Informatica®

Informatica® Cloud Data Integration

Hadoop Files V2 Connector

Informatica Cloud Data Integration Hadoop Files V2 Connector
October 2022

Publication Date: 2023-10-30

Table of Contents

Preface

Use the *Hadoop Files V2 Connector* to learn how to read from or write to Hadoop Distributed File System by using Cloud Data Integration. Learn to create a connection and 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, on the Informatica Intelligent Cloud Services Status page, click **SUBSCRIBE TO UPDATES**. You can 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 Global Support Center through the Informatica Network or by telephone.

To find online support resources on the Informatica Network, click **Contact Support** in the Informatica Intelligent Cloud Services Help menu to go to the **Cloud Support** page. The **Cloud Support** page includes system status information and community discussions. Log in to Informatica Network and click **Need Help** to find additional resources and to contact Informatica Global Customer Support through email.

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 Hadoop Files V2 Connector

You can use Hadoop Files V2 Connector to securely read data from and write data to complex files on local system or to HDFS (Hadoop Distributed File System). You can read or write structured, semi-structured, and unstructured data.

You can create a Hadoop Files V2 connection and use the connection in mappings or mapping tasks. You can switch mappings to advanced mode to include transformations and functions that enable advanced functionality. Create a mapping task to process data based on the data flow logic defined in a mapping or integration template.

Hadoop Files V2 Connector example

You are a data analyst in a large-scale social media enterprise. The enterprise stores departmental data in multiple formats such as PDF and JSON. A large amount of data is stored in the local file system of multiple computers. You need to consolidate this large amount of data and analyze it for business-critical decisions in near future.

You can consolidate data in the Hadoop environment and run various analytics tools to analyze the data quickly. You can use Hadoop Files V2 Connector to read data from complex files in multiple formats from the local file systems and write data to HDFS. After you write data to HDFS, use tools such as Hadoop Analytics to analyze the data.

Hadoop Files V2 Connector assets

Create assets in Data Integration to integrate data using Hadoop Files V2 Connector.

When you use Hadoop Files V2 Connector, you can include the following Data Integration assets:

- Mapping
- Mapping task

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

Hadoop Files V2 source and target objects

You can specify a single complex file or multiple complex files as source and target objects.

Specify the complex file name in the source and target object if you want to perform the read or write operation to a single complex file. Specify a directory if you want to perform read or write operations to multiple files in a directory and its sub directories.

To specify a directory as a source object, configure **File Path** in the advanced source properties.

To specify a directory as a target object, configure **File Directory** in the advanced target properties.

Hadoop Files V2 compression formats

You can read or write compressed complex files, specify compression formats for the target objects, and decompress the output files.

You can use compression formats such as Gzip and DEFLATE. The following table describes the complex file compression formats:

Compression Options	Description
None or empty	The file is not compressed.
Auto	The Data Integration Service detects the compression format of the file based on the file extension.
DEFLATE	The DEFLATE compression format that uses a combination of the LZ77 algorithm and Huffman coding.
Gzip	The GNU zip compression format that uses the DEFLATE algorithm.
Bzip2	The Bzip2 compression format that uses the Burrows–Wheeler algorithm.
LZO	The Lempel-Ziv-Oberhumer(LZO) compression format uses the LZO data-compression algorithm.
Snappy	The Snappy compression format uses the Snappy compression algorithm.

Hadoop Files V2 connector administration

Before you use Hadoop Files V2 Connector in tasks, an administrator must verify the following prerequisites:

- Use Hadoop Files V2 Connector on the Linux 64-bit operating system.
- Before you connect to the Hadoop cluster, enable all the required ports for the data node, name node, KMS, and Zookeeper services.

Access Kerberos-enabled Hadoop cluster

Configure the `/etc/hosts` file and copy the Kerberos configuration file for HDFS instances that use Kerberos authentication.

