Informatica

Informatica® Informatica Big Data Management

10.2.2

How to Migrate Mappings from the Hive Engine

Informatica Informatica Big Data Management How to Migrate Mappings from the Hive Engine
10.2.2
June 2019

Publication Date: 2020-11-19

Table of Contents

Abstract

Effective in version 10.2.2, Informatica dropped support for the Hive engine. You can run mappings on the Blaze and Spark engines in the Hadoop environment or on the Databricks Spark engine in the Databricks environment. This article tells how to change the validation and run-time environments for mappings, and it describes processing differences for mappings that run on the Blaze and Spark engines versus the Hive engine.

CHAPTER 1

How to Migrate Mappings from the Hive Engine

Enter a short description of the concept here (required).

This is the start of the concept.

Overview

After you upgrade to 10.2.2, you need to run a set of infacmd commands to update mappings to set the validation and execution environments to remove Hive engine configuration. Read this article to learn how to migrate mappings and to understand the processing differences between the Hive engine and the Spark and Blaze engines.

Update Mapping Environments

When you update mappings through infacmd, Informatica recommends that you choose all run-time engines for validation. When all engines are selected for validation, Informatica can analyze each engine to determine the best engine to run the mapping based on functionality, performance, and heuristics. For example, if you run a mapping that contains a Python transformation, the Data Integration Service will push the mapping to the Spark engine because the Python transformation is not supported on the Blaze engine. Read this article to learn about the commands that you can run to change the environment.

Understand Engine Processing Differences

When you push a mapping to the Hadoop environment, the engine that processes the mapping uses a set of rules that differ from the Data Integration Service, and the results might vary based on the rules that the engine uses. This article provides information about processing differences between the Hive engine and the Spark and Blaze engines as they relate to Data Integration Service processing. This article does not provide information about processing differences between the Data Integration Service and the Spark and Blaze engines if the Spark and Blaze engine processing is the same as the Hive engine.

If you want more detailed information about the transformation processing for each engine, you can click links to the user guides throughout this article. For information about known limitations related to the engines, you can refer to the 10.2.2 Release Notes.

Note: Informatica documents differences that are discovered through internal testing and usage. Informatica does not test all the rules of the third-party engines and cannot provide an extensive list of the differences.

Engine Support Summary for Transformations

The following table lists how the Spark and Blaze engine support differs from the Hive engine for each transformation:

Transformation	Spark Engine	Blaze Engine
Address Validator	No processing differences	No processing differences
Aggregator	No processing differences	Some processing differences
Case Converter	No processing differences	No processing differences
Classifier	No processing differences	No processing differences
Comparison	No processing differences	No processing differences
Consolidation	No processing differences	No processing differences
Data Masking	No processing differences	No processing differences
Data Processor	Not supported	Some processing differences
Decision	Some processing differences	Supported without restrictions
Expression	No processing differences	Some processing differences
Filter	No processing differences	Some processing differences
Java	Some processing differences	Some processing differences
Joiner	Some processing differences	Some processing differences
Key Generator	No processing differences	No processing differences
Labeler	Supported without restrictions	Supported without restrictions
Lookup	Some processing differences	Some processing differences
Match	No processing differences	No processing differences
Merge	No processing differences	No processing differences
Normalizer	No processing differences	No processing differences
Parser	Supported without restrictions	Supported without restrictions
Python*	Additional support	Not supported
Rank	Some processing differences	Some processing differences
Router	No processing differences	No processing differences
Sequence Generator*	Additional support	Additional support
Sorter	No processing differences	Some processing differences

Transformation	Spark Engine	Blaze Engine
Standardizer	No processing differences	No processing differences
Union	No processing differences	No processing differences
Update Strategy	Some processing differences	Some processing differences
Weighted Average	No processing differences	No processing differences
Window*	Additional support	Not supported
** Previously not supported on the Hive engine.*		

General Processing Differences for Non-native Engines

This article provides processing differences based on transformation processing. However, the different engines in the Hadoop environment process some functionality different than the Data Integration Service regardless of the transformation. General processing differences can include processing of data types and functions. For more information about general processing differences, refer to the following links:

Function Support in a Non-native Environment

Rules and Guidelines for Spark Engine Processing

Rules and Guidelines for Blaze Engine Processing

Function and Data Type Processing on the Spark Engine

Update Mappings for Dropped Hive Engine Support

After you upgrade to version 10.2.2, you need to update mappings that have the Hive engine configured within the Hadoop validation environment. Run a series of infacmd commands to update mappings to change the Hive engine configuration. Informatica continues to support the Blaze and Spark engines in the Hadoop environment.

