



Informatica® Mass Ingestion
April 2024

一括取り込みコマンドライ ンインタフェース

Informatica Mass Ingestion 一括取り込みコマンドラインインタフェース
April 2024

© 著作権 Informatica LLC 2019, 2024

本ソフトウェアおよびマニュアルは、使用および開示の制限を定めた個別の使用許諾契約のもとでのみ提供されています。本マニュアルのいかなる部分も、いかなる手段（電子的複製、写真複製、録音など）によっても、Informatica LLC の事前の承諾なしに複製または転載することは禁じられています。

米政府の権利プログラム、ソフトウェア、データベース、および関連文書や技術データは、米国政府の顧客に配信され、「商用コンピュータソフトウェア」または「商業技術データ」は、該当する連邦政府の取得規制と代理店固有の補足規定に基づきます。このように、使用、複製、開示、変更、および適応は、適用される政府の契約に規定されている制限およびライセンス条項に従うものとし、政府契約の条項によって適当な範囲において、FAR 52.227-19、商用コンピュータソフトウェアライセンスの追加権利を規定します。

Informatica、Informatica Cloud、Informatica Intelligent Cloud Services、PowerCenter、PowerExchange、および Informatica ロゴは、米国およびその他の国における Informatica LLC の商標または登録商標です。Informatica の商標の最新リストは、Web (<https://www.informatica.com/trademarks.html>) にあります。その他の企業名および製品名は、それぞれの企業の商標または登録商標です。

本ソフトウェアまたはドキュメンテーション（あるいはその両方）の一部は、第三者が保有する著作権の対象となります。必要な第三者の通知は、製品に含まれています。

本マニュアルの情報は、予告なしに変更されることがあります。このドキュメントで問題が見つかった場合は、infa_documentation@informatica.com までご報告ください。

Informatica 製品は、それらが提供される契約の条件に従って保証されます。Informatica は、商品性、特定目的への適合性、非侵害性の保証等を含めて、明示的または黙示的ないかなる種類の保証をせず、本マニュアルの情報を「現状のまま」提供するものとします。

発行日: 2024-05-20

目次

序文	4
Informatica のリソース.....	4
Informatica マニュアル.....	4
Informatica Intelligent Cloud Services Web サイト.....	4
Informatica Intelligent Cloud Services コミュニティ.....	4
Informatica Intelligent Cloud Services マーケットプレイス.....	5
データ統合のコネクタのドキュメント.....	5
Informatica ナレッジベース.....	5
Informatica Intelligent Cloud Services Trust Center.....	5
Informatica グローバルカスタマサポート.....	5
 第 1 章 : クラウド一括取り込みコマンドラインインタフェース	6
クラウド一括取り込み CLI の概要.....	6
必要なユーザーロールと権限.....	7
クラウド一括取り込み CLI の実行.....	7
クラウド一括取り込み CLI の共通オプション.....	8
クラウド一括取り込み CLI のコマンド.....	8
入力ファイル形式.....	17
デフォルト値.....	30
 索引	31

序文

『一括取り込みコマンドラインインタフェース』では、タスクの作成、表示、およびデプロイや、アプリケーション取り込みジョブとデータベース統合ジョブの一覧表示、管理、および監視を行うために使用できる CLI コマンドについて説明します。

Informatica のリソース

Informatica は、Informatica Network やその他のオンラインポータルを通じてさまざまな製品リソースを提供しています。リソースを使用して Informatica 製品とソリューションを最大限に活用し、その他の Informatica ユーザーや各分野の専門家から知見を得ることができます。

Informatica マニュアル

Informatica マニュアルポータルでは、最新および最近の製品リリースに関するドキュメントの膨大なライブラリを参照できます。マニュアルポータルを利用するには、<https://docs.informatica.com> にアクセスしてください。

製品マニュアルに関する質問、コメント、ご意見については、Informatica マニュアルチーム (infa_documentation@informatica.com) までご連絡ください。

Informatica Intelligent Cloud Services Web サイト

Informatica Intelligent Cloud Services Web サイト (<http://www.informatica.com/cloud>) にアクセスできます。このサイトには、Informatica Cloud 統合サービスに関する情報が含まれます。

Informatica Intelligent Cloud Services コミュニティ

Informatica Intelligent Cloud Services コミュニティを使用して、技術的な問題について議論し、解決します。また、技術的なヒント、マニュアルの更新情報、FAQ（よくある質問）への答えを得ることもできます。