1. Open the `/etc/hosts` file located in the `/etc` directory on the Secure Agent machine on Linux.
2. To configure the Secure Agent to work with the Kerberos Key Distribution Center (KDC), make an entry of the KDC hosts in the `/etc/hosts` file.
3. Copy the `krb5.conf` configuration file from the `/etc` directory in the Hadoop cluster node to the following location:
   `<Secure Agent installation directory>/apps/jdk/zulu<latest_version>/jre/lib/security`

   If the Secure Agent is already installed, copy to: `<Secure Agent installation directory>/apps/jdk/jre/lib/security`
4. If the cluster is SSL enabled, import the certificate alias file to the following location:
   `<Secure Agent installation directory>/jdk/jre/lib/security/cacerts`

   If the Secure Agent is already installed, import to: `<Secure Agent installation directory>/apps/jdk/zulu<latest_version>/jre/lib/security/cacerts`
5. Restart the Secure Agent.

Access the non-Kerberos enabled Hadoop cluster

If you have a Cloudera, Amazon EMR, Hortonworks, or Microsoft HDInsight instance that does not use Kerberos authentication and runs in a Hadoop cluster environment, perform the following steps:

1. Copy the `conf` configuration files from the `/etc/hadoop/conf` directory in the Hadoop cluster node and place it in the Secure Agent location with full permission.

Supported distributions for Hadoop Files V2 Connector

The following distribution versions are supported for Hadoop Files V2 Connector:

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

CHAPTER 2

Hadoop Files V2 connections

You can use a Hadoop Files V2 connection in mappings and mapping tasks.

Create a Hadoop Files V2 connection so that the agent can read complex files from and write complex files to a local file system or HDFS. You can create a Hadoop Files V2 connection in the **Connections** page. Use the connection when you create a mapping or a mapping task.

Hadoop Files V2 connection properties

When you set up a Hadoop Files V2 connection, you must configure the connection properties.

The following table describes the Hadoop Files V2 connection properties:

Connection property	Description
Connection Name	Name of the Hadoop Files V2 connection.
Description	Description of the connection. The description cannot exceed 765 characters.
Type	Type of connection. Select **Hadoop Files V2**.
Runtime Environment	The name of the runtime environment where you want to run the tasks.
User Name	Required to read data from HDFS. Enter a user name that has access to the single-node HDFS location to read data from or write data to.

Connection property	Description

NameNode URI	The URI to access HDFS.
	Use the following format to specify the name node URI in Cloudera, Amazon EMR, and Hortonworks distributions:
	`hdfs://<namenode>:<port>/`
	where,
	- `<namenode>` is the host name or IP address of the name node.
	- `<port>` is the port that the name node listens for remote procedure calls (RPC).
	To connect to the Hadoop cluster, specify the name node port `fs.defaultFS`.
	If the Hadoop cluster is configured for high availability, you must copy the `fs.defaultFS` value in the `core-site.xml` file and append `/` to specify the name node URI.
	For example, the following snippet shows the `fs.defaultFS` value in a sample `core-site.xml` file:
	`<property>`
	`<name>fs.defaultFS</name>`
	`<value>hdfs://nameservice1</value>`
	`<source>core-site.xml</source>`
	`</property>`
	In the above snippet, the `fs.defaultFS` value is
	`hdfs://nameservice1`
	and the corresponding name node URI is
	`hdfs://nameservice1/`
	Note: Specify either the name node URI or the local path. Do not specify the name node URI if you want to read data from or write data to a local file system path.

Local Path	A local file system path to read and write data. Read the following conditions to specify the local path: - You must enter **NA** in local path if you specify the name node URI. If the local path does not contain **NA**, the name node URI does not work. - If you specify the name node URI and local path, the local path takes the preference. The connection uses the local path to run all tasks. - If you leave the local path blank, the agent configures the root directory (/) in the connection. The connection uses the local path to run all tasks. - If the file or directory is in the local system, enter the fully qualified path of the file or directory. For example, `/user/testdir` specifies the location of a directory in the local system. Default value for Local Path is NA.
Configuration Files Path	The directory that contains the Hadoop configuration files. **Note:** Copy the core-site.xml, hdfs-site.xml, and hive-site.xmlfrom the Hadoop cluster and add them to a folder in Linux Box.
Keytab File	The file that contains encrypted keys and Kerberos principals to authenticate the machine.
Principal Name	Users assigned to the superuser privilege can perform all the tasks that a user with the administrator privilege can perform.
Impersonation Username	You can enable different users to run mappings in a Hadoop cluster that uses Kerberos authentication or connect to sources and targets that use Kerberos authentication. To enable different users to run mappings or connect to big data sources and targets, you must configure user impersonation.

