



Informatica® Data Ingestion and Replication  
July 2024

# Data Ingestion and Replication Command-Line Interface

© Copyright Informatica LLC 2019, 2024

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](mailto:infa_documentation@informatica.com).

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

Publication Date: 2024-07-12

# Table of Contents

- Preface ..... 4**
  - Informatica Resources. .... 4
    - Informatica Documentation. .... 4
    - Informatica Intelligent Cloud Services web site. .... 4
    - Informatica Intelligent Cloud Services Communities. .... 4
    - Informatica Intelligent Cloud Services Marketplace. .... 4
  - Data Integration connector documentation. .... 5
  - Informatica Knowledge Base. .... 5
  - Informatica Intelligent Cloud Services Trust Center. .... 5
  - Informatica Global Customer Support. .... 5
- Chapter 1: Data Ingestion and Replication Command-Line Interface..... 6**
  - Data Ingestion and Replication CLI overview. .... 6
  - Required user roles and permissions. .... 7
  - Running the Data Ingestion and Replication CLI. .... 7
  - Data Ingestion and Replication CLI common options. .... 8
  - Data Ingestion and Replication CLI commands. .... 9
  - Input file format. .... 17
    - Default values. .... 30
- Index..... 31**

# Preface

Use *Data Ingestion and Replication Command-Line Interface* to get information about the CLI commands you can use to create, view, and deploy tasks and to list, manage, and monitor application ingestion and replication and database ingestion and replication jobs.

## 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](mailto: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](mailto: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

# Data Ingestion and Replication Command-Line Interface

You can run commands in the Data Ingestion and Replication Command Line Interface (CLI) to create, view, and deploy application ingestion and replication and database ingestion and replication tasks and to list, start, stop, redeploy, undeploy, and monitor ingestion jobs.

## Data Ingestion and Replication CLI overview

The Data Ingestion and Replication Command-Line Interface (CLI) files are installed in the *Informatica Cloud Secure Agent installation/apps/Database\_Ingestion/version/dbmicli* directory. Informatica recommends that you copy the dbmicli directory to another Linux or Windows server to preserve the CLI configuration files that you will create across Data Ingestion and Replication upgrades.

You can use the Data Ingestion and Replication CLI commands to perform the following tasks for application ingestion and replication and database ingestion and replication jobs and tasks:

- Create a new task or create tasks based on one or more existing tasks.
- Deploy a task.
- View a task.
- View status information for your organization's jobs or a specific job.
- Start or resume a job.
- Stop or abort a job.
- Replace an existing task definition with an updated version of the task.
- Undeploy a job.
- Redeploy a job.
- Resynchronize the target with the source in a combined load job.
- Monitor a job.

For database ingestion and replication tasks only, you can use the CLI to enable the database CDC option on the source tables.

Commands are also available to get information about the CLI:

- Get help information about the Data Ingestion and Replication CLI commands.
- Get the current version of the CLI.

# Required user roles and permissions

For application ingestion and replication tasks and database ingestion and replication tasks, you must have certain roles and permissions to perform tasks by using the CLI. Configure user roles and privileges in Administrator.

The following system-defined user roles are typically required:

- To list all of the application ingestion and replication and database ingestion and replication jobs in your organization, make sure you have one of the following system-defined roles or a custom role that has at least the same privileges:
  - Admin
  - Data Integration Task Executor
  - Designer
  - Monitor

If you do not have any of these roles, you can list only the jobs that you own.

- To start, resume, stop, or abort jobs for an ingestion and replication task, make sure you have one of the following system-defined roles or a custom role that has at least the same privileges:
  - Admin
  - Designer

For application ingestion and replication and database ingestion and replication tasks, you must also have the execute permission on the task.

- To create ingestion and replication tasks, make sure you have one of the following system-defined roles or a custom role that has at least the same privileges:
  - Admin
  - Designer
- To deploy ingestion and replication tasks, make sure you have one of the following system-defined roles or a custom role that has at least the same privileges:
  - Admin
  - Designer
  - Deployer

For application ingestion and replication and database ingestion and replication tasks, you must also have the execute permission on the task.

## Running the Data Ingestion and Replication CLI

If you copied the Data Ingestion and Replication dbmicli files to a Linux or Windows server that is remote from the Secure Agent, as recommended, run the CLI from there. If the dbmicli files are located only on the Secure Agent system, run the CLI from the *Informatica Cloud Secure Agent installation/apps/Database\_Ingestion/version/dbmicli* directory.

To start the CLI, run one of the following scripts:

- On Linux, run dbmicli.sh
- On Windows, run dbmicli.bat

## Syntax

dbmicli [common options] command [command-specific options]

# Data Ingestion and Replication CLI common options

You must provide the information about a user name, password, and pod URL for the CLI in one of the following ways:

- Specify the values in the command-line options. Syntax is:

```
Secure_Agent_installation_directory/apps/Database_Ingestion/version/dbmicli/  
dbmicli.bat -P pod_URL -u username -p password CLI_command
```

- Define the values in the environment variables. Syntax is:

```
[root@root_user_ID dbmicli]# export DBMIPOD=pod_URL  
[root@root_user_ID dbmicli]# export DBMIUSER=username  
[root@root_user_ID dbmicli]# export DBMIPASSWORD=password  
[root@root_user_ID dbmicli]# ./dbmicli.sh CLI_command
```

- Enter them manually in interactive mode. To use the interactive mode, enter the username and password after running the following command:

```
Secure_Agent_installation_directory/apps/Database_Ingestion/version/dbmicli/  
dbmicli.bat -P pod_URL CLI_command
```

The following table describes the common options of the Data Ingestion and Replication CLI:

Common Option	Environment Variable	Description
-u, --user	DBMIUSER	Name of a Data Ingestion and Replication user.
-p, --password	DBMIPASSWORD	Password for the Data Ingestion and Replication user.
-P, --pod	DBMIPOD	Domain name for your Informatica Intelligent Cloud Services pod. <b>Important:</b> You can get the domain name of your pod from the web address of your POD. From the web address, omit the first part of the domain name (fourth level domain) and the path. For example, if you have the web address <code>https://usw1.dm1-us.informaticacloud.com/cloudshell/showProducts</code> , the domain name of your pod is <code>dm1-us.informaticacloud.com</code> .

The Data Ingestion and Replication CLI tries to get the user name, password, and pod URL in the following order:

- Get the values from the command-line options.
- If a command-line option is not specified, get the value from the corresponding environment variable, if defined.
- If the environment variable is not defined, the CLI switches to interactive mode and prompts a user to enter the value.

**Important:** If the pod URL is not specified with the -P option and is not defined in the DBMIPOD environment variable, the CLI ends with an error without switching to the interactive mode.



# Data Ingestion and Replication CLI commands

The following table describes the commands of the Data Ingestion and Replication CLI:

**Note:** Unless otherwise indicated, the CLI commands apply to both Application Ingestion and Replication and Database Ingestion and Replication.