Run commands using the following infacmd plugins.

- **infacmd dis plugin**. Run commands with the dis plugin to update mappings that are deployed to the Data Integration Service. For example, dis enableMappingValidationEnvironment.
- **infacmd mrs plugin**. Run commands with the mrs plugin to update mappings that are not deployed to the Data Integration Service. For example, mrs enableMappingValidationEnvironment.

Note: When you run the commands, the -sn (Service Name) parameter depends on the plugin that you use. Use the name of the Data Integration Service when you run dis commands, and use the name of the Model Repository Service when you run mrs commands.

Run the following commands against both the dis and the mrs plugins.

listMappingEngines

To identify mappings that have the Hive engine configured for validation, run the listMappingEngines command with the `-vef` parameter set to `hive`. Consider the following sample syntax:

```
mrs|dis listMappingEngines -dn domain_3987 -un Administrator -pd Password -vef hive -sn
SN_3986
```

For more information, see dis listMappingEnginesand mrs listMappingEngines.

enableMappingValidationEnvironment

If you want to enable other validation environments, run the enableMappingValidationEnvironment command for each environment that you want to enable. You can enable the following environments: native, blaze, spark, or spark-databricks. Consider the following sample syntax examples based on different command filters:

- Modify all mappings.
  ```
  mrs|dis enableMappingValidationEnvironment -dn domain_3987 -un Administrator -pd
  Password -sn SN_3986 -ve spark -cn HADOOP_cco_hdp619
  ```
- Modify mappings based on mapping name.
  ```
  mrs|dis enableMappingValidationEnvironment -dn domain_3987 -un Administrator -pd
  Password -sn SN_3986 -ve spark -cn HADOOP_cco_hdp619 -mnf m_nav327,m_nav376
  ```
- Modify mappings based on execution environment, mapping name, and project name.
  ```
  mrs|dis enableMappingValidationEnvironment -dn domain_3987 -un Administrator -pd
  Password -sn SN_3986 -ve spark -cn HADOOP_cco_hdp619 -eef hadoop -mnf
  m_nav327,m_nav376 -pn project1
  ```

For more information, see dis enableMappingValidationEnvironment and mrs enableMappingValidationEnvironment.

setMappingExecutionEnvironment

If you want to change the execution environment, run the setMappingExecutionEnvironment. Consider the following sample syntax based on mapping name filter:

```
mrs|dis setMappingExecutionEnvironment -dn domain_3987 -un Administrator -pd Password -sn
SN_3986 -ee Databricks -mnf m_nav327,m_nav376 -cn DATABRICKS_cco_db619
```

For more information, see dis setMappingExecutionEnvironment and mrs setMappingExecutionEnvironment.

disableMappingValidationEnvironment

Update all mappings in the Model repository to disable the Hive engine from the Hadoop validation environment. Consider the following sample syntax:

```
mrs|dis disableMappingValidationEnvironment -dn domain_3987 -un Administrator -pd
Password -sn SN_3986 -ve hive
```

For more information, see dis disableMappingValidationEnvironment and mrs disableMappingValidationEnvironment.

listMappingEngines

Run listMapping Engines again to verify that all Hive validation environments are disabled.

Warnings

Consider the following points of failure *if you do not update the environments*:

- Mappings fail at run time if configured with the Hive engine as the *only* validation environment.

- If you edit the validation environment in the Developer tool that has the Hive engine as the *only* validation environment, the Hadoop connection in the mapping is lost. You need to set the validation environments and select the Hadoop connection again. This can happen when you upgrade from a previous version or when you import a mapping from a previous version.

Address Validator Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior is the same as the Hive engine.
- Blaze engine. Processing behavior is the same as the Hive engine.

Related Links

10.2.2 Address Validator Transformation for the Spark and Blaze Engines

10.2.1 Address Validator Transformation for the Hive Engine

Aggregator Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior is the same as the Hive engine.
- Blaze engine. Processing behavior differs from the Hive engine.

Blaze Engine

Data cache optimization is different on the Blaze engine for the Aggregator transformation. The data cache for the Aggregator transformation is optimized to use variable length to store binary and string data types that pass through the Aggregator transformation. The optimization is enabled for record sizes up to 8 MB. If the record size is greater than 8 MB, variable length optimization is disabled.

