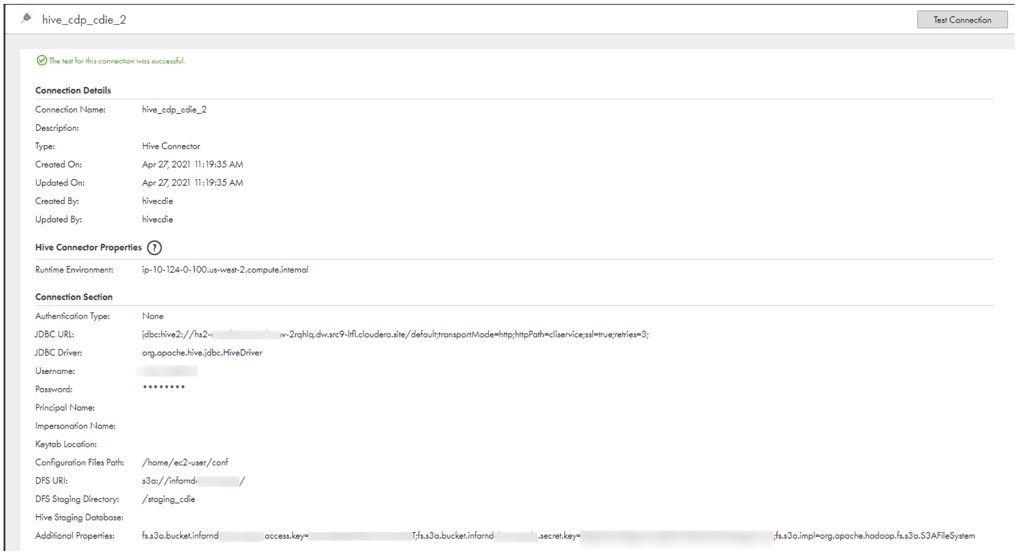
Perform the following steps to create a Hive connection in Data Integration:

1. On the **Connections** page, click **New Connection**.

   The **New Connection** page appears.

2. On the **New Connections** page, configure the required connection properties.
3. Click **Test Connection** to evaluate the connection.

   The following image shows the connection page details after the test is successful:

   hive_cdp_cdie_2

   ✓ The test for this connection was successful.

   Connection Details

Connection Name:	hive_cdp_cdie_2
Description:	
Type:	Hive Connector
Created On:	Apr 27, 2021 11:19:35 AM
Updated On:	Apr 27, 2021 11:19:35 AM
Created By:	hivecdie
Updated By:	hivecdie

   Hive Connector Properties ?

Runtime Environment:	ip-10-124-0-100.us-west-2.compute.internal

   Connection Section

Authentication Type:	None
JDBC URL:	jdbc:hive2://hs2-[redacted]v-2rqhlq.dw.src9-ltfl.cloudera.site/default;transportMode=http;httpPath=cliservice;ssl=true;retries=3;
JDBC Driver:	org.apache.hive.jdbc.HiveDriver
Username:	[redacted]
Password:	********
Principal Name:	
Impersonation Name:	
Keytab Location:	
Configuration Files Path:	/home/ec2-user/conf
DFS URI:	s3a://infarnd[redacted]/
DFS Staging Directory:	/staging_cdie
Hive Staging Database:	
Additional Properties:	fs.s3a.bucket.infarnd[redacted].access.key=[redacted]T;fs.s3a.bucket.infarnd[redacted].secret.key=[redacted];fs.s3a.impl=org.apache.hadoop.fs.s3a.S3AFileSystem

4. Click **Save** to save the connection.

CHAPTER 3

Mappings and mapping tasks with Hive Connector

Use the Data Integration Mapping Designer to create a mapping. When you create a mapping, you configure a source or target to represent a Hive object.

Describe the flow of data from source and target along with the required transformations before the agent writes data to the target.

Select the mapping that you want to use in a mapping task. Use the Mapping Task wizard to create a mapping task. The mapping task processes data based on the data flow logic you define in the mapping.

In advanced mode, the Mapping Designer updates the mapping canvas to include transformations and functions that enable advanced functionality.

You can use Hive Connector to perform the following tasks:

- Connect to Hive to perform the relevant data operations.
- Access Hive from Kerberos or non-Kerberos clusters.
- Use mappings or mappings in advanced mode to access Hive on an Amazon EMR 6.3 kerberized cluster where the Hive metastore is on AWS Glue.
- Use mappings in advanced mode to access Hive on an Amazon EMR 6.3 kerberized cluster where the Hive metastore is on MySQL.
- Use all the operators supported in HiveQL.
- Use the AND conjunction in filters. You can use both the AND and OR conjunctions in advanced filters.
- Apply filter on all filterable columns in Hive tables.
- You can configure only Informatica passthrough partitioning for Hive target objects.

Configure preSQL and postSQL commands

You can specify **preSQL** and **postSQL** advanced properties for Hive sources and targets.

You can perform the following operations by using preSQL and postSQL commands:

- INSERT
- ALTER
- TRUNCATE

- DROP

Hive variables are not honored through preSQL and postSQL commands.

Hive sources in mappings

In a mapping, you can configure a Source transformation to represent a single Hive source or multiple Hive sources.

The following table describes the Hive source properties that you can configure in a Source transformation in mappings:

Property	Description
Connection	Name of the source connection.
Source type	Type of the source object. Select one of the following types: - **Single Object**. Select to specify a single Hive source object. - **Multiple Objects**. Select to specify multiple Hive source objects. - **Parameter**. Specify a parameter where you define values that you want to update without having to edit the task.
Object	Source object for a single source.
Add Source Object	Primary source object for multiple sources.
Advanced Relationship	Defines the relationship between multiple source objects. You can use an equijoin between the objects.
Filter	Adds conditions to filter records. Configure a **Not Parameterized** or an **Advanced** filter.
Sort	Add conditions to sort records. You can specify from the following sort conditions: - **Not parameterized**. Select the fields and type of sorting to use. - **Parameterized**. Use a parameter to specify the sort option. - **Sort Order**. Sorts data in ascending or descending order, according to a specified sort condition.

The following table describes the Hive source advanced properties that you can configure in a Source transformation in mappings:

Property	Description
SQL Override	When you read data from a Hive source object, you can configure SQL overrides and define constraints.
PreSQL	SQL statement that you want to run before reading data from the source.
PostSQL	SQL statement that you want to run after reading data from the source.
Schema Override	Overrides the schema of the source object at runtime.

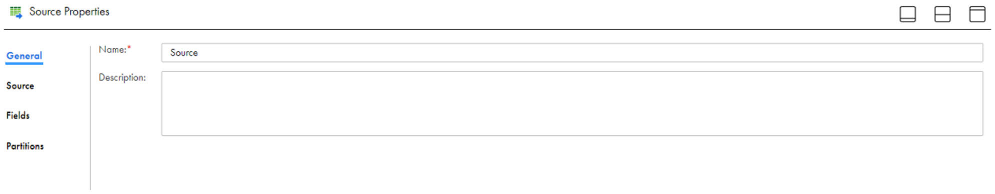
Property	Description
Table Override	Overrides the table of the source object at runtime.
Tracing Level	Amount of detail that appears in the log for the transformation. Use the following tracing levels: - Terse - Normal - Verbose Initialization - Verbose Default is normal.

Adding multiple Hive source objects

You need to create a connection before getting started with a mapping task.

The following steps help you to set up a mapping task in Data Integration:

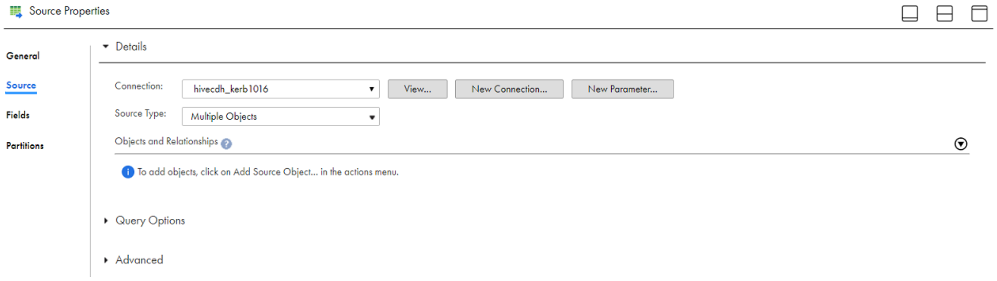
1. Click **New** > **Mappings**.
2. Select **Mapping** and click **Create**.
3. In the **Source Properties** page, specify the name and provide a description in the **General** tab.



Name:* Source

Description:

4. Click the **Source** tab.
5. Select the source connection and source type as **Multiple Objects** to be used for the task.

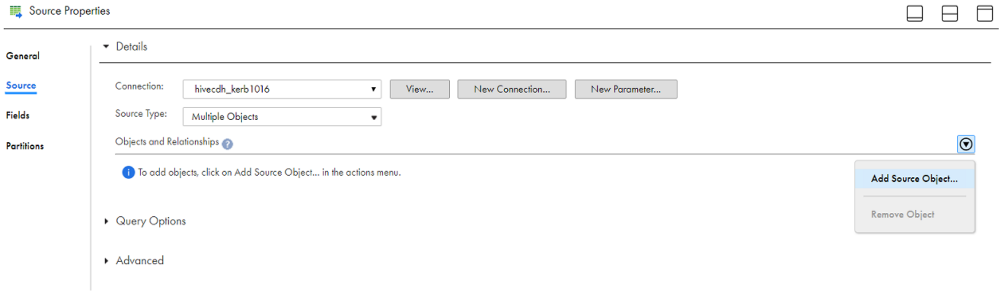


Connection: hivecdh_kerb1016

Source Type: Multiple Objects

To add objects, click on Add Source Object... in the actions menu.

6. In the **Objects and Relationships** section, click the arrow to open the **Action** menu and then select **Add Source Object**.