**Note:** In the CLI, the naming convention of the job name is `<taskname>` and `<job_instance_number>` separated by an underscore (\_):

`<taskname>_<job_instance_number>`

The naming convention is different from the Informatica Intelligent Cloud Services user interface, where the `<taskname>` and `<job_instance_number>` are separated by a hyphen (-):

`<taskname>-<job_instance_number>`

Command	Description
<code>--version, -v</code>	Displays the version of the Data Ingestion and Replication CLI.
<code>help, --help, -h</code>	Displays a list of all of the Data Ingestion and Replication CLI options and commands.
<code>help command</code>	Displays a description of the specified command.
<code>job start job_name</code> <code>[-waitForCompletion]</code> <code>[-period=requestsPeriod]</code> <code>[-retryCount=retryCount]</code>	<p>Starts the job with the specified job name. Optionally monitors the job and resumes the job if it has the Failed status.</p> <p>Syntax is:</p> <pre>job start job_name [--waitForCompletion] [--period=requestsPeriod] [--retryCount=retryCount]</pre> <p>Valid options are:</p> <ul style="list-style-type: none"><li>- <code>waitForCompletion</code>. Waits for a Completed, Aborted, or Stopped status, or resumes the job if it has the Failed status.</li><li>- <code>period=requestsPeriod</code>. The time, in milliseconds, between job status checks. The default value is 5000.</li><li>- <code>retryCount=retryCount</code>. The number of attempts that can be made to resume the job before stopping job monitoring. The default value is 10.</li></ul> <p>For jobs with a <code>job start --waitForCompletion</code> command, the exit codes indicate the current status of the job. To view the exit code for the job, run one of the following commands:</p> <ul style="list-style-type: none"><li>- On Windows, run <code>echo %ERRORLEVEL%</code>.</li><li>- On Linux, run <code>echo \$?</code>.</li></ul> <p>The following exit codes can be returned for a <code>job start --waitForCompletion</code> command to indicate the job status:</p> <ul style="list-style-type: none"><li>- exit code 0 - Completed</li><li>- exit code 1 - An exception occurred</li><li>- exit code 2 - Failed and there is no retry left</li><li>- exit code 3 - Stopped</li><li>- exit code 4 - Aborted</li><li>- exit code 5 - Undeployed</li></ul>
<code>job resume job_name</code>	Resumes the job that has the specified job name.
<code>job stop job_name</code>	Stops the job that has the specified job name.
<code>job abort job_name</code>	Aborts the job that has the specified job name.

Command	Description
<code>job status</code> <code>[-f field_names]</code> <code>[-taskType task_type]</code>	<p>Displays information about your organization's jobs if you have the required permissions. If you do not have sufficient privileges, no jobs are listed.</p> <p>Syntax is:</p> <pre>job status [-f field1,field2,...] [taskType=task_type]</pre> <p>Use the -f option to specify the comma-separated list of fields that you want the CLI to return for each job. By default, the CLI returns the following string of values for all jobs:</p> <pre>name;status;startTime;duration</pre> <p>To customize the fields, enter the -f option with one or more of the following field names:</p> <ul style="list-style-type: none"> <li>- name. The job name.</li> <li>- state. The status of the job. For descriptions of status values, see "Monitoring Ingestion Jobs" &gt; "Job Properties" in the Data Ingestion and Replication online help or documentation on the Doc Portal.</li> <li>- startTime. The date and time when the job was deployed.</li> <li>- endTime. The date and time when the job ended because it completed processing, was stopped, or failed.</li> <li>- undeployTime. The date and time when the job was undeployed.</li> <li>- duration. The amount of time that the job has run since it was deployed.</li> </ul> <p>For example, the following command lists the names and statuses of all jobs:</p> <pre>C:\DBMICLI&gt;dbmicli.bat -P MyPod --user user_name --password user_password job status -f name,state</pre> <p>Use the taskType option to specify the type of ingestion and replication task, which can be either appmi for application ingestion and replication or dbmi for database ingestion and replication. If you omit this option, both task types are included.</p> <p>Example output:</p> <pre>testGCS_2_1611;Completed testGCS_0403_1610;Undeployed testGCS_0403_1607;Undeployed testGCS_1528;Up and Running</pre>
<code>job status [-f field_names]</code> <code>job_name</code>	<p>Displays information for a job with a specific job name.</p> <p>Syntax is:</p> <pre>job status [-f field1,field2,...] job_name</pre> <p>You can use the -f option to specify a comma-separated list of fields that you want the CLI to return for the job.</p> <p>Valid fields are:</p> <ul style="list-style-type: none"> <li>- name. The job name.</li> <li>- state. The status of the job. For descriptions of status values, see "Monitoring Ingestion Jobs" &gt; "Job Properties" in the Data Ingestion and Replication online help or documentation on the Doc Portal.</li> <li>- startTime. The date and time when the job was deployed.</li> <li>- endTime. The date and time when the job ended because it completed processing, was stopped, or failed.</li> <li>- undeployTime. The date and time when the job was undeployed.</li> <li>- duration. The amount of time that the job has run since it was deployed.</li> </ul> <p>For example, the following command lists the start time and duration of the job that has the sample_job name:</p> <pre>C:\DBMICLI&gt;dbmicli.bat -P MyPod --user user_name --password user_password job status -f startTime,duration sample_job</pre> <p>Example output:</p> <pre>2021-03-04T05:39:23.000-0800;02:51:32</pre>

Command	Description
<code>job attach <i>job_name</i></code> <code>[-period=<i>requestsPeriod</i>]</code> <code>[-retryCount=<i>retryCount</i>]</code>	<p>Monitors the job with the specified job name and resumes the job if it has the Failed status. The cycle ends when one of the following conditions is met:</p> <ul style="list-style-type: none"> <li>- A Completed, Aborted, or Stopped status is detected for an initial load job.</li> <li>- An Aborted or Stopped status is detected for an incremental load or combined initial and incremental load job.</li> <li>- The current retry count equals or is greater than the specified retryCount.</li> </ul> <p>Syntax is:</p> <pre>job attach <i>job_name</i> [--period=<i>requestsPeriod</i> [--retryCount=<i>retryCount</i>]</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- <code>period=<i>requestsPeriod</i></code>. The time, in milliseconds, between job status checks. The default value is 5000.</li> <li>- <code>retryCount=<i>retryCount</i></code>. The number of attempts that can be made to resume the job before stopping job monitoring. The default value is 10.</li> </ul> <p>The exit codes indicate the current status of the job. To view the exit code for the job, run one of the following commands:</p> <ul style="list-style-type: none"> <li>- On Windows, run <code>echo %ERRORLEVEL%</code>.</li> <li>- On Linux, run <code>echo \$?</code>.</li> </ul> <p>The following exit codes can be returned for a job attach command to indicate the job status:</p> <ul style="list-style-type: none"> <li>- exit code 0 - Completed</li> <li>- exit code 1 - An exception occurred</li> <li>- exit code 2 - Failed and there is no retry left</li> <li>- exit code 3 - Stopped</li> <li>- exit code 4 - Aborted</li> <li>- exit code 5 - Undeployed</li> </ul>
<code>job undeploy <i>job_name</i></code>	<p>Undeploys an ingestion job that was previously deployed.</p> <p>Before you attempt to undeploy a job, ensure that it is not running.</p> <p>You might need to undeploy a job when you no longer need to run the job, the job is in the Failed state, or you need to change a connection or property in the associated task that cannot be edited until the job is undeployed.</p> <p>After the job is undeployed, you cannot run it again or redeploy it. If you want to run a job for the associated ingestion and replication task again, you must deploy the task again to create a new job instance.</p>
<code>job redeploy <i>job_name</i></code>	<p>Redeploys a job after you've edited available fields in the associated ingestion and replication task so that the new settings can take effect, without first undeploying the job.</p> <p>You can update the task configuration details only in the user interface.</p>
<code>job resumeWithOptions resync <i>job_name</i></code>	<p>For combined load jobs that have Failed or Running With Warnings state due to a schema drift error, resynchronizes the tables that are in error.</p> <p>Applicable for jobs with sources other than Oracle, Salesforce, or SQL Server.</p> <p><b>Note:</b> This command is not valid if the job is in the Failed state because the task deployment failed.</p>
<code>job resumeWithOptions resyncRefresh <i>job_name</i></code>	<p>For combined load jobs that have an Oracle, Salesforce, or SQL Server source and are in the Stopped, Aborted, or Failed state, resynchronizes the target tables with the latest source table definitions, including any DDL changes. After the target tables are refreshed, the structure of source and target tables match. This command mimics the behavior of the resumeWithOptions resync command.</p> <p><b>Note:</b> This command is not valid if the job is in the Failed state because the task deployment failed.</p>