次の Informatica Intelligent Cloud Services コミュニティにアクセスします。

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

開発者は、次の Cloud 開発者コミュニティで詳細情報を確認したり、ヒントを共有したりできます。

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

Informatica Intelligent Cloud Services マーケットプレイス

Informatica マーケットプレイスにアクセスすると、データ統合コネクタ、テンプレート、およびマップレットを試用したり購入したりできます。

<https://marketplace.informatica.com/>

データ統合のコネクタのドキュメント

データ統合のコネクタのドキュメントには、マニュアルポータルからアクセスできます。マニュアルポータルを利用するには、<https://docs.informatica.com> にアクセスしてください。

Informatica ナレッジベース

Informatica ナレッジベースを使用して、ハウツー記事、ベストプラクティス、よくある質問に対する回答など、製品リソースを見つけることができます。

ナレッジベースを検索するには、<https://search.informatica.com> にアクセスしてください。ナレッジベースに関する質問、コメント、ご意見の連絡先は、Informatica ナレッジベースチーム (KB_Feedback@informatica.com) です。

Informatica Intelligent Cloud Services Trust Center

Informatica Intelligent Cloud Services Trust Center は、Informatica のセキュリティポリシーおよびリアルタイムでのシステムの可用性について情報を提供します。

Trust Center (<https://www.informatica.com/trust-center.html>) にアクセスします。

Informatica Intelligent Cloud Services Trust Center にサブスクライブして、アップグレード、メンテナンス、およびインシデントの通知を受信します。[Informatica Intelligent Cloud Services Status](#) ページには、すべての Informatica Cloud 製品の実稼働ステータスが表示されます。メンテナンスの更新はすべてこのページに送信され、停止中は最新の情報が表示されます。更新と停止の通知がされるようにするには、Informatica Intelligent Cloud Services の 1 つのコンポーネントまたはすべてのコンポーネントについて更新の受信をサブスクライブします。すべてのコンポーネントにサブスクライブするのが、更新を逃さないようにするための最良の方法です。

サブスクライブするには、[Informatica Intelligent Cloud Services Status](#) ページで **【サブスクライブして更新】** をクリックします。電子メール、SMS テキストメッセージ、Webhook、RSS フィード、またはこの 4 つの任意に組み合わせとして送信される通知を受信するという選択ができます。

Informatica グローバルカスタマサポート

グローバルサポートセンターには、Informatica Network または電話でお問い合わせください。

Informatica Network でオンラインサポートリソースを検索するには、Informatica Intelligent Cloud Services のヘルプメニューで **【サポートにお問い合わせください】** をクリックして、**Cloud Support** ページに移動します。**Cloud Support** ページには、システムステータス情報とコミュニティディスカッションが記載されています。追加のリソースを検索する場合や電子メールで Informatica グローバルカスタマサポートに問い合わせる場合は、Informatica Network にログインし、**【サポートが必要な場合】** をクリックしてください。

Informatica グローバルカスタマサポートの電話番号は、Informatica の Web サイト <https://www.informatica.com/services-and-training/support-services/contact-us.html> に掲載されています。

第 1 章

クラウド一括取り込みコマンドラインインタフェース

クラウド一括取り込みコマンドラインインタフェース (CLI) でコマンドを実行して、アプリケーション取り込みおよびデータベース統合タスクを作成、表示、およびデプロイし、取り込みジョブを一覧表示、開始、停止、デプロイ、デプロイ解除、および監視できます。

クラウド一括取り込み CLI の概要

クラウド一括取り込みコマンドラインインタフェース (CLI) ファイルは、*Informatica Cloud Secure Agent installation/apps/Database_Ingestion/version/dbmicli* ディレクトリにインストールされます。dbmicli ディレクトリを別の Linux または Windows サーバーにコピーして、一括取り込みのアップグレード後に作成する CLI 設定ファイルを保持することをお勧めします。

クラウド一括取り込み CLI のコマンドを使用して、アプリケーション取り込みおよびデータベース統合のジョブやタスクのために次のタスクを実行できます。

- 新しいタスクを作成する、または 1 つ以上の既存のタスクに基づいてタスクを作成する。
- タスクをデプロイする。
- タスクを表示する。
- 組織のジョブまたは特定のジョブのステータス情報を表示する。
- ジョブを開始または再開する。
- ジョブを停止または中止する。
- ジョブを再デプロイします。
- 既存のタスク定義をタスクの更新バージョンに置き換えます。
- ジョブをデプロイ解除する。
- ジョブを監視する。