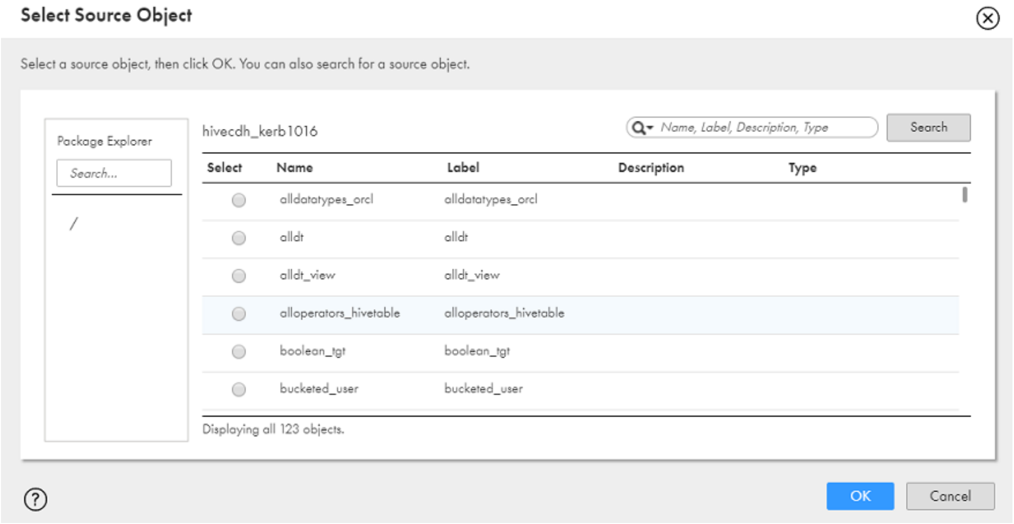


Connection: hivecdh_kerb1016

Source Type: Multiple Objects

To add objects, click on Add Source Object... in the actions menu.

7. Select a source object from the list and click **OK**.



Select Source Object

Select a source object, then click OK. You can also search for a source object.

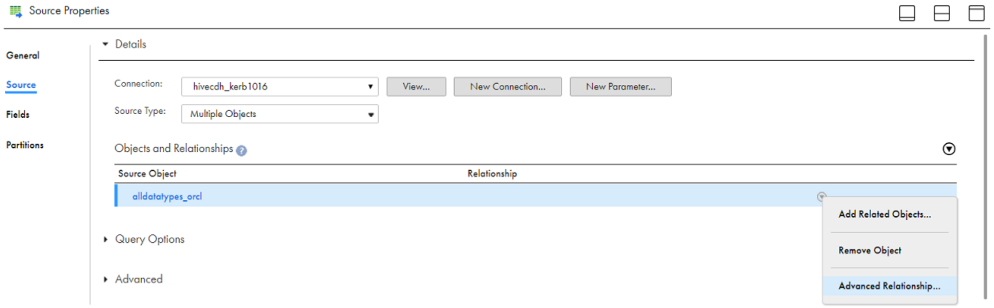
Package Explorer

hivecdh_kerb1016

Select	Name	Label	Description	Type
○	alldatatypes_orcl	alldatatypes_orcl		
○	alldt	alldt		
○	alldt_view	alldt_view		
○	alloperators_hivetable	alloperators_hivetable		
○	boolean_tgt	boolean_tgt		
○	bucketed_user	bucketed_user		

Displaying all 123 objects.

8. In the **Objects and Relationships** section, click the arrow next to the source object and then click **Advanced Relationship** to add related objects.



Source Properties

Details

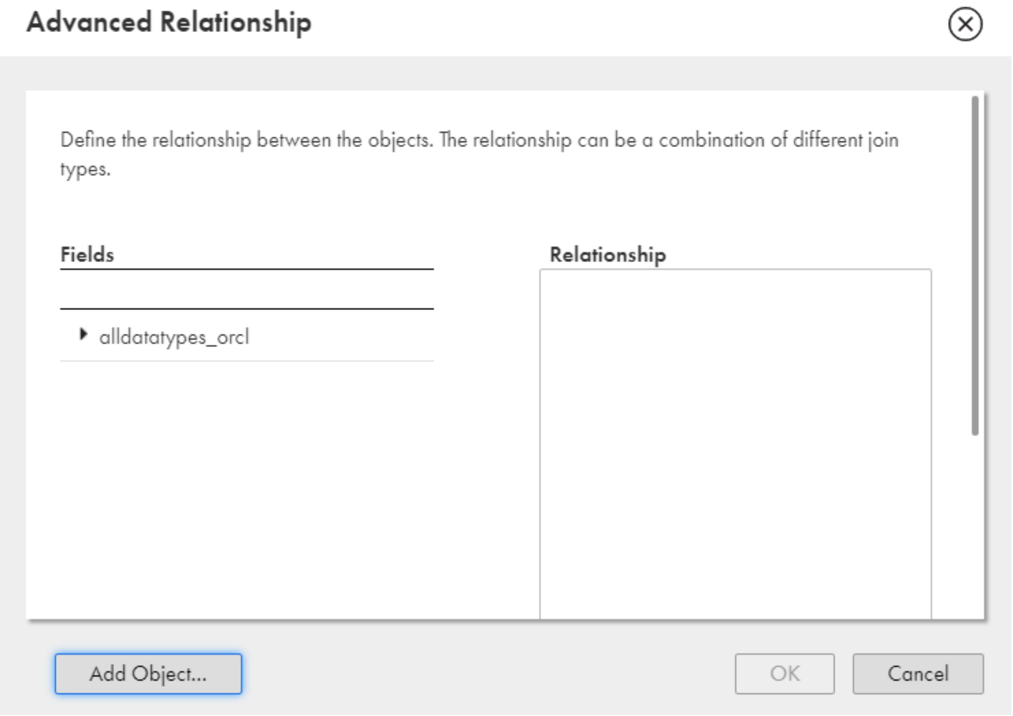
Connection: hivecdh_kerb1016

Source Type: Multiple Objects

Objects and Relationships

Source Object	Relationship
alldatatypes_orcl	

9. In the **Advanced Relationship** page, click **Add Object**.

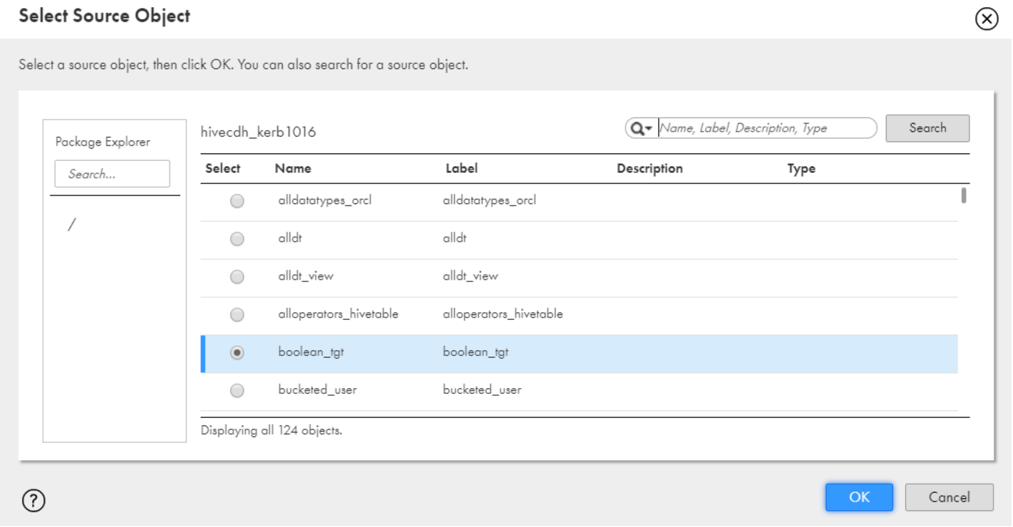


Advanced Relationship

Define the relationship between the objects. The relationship can be a combination of different join types.

The **Select Source Object** page appears.

10. Select an object from the list and click **OK**.



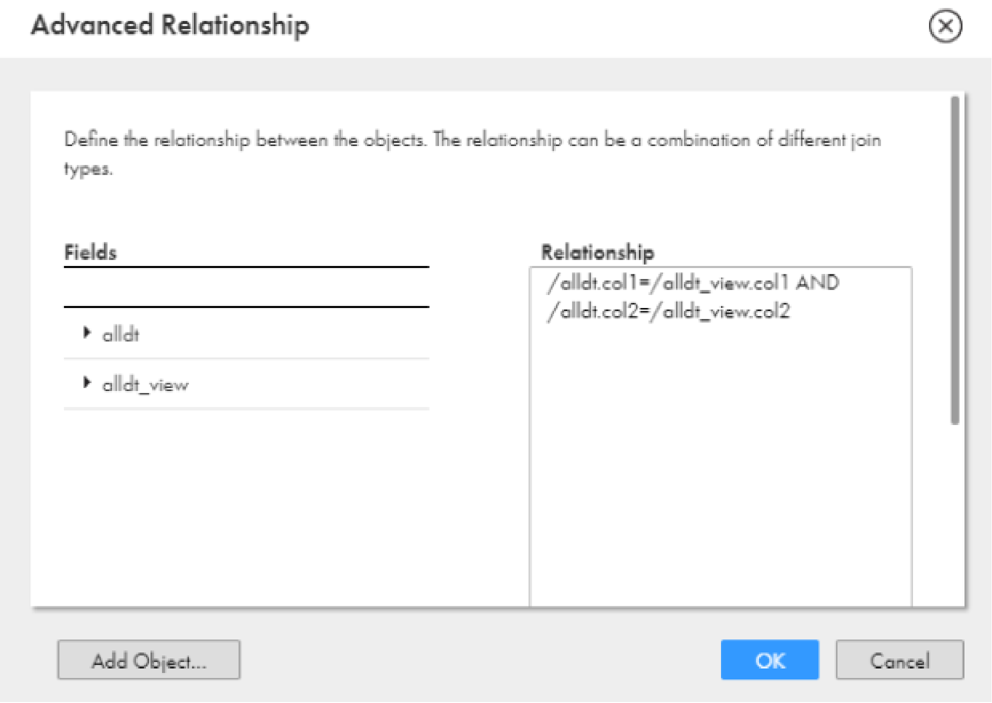
Select Source Object

Select a source object, then click OK. You can also search for a source object.

