



Informatica® Mass Ingestion
January 2023

Troubleshooting

© Copyright Informatica LLC 2019, 2023

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, Informatica Cloud, Informatica Intelligent Cloud Services, PowerCenter, PowerExchange, and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

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: 2023-01-09

Table of Contents

Preface 4

Chapter 1: Troubleshooting..... 5

 Troubleshooting an application ingestion task. 5

 Troubleshooting a database ingestion task. 6

 Troubleshooting a streaming ingestion task. 10

Index..... 12

Preface

Read *Troubleshooting* to get information about troubleshooting errors in the Mass Ingestion service.

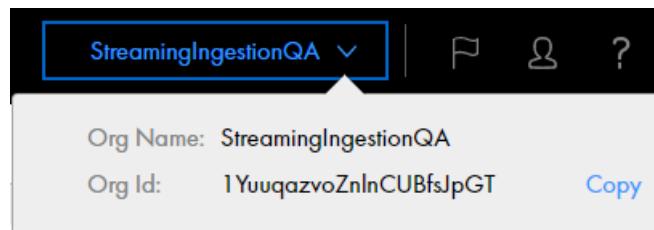
CHAPTER 1

Troubleshooting

Use the following sections to troubleshoot errors in Mass Ingestion.

Note: To get support for Mass Ingestion, you might need to give your organization ID to Informatica Global Customer Support. You can find your organization ID through the **Organization** menu in the upper right corner.

The following image shows the **Organization** menu:



To copy the organization ID, click the **Copy** option that appears when you hover the cursor to the right of the **Org ID** field.

You can also find your organization ID on the **Organization** page in Administrator.

Troubleshooting an application ingestion task

If you change the unsupported data type of a source field to a supported data type, the change might not be replicated to the target.

This problem occurs when the **Modify field** schema drift option is set to **Replicate** and the **Add field** option is set to **Ignore**.

When you deploy an application ingestion job, Mass Ingestion Applications does not create target columns for the source fields of unsupported data types. If you change the unsupported data type to a supported data type for the source field, Mass Ingestion Applications processes the modification made to the source field but does not replicate the change to the target. When Mass Ingestion Applications tries to add a field with the supported data type to the target, the operation is ignored because the schema drift option **Add column** is set to **Ignore**.

To handle this situation, perform the following steps:

1. On the **Schedule and Runtime Options** page in the application ingestion task wizard, in the **Schema Drift Options** section, set the **Add field** option to **Replicate**.

2. Change the source field data type to a supported data type again so that the application ingestion job can detect the schema change.

Note: The application ingestion job does not propagate the field values that were added prior to changing the data type of the source field.

3. If you want to propagate all the values from the source field to the target, resynchronize the target table with the source.

If you change a primary key constraint on the source, Mass Ingestion Applications stops processing the source object on which the DDL change occurred.

This problem occurs if you add or drop a primary key constraint, or if you add or drop a field from an existing primary key.

To resume processing the source object for combined initial and incremental jobs, resynchronize the target table with the source.

To resume processing the source object for incremental jobs, perform the following steps:

1. On the **Source** page of the application ingestion task wizard, add an object selection rule to exclude the source object.
2. Redeploy the task.
Mass Ingestion Applications deploys the modified task and deletes the information about the primary keys of the excluded object.
3. Edit the task again to delete the object selection rule that excluded the source object.
4. Redeploy the task.

An application ingestion initial load job that processes many source objects and has a Google BigQuery target might fail with the following error:

`The job has timed out on the server. Try increasing the timeout value.`

This problem occurs when the job is configured to process many source objects and the connection configured for the Google BigQuery target times out before completing the processing of the source objects. To resolve this issue, you must increase the timeout interval for the Google BigQuery V2 connection configured for the target.

To increase the timeout interval for the connection, perform the following steps:

1. In Administrator, open the Google BigQuery V2 connection associated with the application ingestion job in edit mode.
2. In the **Provide Optional Properties** field, set the timeout property to the required timeout interval in seconds. Use the following format:

```
"timeout": "<timeout_interval_in_seconds>"
```
3. Save the connection.
4. Redeploy the application ingestion job.

Troubleshooting a database ingestion task

If you change an unsupported data type of a source column to a supported data type, the change might not be replicated to the target.

This problem occurs when the **Modify column** schema drift option is set to **Replicate** and the **Add column** option is set to **Ignore**.