When variable length is used to store data that passes through the Aggregator transformation in the data cache, the Aggregator transformation is optimized to use sorted input and a pass-through Sorter transformation is inserted before the Aggregator transformation in the run-time mapping.

To view the Sorter transformation, view the optimized mapping or view the execution plan in the Blaze validation environment.

During data cache optimization, the data cache and the index cache for the Aggregator transformation are set to Auto. The sorter cache for the Sorter transformation is set to the same size as the data cache for the Aggregator transformation. To configure the sorter cache, you must configure the size of the data cache for the Aggregator transformation.

Related Links

10.2.2 Aggregator Transformation for the Spark and Blaze Engines

10.2.1 Aggregator Transformation for the Hive Engine

Case Converter Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Classifier Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Comparison Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Consolidation Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior is the same as the Hive engine.
- Blaze engine. Processing behavior is the same as the Hive engine.

Related Links

10.2.2 Consolidation Transformation for the Spark and Blaze Engines

10.2.1 Consolidation Transformation for the Hive Engine

Data Masking Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior is the same as the Hive engine.
- Blaze engine. Processing behavior is the same as the Hive engine.

Related Links

10.2.2 Data Masking Transformation for the Spark and Blaze Engines

10.2.1 Data Masking Transformation for the Hive Engine

Data Processor Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Not supported for this transformation.
- Blaze engine. Processing behavior differs from the Hive engine.

Blaze Engine

Mapping validation fails when the transformation data processor mode is set to **Input Mapping** or **Service and Input Mapping**.

Related Links

10.2.2 Data Processor Transformation for the Spark and Blaze Engines

10.2.1 Data Processor Transformation for the Hive Engine

Decision Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior differs from the Hive engine.
- Blaze engine. Processing behavior is the same for the Hive engine, the Blaze engine, and the Data Integration Service.

Spark Engine

You must configure the Decision transformation properties to be partitionable.

Related Links

10.2.2 Decision Transformation for the Spark and Blaze Engines

10.2.1 Decision Transformation for the Hive Engine

Expression Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior differs from the Hive engine.
- Blaze engine. Processing behavior is the same as the Hive engine.

Spark Engine

If an expression results in numerical errors, such as division by zero or SQRT of a negative number, it returns a null value.

Related Links

10.2.2 Expression Transformation for the Spark and Blaze Engines

10.2.1 Expression Transformation for the Hive Engine

Filter Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior is the same for the Hive engine, the Spark engine, and the Data Integration Service.
- Blaze engine. Processing behavior differs from the Hive engine.

Blaze Engine

When a mapping contains a Filter transformation on a partitioned column of a Hive source, the Blaze engine can read only the partitions that contain data that satisfies the filter condition. To push the filter to the Hive source, configure the Filter transformation to be the next transformation in the mapping after the source.

Related Links

10.2.2 Filter Transformation on the Spark and Blaze Engines

10.2.1 Filter Transformation on the Hive Engine

Java Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior differs from the Hive engine.
- Blaze engine. Processing behavior differs from the Hive engine.

Spark engine

Consider the following Spark engine processing differences from the Hive engine:

- The partitionable property must be enabled, as the transformation cannot run in one partition.
- If the transformation scope is row, you must enable the stateless advanced property.
- The Spark engine supports precision up to microseconds. If a date/time value contains nanoseconds, the trailing digits are truncated.
- Mapping validation fails in the following situations:
  - You select a port of a complex data type as the partition or sort key.
  - You enable nanosecond processing in date/time and the Java transformation contains a port of complex data type with an element of a date/time type. For example, a port of type `array<data/time>` is not valid if you enable nanosecond processing in date/time.
  - You enable high precision and the transformation contains a port of a decimal data type.
  - You enable high precision and the transformation contains a complex data type with an element of a decimal data type.

Related Links

10.2.2 Java Transformation on the Spark and Blaze Engines

10.2.1 Java Transformation on the Hive Engine

Joiner Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior differs from the Hive engine.
- Blaze engine. Processing behavior differs from the Hive engine.

Spark Engine

Mapping validation fails if the join condition is of binary data type or contains binary expressions.

Blaze Engine

Mapping validation fails in the following situations:

- The transformation contains an inequality join and a map-side join is disabled.
- The transformation expression references an unconnected Lookup transformation.