Command	Description
<code>job resumeWithOptions resyncRetain <i>job_name</i></code>	<p>For combined load jobs that have an Oracle, Salesforce, or SQL Server source and are in the Stopped, Aborted, or Failed state, resynchronizes the same columns that have been processed for CDC, retaining the current structure of the source and target tables. No checks for changes to the source or target table definitions are performed. If source DDL changes affected the source table structure, those changes are not processed.</p> <p><b>Note:</b> This command is not valid if the job is in the Failed state because the task deployment failed.</p>
<code>job resyncSubtask resync <i>job_name</i> --srcSchemaName <i>source_schema_name</i> --srcTableName <i>source_table_name</i></code>	<p>For a subtask that is part of a running combined initial and incremental load job, resynchronizes the target table with the latest source table definition, including any DDL changes. The subtask must be in a state other than Queued or Starting.</p> <p>Applicable for jobs with sources other than Oracle, Salesforce, or SQL Server.</p> <p>Syntax is:</p> <pre>job resyncSubtask resync &lt;job_name&gt; --srcSchemaName &lt;source_schema_name&gt; --srcTableName &lt;source_table_name&gt;</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- srcSchemaName. Required. Source schema name.</li> <li>- srcTableName. Required. Source table name.</li> </ul>
<code>job resyncSubtask resyncRefresh <i>job_name</i> --srcSchemaName <i>source_schema_name</i> --srcTableName <i>source_table_name</i></code>	<p>For a subtask in a running combined initial and incremental load job that has an Oracle, Salesforce, or SQL Server source, resynchronizes the target table with the latest source table definition, including any DDL changes. The subtask must be in a state other than Queued or Starting. After the target table is refreshed, the target table structure matches the current source table structure. This command mimics the behavior of the resyncSubtask resync command.</p> <p>Syntax is:</p> <pre>job resyncSubtask resyncRefresh &lt;job_name&gt; --srcSchemaName &lt;source_schema_name&gt; --srcTableName &lt;source_table_name&gt;</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- srcSchemaName. Required. Source schema name.</li> <li>- srcTableName. Required. Source table name.</li> </ul>
<code>job resyncSubtask resyncRetain <i>job_name</i> --srcSchemaName <i>source_schema_name</i> --srcTableName <i>source_table_name</i></code>	<p>For a subtask in a running combined initial and incremental load job that has an Oracle, Salesforce, or SQL Server source, resynchronizes the same columns that have been processed for CDC, retaining the current structure of the source and target tables. The subtask must be in a state other than Queued or Starting. No checks for changes in the source or target table definitions are performed. If source DDL changes affected the source table structure, those changes are not processed.</p> <p>Syntax is:</p> <pre>job resyncSubtask resyncRetain &lt;job_name&gt; --srcSchemaName &lt;source_schema_name&gt; --srcTableName &lt;source_table_name&gt;</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- srcSchemaName. Required. Source schema name.</li> <li>- srcTableName. Required. Source table name.</li> </ul>

Command	Description
<pre>task create --pathToConfig path_to_configuration_file [--taskName task_name] [--taskLocation task_location] [-taskType task_type] [--deploy]</pre>	<p>Creates an ingestion and replication task by using the information from the input configuration file and optionally deploys the task.</p> <p>Syntax is:</p> <pre>task create --pathToConfig path_to_configuration_file [--taskName task_name][--taskLocation task_location][-taskType task_type][-- deploy]</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- pathToConfig. Required. The path to the YAML input file.</li> <li>- taskName. The task name.</li> <li>- taskLocation. The project or project\folder that will contain the task definition. The default value is Default.</li> <li>- taskType. The type of task, which can be either appmi for application ingestion and replication or dbmi for database ingestion and replication.</li> </ul> <p>If you do not specify the task type in the command, you can include it at the beginning of the YAML file.</p> <p>Default is dbmi.</p> <ul style="list-style-type: none"> <li>- deploy. Deploys the tasks after they are created.</li> </ul> <p>For example, the following command creates a task by using the sample_task.yaml file:</p> <pre>C:\DBMICLI&gt;dbmicli.bat -P MyPod --user user_name --password user_password task create --pathToConfig C:\DBMI_TASKS \sample_task.yaml</pre>
<pre>task createFrom --taskName task_name [--taskLocation task_location] [-taskType task_type] --override path_to_override_file [--deploy]</pre>	<p>Creates a task from another specified task using the field overrides in an override file. Also optionally deploys the task.</p> <p>For example, if the override file specifies the location "project1" to override the Default location, the new task definition will be in the project1 location.</p> <p>Syntax is:</p> <pre>task createFrom --taskName task_name [--taskLocation task_location][--taskType task_type] --override path_to_override_file [--deploy]</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- taskName. Required. The task name.</li> <li>- taskLocation. The project or project\folder that will contain the task definition. The default value is Default.</li> <li>- taskType. The type of task, which can be either appmi for application ingestion and replication or dbmi for database ingestion and replication.</li> </ul> <p>If you do not specify the task type in the command, you can include it at the beginning of the YAML file.</p> <p>Default is dbmi.</p> <ul style="list-style-type: none"> <li>- override. Required. The path to the YAML file that contains the fields that are used to override fields of the original task.</li> <li>- deploy. Deploys the tasks after they are created.</li> </ul>