Select	Name	Label	Description	Type
○	alldatatypes_orcl	alldatatypes_orcl		
○	alldt	alldt		
○	alldt_view	alldt_view		
○	alloperators_hivetable	alloperators_hivetable		
●	boolean_tgt	boolean_tgt		
○	bucketed_user	bucketed_user		

Displaying all 124 objects.

11. In the **Advanced Relationship** page, select the required fields for the objects and establish an equijoin relationship between them.



Advanced Relationship

Define the relationship between the objects. The relationship can be a combination of different join types.

/alldt.col1=/alldt_view.col1 AND
/alldt.col2=/alldt_view.col2

12. Click **OK**.

Hive targets in mappings

In a mapping, you can configure a Target transformation to represent a Hive target object. You can use a mapping task to write data to Hive targets.

The following table describes the Hive target properties that you can configure in a Target transformation in mappings:

Property	Description
Connection	Name of the target connection.
Target type	Type of the target object. Select one of the following types: - **Single Object**. Select to specify a single Hive target object. - **Parameter**. Specify a parameter where you define values that you want to update without having to edit the task.
Object	Target object for a single target. You can select an existing target object or create a new target at runtime.

Property	Description
Operation	The target operation. You can choose from the following options: **Insert** Inserts data to a Hive target. **Upsert (Update or Insert)**[1] Performs an upsert operation to the Hive target. You must also select the **Update as Upsert** property for upsert to work. The Secure Agent performs the following tasks: - If the entries already exist in the Hive target, the Secure Agent updates the data. - If the entries do not exist in the Hive target, the Secure Agent inserts the data. **Update**[1] Updates data to the Hive target. **Delete**[1] Deletes data in the Hive target. **Data Driven**[1] Determines if the agent inserts, updates, or deletes records in the Hive target table based on the expression you specify. **Note:** Reject operation is ignored for the data driven operation type.
Update Columns[1]	Columns that identify rows in the target table to update or upsert data. Select the key columns where you want to upsert or update data in the Hive target table. Required if you select the **Upsert (Update or Insert)** option.
Data Driven Condition[1]	Enables you to define expressions that flag rows for an insert, update, or delete operation when you select the Data Driven operation type. **Note:** Reject operation is ignored for the data driven operation type.
[1] Applies only to mappings in advanced mode.	

The following table describes the properties that you can configure when you use the **Create New at Runtime** option in a Target transformation in mappings:

Property	Description
Object Name	The name for the target table.
External Table	The type of Hive tables such as managed or external to write the data. Select the check box if you want to create an external table. Clear the checkbox if you want to create a managed table.
Table Location	The path to the managed or external table in the Hive target to store the data. If you do not specify a path, Data Integration uses the default warehouse directory configured in the Hive server.
Number of Buckets	The number of buckets to create if the table contains bucket columns.

Property	Description
Stored As	The format to store the data in the table location you specify. You can choose from the following formats: - Avro - Orc - Parquet - RC file - Sequence file - Text file
Additional Table Properties	List of key-value comma-separated pairs of additional properties that you want to configure to create the target table. Enclose both the key and value within double quotes and specify the following format to include additional properties: `"<property name>"="<value>"` For example, you can configure additional properties such as compression formats or to include comments in the Hive target table by specifying the following properties: `"avro.compression"="BZIP2","orc.compress"="ZLIB", "comment"="table_comment"`
Path	The Hive target database name to write the data.

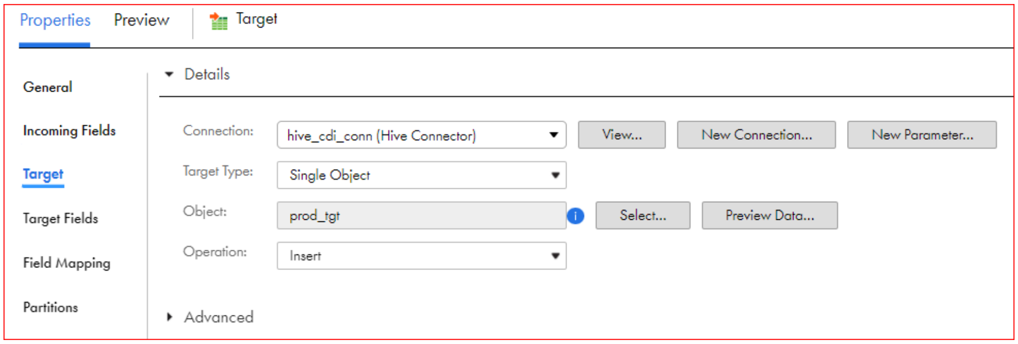
The following table describes the Hive target advanced properties that you can configure in a Target transformation in mappings:

Property	Description
Update as Upsert[1]	Upserts any records flagged for upsert. This property is required when you select the **Upsert (Update or Insert)** option and you want to upsert data. **Important:** When you select the Update operation and also provide the Update as Upsert flag, the agent supports the upsert operation, not the update operation.
Truncate Target	Truncates the database target table before inserting new rows. Select the **Truncate Target** check box to truncate the target table before inserting all rows. By default, the **Truncate Target** check box is not selected.
PreSQL	SQL statement that you want to run before writing data to the target.
PostSQL	SQL statement that you want to run after writing the data to the target.
Schema Override	Overrides the schema of the target object at runtime.
Table Override	Overrides the table of the target object at runtime.
Forward Rejected Rows	Determines whether the transformation passes rejected rows to the next transformation or drops rejected rows. By default, the mapping task forwards rejected rows to the next transformation. If you select the **Forward Rejected Rows** option, the Secure Agent flags the rows for reject and writes them to the reject file. If you do not select the **Forward Rejected Rows** option, the Secure Agent drops the rejected rows and writes them to the session log file. The Secure Agent does not write the rejected rows to the reject file.
[1]Applies only to mappings in advanced mode.	

Writing data to a Hive target

You can use an existing target or create a new target at runtime to write data to Hive.

1. Click **New** > **Mappings**, and then click **Create**.
2. Based on your requirement, you can click **Switch to Advanced**
3. In the **Target Properties** page, specify the name and provide a description in the **General** tab.
4. Click the **Target** tab.



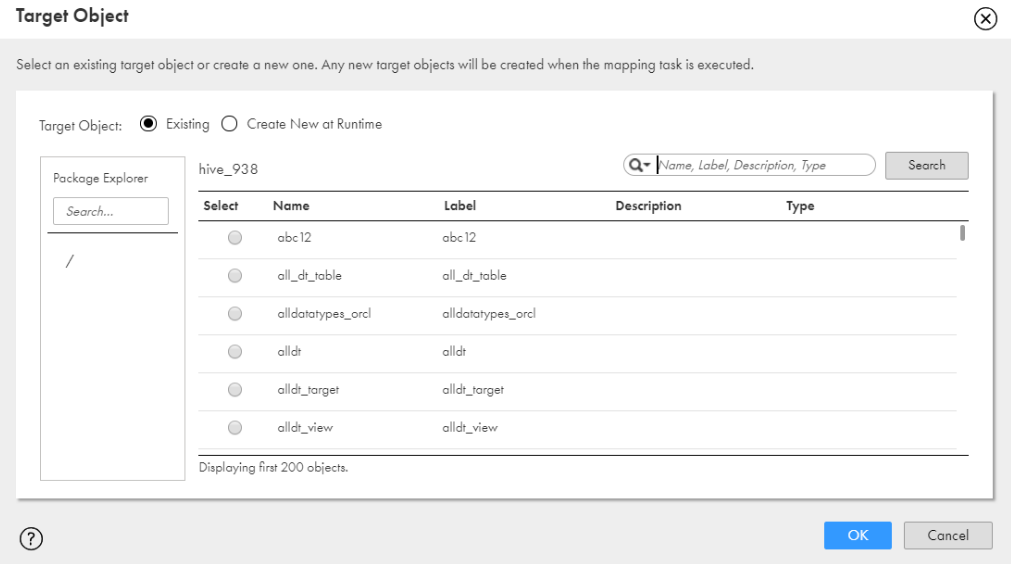
Connection: hive_cdi_conn (Hive Connector)

Target Type: Single Object

Object: prod_tgt

Operation: Insert

5. Select the target connection and the target type.
6. Click **Select** to select a target object.
7. Select an existing target object from the list and click **OK**.



Select an existing target object or create a new one. Any new target objects will be created when the mapping task is executed.

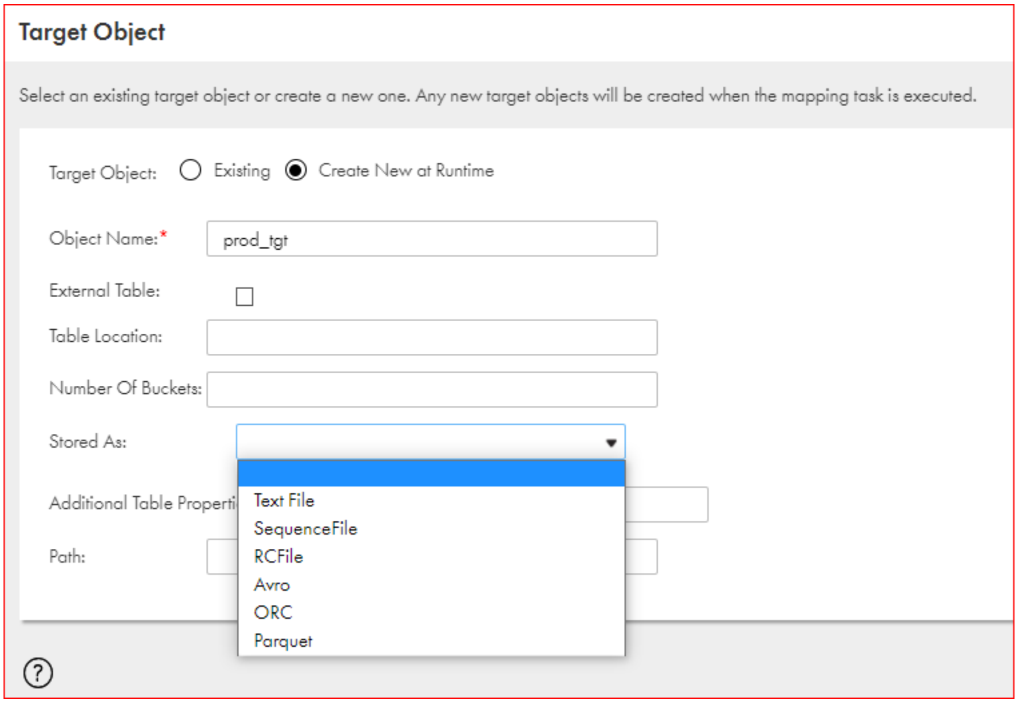
Target Object: ● Existing ○ Create New at Runtime

hive_938

Select	Name	Label	Description	Type
○	abc12	abc12		
○	all_dt_table	all_dt_table		
○	alldatatypes_orcl	alldatatypes_orcl		
○	alldt	alldt		
○	alldt_target	alldt_target		
○	alldt_view	alldt_view		

Displaying first 200 objects.