Map-side join is disabled when the Joiner transformation is configured for a detail outer join or a full outer join.

Related Links

10.2.2 Joiner Transformation on the Spark and Blaze Engines

10.2.1 Joiner Transformation on the Hive Engine

Key Generator Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Labeler Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Lookup Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior differs from the Hive engine.
- Blaze engine. Processing behavior differs from the Hive engine.

Spark Engine

Mapping validation fails in the following situations:

- Case sensitivity is disabled.
- The lookup condition contains a binary data type.

Mappings fail when a relational lookup has a custom query. Consider the following guidelines to configure custom queries:

- To override the default query in a mapping with an advanced query, you must define a mapping parameter and set its value to $CONDITIONS. You must then include $CONDITIONS in the WHERE clause of the custom query.
- If you define a custom query, you must verify that the metadata of the custom query matches the metadata of the source object. Otherwise, Sqoop might write blank values to the target.
- When you enable OraOop and configure an advanced query to read data from an Oracle source through Sqoop, the mapping fails on the Spark engine.

For more information, see the Knowledge Base article
HOW TO: Configure a Custom Query for a Sqoop based execution.

Consider the following restrictions related to configuring multiple matches:

- If you choose to return the first, last, or any value on multiple matches, the Spark engine returns any value.
- If you configure the transformation to report an error on multiple matches, the Spark engine drops the duplicate rows and does not include the rows in the logs.

Blaze Engine

- Mapping validation fails if the cache is configured to be shared, named, persistent, dynamic, or uncached. The cache must be a static cache.
- If you add a data object that uses Sqoop as a Lookup transformation in a mapping, the Data Integration Service does not run the mapping through Sqoop. It runs the mapping through JDBC.

Related Links

10.2.2 Lookup Transformation on the Spark and Blaze Engines

10.2.1 Lookup Transformation on the Hive Engine

Match Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior is the same as the Hive engine.
- Blaze engine. Processing behavior is the same as the Hive engine.

Related Links

10.2.2 Match Transformation for the Spark and Blaze Engines

10.2.1 Match Transformation for the Hive Engine

Merge Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Normalizer Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Parser Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Python Transformation

Consider the processing differences between the Data Integration Service and the following engines:

- Spark engine. Supported with restrictions.
- Blaze engine. Not supported for this transformation.

Note: The Hive engine did not support the Python transformation.

Spark Engine

Mapping validation fails if a user-defined default value is assigned to an output port.

Mapping execution fails in the following situations:

- An output port is not assigned a value in the Python code.
- The data types in corresponding input and output ports are not the same, and the Python code does not convert the data type in the input port to the data type in the output port.

The Data Integration Service does not validate Python code.

Related Links

10.2.2 Python Transformation for the Spark Engine

Rank Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior differs from the Hive engine.
- Blaze engine. Processing behavior differs from the Hive engine.

Spark Engine

Mapping validation fails if the rank port is of binary data type.

Blaze Engine

Data cache optimization is different on the Blaze engine for the Rank transformation. The data cache for the Rank transformation is optimized to use variable length to store binary and string data types that pass through the Rank transformation. The optimization is enabled for record sizes up to 8 MB. If the record size is greater than 8 MB, variable length optimization is disabled.

When variable length is used to store data that passes through the Rank transformation in the data cache, the Rank transformation is optimized to use sorted input and a pass-through Sorter transformation is inserted before the Rank transformation in the run-time mapping. To view the Sorter transformation, view the optimized mapping or view the execution plan in the Blaze validation environment.

During data cache optimization, the data cache and the index cache for the Rank transformation are set to Auto. The sorter cache for the Sorter transformation is set to the same size as the data cache for the Rank transformation. To configure the sorter cache, you must configure the size of the data cache for the Aggregator transformation.

Related Links

10.2.2 Rank Transformation for the Spark and Blaze Engines

10.2.1 Rank Transformation for the Hive Engine

Router Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Sequence Generator Transformation

Consider the processing differences between the Data Integration Service and the following engines:

- Spark engine. Processing behavior differs from the Data Integration Service.
- Blaze engine. Processing behavior differs from the Data Integration Service.

Note: The Hive engine did not support the Sequence Generator transformation.

Spark Engine

The Sequence Generator transformation does not maintain row order in output data. If you enable the Maintain Row Order property on the transformation, the Data Integration Service ignores the property.