Command	Description
<code>task multipleCreateFrom</code> <code>--taskName <i>taskName</i></code> <code>--override</code> <code><i>path_to_override_file</i></code> <code>[--taskLocation</code> <code><i>task_location</i>]</code> <code>[--taskType <i>task_type</i>]</code> <code>[--deploy]</code> <code>[--deployTimeout</code> <code><i>wait_time_for_status</i>]</code> <code>[--continueOnFail]</code>	<p>Creates multiple tasks from another task using the field overrides specified in the override file. Also optionally deploys the tasks.</p> <p>Syntax is:</p> <pre>task multipleCreateFrom --taskName <i>task_name</i> --override <i>path_to_override_file</i> [--deploy] [--deployTimeout <i>wait_time_for_status</i>] [--continueOnFail][--taskLocation <i>task_location</i>] [--taskLocation <i>task_location</i>] [--taskType <i>task_type</i>]</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- <code>taskName</code>. Required. The name of task from which you're creating other tasks.</li> <li>- <code>override</code>. Required. The YAML file that contains the field overrides.</li> <li>- <code>taskLocation</code>. The project or project\folder that will contain the new task definitions. The default value is Default.</li> <li>- <code>taskType</code>. The type of task, which can be either appmi for application ingestion and replication or dbmi for database ingestion and replication. If you do not specify the task type in the command, you can include it at the beginning of the YAML file. Default is dbmi.</li> <li>- <code>deploy</code>. Deploys the new tasks after they are created.</li> <li>- <code>deployTimeout</code>. The time interval, in seconds, that Data Ingestion and Replication waits for the deployment status, which can be either deployed or failed, before timing out. You can use this option when you are creating and deploying more than 100 tasks to help keep the Secure Agent running. If a failure occurs, a message is written in the output and to the error log to indicate the status of the job. For example: Failed;Mass Ingestion could not deploy the task because supplemental logging is not enabled for source table 'TEST169'. Valid values are 0 to 86400. If you do not specify this option when you are creating and deploying many tasks, requests are sent asynchronously and do not wait for a response, which can lead to increased resource consumption.</li> <li>- <code>continueOnFail</code>. If the creation of a task fails, continues creating the other tasks. For example, if you create 10 tasks and the second task fails, the other tasks will be created.</li> </ul>

Command	Description
<code>task cdc</code> <code>--taskName=taskName</code> <code>[--taskLocation=Default]</code> <code>--scope=pk all</code> <code>[--execute]</code> <code>[--download] [--path=path]</code> <code>[--msRoleName=role]</code> <code>[--msFileGroupName=group]</code>	<p>For database ingestion and replication tasks, enables the database CDC option on the source tables. You can optionally generate a CDC script and then either run the script or download it and save it in a file.</p> <p>Syntax is:</p> <pre>task cdc --taskName=taskName[--taskLocation=Default] --scope=pk all [--execute] [--download] [--path] [--msRoleName=role] [--msFileGroupName=group]</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- <code>taskName</code>. Required. The task name.</li> <li>- <code>taskLocation</code>. The project or project\folder that will contain the task definition. The default value is Default.</li> <li>- <code>scope</code>. Required. Generate or run a CDC script for either primary key (PK) columns or for all (ALL) columns.</li> <li>- <code>execute</code>. Execute the CDC script.</li> <li>- <code>download</code>. Download the CDC script.</li> <li>- <code>path</code>. An optional path to the CDC script location. You can specify a file or folder. For example, to specify a script file name and location, enter <code>C:\cli\cdc.txt</code>. The CLI then saves the CDC script to the <code>cdc.txt</code> file in the <code>cli</code> directory. To specify a folder only, enter <code>C:\cli\</code>. The CLI then saves the generated CDC script file to the <code>cli</code> folder. If a path value is not specified, the script file is saved to the CLI location and file: <code>cliLocation\cdc_script_taskName_currentTimestamp.txt</code></li> </ul> <p>The following options are valid for Microsoft SQL Server database ingestion and replication sources only:</p> <ul style="list-style-type: none"> <li>- <code>msRoleName</code>. The name of the database role that is used to gate access to change data. If you do not specify a role name, the database does not use the gating role.</li> <li>- <code>msFileGroupName</code>. The name of the filegroup to be used for the change table that is created for the capture. If you do not specify a filegroup name, the change table is located in the default filegroup if the database.</li> </ul>
<code>task deploy</code> <code>[--taskName taskName]</code> <code>[--taskLocation task_location]</code>	<p>Deploys an application ingestion and replication or database ingestion and replication task.</p> <p>Syntax is:</p> <pre>task deploy [--taskName taskName] [--taskLocation task_location]</pre> <p>Valid options are:</p> <ul style="list-style-type: none"> <li>- <code>taskName</code>. The task name.</li> <li>- <code>taskLocation</code>. The project or project\folder that contains the task definition.</li> </ul>

Command	Description
<pre>task replace --pathToConfig <i>yaml_file</i> [--deploy] [--noprompt]</pre>	<p>Replaces a task definition with an updated version of the task, only for undeployed jobs and for tasks that have never been deployed. Optionally, you can include the <code>--deploy</code> option to also deploy the updated task.</p> <p>Syntax is:</p> <pre>task replace --pathToConfig &lt;yaml_file&gt; [--deploy] [--noprompt]</pre> <ul style="list-style-type: none"> <li>- <code>pathToConfig</code>. Points to the CLI YAML configuration file that you updated or created. You cannot update the task name or location.</li> <li>- <code>deploy</code>. Optional. Deploys the updated task definition.</li> <li>- <code>noprompt</code>. Optional. Causes the following confirmation prompt to <i>not</i> be displayed for a replace operation so that automated scripts and tasks can proceed: The "Replace" command overwrites the current task definition with the given configuration file. Type "Y" to confirm your request:</li> </ul> <p>If a CLI YAML configuration file already exists, you can modify it. If you created the task from the task wizard and a CLI configuration file does not exist, you can create the YAML configuration file by using the example template file provided in the DBMI package here:</p> <pre>&lt;secure_agent_location&gt;\apps\Database_Ingestion\&lt;version&gt;\dbmicli \examples</pre> <p><b>Note:</b> No validation or comparison of existing task with the updated configuration file occurs during command execution. Make sure the replacement configuration file is correct.</p>
<pre>task status [-f <i>field_names</i>] [--taskName <i>taskName</i>] [--taskLocation <i>task_location</i>] [--taskType <i>task_type</i>]</pre>	<p>Displays information about ingestion and replication tasks.</p> <p>Syntax is:</p> <pre>task status     [-f <i>field1,field2,...</i>]     [--taskName <i>taskName</i>]     [--taskLocation <i>task_location</i>]     [--taskType <i>task_type</i>]</pre> <p>By default, the CLI returns the following string of values for each task:</p> <pre>name;loadType;sourceConnection;targetConnection;jobName;jobStatus</pre> <p>To modify the fields for which values are returned, use the <code>-f</code> option with one or more of the following field names:</p> <ul style="list-style-type: none"> <li>- <code>name</code>. The task name.</li> <li>- <code>location</code>. The project or project\folder that will contain the task definition.</li> <li>- <code>runtimeEnvironment</code>. The runtime environment in which the task will run.</li> <li>- <code>loadType</code>. The type of load operation that the database ingestion and replication task performs.</li> <li>- <code>sourceConnection</code>. The connection for the source system.</li> <li>- <code>sourceSchema</code>. The source schema that includes the source tables or objects.</li> <li>- <code>targetConnection</code>. The connection for the target type.</li> <li>- <code>targetSchema</code>. The target schema in which the target tables are created.</li> <li>- <code>schedule</code>. The name of a predefined schedule that you want to use for running job instances for an initial load task.</li> <li>- <code>jobName</code>. The job name.</li> <li>- <code>jobStatus</code>. The job status.</li> </ul> <p>To filter the tasks, include one or more of the following options:</p> <ul style="list-style-type: none"> <li>- <code>taskName</code>. The name of the task for which to display information.</li> <li>- <code>taskLocation</code>. The project or project\folder that contains the task definition.</li> <li>- <code>taskType</code>. The type of task, which can be either <code>appmi</code> for application ingestion and replication or <code>dbmi</code> for database ingestion and replication. If you do not specify this option, both ingestion types are included.</li> </ul>



# Input file format

The Data Ingestion and Replication Command-Line Interface uses input files that have the YAML format to create application ingestion and replication or database ingestion and replication tasks. The YAML (.yaml) files are validated for possible errors such as missing required fields or incorrect field names. Examples of .yaml files are available under the dbmicli/examples directory, as a part of secure agent files. Copy an example file under another name and then customize it, as needed.