8. To create a new Hive target at runtime, select **Create New at Runtime**, and specify the required properties for the Hive target object.



Target Object

Select an existing target object or create a new one. Any new target objects will be created when the mapping task is executed.

Target Object: ○ Existing ● Create New at Runtime

Object Name:* prod_tgt

External Table: □

Table Location:

Number Of Buckets:

Stored As:

9. Select the target operation that you want to use.
10. Select the advanced properties that you want to configure for the target object.
11. Click **Save**.

Column partitioning for targets

You can organize tables or data sets into partitions to group the same type of data based on a column.

You can use an existing Hive target that has partitioned columns to write the data or you can configure partitions when you create a new Hive target at runtime.

When you create a new target at runtime, you can select the incoming columns that you want to add as partition columns. Include the partition columns from the list of fields that display in **Partition Fields** on the **Partitions** tab in the Target transformation.

When you select the columns as partitioned columns, the default data type for the columns is set to String. You cannot edit the data type of a partitioned column in the Hive target object. You can add, delete, and change the order of the partition fields, if required. You must not give the same partition order for multiple columns.

You can also create buckets to divide large data sets into more manageable parts. To configure buckets, you must first specify the number of buckets when you create a new target at runtime. After you specify the number, select the bucket fields, and then select the partitioning fields.

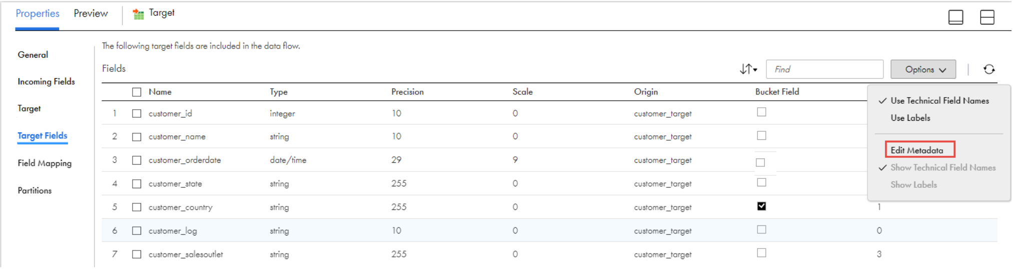
Data Integration creates all the fields that you select for partitioning in the target based on the partition order you specify. For example, you can create a table to write employee joining details categorized in the following hierarchical order, such as month, hour, year, and date.

Adding columns as partitions to the target

In this example, you want to create a table to write sales data from retail stores spread across Asia Pacific to a Hive table. You want to write data to partitions categorized based on country, state, and outlet names so that you can run queries easily across data split in partitions.

1. Configure a target transformation with the **Create New at Runtime** option.
2. To include a bucket, in the **Create New at Runtime** properties, specify the number of buckets you want in the target.

   Include a bucket if you want to split incoming data into a bucket.
3. On the **Target Fields** tab, edit the metadata, and then select the column that you want to include as a bucket field in the Hive target table.

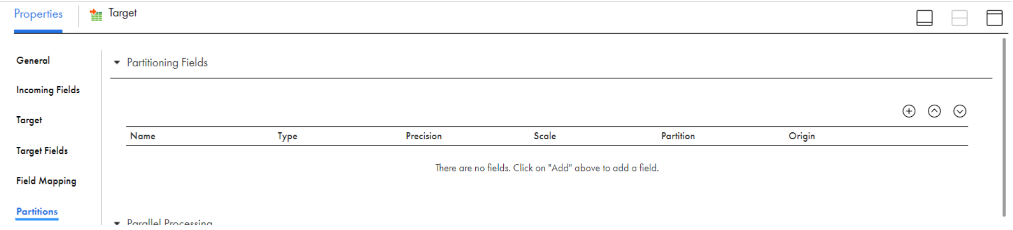


The following target fields are included in the data flow.

	□ Name	Type	Precision	Scale	Origin	Bucket Field	
1	□ customer_id	integer	10	0	customer_target	□	
2	□ customer_name	string	10	0	customer_target	□	
3	□ customer_orderdate	date/time	29	9	customer_target	□	
4	□ customer_state	string	255	0	customer_target	□	
5	□ customer_country	string	255	0	customer_target	☑	1
6	□ customer_log	string	10	0	customer_target	□	0
7	□ customer_salesoutlet	string	255	0	customer_target	□	3

   You cannot edit the metadata for a column that you select for partitioning. If you need to include buckets, you must select the column that you want as a bucket on the **Target Fields** tab, and then select those columns for partitioning on the **Partitions** tab.
4. Click the ⊕ icon in the **Partitions** tab to add the partition columns for a target.

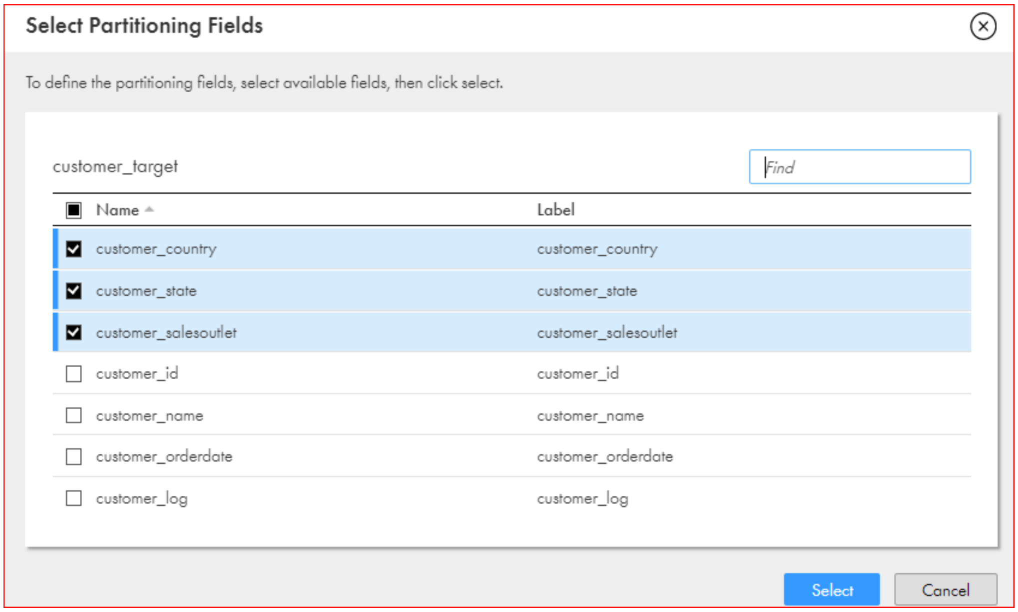
   The following image shows the **Partitions** tab where you can add the partition columns:



Name	Type	Precision	Scale	Partition	Origin

There are no fields. Click on "Add" above to add a field.

5. On the **Partitions** tab, select the required partitioning fields from the list of incoming fields from the source:



Select Partitioning Fields

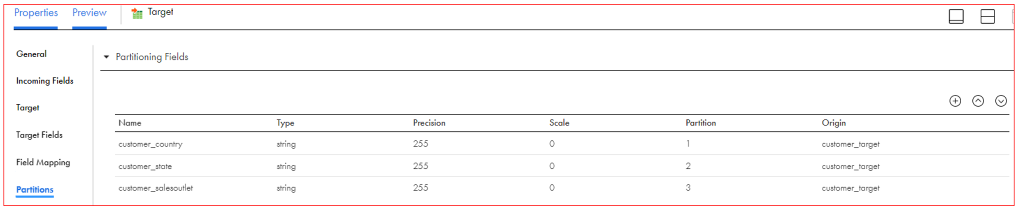
To define the partitioning fields, select available fields, then click select.

customer_target

■	Name	Label
☑	customer_country	customer_country
☑	customer_state	customer_state
☑	customer_salesoutlet	customer_salesoutlet
☐	customer_id	customer_id
☐	customer_name	customer_name
☐	customer_orderdate	customer_orderdate
☐	customer_log	customer_log

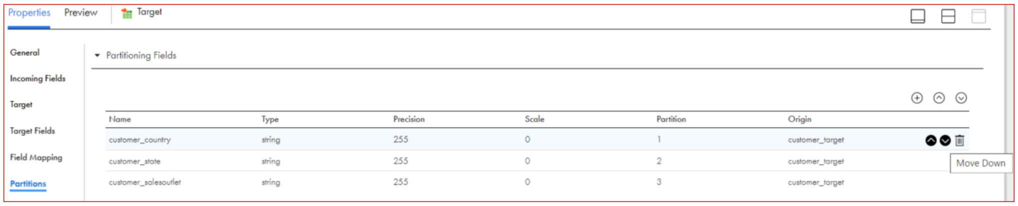
6. Click **Select**.

   The **Partitions** tab shows the partition columns that you selected:



Name	Type	Precision	Scale	Partition	Origin
customer_country	string	255	0	1	customer_target
customer_state	string	255	0	2	customer_target
customer_salesoutlet	string	255	0	3	customer_target

7. If required, change the partition order using the up and down arrows as shown in the following image:



Name	Type	Precision	Scale	Partition	Origin
customer_country	string	255	0	1	customer_target
customer_state	string	255	0	2	customer_target
customer_salesoutlet	string	255	0	3	customer_target

   Note: Do not change the partitioning order in the **Target Fields** tab in the Target transformation. You can change the partitioning order only from the **Partitions** tab.