Note: When you read from or write to remote files, the **NameNode URI** and **Configuration Files Path** fields are mandatory. When you read from or write to local files, you require only the **Local Path** field.

Creating a Hadoop Files V2 connection

To use Hadoop Files V2 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 following connection properties:

Connection property	Description
Connection Name	Name of the Hadoop Files V2 connection.
Description	Description of the connection. The description cannot exceed 765 characters.
Type	Type of connection. Select Hadoop Files V2.
Runtime Environment	The name of the runtime environment where you want to run the tasks.
User Name	Required to read data from HDFS. Enter a user name that has access to the single-node HDFS location to read data from or write data to.

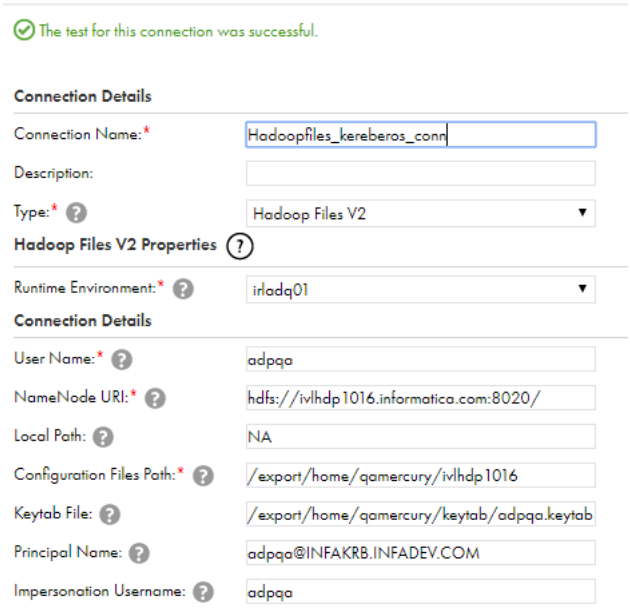
Connection property	Description

NameNode URI	The URI to access HDFS.
	Use the following format to specify the name node URI in Cloudera, Amazon EMR, and Hortonworks distributions:
	`hdfs://<namenode>:<port>/`
	Where
	- `<namenode>` is the host name or IP address of the name node.
	- `<port>` is the port that the name node listens for remote procedure calls (RPC).
	If the Hadoop cluster is configured for high availability, you must copy the `fs.defaultFS` value in the `core-site.xml` file and append `/` to specify the name node URI.
	For example, the following snippet shows the `fs.defaultFS` value in a sample `core-site.xml` file:
	`<property>`
	`<name>fs.defaultFS</name>`
	`<value>hdfs://nameservice1</value>`
	`<source>core-site.xml</source>`
	`</property>`
	In the above snippet, the `fs.defaultFS` value is
	`hdfs://nameservice1`
	and the corresponding name node URI is
	`hdfs://nameservice1/`
	Note: Specify either the name node URI or the local path. Do not specify the name node URI if you want to read data from or write data to a local file system path.