**Tip:** Informatica recommends that you copy the dbmicli directory that is installed under the Informatica Cloud Secure Agent to a remote location and then add your YAML files in a folder there. This practice prevents subsequent Secure Agent upgrades from overwriting your YAML files. When you issue commands such as task create, you can specify the path to the YAML file that you want to use.

## Initial parameter

You can enter the following parameters at the beginning of the YAML file, before the General section.

### **taskType**

The type of task, which can be either "appmi" for application ingestion and replication tasks or "dbmi" for database ingestion and replication tasks.

If you do not specify the task type in the input file, you can include it in the task create, task createFrom, or task multipleCreateFrom command.

Default is dbmi.

## General section parameters

The general section of the YAML input file specifies some basic information about the task, such as a task name, project or project folder location, and load operation type.

### **name**

A name for the ingestion and replication task.

### **description**

An optional description for the task.

### **location**

The project or project\folder that will contain the task definition.

### **runtimeEnvironment**

The runtime environment in which you want to run the task.

### **type**

The type of load operation that the application ingestion and replication or database ingestion and replication task performs. Valid values are:

- initial
- incremental
- combined (for initial and incremental)

## Source section parameters

The source section of the YAML input file specifies the source options. Unless otherwise indicated, the options pertain to both application ingestion and replication and database ingestion and replication.

### **connection**

**Sources:** All

The name of the connection for the source system.

#### **salesforceAPI**

**Sources:** Salesforce, initial load tasks and combined initial and incremental load tasks.

For application ingestion and replication tasks only, the type of Salesforce API that you want to use to retrieve the source data. Valid values are:

- **Standard (REST) API:** Replicates source fields of Base64 data type.
- **Bulk API 2.0:** Excludes replication of source fields of Base64 data type. By default, Bulk API 2.0 is set as the API type in the YAML file.

#### **schema**

**Sources:** All database ingestion and replication sources except MongoDB

For database ingestion and replication tasks only, the source schema that includes the source tables.

#### **database**

**Sources:** MongoDB

For database ingestion and replication tasks only, the MongoDB database that stores the collections with the source data.

#### **journalName**

**Sources:** Db2 for i, incremental loads

For database ingestion and replication tasks only, the name of the journal that records the changes made to the source tables.

#### **replicationSlotName**

**Sources:** PostgreSQL

For database ingestion and replication tasks only, the unique name of a PostgreSQL replication slot.

#### **replicationPlugin**

**Sources:** PostgreSQL

For database ingestion and replication tasks only, a PostgreSQL replication plugin. Valid values are:

- pgoutput
- wal2json

#### **publication**

**Sources:** PostgreSQL

For database ingestion and replication tasks only, the publication name that is used by the pgoutput plugin. Use this parameter only if you specified pgoutput as the replicationPlugin value.

#### **selectionRules**

**Sources:** All

Optional object or table selection rules to select a subset of the source objects or tables. By default, an Include rule that contains only the asterisk (\*) wildcard character is provided. This rule selects all objects on the source for an application ingestion and replication task or all tables in the source schema for a database ingestion and replication task. To narrow the source objects or tables to be processed by the task, you can define additional Include rules, Exclude rules, or both types of rules.

**Example:**

```
selectionRules:
  - include: TABLE_1
  - exclude: TABLE_2
```

**restartPointForIncrementalLoad**

**Sources:** SAP ECC sources of application ingestion incremental load jobs. All sources of database ingestion incremental load and combined initial and incremental load jobs.

The position in the source change stream or logs from which an application ingestion and replication or database ingestion and replication job starts reading change records the first time it runs. Valid types are:

- earliest
- latest

For example:

```
restartPointForIncrementalLoad: earliest
restartPointForIncrementalLoad: latest
```

For database ingestion and replication jobs, the default is latest.

For application ingestion and replication jobs, the **earliest** option is not supported and the **latest** option applies only to SAP ECC sources. The default is the restartPointForIncrementalLoadTimestamp value.

If you specify this parameter, do not also specify the restartPointForIncrementalLoadPosition or restartPointForIncrementalLoadTimestamp parameter.

**restartPointForIncrementalLoadPosition**

**Sources:** All sources of database ingestion incremental load and combined initial and incremental load jobs.

The RBA position in the source change stream or logs from which a database ingestion and replication job starts reading change records the first time it runs. For example:

```
restartPointForIncrementalLoadPosition: 0
```

This parameter is not supported for application ingestion jobs.

If you specify this parameter, do not also specify the restartPointForIncrementalLoad or restartPointForIncrementalLoadTimestamp parameter.

**restartPointForIncrementalLoadTimestamp**

**Sources:** All sources of application incremental load jobs and of database ingestion incremental load and combined initial and incremental load jobs.

The date and time, including AM or PM, in the source change stream or logs from which a application ingestion and replication or database ingestion and replication job starts reading change records the first time it runs. For example:

```
restartPointForIncrementalLoadTimestamp: 2021-08-18 02:50:00 PM
```

For application ingestion jobs, this parameter determines the default behavior.

If you specify this parameter, do not also specify the restartPointForIncrementalLoad or restartPointForIncrementalLoadPosition parameter.

**cdcinterval**

**Sources:** Salesforce, incremental load tasks and combined initial and incremental load tasks

For application ingestion and replication tasks only, the time interval in which the application ingestion and replication job runs to retrieve change records for incremental load. You can configure the `cdcIntervalDays`, `cdcIntervalHours`, and `cdcIntervalMins` parameters to specify the interval.

#### **fetchSize**

**Sources:** Salesforce, initial load tasks and incremental load tasks

For application ingestion and replication tasks only, the number of records that the application ingestion and replication job reads at a time from the source. The default value for initial load tasks is 50000 and the default value for incremental load tasks is 2000. If you specified Standard (REST) API for the `salesforceAPI` parameter, you must change the `fetchSize` value to 2000.

#### **fetchSizeForInitialLoad**

**Sources:** Salesforce, combined initial and incremental load tasks

For application ingestion and replication tasks only, the number of source records that the application ingestion and replication job reads at a time during the initial loading of data. The default value is 50000. If you specified Standard (REST) API for the `salesforceAPI` parameter, you must change the `fetchSize` value to 2000.

#### **fetchSizeForIncrementalLoad**

**Sources:** Salesforce, combined initial and incremental load tasks

For application ingestion and replication tasks only, the number of source records that the application ingestion and replication job reads at a time during the incremental loading of change data. The default value is 2000. For initial load tasks or combined initial and incremental load tasks, if you specified Standard (REST) API for the `salesforceAPI` parameter, you must change the `fetchSize` value to 2000.