   The columns that you select for partitioning are set by default to string while writing the data to the target.

Rules and guidelines for adding partitioning columns

Consider the following general rules and guidelines when you include partitioned columns to a target created at runtime:

- You cannot use an existing table to add partitioned columns.
- Parallel processing is not applicable for Hive Connector.

- If you specify a value greater than 1 for the number of buckets to include in the target, but you do not select the bucket columns in the **Target Fields** tab, Data Integration does not create the buckets in the target Hive table.
- You cannot create more than two partitions for a Hive target on the Cloudera CDP version 7.1.1 cluster.
- If you create a Hive target on the Cloudera EMR cluster and the data is of the RC file format, you cannot insert null values to the binary data type column.

Mapping

Consider the following rules and guidelines for mappings:

- When you select columns other than the String data type as partition columns for a target that you want to create at runtime, the data type is written as String in the target. You cannot select a field of the Binary data type for partitioning when you create a new target at runtime.

Mappings in advanced mode

You cannot select a field of the Binary data type for partitioning when you write data to the target.

Hive lookups in mappings

You can create lookups for Hive objects. You can retrieve data from a Hive lookup object based on the specified lookup condition.

You can configure an unconnected with cached lookup in a Hive mapping. When you configure a lookup, you select the lookup connection and lookup object. You can define the behavior when a lookup condition returns more than one match.

The following table describes the Hive lookup object properties that you can configure in a Lookup transformation in mappings:

Property	Description
Connection	Name of the lookup connection. You can select an existing connection, create a new connection, or define parameter values for the lookup connection property. If you want to overwrite the lookup connection properties at runtime, select the **Allow parameter to be overridden at run time** option.
Source Type	Type of the source object. Select Single Object or Parameter.
Parameter	A parameter file where you define values that you want to update without having to edit the task. Select an existing parameter for the lookup object or click **New Parameter** to define a new parameter for the lookup object. The **Parameter** property appears only if you select parameter as the lookup type. If you want to overwrite the parameter at runtime, select the **Allow parameter to be overridden at run time** option. When the task runs, the Secure Agent uses the parameters from the file that you specify in the advanced session properties.

Property	Description
Lookup Object	Name of the lookup object for the mapping.
Multiple Matches	Behavior when the lookup condition returns multiple matches. You can return all rows, any row, or an error. You can select from the following options in the lookup object properties to determine the behavior: - Return any row - Return all rows - Report error

The following table describes the Hive lookup object advanced properties that you can configure in a Lookup transformation in mappings:

Advanced Property	Description
SQL Override	When you read data from a Hive source object, you can configure SQL overrides and define constraints.
PreSQL	SQL statement that you want to run before reading data from the source.
PostSQL	SQL statement that you want to run after reading data from the source.

Data processing using Hierarchy Processor transformation

Use a Hierarchy Processor transformation in a mapping in advanced mode to process data from complex data sources.

You can use the Hierarchy Processor transformation to read relational or hierarchical input and convert it to relational or hierarchical output. You can use hierarchical fields that represent a struct or an array.

In a mapping that converts hierarchical data to relational output, you can flatten the hierarchical data source and write the data to the target. In a mapping that converts relational data to hierarchical output, you can read from relational data sources and write to an hierarchical target output file. This transformation allows you to create structs and arrays. In a mapping that converts hierarchical data to hierarchical data, you can read from one or more hierarchical data sources and write to a hierarchical data file.

For more information about using Hierarchy Processor transformation, see *Transformations* in the Data Integration documentation.

Configure session recovery for a task that reads from Kafka

You can configure session recovery for a mapping task that reads from a Kafka source in batch mode and writes to a Hive target. If the task crashes while staging or after staging the data, you can rerun the task to recover the task from the last check point.

When you configure a recovery strategy for the task, the Secure Agent can recover unprocessed messages from a failed mapping. The Secure Agent stores source messages in a recovery file in the staging directory specified in the Hive connection properties. If the mapping task fails, run the mapping task in recovery mode to recover the messages that the Secure Agent did not process.

When you configure a mapping task, you can configure the **Recovery Strategy** property in the **Advanced Session Properties** section on the **Runtime Options** tab of the mapping task and select **Resume from the last checkpoint**. The Secure Agent saves the mapping state of operation and maintains target recovery tables. If the mapping aborts, stops, or terminates, the Secure Agent uses the saved recovery information to resume the mapping from the point of interruption.

Steps to enable message recovery

Verify that you have specified the DFS staging directory in the Hive connection properties and then perform the following tasks to enable message recovery for a mapping task:

1. In the **Advanced Session Properties** on the **Runtime Options** tab of the mapping task, add the following properties:

Session Property Name	Session Property Value
Commit on End of File	No
Commit Type	Source
Recovery Strategy	Resume from last checkpoint

2. Click **Finish**.

The Secure Agent stores the messages in the recovery file and a backup of the recovery file in the **DFS Staging Directory** field that you specified in the Hive connection properties. The Secure Agent uses the following format for the recovery file name: `<mappingID>_<databaseName>_<tableName>_recovery`

For example, the Secure Agent creates the recovery file with the following file name: `/tmp/stage/s_mtt_014TMY0Z0000000000F7_default_kaf_real_recovery`

Rules and guidelines for Hive Connector in mappings

Consider the following rules and guidelines for Hive Connector in mappings and mapping tasks:

General guidelines

When you select a connection in the mapping and leave it unused for a long duration, and then select the object in the mapping, an error occurs. You must reselect the connection in the mapping and then select the object.

Create target at runtime

When you create a target at runtime, consider the following guidelines:

- When you re-run a mapping task, the task fails when the table is dropped at the backend. You must edit the mapping or mapping task and then run the mapping task.
- If you create a Hive target and the data is of the RC file format, you cannot insert null values to the binary data type column.
- You cannot write float and double data types that contain exponential data to a Hive target on the Cloudera EMR cluster.
- When you use the Create New at Runtime option to create a Hive target, you can parameterize only the target connection and the object name using a parameter file. You cannot parameterize the other properties such as the path, location, file format, and additional properties.
- When you configure a mapping to create a new Hive target at runtime to write data from a Kafka source, where the selected objects in the mapping are parameterized, the mapping fails with the following error: com.informatica.powercenter.sdk.SDKException: Invalid operation field name
- If you enable the Alter and Apply Changes schema handling option for a mapping that creates a new target at runtime and the source column names contain upper or mixed case characters, the alter command does not trigger and the mapping fails.

Data types

- When you run a mapping to write data to a Hive table and the columns consist of complex data types, the mapping fails.
- You cannot preview a column in a table that uses the binary data type.
- If the data is of the Avro file format and contains the timestamp data types, Data Integration writes the data only up to milliseconds.
- You cannot read or write hierarchical data with decimal data types. The mapping fails with the following error: `java.lang.RuntimeException`

Parameterization

- You can enter a value up to milliseconds in a filter that is not parameterized. For example:
  `hive_timestamp_only.c.timestamp > 2085-11-06 06:24:12.018`
  You cannot enter a value in nanoseconds in a filter that is not parameterized. For example:
  `hive_timestamp_only.c.timestamp > 2085-11-06 06:24:12.01888`

  You can use an advanced filter to enter a value in nanoseconds.
- To enter a date without specifying the time in a filter that is not parameterized, use the format specified in the following example:
  `hive_date_only.c.date > 2085-11-06 00:00:00.00`
- If the target object is parameterized and the field mapping is not set to automap, the option to create a target option does not display in the task properties. You must set the field mapping to automap when you use a parameterized object.
- When you read from multiple databases, if the connection does not have the database name, you cannot use a parameter file from the mapping task.
- Avoid using an existing connection to read from multiple databases in the new mapping. If you edit an existing connection to remove the database name so that you can read from multiple databases, do not use that connection for the existing mapping. If you use the same connection in the existing mapping, it fails.

- If you parameterize the source or target connection and object and then specify an override in the advance properties with in-out parameters, and also use a parameter file in the mapping task, the data is read from the imported table at design time and not from the overridden table.

Special characters

- You cannot import data from Hive that contains Unicode characters in the table or column names. The mapping fails to run.
- When you read from or write data to a Hive table, ensure that the column name in the Hive table does not exceed 74 characters.
- When you import data from Hive tables, special characters in the table or column names appear as '_' in the source object and in the data preview in the Source transformation.
- When you import a Hive source or target table where the table or column name contains special characters such as a slash '/', the staging files do not get deleted from the staging location.

SQL override

- When you specify an SQL override to read data from a Hive source, ensure that the data type specified in the query is compatible with the data type in the selected source.
- Specify the query without a semicolon ';' at the end of the SQL statement. If you insert a semi-colon at the end of the SQL statement, the task fails.
- The SQL override in the source can contain only SELECT queries with Join or Union clauses. If the SQL override contains any other clause such as WHERE and WITH, the mapping fails.

Multiple objects

When the source type selected in a mapping is multiple objects, consider the following restrictions:

- You cannot override the table name.
- When you specify an override to the schema, ensure that all the objects selected in the Source transformation belong to the same schema.
- You cannot join the tables from different databases using the advanced relationship.

Session logs

Ignore log4j initialization warnings in the session logs.

Rules and guidelines for Hive Connector in mappings configured in advanced mode

In advanced mode, consider the following rules and guidelines for Hive Connector in mappings configured in advanced mode:

Advanced cluster

When you configure an advanced cluster to enable and run mappings in advanced mode, you cannot use the local cluster. For more information about clusters, see *Advanced Clusters* in the Administrator help.

Data types

- When you write data with the Timestamp data type from a Hive table stored in either the ORC or Parquet file format, the Timestamp data type is written only up to microseconds and the rest of the data is omitted.