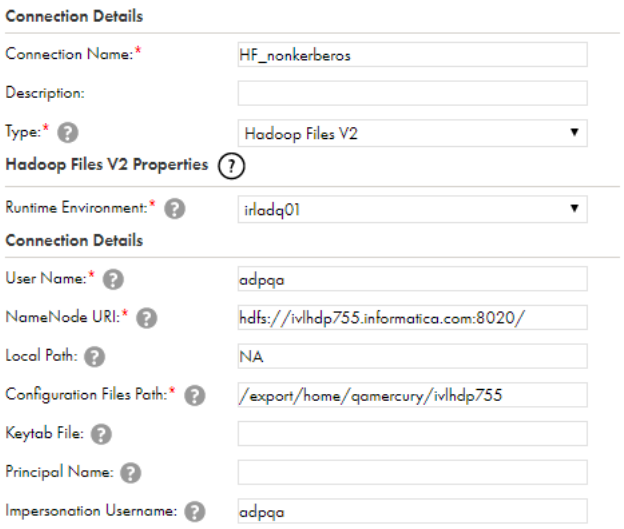
Local Path	A local file system path to read data from or write data to. Do not specify local path if you want to read data from or write data to HDFS. Read the following conditions to specify the local path: - You must enter **NA** in local path if you specify the name node URI. If the local path does not contain **NA**, the name node URI does not work. - If you specify the name node URI and local path, the local path takes the preference. The connection uses the local path to run all tasks. - If you leave the local path blank, the agent configures the root directory (/) in the connection. The connection uses the local path to run all tasks. Default value for Local Path is NA.
Configuration Files Path	The directory that contains the Hadoop configuration files for the client.
Keytab File	The file that contains encrypted keys and Kerberos principals to authenticate the machine.
Principle Name	Users assigned to the superuser privilege can perform all the tasks that a user with the administrator privilege can perform.
Impersonation Username	You can enable different users to run mappings in a Hadoop cluster that uses Kerberos authentication or connect to sources and targets that use Kerberos authentication. To enable different users to run mappings or connect to big data sources and targets, you must configure user impersonation.

3. Click **Test Connection** to evaluate the connection.

   The following image shows the connection page details in Kerberos distribution:



The following image shows the connection page details in non-Kerberos distribution:



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

Hadoop Files V2 Connector rules and guidelines

Consider the following rules and guidelines when you create a Hadoop Files V2 connection:

- Ensure that you restart the Secure Agent if you update the following connection properties:
  - **Impersonation Username**
  - **Keytab File**
  - **Principal Name**

- **NameNode URI**
- **Configuration Files Path**

- Do not use a colon (:) and double forward slash (//) characters in the **File path** field of a complex file source object.

CHAPTER 3

Mappings and tasks with Hadoop Files V2 Connector

When you create a mapping, you can configure a Source or Target transformation to represent a Hadoop Files V2 object.

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

After you configure a mapping, deploy the mapping in a mapping task. If you parameterize the connection or object in a mapping, you must specify the parameterized values when you create the mapping task.

Hadoop Files V2 sources in mappings

To read data from a complex file, configure a Hadoop Files V2 object as the Source transformation in a mapping. You can configure a Source transformation to represent a single complex file source.

Specify the name and description of the Hadoop Files V2 source. Configure the source and advanced properties for the source object.

The following table describes the Hadoop Files V2 source properties that you can configure in a Source transformation:

Source Property	Description
Connection	Name of the source connection or create a connection parameter.
Source Type	Type of source object. Select Single or Parameter as the source type.
Object	Select the source object from which you want to read data. Though selecting a source object is mandatory, the agent ignores this object. The agent processes the source object specified in File Path in advanced source properties.

Source Property	Description
Format	File format of the source object. You can select from the following file format types: - None - Avro - Parquet - JSON Default is **None**. If you select **None** as the format type, the Secure Agent reads data in binary format. In advanced mode, **None** is not applicable. **Note:** You can only use Avro, Parquet, and JSON file format types in Hadoop Files V2 Connector. You cannot read data from ORC file format types even though they are listed in the **Formatting Options**.
Parameter	The parameter for the source object. Create or select the parameter for the source object. **Note:** The parameter property appears if you select parameter as the source type.