#### **includeArchivedAndDeletedRows**

**Sources:** Salesforce, initial load tasks and combined initial and incremental load tasks

For application ingestion and replication tasks only, ingests archived and soft-deleted rows from the source during the initial loading of data. The default value is false.

#### **includeBase64Fields**

**Sources:** Salesforce

For application ingestion and replication tasks only, ingests the source fields of Base64 data type. The default value is false.

**Note:** Configure this parameter only if you set the `salesforceAPI` parameter to Standard (REST) API .

#### **maximumBase64BodySize**

**Sources:** Salesforce

For application ingestion and replication tasks only, the maximum body size of Base64 encoded data. The default value is 7 MB.

**Note:** Configure this parameter only if you set the `includeBase64Fields` parameter to yes.

#### **includeViews**

**Sources:** Microsoft SQL Server, Oracle

For database ingestion and replication tasks only, indicates whether to include views in the table counts and list of table names for database ingestion and replication tasks. Valid values:

- true. Include views.
- false. Do not include views.

## customProperties

**Sources:** All

Custom properties that Informatica provides to meet special requirements. Specify these properties only at the direction of Informatica Global Customer Support.

**Example:**

```
customProperties:
  "readerInputIsPersisted": true
  "propertyName": "propertyValue"
```

## Target section parameters

The target section of the YAML input file specifies the target options.

### connection

**Targets:** All

The name of the connection for the target system.

### schema

**Targets:** Amazon Redshift, Databricks Delta, Google BigQuery, Microsoft Azure Synapse Analytics, Oracle, Snowflake

The target schema in which Data Ingestion and Replication creates the target objects or tables.

### bucket

**Targets:** Amazon Redshift, Google Cloud Storage, Google BigQuery

Specifies the name of an existing bucket container that stores, organizes, and controls access to the data objects that you load to the target.

### directory

**Targets:** Databricks Delta, Amazon Redshift, Google Cloud Storage, Google BigQuery

Specifies the virtual directory for the target objects that contain the data.

### useTableNameAsTopicName

**Targets:** Apache Kafka

Indicates whether Data Ingestion and Replication writes messages that contain source data to separate topics, one for each source table or object, or writes all messages to a single topic. Valid values are:

- true. Write messages to separate table-specific topics.
- false. All messages are written to the single topic which name is specified in the topicName parameter.

### includeSchemaName

**Targets:** Apache Kafka

When you set useTableNameAsTopicName to true, this setting adds the source schema name in the table-specific topic names. The topic names then have the format *schemaname\_tablename*.

Valid values are:

- true. Add the source schema name in the table-specific topic names.
- false. Do not add the source schema name in the table-specific topic names.

### tablePrefix

**Targets:** Apache Kafka

When you set `useTableNameAsTopicName` to `true`, this parameter specifies an optional prefix to add to the table-specific topic names. For example, if you specify `myprefix_`, the topic names have the format `myprefix_tablename`. If you omit the underscore (`_`) after the prefix, the prefix is prepended to the table name.

#### **tableSuffix**

**Targets:** Apache Kafka

When you set `useTableNameAsTopicName` to `true`, this parameter specifies an optional suffix to add to the table-specific topic names. For example, if you specify `_mysuffix`, the topic names have the format `tablename_mysuffix`. If you omit the underscore (`_`) before the suffix, the suffix is appended to the table name.

#### **topicName**

**Targets:** Apache Kafka

The name of the single Kafka topic to which all messages that contain source data will be written. Use this parameter if `useTableNameAsTopicName` is set to `false`.

#### **stage**

**Targets:** Snowflake

The name of internal staging area that holds the data read from the source before the data is written to the target tables. This name must not include spaces. If the staging area does not exist, it will be automatically created.

#### **outputFormat**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Microsoft Azure Data Lake Storage, Kafka

The format of the output file. Valid values are:

- AVRO
- CSV
- PARQUET

The default value is CSV.

**Note:** Output files in CSV format use double-quotation marks (") as the delimiter for each field.

#### **parquetFormat**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Microsoft Azure Data Lake Storage, Kafka

If AVRO is specified as the output format in the `outputFormat` parameter, set this parameter to `true` to write data in uncompressed Parquet format. Alternatively, you can just set the `outputFormat` parameter to PARQUET and not include this parameter.

**Important:** If you set this option to `true`, you must install Visual C++ Redistributable Packages for Visual Studio 2013 on the computer where the Secure Agent runs.

Valid values are:

- `true`. Write data in uncompressed Parquet format.
- `false`. Write data in AVRO format.

#### **avroFormat**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Microsoft Azure Data Lake Storage, Kafka

If AVRO is specified as the output format, specify the format of the Avro schema that will be created for each source table or object. Valid values are:

- Avro-Flat. This Avro schema format lists all Avro fields in one record.
- Avro-Generic. This Avro schema format lists all columns or fields from a source table or object in a single array of Avro fields.
- Avro-Nested. This Avro schema format organizes each type of information in a separate record.

The default value is Avro-Flat.

#### **avroSerializationFormat**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Microsoft Azure Data Lake Storage, Kafka

If AVRO is specified as the output format, specify the serialization format of the Avro output file. Valid values are:

- Binary
- JSON

The default value is Binary.

#### **avroSchemaDirectory**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Microsoft Azure Data Lake Storage, Kafka

If AVRO is specified as the output format, specifies the local directory where Data Ingestion and Replication stores Avro schema definitions for each source table or object. Schema definition files have the following naming pattern: *schemaname\_tablename.txt*

**Note:** If this directory is not specified, no Avro schema definition file is produced.

#### **fixedDirectoryForEachTable**

**Targets:** Amazon S3, Google Cloud Storage, or Microsoft Azure Data Lake Storage Gen2

For initial load tasks, indicates whether to use the source table or object names as the names of the directories to which Data Ingestion and Replication writes flat files that contain source data, for all job runs.

Valid values are:

- true. Use the source table or object names as the names of the directories for each job run.
- false. Create a new set of directories for each job run by using the following naming pattern: *tablename\_timestamp*.

#### **fileCompressionType**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Microsoft Azure Data Lake Storage

If the output format is set to AVRO or CSV format in the `outputFormat` parameter, optionally specifies a file compression type. Valid values are:

- Deflate
- gzip
- Snappy

**Note:** If you do not specify this parameter, the output files are not compressed.

#### **avroCompressionType**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Kafka, Microsoft Azure Data Lake Storage

If the output format is set to AVRO in the outputFormat parameter, optionally specifies an Avro compression type. Valid values are:

- None
- Bzip2
- Deflate
- Snappy

The default value is **None**, which means no compression is used.

#### **parquetCompressionType**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Kafka, Microsoft Azure Data Lake Storage

If the outputFormat parameter is set to PARQUET, or if the outputFormat parameter is set to AVRO and the parquetFormat parameter is set to true, optionally specifies a Parquet compression type. Valid values are:

- None
- gzip
- Snappy

#### **deflateCompressionLevel**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Kafka, Microsoft Azure Data Lake Storage

If Deflate is specified in the avroCompressionType parameter, specifies a compression level from 0 to 9. The default value is 0.