- When you use a simple source filter to import a Hive source object that contains the Date or Timestamp data types, you can use the following formats in the filter condition:

  ```
  Date = YYYY-MM-DD HH24:MM:SS.US
  Timestamp = YYYY-MM-DD HH24:MM:SS.US
  ```
- Unicode data from a Hive table is written incorrectly to the target.
- You cannot read or write Array or Struct complex data types that contain nested data types such as Map that the Hive Connector cannot read. Data Integration fails to invalidate the mapping both during the design time and run time and fails with the following error: `java.sql.SQLException`
- If the data is of the Avro file format and contains the timestamp data types, Data Integration writes the data only up to milliseconds.
- When you override the source or target object in a mapping from a task with objects that contain complex data type columns, data preview fails. The error message that appears does not contain helpful information.
- If you use a parameterized source or target object in the mapping to run a dynamic mapping task, and the objects contain complex data type columns, the dynamic mapping task fails.
- When you specify a simple filter to import data that contains the Decimal data type with a precision above 28, the mapping fails with the following error: java.lang.RuntimeException

Multiple objects

When the source type selected in a mapping is multiple objects,

- You cannot override the table name.
- When you specify an override to the schema, ensure that all the objects selected in the Source transformation belongs to the same schema.
- When you read from multiple Hive databases, you cannot join the tables from different databases using the advanced relationship.
- If the connection does not have the database name, you cannot specify the database name through the parameter file from the mapping task.
- If you parameterize the source or target connection and object and then specify an override in the advance properties with in-out parameters, and also use a parameter file in the mapping task, the data is read from the imported table at design time and not from the overridden table.
- When you parameterize the connection and source object and select "Allow parameter to be overridden at runtime" in the Source transformation, you cannot select the multiple object source type in the mapping task.
- Avoid using an existing connection to read from multiple objects in the new mapping. If you edit an existing connection to remove the database name so that you can read from multiple databases, do not use that connection for the existing mapping. If you use the same connection in the existing mapping, it fails.
- When you configure a Joiner transformation in a mapping to join two Hive tables, ensure that the field names for the complex data type column in both the tables are different.

Special characters, column types, and column names

- When you read from or write data to a Hive table that contains the INFA_META_ROW_TYPE as one of its columns, data corruption is encountered.
- When you read from or write data to a Hive table, ensure that the column name in the Hive table does not exceed 74 characters.

- When you import data from Hive tables, special characters in the table or column names appear as '_' in the source object and in the data preview in the Source transformation and the mapping fails to run.
- When you import a Hive source or target table where the table or column name contains special characters such as a slash '/', the staging files do not get deleted from the staging location.
- Mappings that read from or write data to more than 500 columns fail with the following error: `HTTP POST request failed due to IO error`

SQL override

- When you specify an SQL override to read data from a Hive source, ensure that the data type specified in the query is compatible with the data type in the selected source. Specify the query without a semicolon ';' at the end of the SQL statement.
- The SQL override in the source must contain only SELECT queries. If the SQL override contains any other clause, the mapping fails.

Target operations

- When you configure an upsert operation, ensure that you select all the columns in the field mapping. If there are unconnected columns, null is inserted to the target table.
- You cannot upsert data to an external or non-transactional Hive table.
- A mapping configured with a data driven operation might encounter errors in the following scenarios:
  - When the length of the update column name exceeds 128 characters, the name is truncated. The agent fails to recognize the column name as a connected field and the mapping fails with the following error: `com.informatica.sdk.dtm.InvalidMappingException`
  - When the column name in a data driven operation exceeds 74 characters, the name is truncated.
- When you configure an upsert operation to write data that contains non-unique columns to a Hive target, set the following property in the **Pre-SQL** field in the Target transformation: `set hive.merge.cardinality.check=false`
- Do not configure an override to the update strategy from the task properties. The agent does not honor the order of precedence and considers the update strategy value specified in the mapping.
- When you configure the Upsert (Update or Insert) operation, you must always keep the **Update as Upsert** selected for the upsert operation to work. When selected, the **Update as Upsert** upserts data. When you deslect the **Update as Upsert** option and perform an update, it does not work.
- When you configure a data driven expression to update the rows and do not specify a third argument in the IIF condition, the agent treats the operation as an insert for all non-matching rows.
  For example, if you give a condition `IIF(Update_column=2,DD_UPDATE)` without including the third argument, such as `DD_DELETE` in the following expression `IIF(Update_column=2,DD_UPDATE,DD_DELETE)`, for any other row where the value of the update_column is not equal to 2, the agent performs an insert by default.

  In the example, from the expression `IIF(COL_INTEGER = 2, DD_UPDATE)`, COL_INTEGER=2 resolves to 1 and COL_INTEGER!=2 resolves to 0,

  where 1 is the internal ID for the Update operation and 0 is the internal ID for the Insert operation. The value of the third argument when not specified defaults to 0.
- When you use a parameterized Hive connection in the Target transformation, the Update Column field does not display in the target properties. In the Target transformation, select a valid connection and object that display in the list and then select the operation.
- When you include an unassigned data field from an ISD source to write to a Hive target, the mapping fails with the following error:

```
[UnassignedData] in the transformation [Target] because the type configurations do not
match
```

- When you configure the update, upsert, or data driven update strategy in mappings and the following conditions are true, the output can be unpredictable:
  - More than one row in the source matches the data in target.
  - You have set the flag `hive.merge.cardinality.check` to false in the PreSQL field in the source or target transformation properties.

  If you set the flag to true, the mapping fails and an error displays in the logs.

Truncate

- When you configure a mapping to write data to a Hive external table, the truncate target option is not applicable.
- When you configure an insert operation in a mapping and choose to truncate the table before inserting new rows, the Secure Agent truncates only the partitions in the Hive table for which the transformation received the input data. As a workaround, when you want to truncate the entire Hive table, specify the following pre-SQL statement in the Target transformation: `truncate table <table_name>;`
- Truncate table is ignored for update and delete operations.

Dynamic mapping task

When the Hive target object is parameterized and the selected operation is data driven or upsert in the mapping, the Update Column field does not display in the dynamic mapping task target properties.

Data preview for transformations

- When you create a mapping, you cannot preview data for individual transformations to test the mapping logic.

Parameterization

If the target object is parameterized and the field mapping is not set to automap, the option to create a target option does not display in the task properties. You must set the field mapping to automap when you use a parameterized object.

Create target

- When you re-run a mapping task, the task fails when the table is dropped at the backend. You must edit the mapping or mapping task and then run the mapping task.
- If you create a Hive target and the data is of the RC file format, you cannot insert null values to the binary data type column.
- You cannot write float and double data types that contain exponential data to a Hive target on the Cloudera EMR cluster.
- When you create a new Hive target at runtime, you can parameterize only the target connection and the object name. You cannot parameterize the other properties such as the partitioning, bucketing, path, location, file format, and additional properties.
- If the source data contains complex fields, the metadata fails to appear on the **Target Fields** tab. If the selected operation is update, upsert, delete, or data driven operation in the Hive Target transformation, the metadata does not appear in the **Update Columns** tab.
- When you run a task to create a new Hive target at runtime, the mapping cannot fetch records from the source and the mapping fails when both of the following conditions are true:
  - Source or target contains complex data type columns.

- Source or target selected in the mapping is parameterized.

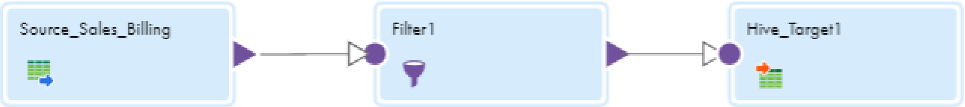
- A mapping that contains 15 or more Hive targets selected to create at runtime fails with a translation error.
- If the target table column name contains special characters and the update strategy condition is parameterized, the mapping fails.

Mapping in advanced mode example

You can create a mapping in advanced mode to achieve faster performance when you read from and write data to a Hive table.

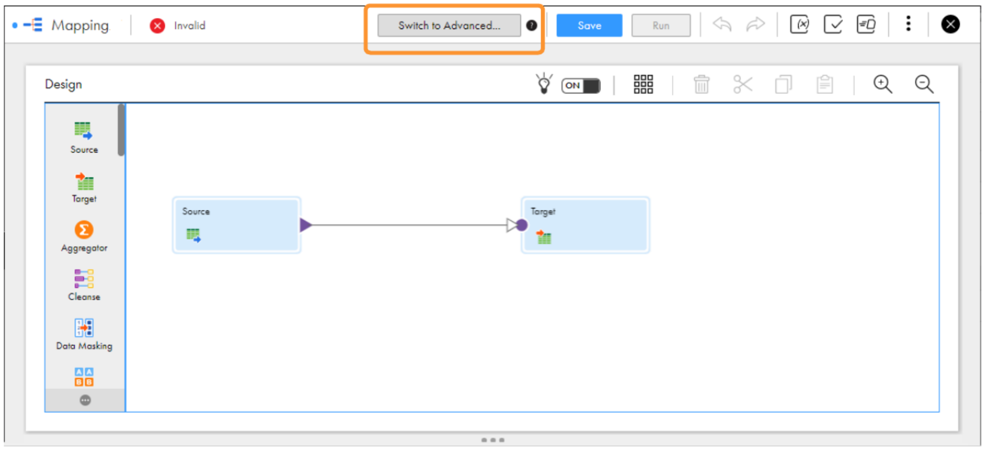
To create a mapping in advanced mode, create a mapping and then update the mapping canvas in the Mapping Designer. You can then choose to add transformations to process the data that you read from the source and then write the data to the Hive table.

The following example illustrates how to configure a mapping in advanced mode that reads data from a Hive table, filters the data based on a customer name before writing to the Hive target table:



1. In Data Integration, click **New** > **Mappings** > **Mapping**.
2. In the Mapping Designer, click **Switch to Advanced**.