The following table describes the Hadoop Files V2 source advanced properties that you can configure in a Source transformation:

Advanced Property	Description
File path	Mandatory. Location of the file or directory from which you want to read data. Maximum length is 255 characters. If the path is a directory, all the files in the directory must have the same file format. If the file or directory is in HDFS, enter the path without the node URI. For example, /user/lib/testdir specifies the location of a directory in HDFS. The path must not contain more than 512 characters. If the file or directory is in the local system, enter the fully qualified path. For example, /user/testdir specifies the location of a directory in the local system.
File Format	Mandatory. Name and format of the file from which you want to read data. Specify the value in the following format: `<filename>.<format>` For example, `customer.avro`
Allow Wildcard Characters	Indicates whether you want to use wildcard characters for the source directory name or the source file name. If you select this option, you can use asterisk (*) wildcard character for the source directory name or the source file name in the **File path** field.
Allow Recursive Read	Indicates whether you want to use wildcard characters to read complex files of the Parquet, Avro, or JSON formats recursively from the specified folder and its subfolders and files. You can use the wildcard character as part of the file or directory. For example, you can use a wildcard character to recursively read data from the following folders: - **/myfolder*/**. Returns all files within any folder or subfolder that has a pattern myfolder in the path name. - **/myfolder*/*.csv**. Returns all .csv files within any folder or subfolder that has a pattern myfolder in the path name. - **/myfolder*/ and file name is abc***. Returns all files that have a pattern abc within any folder or subfolder that has a pattern myfolder in the path name.

Advanced Property	Description
File Format	Specifies a file format of a complex file source. Select one of the following options: - Binary - Custom Input - Sequence File Format Default is Binary.
Input Format	The class name for files of the input file format. If you select input file format in the **File Format** field, you must specify the fully qualified class name implementing the `InputFormat` interface. To read files that use the Avro format, use the following input format: `com.informatica.avro.AvroToXML`
Input Format Parameters	Parameters for the input format class. Enter name-value pairs separated with a semicolon. Enclose the parameter name and value within double quotes. For example, use the following syntax: `"param1"="value1";"param2"="value2"`
Compression Format	Compression format of the source files. Select one of the following options: - None - Auto - DEFLATE - gzip - bzip2 - Lzo - Snappy - Custom
Custom Compression Codec	Required if you use custom compression format. Specify the fully qualified class name implementing the `CompressionCodec` interface.
Tracing Level	Sets the amount of detail that appears in the log file. You can choose terse, normal, verbose initialization, or verbose data. Default is normal.

Hadoop Files V2 targets in mappings

To write data to a Hadoop Files V2, configure a Hadoop Files V2 object as the Target transformation in a mapping. You can configure a Target transformation to represent a single Hadoop Files V2 target.

When you use an Hadoop Files V2 target object, select an Hadoop Files V2 object as the target.

The following table describes the Hadoop Files V2 target properties that you can configure in a Target transformation:

Target Property	Description
Connection	Name of the target connection or create a connection parameter.
Target Type	Type of target object. Select Single Object or Parameter.

Target Property	Description
Object	Select the file to which you want to write data. Though selecting a target object is mandatory, the agent ignores this object. The agent processes the target object specified in File Directory and File Name in advanced target properties. You can select an existing object or create an object at runtime.
Create New at Runtime	Creates a complex file target object at runtime. Enter a name for the target object and map the source fields that you want to use. By default, all source fields are mapped. You can use parameters defined in a parameter file in the target name.
Format	File format of the target object. You can select from the following file format types: - None - Avro - Parquet - JSON Default is **None**. If you select **None** as the format type, the Secure Agent writes data in binary format. In advanced mode, **None** is not applicable. **Note:** You can only use Avro, Parquet, and JSON file format types in Hadoop Files V2 Connector. You cannot write data to ORC file format types even though they are listed in the **Formatting Options**.
Parameter	The parameter for the target object. Create or select the parameter for the target object. **Note:** The parameter property appears if you select parameter as the target type.
Operation	Select the target operation. You can select insert, update, upsert, delete, or data driven in a complex file target.