#### **addDirectoryTags**

**Targets:** Amazon S3, Google Cloud Storage, Microsoft Azure Data Lake Storage

Indicates whether Data Ingestion and Replication adds the "dt=" prefix to the names of apply cycle directories to be compatible with the naming convention for Hive partitioning. Valid values are:

- true. Add the "dt=" prefix to the names of apply cycle directories.
- false. Do not change the names of apply cycle directories.

The default value is false.

#### **directoryTags**

**Targets:** Amazon S3, Google Cloud Storage, Microsoft Azure Data Lake Storage

This field is dependent on the addDirectoryTags field.

Example:

```
addDirectoryTags: true
directoryTags: \\directory
```

#### **renamingRules**

**Targets:** All

Optional rules for renaming the target tables that correspond to the selected source tables or objects.

Example:

```
- source: "*"
  target: "*_1"
```

#### **dataTypeRules**

**Targets:** Amazon Redshift, Databricks Delta, Google BigQuery, Oracle, Snowflake, Synapse Analytics



Optional data type mapping rules to override the default mappings of source data types to target data types. The default mappings are described under Default Data Type Mappings.

Example:

```
- source: Int
  target: String
```

#### **cdcCompatibleFormat**

**Targets:** Amazon S3, Flat file, Google Cloud Storage, Kafka, Microsoft Azure Data Lake Storage

Indicates whether Data Ingestion and Replication includes UNDO data in the output that a job writes to the target. The default value is false.

#### **Advanced parameters**

##### **addOperationType**

**Targets:** Amazon S3, Apache Kafka, Flat File, Google Cloud Storage, Microsoft Azure Data Lake Storage

For incremental load and combined initial and incremental load tasks, set this parameter to true to add a metadata column that includes the source SQL operation type in the output that the job propagates to the target. For incremental load and combined initial and incremental load tasks, default is true. For initial load tasks, default is false.

##### **addOperationTime**

**Targets:** Amazon S3, Apache Kafka, Flat File, Google Cloud Storage, Microsoft Azure Data Lake Storage

For incremental load and combined initial and incremental load tasks, set this parameter to true to add a metadata column that includes the source SQL operation time in the output that the job propagates to the target. For initial loads, the job always writes the current date and time. Default is false for all load types.

##### **addOperationOwner**

**Targets:** Amazon S3, Apache Kafka, Flat File, Google Cloud Storage, Microsoft Azure Data Lake Storage

For incremental load and combined initial and incremental load tasks, set this parameter to true to add a metadata column that includes the owner of the source SQL operation in the output that the job propagates to the target. For initial loads, the job always writes "INFA" as the owner. Default is false. Not used for application ingestion and replication jobs.

##### **addOperationTransactionId**

**Targets:** Amazon S3, Apache Kafka, Flat File, Google Cloud Storage, Microsoft Azure Data Lake Storage

For incremental load and combined initial and incremental load tasks, set this parameter to true to add a metadata column that includes the source transaction ID in the output that the job propagates to the target for SQL operations. Default is false. Not used for application ingestion and replication jobs.

##### **addBeforeImages**

**Targets:** Amazon S3, Apache Kafka, Flat File, Google Cloud Storage, Microsoft Azure Data Lake Storage

For incremental load and combined initial and incremental load tasks, set this parameter to true to include UNDO data in the output that an incremental load job writes to the target. For initial loads, the job writes nulls. Default is false.

##### **asyncWrite**

**Targets:** Kafka

Controls whether to use synchronous delivery of messages to Kafka. Valid values are:

- true. Use asynchronous delivery. Data Ingestion and Replication sends messages as soon as possible, without regard for the order in which the changes were retrieved from the source.
- false. Use synchronous delivery. Kafka must acknowledge each message as received before Data Ingestion and Replication sends the next message. In this mode, Kafka is unlikely to receive duplicate messages. However, performance might be slower.

The default value is true.

#### **producerConfigurationProperties**

**Targets:** Kafka

A comma-separated list of *key=value* pairs to enter Kafka producer properties for Apache Kafka, Confluent Kafka, or Kafka-enabled Event Hubs targets.

If you have a Confluent target that uses Confluent Schema Registry to store schemas, you must specify the following properties:

```
schema.registry.url=url,  
key.serializer=org.apache.kafka.common.serialization.StringSerializer,  
value.serializer=io.confluent.kafka.serializers.KafkaAvroSerializer
```

You can specify Kafka producer properties in either this parameter or in the **Additional Connection Properties** field in the Kafka connection.

If you specify the producer properties in this parameter, the properties pertain to the ingestion and replication jobs associated with this task only. If you enter the producer properties for the connection, the properties pertain to jobs for all tasks that use the connection definition, unless you override the connection-level properties for specific tasks by also specifying properties in the **Producer Configuration Properties** field.

For information about Kafka producer properties, see the Apache Kafka documentation or Confluent Kafka documentation.

#### **Custom parameters**

##### **customProperties**

**Targets:** All

Custom properties that Informatica provides to meet special requirements. Specify these properties only at the direction of Informatica Global Customer Support.

**Example:**

```
customProperties:  
  # cdc, s3, add target directory checkbox  
  rat.cdcfile.target.tag.directories: true  
  # only if CSV format selected as Output Format  
  # adlsgen2, azureDLS, flat file, google, s3, kafka  
  formatEncoderPrintHeader: false  
  # adlsgen2, azureDLS, flat file, google, s3, kafka  
  rat.audit.columns.optype: true  
  rat.audit.columns.optime: true  
  rat.audit.columns.opowner: true  
  rat.audit.columns.optxid: true  
  # kafka, actually located on step4  
  checkpointAllRows: true  
  checkpointEveryCommit: false  
  checkpointRowCount: 0  
  checkpointFrequencySecs: 0  
  # any other values  
  propertyName: propertyValue
```

## RuntimeOptions section parameters

The runtimeOptions section of the YAML input file configures runtime options for jobs of any load type.

### schemaDriftOptions

**Sources:** Microsoft SQL Server or Oracle, incremental load tasks and combined initial and incremental load tasks. Synapse target support ignores the value for the rename column DDL operation. Microsoft SQL Server, Oracle or PostgreSQL, incremental load tasks and combined initial and incremental load tasks are also supported.

The schema drift option to use for each supported type of DDL operation:

- addColumn
- modifyColumn
- dropColumn
- renameColumn

Valid values are:

- ignore
- replicate
- stop job
- stop table

#### Example:

```
schemaDriftOptions:
  addColumn: ignore
  modifyColumn: replicate
  dropColumn: replicate
  renameColumn: replicate
```

### numberOfRowsInOutputFile

**Targets:** Amazon Redshift, Amazon S3, Google Big Query, Google Cloud Storage, Microsoft Azure Data Lake Storage, Microsoft Azure Synapse Analytics, Oracle and Snowflake

Specifies the maximum number of rows that the application ingestion and replication or database ingestion and replication task writes to an output data file on a target.

### fileExtensionBasedOnFileType

**Targets:** Flat File, Amazon S3, Google Cloud Storage, and Microsoft Azure Data Lake Storage

Indicates whether you want the output data files to have the .dat extension.

- true. The output files have file-name extensions based on their file types.
- false. The output files have the .dat extension.