   The following image shows the **Switch to Advanced** button in the Mapping Designer:

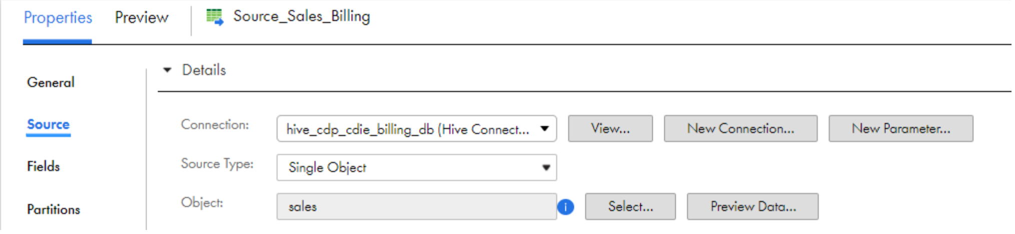


3. In the **Switch to Advanced** dialog box, click **Switch to Advanced**.

   The Mapping Designer updates the mapping canvas to display the transformations and functions that are available in advanced mode.
4. Enter a name, location, and description for the mapping.
5. Add a Source transformation, and specify a name and description in the general properties.

6. On the **Source** tab, perform the following steps to read data from the Hive source:
   a. In the **Connection** field, select the Hive connection.
   b. In the **Source Object** field, select single object as the source type.
   c. In the **Object** field, select the Sales object that has the billing details.
   d. In the **Advanced Properties** section, specify the required parameters.

   The following image shows the configured Source transformation properties that reads the billing data from the Hive object:



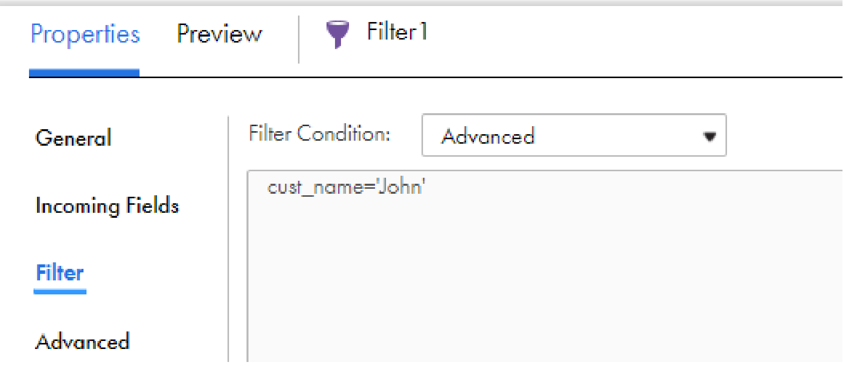
Connection: hive_cdp_cdie_billing_db (Hive Connect...

Source Type: Single Object

Object: sales

7. Add a Filter transformation, and on the **Filter** tab, define an expression to filter the records based on the customer name:

   The following image shows the configured Filter transformation properties:

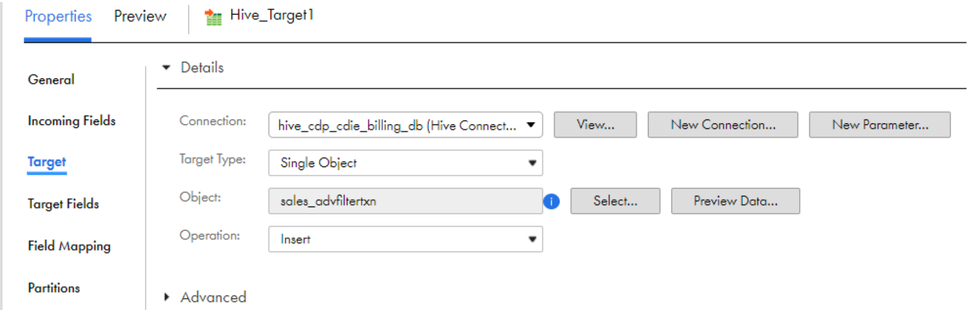


Filter Condition: Advanced

cust_name='John'

8. Add a Target transformation, and specify a name and description in the general properties.
9. On the **Target** tab, specify the details to write data to Hive:
   a. In the **Connection** field, select the Hive target connection.
   b. In the **Target Type** field, select single object.
   c. In the **Object** field, select the Hive object to which you want to write the filtered data.
   d. In the **Operation** field, select the insert operation.
   e. In the **Advanced Properties** section, specify the required advanced target properties.

      The following image shows the configured Hive Target transformation properties:



Connection: hive_cdp_cdie_billing_db (Hive Connect...

Target Type: Single Object

Object: sales_advfiltertxn

Operation: Insert

10. Click **Save** > **Run** to validate the mapping.

    In Monitor, you can monitor the status of the logs after you run the task.

Dynamic schema handling

When you add a mapping or a mapping configured in advanced mode to a mapping task, you can choose how Data Integration handles changes in the data object schemas. To refresh the schema every time the task runs, you can enable dynamic schema handling in the task.

A schema change includes one or more of the following changes to the data object:

- Fields added, deleted, or renamed.
- Fields updated for precision.

Configure schema change handling on the **Runtime Options** page when you configure the task.

The following table describes the schema change handling options:

Option	Description
Asynchronous	Default. Data Integration refreshes the schema when you edit the mapping or task, and when Informatica Intelligent Cloud Services is upgraded.
Dynamic	Data Integration refreshes the schema every time the task runs. You can choose from the following options to refresh the schema: - **Alter and apply changes.** Data Integration applies the following changes from the source schema to the target schema: - New fields: Alters the target schema and adds the new fields from the upstream transformation. - Renamed fields: Adds renamed fields as new columns in the target. - Precision updates. Applies these changes to the target. - Deleted fields: Ignores deleted fields. - **Drop current and recreate.** Drops the existing target table and then recreates the target table at runtime using all the incoming metadata fields from the upstream transformation. - **Don't apply DDL changes.** Data Integration does not apply the schema changes to the target.

For more information, see the "Schema change handling" topic in *Tasks* in the Data Integration help.

Rules and guidelines for dynamic schema handling in mappings

Consider the following general rules and guidelines when you enable dynamic schema change handling for Hive mappings:

- You cannot configure dynamic schema handling if the target table is partitioned.
- When you use a Hive partitioned table as the target, you cannot enable the Alter and Apply Changes and Drop Current and Recreate options for the task. If you enable the options and run the task, the task fails.

Alter and Apply Changes

Consider the following guidelines when you choose to alter and apply the changes to the target:

- You cannot change the precision of the data type, except for target columns that are of the Varchar and Char data types. If the target is of the Varchar, String, and Char data types, you can only increase the precision but you cannot decrease the precision.

- When you create a new target at runtime and you specify the Path property in the Create Target window and run the mapping using the Alter and Apply Changes dynamic schema handling option, the mapping fails. You must not specify the path to run the mapping successfully.

Drop Current and Recreate

Consider the following guidelines when you choose to drop and recreate the table:

- If you configure a mapping to write to an existing Hive target and you choose to drop and recreate the table, the following properties are lost:
  - Field extensions such as bucket column and partition column metadata.
  - Record extensions such as the number of buckets, external or managed table, location, and table properties.

  In this scenario, Data Integration creates a plain table in the Hive target.
- If you configure a mapping to create a new target at runtime and you choose to drop and recreate the table, the partition and bucket information in the table is lost. Data Integration creates the table without the partition and bucket columns.

View log messages

When you run a task, Data Integration logs messages Tomcat log file in the following directory:

```
<Secure Agent installation directory>/apps/Data_Integration_Server/logs/tomcat/<version>.log
```

For example, see the following log messages to verify if the job configured for create target and dynamic schema handling queries ran successfully :

```
2021-12-22 17:51:57 INFO
[com.informaticallc.adapter.hive.metadata.adapter.HiveMetadataAdapter] - Executing
create table query for target object
: CREATE EXTERNAL TABLE IF NOT EXISTS `hive_tgt_from_rs13`(`col1` string,`col2` int)
LOCATION '/tmp/hive_tgt_from_rs13'

2021-12-23 10:34:43 INFO
[com.informaticallc.adapter.hive.metadata.adapter.HiveMetadataAdapter] - Executing hive
query :
ALTER TABLE `hive_alter_table_tgt112` ADD COLUMNS(`col4` string)

2021-12-23 10:21:13 INFO
[com.informaticallc.adapter.hive.metadata.adapter.HiveMetadataAdapter] - Executing hive
query :
 ALTER TABLE `hive_alter_table_tgt112` CHANGE `col1` `col1` varchar(15)
```

CHAPTER 4

Migrating a mapping

When you configure a connection and mapping in one environment and then migrate and run them in another environment, Data Integration uses the configured runtime attributes from the earlier environment and runs the mapping successfully in the new environment.

You can also migrate mappings configured in advanced mode. After the migration, if you want to change the properties in the new environment, you can change the connection properties from Administrator, but you do not need to modify the mapping.

Plan the migration

Consider a scenario where you develop a mapping in the development organization and you then migrate the mapping to the production organization.

When you run the mapping in the migrated environment, you might want to use the same or different object path. Based on your requirement, follow the guidelines in this chapter before you migrate to new environment.

The following table lists the override properties that you can configure for an object type before you migrate a mapping:

Object Type	Advanced properties
Single object	Database, table
Multiple objects	Database

Migrate a mapping within the same path

If you want the migrated mapping to use the same object path as in the earlier environment, you must maintain the same database and table in the Hive account for both the organizations.

For example, if you have two different accounts, Account1 used for Org 1 and Account2 used for Org 2, the object path for the database and table name must be the same in both the accounts:

Account1: DB1//TABLE1

Account2: DB1//TABLE1

Migrate a mapping to a different path

You can use a different object path to run the mapping from the new environment. You can override an object if the object specified in the override contains the same metadata as the object specified at design time.

Before you migrate the mapping, you can change the object metadata and runtime attributes to reflect the object path in the migrated environment. You do not have to edit or update the mapping in the new environment.

When you specify the database or table in the advanced properties or object properties, Data Integration honors the attributes in the following order of precedence:

1. **Runtime advanced attributes**. The advanced properties such as database and table in the Source or Target transformation in a mapping.
2. **Object metadata**. The object selected in the Source or Target transformation in a mapping.

Migration options

When you migrate to a different path, you can choose from one of the following options to update the object path:

Option 1. Override the properties from the advanced properties

Before the migration, specify the required database and table name for the object from Org 2 in the advanced properties of the Org 1 mapping.

After the migration, when you run the mapping, the Secure Agent uses the configured advanced parameters to override the object specified in the mapping imported from Org 1.