データベース取り込みタスクの場合のみ、CLI を使用してソーステーブルでデータベース CDC オプションを有効にできます。

CLI に関する情報を取得するためのコマンドも使用できます。

- クラウド一括取り込み CLI コマンドに関するヘルプ情報を取得する。
- CLI の現在のバージョンを取得する。

必要なユーザーロールと権限

CLI を使用してアプリケーション取り込みタスクとデータベース統合タスクを実行するには、特定のロールと権限が必要です。Administrator でユーザーロールと特権を設定します。

通常、次のシステム定義のユーザーロールが必要です。

- 組織内のすべてのアプリケーション取り込みジョブとデータベース取り込みジョブを一覧表示するには、次のいずれかのシステム定義ロール、または少なくとも同じ特権を持つカスタムロールがあることを確認してください。

- 管理者
- データ統合タスク実行者
- デザイナ
- モニタ

これらのロールのいずれもない場合は、所有しているジョブのみを一覧表示できます。

- 取り込みタスクのジョブを開始、再開、停止、または強制終了するには、次のいずれかのシステム定義ロール、または少なくとも同じ特権を持つカスタムロールがあることを確認してください。

- 管理
- デザイナ

アプリケーション取り込みタスクとデータベース取り込みタスクに対する実行権限も必要です。

- 取り込みタスクを作成するには、次のいずれかのシステム定義ロール、または少なくとも同じ特権を持つカスタムロールがあることを確認してください。

- 管理
- デザイナ

- 取り込みタスクをデプロイするには、次のいずれかのシステム定義ロール、または少なくとも同じ特権を持つカスタムロールがあることを確認してください。

- 管理
- デザイナ
- デプロイヤー

アプリケーション取り込みタスクとデータベース取り込みタスクに対する実行権限も必要です。

クラウド一括取り込み CLI の実行

Cloud 一括取り込み dbmicli ファイルを Secure Agent ではなくリモートの場所にある Linux または Windows サーバーにコピーした場合は、推奨どおり、そこから CLI を実行します。dbmicli ファイルが Secure Agent システムにのみ存在する場合は、*Informatica Cloud Secure Agent installation/apps/Database_Ingestion/version/dbmicli* ディレクトリから CLI を実行します。

CLI を開始するには、次のいずれかのスクリプトを実行します。

- Linux の場合は、dbmicli.sh
- Windows の場合は、dbmicli.bat

構文

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

クラウド一括取り込み CLI の共通オプション

次のいずれかの方法で、CLI のユーザー名、パスワード、およびポッド URL に関する情報を指定する必要があります。

- コマンドラインオプションで値を指定する。構文:

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

- 環境変数で値を定義する。構文:

```
[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
```

- 対話モードで手動で入力する。対話モードを使用するには、次のコマンドを実行した後、ユーザー名とパスワードを入力します。

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

次の表に、クラウド一括取り込み CLI の共通オプションを示します。

共通オプション	環境変数	説明
-u, --user	DBMIUSER	クラウド一括取り込みユーザーの名前。
-p, --password	DBMIPASSWORD	クラウド一括取り込みユーザーのパスワード。
-P, --pod	DBMIPOD	Informatica Intelligent Cloud Services ポッドのドメイン名。 重要: ポッドのドメイン名は、POD の Web アドレスから取得できます。Web アドレスから、ドメイン名の最初の部分（第 4 レベルドメイン）とパスを省略します。たとえば、次の Web アドレス https://usw1.dm1-us.informaticacloud.com/cloudshell/showProducts がある場合、ポッドのドメイン名は dm1-us.informaticacloud.com です。

クラウド一括取り込み CLI は、ユーザー名、パスワード、ポッド URL を次の順序で取得しようとします。

1. 値をコマンドラインオプションから取得する。
2. コマンドラインオプションが指定されていない場合、対応する環境変数が定義されていればその環境変数から値を取得する。
3. 環境変数が定義されていない場合、CLI は対話モードに切り替わり、ユーザーに値の入力を求める。

重要: ポッド URL が -P オプションで指定されておらず、DBMIPOD 環境変数で定義されていない場合、CLI は対話モードに切り替えずにエラーで終了します。