The following table describes the Hadoop Files V2 target advanced properties that you can configure in a Target transformation:

Advanced Property	Description
File Directory	Optional. The directory location of one or more output files. Maximum length is 255 characters. If you do not specify a directory location, the output files are created at the location specified in the connection. If the directory is in HDFS, enter the path without the node URI. For example, /user/lib/testdir specifies the location of a directory in HDFS. The path must not contain more than 512 characters. If the file or directory is in the local system, enter the fully qualified path. For example, /user/testdir specifies the location of a directory in the local system.
File Name	Optional. Renames the output file. The file name is not applicable when you read or write multiple Hadoop Files V2s.
Overwrite Target	Indicates whether the Secure Agent must first delete the target data before writing data. If you select the **Overwrite Target** option, the Secure Agent deletes the target data before writing data. If you do not select this option, the Secure Agent creates a new file in the target and writes the data to the file.

Advanced Property	Description
File Format	Specifies a file format of a complex file source. Select one of the following options: - Binary - Custom Input - Sequence File Format Default is Binary.
Output Format	The class name for files of the output format. If you select Output Format in the **File Format** field, you must specify the fully qualified class name implementing the `OutputFormat` interface.
Output Key Class	The class name for the output key. If you select Output Format in the **File Format** field, you must specify the fully qualified class name for the output key. You can specify one of the following output key classes: - BytesWritable - Text - LongWritable - IntWritable **Note:** Hadoop Files V2 generates the key in ascending order.
Output Value Class	The class name for the output value. If you select Output Format in the **File Format** field, you must specify the fully qualified class name for the output value. You can use any custom writable class that Hadoop supports. Determine the output value class based on the type of data that you want to write. **Note:** When you use custom output formats, the value part of the data that is streamed to the complex file data object write operation must be in a serialized form.
Compression Format	Compression format of the source files. Select one of the following options: - None - Auto - DEFLATE - gzip - bzip2 - LZO - Snappy - Custom
Custom Compression Codec	Required if you use custom compression format. Specify the fully qualified class name implementing the `CompressionCodec` interface.
Sequence File Compression Type	Optional. The compression format for sequence files. Select one of the following options: - None - Record - Block
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.

Writing to multiple target objects

When you import target objects, the Secure Agent appends a FilePath field to the imported target object. When you map the FilePath field in the target object to an incoming field, the Secure Agent creates the folder structure and the target files based on the FilePath field. For example:

Syntax:

```
<tgt_FilePath_folder>/<tgt_FilePath=incoming_value_folder>/part_file
```

Sample:

```
emp_tgt.parquet/emp_tgt.parquet=128000/part-0000-e9ca8-6af-efd43-455c-8709.c000.parquet
```

The FilePath field is applicable to the following file formats:

- Avro
- Parquet

Consider the following guidelines when using the target FilePath field in mappings:

- Do not map the source object FilePath field to the target object FilePath field. If you map the FilePath field in the target object to an incoming field, the Secure Agent does not create directory structure as expected.
- When you map a date type incoming field to the FilePath field in the target object, the Secure Agent creates a nested folder structure based on the incoming date value for target objects.

Rules and guidelines for Hadoop Files V2 objects in mappings

When you use Hadoop Files V2 objects as sources and targets in mappings, you cannot preview the source and target objects.

Hadoop Files V2 file formatting options for create target

When you create a new complex file target, you can configure format options and specify the file format type of the target object.

Format Type

You can select the following file format types:

- None
- Avro
- Parquet
- JSON

Default is **None**. If you select **None** as the format type, the Secure Agent writes data in binary format.

Note: You can only use Avro, Parquet, and JSON file format types in Hadoop Files V2 Connector. You cannot write data to ORC file format types even though they are listed in the **Formatting Options**.

Rules and guidelines for file formats

You must set the appropriate source and target properties when you select the file format types.