Blaze Engine

A mapping with a Sequence Generator transformation consumes significant resources when the following conditions are true:

- You configure the transformation to maintain row order.
- The mapping runs in a single partition.

Related Links

10.2.2 Sequence Generator Transformation for the Spark and Blaze Engines

Sorter Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior is the same as the Hive engine.
- Blaze engine. Processing behavior differs from the Hive engine.

Blaze Engine

Mapping validation fails if the target is configured to maintain row order and the Sorter transformation is not connected directly to a flat file target.

If a Sorter transformation is inserted before an Aggregator or Rank transformation to optimize the Aggregator or Rank data cache, the size of the sorter cache is the same size as the data cache for the Aggregator or Rank transformation. To configure the sorter cache, you must configure the size of the data cache for the Aggregator or Rank transformation.

The Blaze engine can perform a global sort in the following situations:

- The Sorter transformation is connected directly to flat file targets.
- The target is configured to maintain row order.
- The sort key is not a binary data type.

If any of the conditions are not true, the Blaze engine performs a local sort.

Related Links

10.2.2 Sorter Transformation on the Spark and Blaze Engines

10.2.1 Sorter Transformation on the Hive Engine

Standardizer Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Union Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Update Strategy Transformation

Consider the processing differences between the Hive engine and the following engines:

- Spark engine. Processing behavior differs from the Hive engine.
- Blaze engine. Processing behavior differs from the Hive engine.

Spark Engine

Consider the following general restrictions:

- The Update Strategy transformation does not forward rejected rows to the next transformation.
- If the Update Strategy transformation receives multiple update rows for the same primary key value, the transformation selects one random row to update the target.
- If multiple Update Strategy transformations write to different instances of the same target, the target data might be unpredictable.
- If you configure the Update Strategy expression to update partitioning or bucketing columns, the mapping ignores the Hive MERGE option and does not update the columns.

Mapping validation fails in the following situations:

- The target is an external ACID table.
- The target does not contain a connected primary key.
- The Hive target property to truncate the target table at run time is enabled.
- The Hive target property to create or replace the target table at run time is enabled.

Mappings fail in the following situations:

- You configure the Hive MERGE option and a single row for delete or update does not match multiple rows in the target.
- The target is not enabled for transactions.

Consider the following configuration requirements to use a Hive target table with an Update Strategy transformation:

- Create the Hive target table with the following clause in the Hive Data Definition Language: TBLPROPERTIES ("transactional"="true").
- Verify that the following properties are configured in the hive-site.xml configuration set associated with the Hadoop connection:

```
- hive.support.concurrency true
- hive.enforce.bucketing true
- hive.exec.dynamic.partition.mode nonstrict
- hive.txn.manager org.apache.hadoop.hive.ql.lockmgr.DbTxnManager
```

- hive.compactor.initiator.on true

- hive.compactor.worker.threads 1

Blaze Engine

Consider the following general restrictions:

- If multiple Update Strategy transformations write to different instances of the same target, the target data might be unpredictable.
- The Blaze engine executes operations in the following order: deletes, updates, inserts. It does not process rows in the same order as the Update Strategy transformation receives them.

Consider the following configuration requirements to use a Hive target table with an Update Strategy transformation:

- Create the Hive target table with the following clause in the Hive Data Definition Language: TBLPROPERTIES ("transactional"="true").
- Verify that the following properties are configured in the hive-site.xml configuration set associated with the Hadoop connection:

  - hive.support.concurrency true

  - hive.enforce.bucketing true

  - hive.exec.dynamic.partition.mode nonstrict

  - hive.txn.manager org.apache.hadoop.hive.ql.lockmgr.DbTxnManager

  - hive.compactor.initiator.on true

  - hive.compactor.worker.threads 1

Related Links

10.2.2 Update Strategy Transformation for the Spark and Blaze Engines

10.2.1 Update Strategy Transformation for the Hive Engine

Weighted Average Transformation

There are no processing differences between the engines in the Hadoop environment and the Data Integration Service.

Window Transformation

Consider the processing differences between the Data Integration Service and the following engines:

- Spark engine. Processing behavior differs from the Data Integration Service.
- Blaze engine. Not supported for this transformation.

Note: The Hive engine did not support the Window transformation.

Related Links

10.2.2 Window Transformation for the Spark Engine