The default value is true.

## Advanced parameters

### applyCycleChangeLimit

**Targets:** Amazon S3, Google Cloud Storage, or Microsoft Azure Data Lake Storage Gen2

For application ingestion or database ingestion and replication incremental load jobs that have an Amazon S3 or Microsoft Azure Data Lake Storage Gen2 target, the number of records that must be processed before the job ends an apply cycle. When this record limit is reached, the job ends the apply cycle and writes the change data to the target. Default is 10000.

**Note:** Either the `applyCycleChangeLimit` parameter or an `applyCycleIntervaltime_unit` parameter must have a non-zero value.

#### **applyCycleIntervalDays**

**Targets:** Amazon S3, Google Cloud Storage, or Microsoft Azure Data Lake Storage Gen2

For application ingestion and replication or database ingestion and replication incremental load jobs that have an Amazon S3 or Microsoft Azure Data Lake Storage Gen2 target, the number of days that must elapse before an application ingestion and replication or database ingestion and replication job ends an apply cycle. Default is 0. You can specify this parameter with `applyCycleIntervalHours`, `applyCycleIntervalMins`, and `applyCycleIntervalSecs` or specify a subset of these parameters.

#### **applyCycleIntervalHours**

**Targets:** Amazon S3, Google Cloud Storage, or Microsoft Azure Data Lake Storage Gen2

For application ingestion and replication or database ingestion and replication incremental load jobs that have an Amazon S3 or Microsoft Azure Data Lake Storage Gen2 target, the number of hours that must elapse before an application ingestion and replication or database ingestion and replication job ends an apply cycle. Default is 0.

#### **applyCycleIntervalMins**

**Targets:** Amazon S3, Google Cloud Storage, or Microsoft Azure Data Lake Storage Gen2

For application ingestion and replication or database ingestion and replication incremental load jobs that have an Amazon S3 or Microsoft Azure Data Lake Storage Gen2 target, the number of minutes that must elapse before an application ingestion and replication or database ingestion and replication job ends an apply cycle. Default is 15.

#### **applyCycleIntervalSecs**

**Targets:** Amazon S3, Google Cloud Storage, or Microsoft Azure Data Lake Storage Gen2

For application ingestion and replication or database ingestion and replication incremental load jobs that have an Amazon S3 or Microsoft Azure Data Lake Storage Gen2 target, the number of seconds that must elapse before an application ingestion and replication or database ingestion and replication job ends an apply cycle. Default is 0.

#### **lowActivityFlushHours**

**Targets:** Amazon S3, Google Cloud Storage, or Microsoft Azure Data Lake Storage Gen2

The amount of time, in hours, that must elapse during a period of no change activity on the source before an application ingestion and replication or database ingestion and replication job ends an apply cycle. You can use this parameter in conjunction with `lowActivityFlushMins` or specify only one of them. When this time limit is reached, the ingestion job ends the apply cycle and writes the change data to the target. Default is 0.

If you do not specify a value for `lowActivityFlushHours` or `lowActivityFlushMins`, the application ingestion and replication job ends apply cycles only after either the **applyCycleChangeLimit** or cycle interval time limit is reached.

#### **lowActivityFlushMins**

**Targets:** Amazon S3, Google Cloud Storage, or Microsoft Azure Data Lake Storage Gen2

The amount of time, in minutes, that must elapse during a period of no change activity on the source before an application ingestion and replication or database ingestion and replication job ends an apply cycle. You can use this parameter in conjunction with `lowActivityFlushHours` or specify only one of them. When the overall time limit is reached, the ingestion job ends the apply cycle and writes the change data to the target. Default is 0.

If you do not specify a value for `lowActivityFlushHours` or `lowActivityFlushMins`, the application ingestion and replication or database ingestion and replication job ends apply cycles only after either the `applyCycleChangeLimit` or cycle interval time limit is reached.

## Checkpoint parameters

### `checkpointAllRows`

**Targets:** Apache Kafka

For application ingestion and database ingestion and replication incremental load jobs that have an Apache Kafka target, indicates whether an ingestion job performs checkpoint processing for every message that is sent to the Kafka target. When this parameter is set to true, the `checkpointEveryCommit`, `checkpointRowCount`, and `checkpointFrequencySecs` parameters are ignored. Default is true.

### `checkpointEveryCommit`

**Targets:** Apache Kafka

For application ingestion and replication and database ingestion and replication incremental load jobs that have an Apache Kafka target, Indicates whether an ingestion job performs checkpoint processing for every commit that occurs on the source. Default is false. If you set this parameter to true, do not also specify `checkpointAllRows`.

### `checkpointRowCount`

**Targets:** Apache Kafka

For application ingestion and replication and database ingestion and replication incremental load jobs that have an Apache Kafka target, the maximum number of messages that a job sends to the target before adding a checkpoint. If you set this option to 0, an ingestion and replication job does not perform checkpoint processing based on the number of messages. If you set this option to 1, an ingestion job adds a checkpoint for each message. Default is 0. If you set this parameter to a non-zero number, do not also specify `checkpointAllRows`.

### `checkpointFrequencySecs`

**Targets:** Apache Kafka

For application ingestion and replication and database ingestion and replication incremental load jobs that have an Apache Kafka target, the maximum number of seconds that must elapse before an ingestion job adds a checkpoint. If you set this option to 0, an ingestion and replication job does not perform checkpoint processing based on elapsed time. Default is 0. If you set this parameter to a non-zero number, do not also specify `checkpointAllRows`.

## Custom parameters

### `customProperties`

Custom properties that Informatica provides to meet special requirements. Specify these properties only at the direction of Informatica Global Customer Support.

#### Example:

```
customProperties:
  rat.cdcfile.cycle.dml.limit: 10000
  rat.cdcfile.cycle.interval.days: 0
  rat.cdcfile.cycle.interval.hours: 0
  rat.cdcfile.cycle.interval.mins: 0
  rat.cdcfile.cycle.interval.secs: 0
  rat.cdcfile.lowact.flush.hours: 0
  rat.cdcfile.lowact.flush.mins: 0
```

### Schedule section parameters

The schedule section of the YAML input file consists of a single parameter that specifies the task schedule. A large number of scheduled jobs can lead to increased resource consumption.

#### **schedule**

The name of a predefined schedule that you want to use for running run job instances for an initial load task instead of manually starting the job after it is deployed from one of the monitoring interfaces.

## Default values

The defaults.yml file is in the /ymls folder. This file contains the default values for location, salesforceAPI, selectionRules, cdcIntervalMins, fetchSize, and restartPointForIncrementalLoad. You can customize this file according to your needs.

For example, if you did not specify selectionRules in the yml file with your task configuration values that you customized, the values from the defaults.yml file will be applied.

# INDEX

## C

Cloud Application Integration community  
URL [4](#)  
Cloud Developer community  
URL [4](#)

## D

Data Integration community  
URL [4](#)

## I

Informatica Global Customer Support  
contact information [5](#)  
Informatica Intelligent Cloud Services  
web site [4](#)

## M

maintenance outages [5](#)

## S

status  
Informatica Intelligent Cloud Services [5](#)  
system status [5](#)

## T

trust site  
description [5](#)

## U

upgrade notifications [5](#)

## W

web site [4](#)