Use the following guidelines when you select a file format type:

- The **Object** field in the source properties from which you want to read data should be same as the source object specified in the **File path** field in advanced source properties.
- If you write data to a complex file target and the File path field is mapped to any of the source fields, the file name in the **Object** field in the target properties should be different from the target **File Name** specified in the in advanced target properties.
- You cannot write data to a complex file object in Avro, Parquet, or JSON file format using **Create Target** option, if one of the field name is FilePath in the source schema.
- You cannot read and write nested and multi-line indented JSON files.
  You can use the following JSON file structure to read data from and write data to a complex file object in JSON file format:

  1.
```
{"Field Name1": "value", "Field Name2": "value", "Field Name3":
"value",.........,"Field Name n": "value"}
```
  2.
```
[\{"Field Name1": "value", "Field Name2": "value", "Field Name3":
"value",.........,"Field Name n": "value"},\{"Field Name1": "value", "Field
Name2": "value", "Field Name3": "value",.........,"Field Name n": "value"},\
{"Field Name1": "value", "Field Name2": "value", "Field Name3":
"value",.........,"Field Name n": "value"}]
```

Data compression in Hadoop Files V2 sources and targets

You can decompress data when you read data from Hadoop Files V2 and compress the data when you write data to Hadoop Files V2.

Configure the compression format in the **Compression Format** option under the advanced source and target properties.

The following table lists the compression format support for various operations and file formats:

Compression format	Avro File	JSON File	Parquet File	Binary File
None	Yes	Yes	Yes	Yes
Auto	Yes	Yes	Yes	Yes
Bzip2	No	No	No	Yes
Deflate	Yes	No	No	Yes
Gzip	No	No	Yes	Yes
LZO	No	No	Yes	Yes
Snappy	Yes	No	Yes	Yes

Note: You cannot use any compression format for a JSON file in Hadoop Files V2 Sources and Targets.

Configuring LZO compression format

To write .jar files in the LZO compression format, you must copy the files for compression to the machine where the Secure Agent runs.

Perform the following steps to configure the Secure Agent for LZO compression:

1. Copy the LZO native binaries from the cluster to one of the following directories on the machine on which the Secure Agent runs:
   - `<agent-root>/downloads/package-<distribution>/package/<distribution name>/lib/native`
2. In the **Control Panel** Window, click **System and Security**.
3. In the **System and Security** Window, click **Advanced system settings**.
4. On the **Advanced** tab, select the **Environment Variables** button.
   The **Edit Environment Variables** dialog box appears.
5. Click **New** to add a new environment variable.
   The **New Environment Variables** dialog box appears.
6. Enter the value of the **Name** field as `LD_LIBARY_PATH`.
7. Enter the following path in the **Value** field:
   `<agent-root>/downloads/package-<distribution>/package/<distribution name>/infaLib/`
8. Restart the Secure Agent.

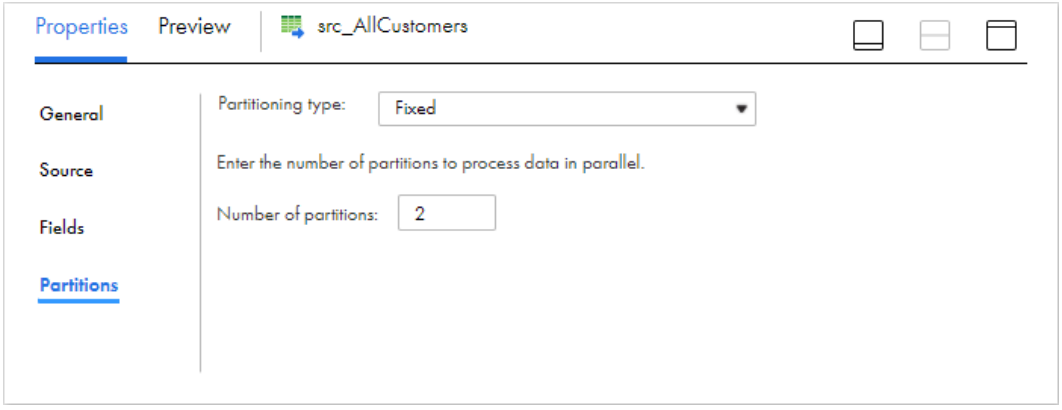
File partitioning to read from complex files

You can configure fixed partitioning in a mapping that reads data from complex files of the Parquet and Avro file formats.

Enable partitioning when you configure the Source transformation in the Mapping Designer.