Mass Ingestion Databases does not create target columns for source columns that have unsupported data types when you deploy a task. If you change the unsupported data type to a supported data type for the source column later, Mass Ingestion Databases processes the modify column operation on the source but does not replicate the change to the target. When Mass Ingestion Databases tries to add a column with the supported data type to the target, the operation is ignored because the schema drift option **Add column** is set to **Ignore**.

To handle this situation, perform the following steps:

1. On the **Schedule and Runtime Options** page in the database ingestion task wizard, under **Schema Drift Options**, set the **Add column** option to **Replicate**.
2. Change the source column data type to a supported data type again so that the database ingestion job can detect this schema change.
The database ingestion job processes the DDL operation and creates the new target column.
Note: The database ingestion job does not propagate the column values that were added prior to changing the source column data type.
3. If you want to propagate all of the values from the source column to the target, resynchronize the target table with the source.

If you change a primary key constraint on the source, Mass Ingestion Databases stops processing the source table on which the DDL change occurred.

This problem occurs if you add or drop a primary key constraint, or if you add or drop a column from an existing primary key.

To resume processing the source table for combined initial and incremental jobs, resynchronize the target table with the source.

To resume processing the source table for incremental jobs, perform the following steps:

1. On the **Source** tab in the database ingestion task definition, add a table selection rule to exclude the source table.
2. Redeploy the task.
Mass Ingestion Databases deploys the edited task and deletes the information about the primary keys of the excluded table.
3. Edit the task again to delete the table selection rule that excluded the source table.
4. Redeploy the task.

If a DDL column-level change causes a source table subtask to stop or be in error and then you resume the database ingestion job, the expected change in the table state is delayed.

If a DDL column-level change on a source table causes a table subtask to stop or be in error and then you resume the database ingestion job, the state of the table subtask might remain unchanged until a DML operation occurs on the table. For example, if you set a schema drift option to **Stop Table** for an incremental or initial and incremental database ingestion task and then deploy and run the job, when a DDL change occurs on a source table, the job monitoring details shows the table subtask to be in the Error state. If you stop the job and then resume it with a schema drift override to replicate the DDL change, the table subtask temporarily remains in the Error state until the first DML operation occurs on the source table.

Mass Ingestion Databases failed to deploy a task that has a Snowflake target with the following error:

```
Information schema query returned too much data. Please repeat query with more selective predicates.
```

This error occurs because of a known Snowflake issue related to schema queries. For more information, see the Snowflake documentation.

In Mass Ingestion Databases, the error can cause the deployment of a database ingestion task that has a Snowflake target to fail when a large number of source tables are selected.

To handle the deployment failure, drop the target tables. Then update the database ingestion task to select fewer source tables for generating the target tables. Then try to deploy the task again.

A database ingestion job that runs on Linux ends abnormally with the following out-of-memory error:

```
java.lang.OutOfMemoryError: unable to create new native thread
```

The maximum number of user processes that is set for the operating system might have been exceeded. If the Linux ulimit value for maximum user processes is not already set to **unlimited**, set it to **unlimited** or a higher value. Then resume the job.

If you copy an asset to another location that already includes an asset of the same name, the operation might fail with one of the following errors:

```
Operation succeeded on 1 artifacts, failed on 1 artifacts.  
Operation did not succeed on any of the artifacts.
```

If you try to copy an asset to another location that already has an asset of the same name, Mass Ingestion Databases displays a warning message that asks if you want to keep both assets, one with a suffix such as "- Copy 1". Note that when you choose to keep both assets, Mass Ingestion Databases validates the name length to ensure that it will not exceed the maximum length of 50 characters after the suffix is added. If the name length will exceed 50 characters, the copy operation will fail. In this case, you must copy the asset to another location, rename the copy, and then move the renamed asset back to the original location.

Schema drift options on the Schedule and Runtime Options page do not match the options in the job log

In the Mass Ingestion Databases Spring 2020 April release, the order of the **Schema Drift** options displayed on the **Schedule and Runtime Options** page changed but their values remained in the original order. If you use a database ingestion task that was created after the Spring 2020 April release and before the Fall 2020 December release, the schema drift options that the associated job uses at runtime, as reported under "schemaChangeRules" in the job log, might not match the **Schema Drift** options displayed on the **Runtime Options** page. In this case, reset each of the **Schema Drift** options on the **Runtime Options** page and save the task again.

A Kafka consumer ends with one of the following errors:

```
org.apache.avro.AvroTypeException: Invalid default for field meta_data: null not a  
{ "type": "array" } ...  
org.apache.avro.AvroTypeException: Invalid default for field header: null not a  
{ "type": "record" } ...
```

This error might occur because the consumer has been upgraded to a new Avro version but still uses the Avro schema files from the older version.