クラウド一括取り込み CLI のコマンド

次の表に、クラウド一括取り込み CLI のコマンドを示します。

注: 特に明記しない限り、CLI コマンドはアプリケーション取り込みとデータベース取り込みの両方に適用されます。

コマンド	説明
--version、-v	クラウド一括取り込み CLI のバージョンを表示します。
help、--help、-h	すべてのクラウド一括取り込み CLI オプションとコマンドのリストを表示します。
help <i>command</i>	指定されたコマンドの説明を表示します。
job start <i>job_name</i> [--waitForCompletion] [--period= <i>requestsPeriod</i>] [--retryCount= <i>retryCount</i>]	<p>指定されたジョブ名のジョブを開始します。必要に応じて、ジョブを監視し、失敗ステータスの場合はジョブを再開します。</p> <p>構文:</p> <pre>job start <i>job_name</i> [--waitForCompletion] [--period=<i>requestsPeriod</i>] [--retryCount=<i>retryCount</i>]</pre> <p>有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> - waitForCompletion。完了、中止、または停止のステータスになるのを待つか、失敗のステータスの場合はジョブを再開します。 - period=<i>requestsPeriod</i>。ジョブステータスのチェックの時間間隔（ミリ秒単位）。デフォルト値は 5000 です。 - retryCount=<i>retryCount</i>。ジョブの監視を停止するまでに実行できるジョブ再開の試行回数。デフォルト値は 10 です。 <p>job start --waitForCompletion コマンドを使用したジョブの場合は、終了コードでジョブの現在のステータスを示します。ジョブの終了コードを表示するには、次のいずれかのコマンドを実行します。</p> <ul style="list-style-type: none"> - Windows では、echo %ERRORLEVEL%を実行します。 - Linux では、echo \$?を実行します。 <p>job start --waitForCompletion コマンドでは次の終了コードが返され、以下のようにジョブのステータスを示します。</p> <ul style="list-style-type: none"> - 終了コード 0 - 完了 - 終了コード 1 - 例外が発生 - 終了コード 2 - 失敗（再試行は残っていない） - 終了コード 3 - 停止 - 終了コード 4 - 強制終了 - 終了コード 5 - デプロイ解除
job resume <i>job_name</i>	指定されたジョブ名のジョブを再開します。
job stop <i>job_name</i>	指定されたジョブ名のジョブを停止します。
job abort <i>job_name</i>	指定されたジョブ名のジョブを中止します。

コマンド	説明
<p>ジョブステータス</p> <p><i>[-f field_names]</i></p> <p><i> [--taskType task_type]</i></p>	<p>必要な権限がある場合は、組織のジョブに関する情報が表示されます。十分な権限がない場合、ジョブはリストに表示されません。</p> <p>構文:</p> <pre>job status [-f field1,field2,...] [taskType=task_type]</pre> <p>-f オプションを使用して、CLI によってジョブごとに返されるようにするフィールドのカンマ区切りリストを指定します。デフォルトでは、CLI はすべてのジョブに対して次の値の文字列を返します。</p> <p><i>name;status;startTime;duration</i></p> <p>フィールドをカスタマイズするには、次のフィールド名を 1 つ以上指定して -f オプションを入力します。</p> <ul style="list-style-type: none"> - name。ジョブ名。 - state。ジョブのステータス。ステータス値の説明については、一括取り込みのオンラインヘルプまたは Doc Portal のドキュメントの「取り込みジョブの監視」 > 「ジョブのプロパティ」を参照してください。 - startTime。ジョブがデプロイされた日時。 - endTime。処理完了、停止、または失敗したためにジョブが終了した日時。 - undeployTime。ジョブがデプロイ解除された日時。 - duration。ジョブがデプロイ後に実行された時間。 <p>例えば、次のコマンドは、すべてのジョブの名前と状態を一覧表示します。</p> <pre>C:\DBMICLI>dbmicli.bat -P MyPod --user user_name --password user_password job status -f name,state</pre> <p>taskType オプションを使用して、取り込みタスクのタイプを指定します。これは、アプリケーション取り込み用の appmi またはデータベース取り込み用の dbmi のいずれかになります。このオプションを省略すると、両方のタスクタイプが含まれます。</p> <p>出力例:</p> <pre>testGCS_2_1611;Completed testGCS_0403_1610;Undeployed testGCS_0403_1607;Undeployed testGCS_1528;Up and Running</pre>