On the **Partitions** tab for the Source transformation, you select fixed partitioning and enter the number of partitions based on the amount of data that you want to read.

The following image shows the configured partitioning:



When you run the mapping, the Secure Agent distributes the source data across the partitions to process parallely based on the number of partitions that you specify in the Source transformation. The maximum number of partitions must not exceed 64.

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 Hadoop Files V2 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.

APPENDIX A

Hadoop Files V2 data type reference

Data Integration uses complex files in mapping tasks with Hadoop Files V2 Connector.

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

Complex file native data types

Complex native data types appear in the Source and the Target 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 mappings, mass ingestion tasks, and mapping tasks.

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.

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

Complex File Native Data Type	Transformation Data Type
Binary	Binary

Avro complex file data types and transformation data types

Avro complex file data types map to transformation data types that the Secure Agent uses to move data across platforms.

The following table lists the Avro complex file data types that the Secure Agent supports and the corresponding transformation data types:

Avro Complex File Data Type	Transformation Data Type	Range and Description
Boolean	Integer	TRUE (1) or FALSE (0)
Bytes	Binary	Precision 4000
Double	Double	Precision 15
Float	Double	Precision 15
Int	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
Long	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
Null	Integer	-2,147,483,648 to 2,147,483,647 Precision 10, scale 0
String	String	-1 to 104,857,600 characters

Parquet complex file data types and transformation data types

Parquet complex file data types map to transformation data types that the Secure Agent uses to move data across platforms.

The following table lists the Parquet complex file data types that the Secure Agent supports and the corresponding transformation data types:

Parquet Complex File Data Type	Transformation Data Type	Range and Description
Binary	Binary	1 to 104,857,600 bytes.
Boolean	Integer	TRUE (1) or FALSE (0).

Parquet Complex File Data Type	Transformation Data Type	Range and Description
Date	Date/Time	January 1, 0001 to December 31, 9999.
Decimal	Decimal	Precision 1 to 28 digits, scale 0 to 28. **Note:** You cannot use decimal values with precision greater than 28.
Double	Double	Precision 15.
Float	Double	Precision 15.
Int32	Integer	-2,147,483,648 to +2,147,483,647.
Int64	Bigint	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807. 8-byte signed integer.
Int96	Date/Time	Jan1,0001 to Dec 31, 9999. Precision of 29, scale of 9.
String	String	-1 to 104,857,600 characters.
Time	Date/Time	Time of the day. Precision to microsecond.
Timestamp	Date/Time	January 1, 0001 00:00:00 to December 31, 9999 23:59:59.997. Precision to microsecond. **Note:** Precision to nanoseconds is not applicable.

The Parquet schema that you specify to read or write a Parquet file must be in smaller case. Case-sensitive schema is not applicable.

JSON complex file data types and transformation data types

JSON complex file data types map to transformation data types that the Secure Agent uses to move data across platforms.

The following table lists the JSON complex file data types that the Secure Agent supports and the corresponding transformation data types:

JSON Complex File Data Type	Transformation Data Type	Range and Description
Double	Double	Precision 15
Integer	Integer	-2,147,483,648 to 2,147,483,647 Precision of 10, scale of 0
Long	BigInt	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Precision 19, scale 0
String	String	-1 to 104,857,600 characters

Rules and guidelines for data types

Consider the following rules and guidelines for data types:

- Parquet data types support:
  - The following table shows the parquet version and the supported data types:

Parquet Version	Supported Data Types	Supported Distributions
2.1.0	Decimal	CDH 5.15
2.3.0	Date, Time_Millis, Timestamp_Millis	HDP 2.6, HDI 3.6
2.3.1	Date, Time_Millis, Time_Micros, Timestamp_Millis, Timestamp_Micros	CDH 6.1
2.4.0	Date, Time_Millis, Time_Micros, Timestamp_Millis, Timestamp_Micros	HDP 3.1, HDI 4.0

  - You can use the Decimal data types when you run a mapping in CDH 5.15, CDH 6.1, HDP 2.6, HDP 3.1, and HDI 4.0 distributions.

INDEX

T

U

W