To resolve the problem, use the new Avro schema files that Mass Ingestion Databases provides.

A database ingestion job that propagates incremental change data to a Kafka target that uses Confluent Schema Registry fails with the following error:

```
io.confluent.kafka.schemaregistry.client.rest.exceptions.RestClientException:  
Register operation timed out; error code: 50002
```

This problem might occur when the job is processing many source tables, which requires Confluent Schema Registry to process many schemas. To resolve the problem, try increasing the value of the Confluent Schema Registry **kafkastore.timeout.ms** option. This option sets the timeout for an operation on the Kafka store. For more information, see the Confluent Schema Registry documentation.

Subtasks of a database ingestion job that has a Google BigQuery target fail to complete initial load processing of source tables with the following error:

```
The job has timed out on the server. Try increasing the timeout value.
```


This problem occurs when the job is configured to process many source tables and the Google BigQuery target connection times out before initial load processing of the source tables is complete. To resolve this problem, increase the timeout interval in the Google BigQuery V2 target connection properties.

1. In Administrator, open the Google BigQuery V2 connection that is associated with the database ingestion job in Edit mode.
2. In the **Provide Optional Properties** field, set the timeout property to the required timeout interval in seconds. Use the following format:

```
"timeout": "<timeout_interval_in_seconds>"
```

3. Save the connection.
4. Redeploy the database ingestion task.

A database ingestion task with an Amazon Redshift target returns one of the following errors during deployment:

```
Mass Ingestion Databases could not find target table 'table_name' which is mapped to  
source table 'table_name' when deploying the database ingestion task.
```

```
com.amazon.redshift.util.RedshiftException: ERROR: Relation "table_name" already  
exists
```

This problem occurs because Amazon Redshift reads table and column names as lowercase by default.

To prevent this error, you can set the `enable_case_sensitive_identifier` parameter to "true" when configuring the database parameter group. For more information about this parameter, see the AWS Amazon Redshift documentation at

https://docs.aws.amazon.com/redshift/latest/dg/r_enable_case_sensitive_identifier.html.

Deployment of a database ingestion task fails if the source table or column names include multibyte or special characters and the target is Databricks Delta.

When a new Databricks Delta target table is created during deployment, an entry is added to the Hive metastore that Databricks Delta uses. The Hive metastore is typically a MySQL database. More specifically, column names are inserted into the `TABLE_PARAMS` field of the metastore. The charset collation of the `PARAM_VALUE` from `TABLE_PARAMS` is `latin1_bin`, and the charset is `latin1`. This charset does not support Japanese characters. To resolve the problem, create an external metastore with `UTF-8_bin` as the collation and `UTF-8` as the charset. For more information, see the Databricks Delta documentation at

<https://docs.microsoft.com/en-us/azure/databricks/kb/metastore/jpn-char-external-metastore> and

<https://kb.databricks.com/metastore/jpn-char-external-metastore.html>.

A database ingestion initial load task with an Azure Synapse Analytics target and an Oracle, PostgreSQL, or SAP HANA source that includes tables with Japanese data might write the Japanese data as ??? characters to the target columns

To resolve this issue, edit the task to add custom data type mappings that map the following source data types to target data types:

- For Oracle sources:
 `VARCHAR2(25) -> NVARCHAR(MAX)`
 `VARCHAR2 (152) -> NVARCHAR(MAX)`
- For PostgreSQL sources:
 `CHARACTER VARYING(25) -> NVARCHAR(MAX)`
 `CHARACTER VARYING(125) -> NVARCHAR(MAX)`
- For SAP HANA sources:
 `TEXT(25) -> NVARCHAR(MAX)`
 `TEXT(125) -> NVARCHAR(MAX)`

A database ingestion initial load job that has a SQL Server source with XML data and a Microsoft Azure Synapse Analytics target fails

When you run a database ingestion job that has a SQL Server source with XML column data of 500,000 or more single-byte characters and a Microsoft Azure Synapse Analytics target, subtasks might fail when Synapse Analytics tries to process SQL queries for creating the target tables. Before writing data to the target tables, Mass Ingestion Databases truncates XML data to 500,000 bytes by default and adds *x* bytes of auxiliary metadata. Synapse Analytics stores each source character as 2 bytes and has a maximum row size of 1000000 bytes. As a result, the number of bytes in a row to be written to a target table can be greater than the maximum target row size. In this case, the subtask fails and a trace message reports information such as:

```
Unexpected error encountered filling record reader buffer: HadoopExecutionException:  
The size of the schema/row at ordinal 1 is 1000050 bytes. It exceeds the maximum  
allowed row size of 1000000 bytes for Polybase.
```

To correct the problem, determine an appropriate lower truncation point and specify it in the unloadClobTruncationSize custom property on the **Target** page of the task wizard. If only one XML column occurs in a row, decrease the truncation point by the difference between <actual schema/row size> and maximum row size. For example, based on the preceding sample message, you'd calculate the lower truncation point for a row with a single XML column as 500000 - 50, or 499950 bytes.