Option 2. Parameterize the properties in the mapping

You can choose to parameterize the advanced attributes, such as the database and table name before the migration. You can configure input parameters, in-out parameters, and parameter files in the mapping. When you use a parameter file, you can save the parameter file on a local machine or in a cloud-hosted directory. After you migrate the mapping, do not edit or update the mapping. If you have used in-out parameters for the advanced attributes such as for the database and table name, you can update these from the parameter file.

Parameterizing only the advanced properties, but not the object

If you want to parameterize only the advanced properties and use them at runtime, select a placeholder object in the object properties in the mapping and then specify an override to this placeholder object from the advanced properties. Ensure that the placeholder object contains the same metadata as the corresponding table that you specify as an override. When you run the mapping, the value specified in the advanced property overrides the placeholder object.

Parameterizing both the object and the advanced properties

If you want to keep both the Hive object type and the advanced fields parameterized, you must leave the **Allow parameter to be overridden at runtime** option unselected in the input parameter window while adding the parameters, and then select the required object at the task level. When you run the task, the values specified in the advanced properties take precedence.

Migration rules and guidelines

Before you migrate, consider the following rules and guidelines:

Parameterization

In advanced mode, you cannot use the parameter file from the mapping task to override the runtime parameters.

In mappings, when you parameterize the source and target object and select the Hive object from the mapping task and also override the schema and table, the agent ignores the overridden values.

Using multiple source objects

All the source objects used in a mappingmust be from the same database. You can override only the schema name but not the table name.

Dynamically refresh the data object schema

You cannot dynamically refresh the data object schema at runtime. You must maintain the same metadata for the table selected in the source, target, or lookup transformations and the corresponding advanced field overrides, as schema change handling is not applicable.

Overriding properties at runtime

Consider the following restrictions when you plan to override a property at runtime:

- To configure an override, the design time and the overridden object table metadata must be the same.
- When you specify an override to the schema or table, and the mapping contains a pre-SQL, post-SQL, and an SQL override, the mapping fails to pick the overridden table and instead uses the schema from the design time.
  If you specify the pre-SQL, post-SQL, and SQL override without providing the schema or table name, the agent considers the schema and table name imported during the design time based on the database value specified in the connection string. If the database name is not provided in the connection, the mapping uses the default database in the Hive backend.
- You cannot configure both an advanced filter in the Source transformation and a schema and table override in the advanced source properties. If both these properties are configured, then the only option is to update the filter condition in the mapping with the overridden table name and then run the mapping.
- In advanced mode, if you specify the database name in the JDBC URL in the Hive connection properties and you also configure an override to the table and schema name in the advanced properties in the mapping, the database name in the connection URL is considered.

Disabling migration

If you do not want to use the migration functionality, you can manually disable the migration for your organization.

Set the *INFA_DEBUG* to `'-DDeployments_Override_Feature_Off'` in the Secure Agent properties.

1. Open Administrator and select **Runtime Environments**.
2. Select the Secure Agent for which you want to configure the DTM property.
3. On the upper-right corner of the page, click **Edit**.
4. In the **System Configuration Details** section, select the service as Data Integration Server and the type as Platform.
5. Enter the name as *INFA_DEBUG* and set the property to the following value: `'-DDeployments_Override_Feature_Off'`

CHAPTER 5

Data type reference

Data Integration uses the following data types in mapping tasks and mappings with Hive:

Hive native data types

Hive data types appear in the Source transformations when you choose to edit metadata for the fields.

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 Secure Agent uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

When Data Integration reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When Data Integration writes to a target, it converts the transformation data types to the comparable native data types.

Hive and transformation data types

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

Hive Data Type	Transformation Data Type
bigint	biginteger
binary	binary
boolean	string
char	string
date	date/time
decimal	decimal[1]
double	double
float	double
integer	integer
smallint	integer

Hive Data Type	Transformation Data Type
string	string
timestamp	date/time
tinyint	integer
varchar	string
array	array[2]
struct	struct[2] **Note:** You can also access data from a struct that is within an array.
map	map[2]

[1] You cannot use decimal values with precision greater than 28. Data Integration supports precision only up to 28 digits.

[2] To write array, map, and struct data types to Hive, use the **Create New at Runtime** option in the Target transformation.

CHAPTER 6

Troubleshooting

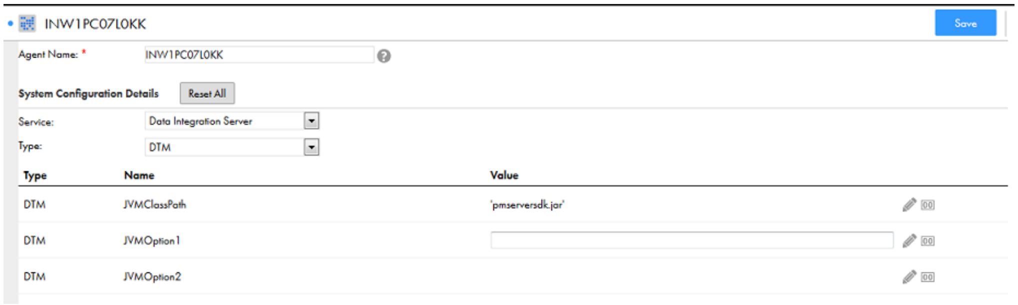
Use this section to troubleshoot errors when you use Hive Connector.

Increasing the Secure Agent memory

To increase performance and avoid runtime environment memory issues, perform the following steps:

1. In Administrator, select **Runtime Environments**.
2. Select the Secure Agent for which you want to increase memory from the list of available Secure Agents.
3. On the upper right corner of the page, click **Edit**.
4. In the **System Configuration Details** section, select the **Type** as **DTM** for the Data Integration Server.
5. Edit **JVMOption1** as **-Xms4056m** and **JVMOption2** as **-Xmx4056m**.

   The following image shows the **Details** page:

   Agent Name: * INW1PC07L0KK

   System Configuration Details

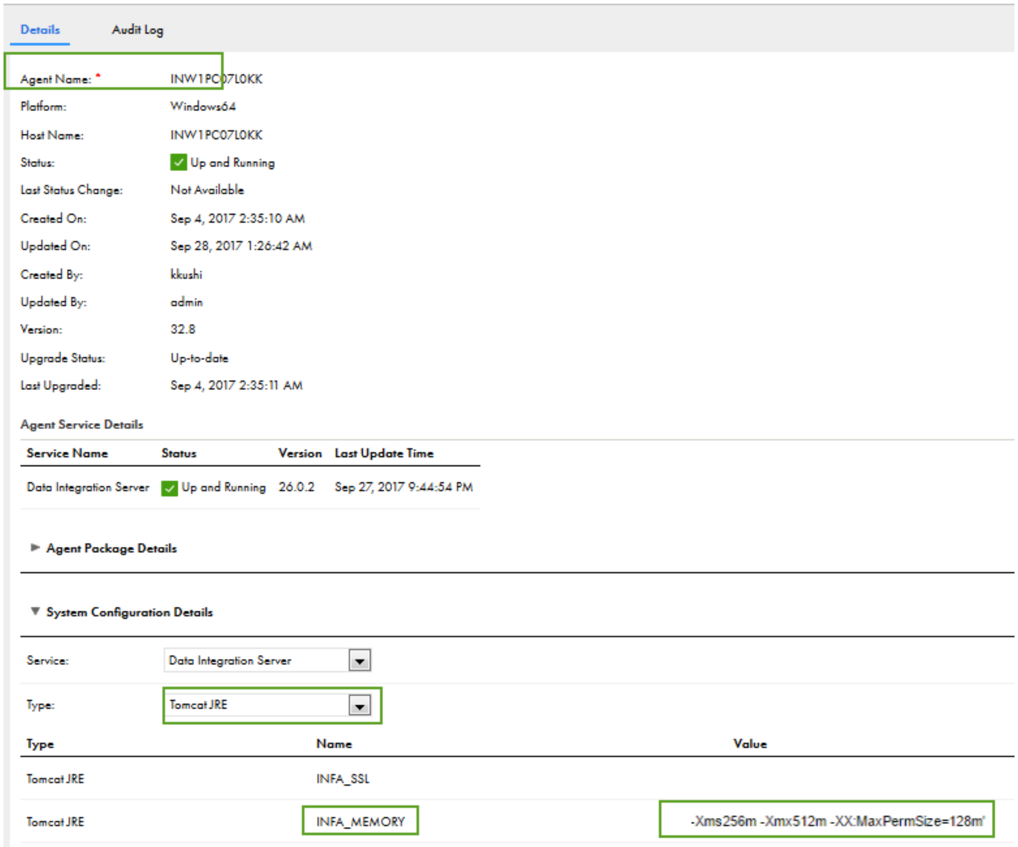
   Service: Data Integration Server

   Type: DTM

Type	Name	Value
DTM	JVMClassPath	'pmserversdk.jar'
DTM	JVMOption1	
DTM	JVMOption2	

6. In the **System Configuration Details** section, select the **Type** as **TomCatJRE**
7. Edit **INFA_memory** as **-Xms256m -Xmx512m**.

The following image shows the **Agent Details** page:



Details Audit Log

Agent Name: *	INW1PC07L0KK
Platform:	Windows64
Host Name:	INW1PC07L0KK
Status:	✓ Up and Running
Last Status Change:	Not Available
Created On:	Sep 4, 2017 2:35:10 AM
Updated On:	Sep 28, 2017 1:26:42 AM
Created By:	kkushi
Updated By:	admin
Version:	32.8
Upgrade Status:	Up-to-date
Last Upgraded:	Sep 4, 2017 2:35:11 AM

Agent Service Details

Service Name	Status	Version	Last Update Time
Data Integration Server	✓ Up and Running	26.0.2	Sep 27, 2017 9:44:54 PM

▶ Agent Package Details

▼ System Configuration Details

Service:	Data Integration Server
Type:	Tomcat JRE

Type	Name	Value
Tomcat JRE	INFA_SSL	
Tomcat JRE	INFA_MEMORY	-Xms256m -Xmx512m -XX:MaxPermSize=128m'

Note: The minimum and maximum values for the Java heap size are given as an example. Specify the size according to your requirements.

8. Restart the Secure Agent.

INDEX