A database ingestion incremental load job with a Db2 for z/OS source ends abnormally when processor resource limits are exceeded during Informatica stored procedure processing

If you use Db2 DSNRLSTxx resource limit table to limit the amount of processor resources that are used by SQL operations, such as SELECT, INSERT, UPDATE, and DELETE, on a z/OS source system, database ingestion incremental load jobs with Db2 for z/OS sources might end abnormally. The jobs end if the default or set resource limits are not large enough to accommodate long-running processing of the WLM stored procedure that database ingestion jobs use to process captured change data. If abends related to resource limits occur, perform the following steps:

1. Add a row to the resource limit table specifically for the database ingestion packages in your runtime environment. Add the row with the following columns:
 - An AUTHID column with the authentication ID that the database ingestion task uses OR an RLFPDG column with the same package name as the database ingestion task OR both of these columns.
 - An ASUTIME column defined with NULL or with a resource limit that is greater than the default limit.
2. For the changes to the resource limit table to take effect, issue the Db2 -START RLIMIT command.

If you do not have the authority to perform these steps, contact your Db2 DBA or z/OS system programmer.

Troubleshooting a streaming ingestion task

While deploying a streaming ingestion task with Amazon Kinesis source, data loss is encountered when you send data immediately after the task status changes to Up and Running.

Workaround: After deploying the task to run as a job, wait for some time before sending the data.

A streaming ingestion job with Kinesis Streams source which consumes streams from Amazon DynamoDB fails to read the data ingested from Kinesis Streams. The streaming ingestion job runs with Up and Running status, but does not return any error.

Provide the following required permissions to the Amazon DynamoDB user:

```
dynamodb:CreateTable
- dynamodb:DescribeTable
- dynamodb:Scan
- dynamodb:PutItem
- dynamodb:GetItem
- dynamodb:UpdateItem
- dynamodb>DeleteItem
Resource:
- !Join ["", ["arn:aws:dynamodb:*:", !Ref 'AWS::AccountId', ":table/
*"]]
```

Provide the following permissions for Amazon CloudWatch:

```
"Action": "cloudwatch:DescribeAlarms"
"Action": "cloudwatch:PutMetricData"
```

While creating a Kafka connection, if the SSL mode is set as disabled, any other additional connection property values declared are not considered.

To override this issue, declare the additional connection properties in the **Additional Security Properties** field.

When you try to ingest high volume of data to a Kafka target, the streaming ingestion job runs with the error:

```
A message in the stream exceeds the maximum allowed message size of 1048576 byte.
```

You get this error message if the received message size is more than 1 MB, which is the maximum message size that a Kafka server can receive.

When you try to ingest high volume of data to an Amazon Kinesis Firehose or Amazon Kinesis Streams target, the streaming ingestion job runs with the error:

```
INFO - Mon Feb 04 07:01:44 UTC 2019PutKinesisStream[id=0421419e-b24f-4e3f-
ad19-9a1fbc7b0f3c] Failed to publish to kinesis records
StandardFlowFileRecord[uuid=36926fcd-dfed-46f9-ae41-
a6d32c3d5633,claim=StandardContentClaim
[resourceClaim=StandardResourceClaim[id=1549263211727-1, container=default,
section=1], offset=2758,
length=1067926],offset=0,name=testKinesis.2758-1070684.txt,size=1067926] because the
size was greater than 1024000 bytes
```

You get this error message if the received message size is more than 1 MB, which is maximum message buffer size of Amazon Kinesis. Any message of size greater than 1 MB never reaches the target and is lost in the transition.

While deploying a streaming ingestion task with Amazon Kinesis source, data loss is encountered.

Workaround: To override this issue, don't send data when the data flow is in Stopped, Deploying, Edit, or Redeploy state, after deploying the task to run as a job.

INDEX

D

Database Ingestion
troubleshooting [6](#)

O

organization ID
finding [5](#)
organizations
finding your organization ID [5](#)

S

Streaming Ingestion
troubleshooting [10](#)

T

Troubleshooting
database ingestion task [6](#)
streaming ingestion task [10](#)