コマンド	説明
<code>job status [-f <i>field_names</i>] <i>job_name</i></code>	<p>特定のジョブ名を持つジョブの情報を表示します。</p> <p>構文:</p> <pre>job status [-f <i>field1,field2,...</i>] <i>job_name</i></pre> <p>-f オプションを使用して、CLI によってジョブに対して返されるようにするフィールドのカンマ区切りリストを指定できます。</p> <p>有効なフィールドは次のとおりです。</p> <ul style="list-style-type: none"> - name。ジョブ名。 - state。ジョブのステータス。ステータス値の説明については、一括取り込みのオンラインヘルプまたは Doc Portal のドキュメントの「取り込みジョブの監視」>「ジョブのプロパティ」を参照してください。 - startTime。ジョブがデプロイされた日時。 - endTime。処理完了、停止、または失敗したためにジョブが終了した日時。 - undeployTime。ジョブがデプロイ解除された日時。 - duration。ジョブがデプロイ後に実行された時間。 <p>たとえば、次のコマンドは、sample_job という名のジョブの開始時刻と期間を一覧表示します。</p> <pre>C:\DBMICLI>dbmicli.bat -P MyPod --user user_name --password user_password job status -f startTime,duration sample_job</pre> <p>出力例:</p> <pre>2021-03-04T05:39:23.000-0800;02:51:32</pre>
<code>job attach <i>job_name</i> [--period=<i>requestsPeriod</i>] [--retryCount=<i>retryCount</i>]</code>	<p>指定されたジョブ名のジョブを監視し、失敗ステータスの場合はジョブを再開します。次の条件のいずれかが満たされると、サイクルは終了します。</p> <ul style="list-style-type: none"> - 初期ロードジョブで、完了、中止、または停止ステータスが検出された。 - 増分ロードジョブ、または初期ロードジョブと増分ロードジョブの組み合わせで、中止または停止ステータスが検出された。 - 現在の再試行回数が、指定された retryCount 以上である。 <p>構文:</p> <pre>job attach <i>job_name</i> [--period=<i>requestsPeriod</i>] [--retryCount=<i>retryCount</i>]</pre> <p>有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> - period=<i>requestsPeriod</i>。ジョブステータスのチェックの時間間隔（ミリ秒単位）。デフォルト値は 5000 です。 - retryCount=<i>retryCount</i>。ジョブの監視を停止するまでに実行できるジョブ再開の試行回数。デフォルト値は 10 です。 <p>終了コードでジョブの現在のステータスを示します。ジョブの終了コードを表示するには、次のいずれかのコマンドを実行します。</p> <ul style="list-style-type: none"> - Windows では、echo %ERRORLEVEL%を実行します。 - Linux では、echo \$?を実行します。 <p>job attach コマンドでは次の終了コードが返され、以下のようにジョブのステータスを示します。</p> <ul style="list-style-type: none"> - 終了コード 0 - 完了 - 終了コード 1 - 例外が発生 - 終了コード 2 - 失敗（再試行は残っていない） - 終了コード 3 - 停止 - 終了コード 4 - 強制終了 - 終了コード 5 - デプロイ解除
<code>job redeploy <i>job_name</i></code>	<p>関連付けられた取り込みタスクで使用可能なフィールドを編集した後にジョブを再デプロイします。これにより、最初にジョブをデプロイ解除しなくても、新しい設定が有効になります。</p> <p>タスク設定の詳細はユーザーインターフェイスでのみ更新できます。</p>

コマンド	説明
<code>job undeploy <i>job_name</i></code>	<p>以前にデプロイされた取り込みジョブをデプロイ解除します。</p> <p>ジョブをデプロイ解除する前に、ジョブが実行されていないことを確認してください。</p> <p>ジョブを実行する必要がなくなった場合、ジョブが失敗状態になっている場合、またはジョブがデプロイ解除されるまで編集できない関連タスクの接続またはプロパティを変更する必要がある場合、ジョブのデプロイ解除が必要になることがあります。</p> <p>ジョブがデプロイ解除された後は、ジョブを再度実行したり、再デプロイしたりすることはできません。関連付けられた取り込みタスクのジョブを再度実行する場合は、タスクを再度デプロイして、新しいジョブインスタンスを作成する必要があります。</p>
<code>task create</code> <code>--pathToConfig</code> <code><i>path_to_configuration_file</i></code> <code>[-taskName <i>task_name</i>]</code> <code>[-taskLocation</code> <code><i>task_location</i>]</code> <code>[-taskType <i>task_type</i>]</code> <code>[---deploy]</code>	<p>入力設定ファイルからの情報を使用して取り込みタスクを作成し、必要に応じてタスクをデプロイします。</p> <p>構文:</p> <pre>task create --pathToConfig <i>path_to_configuration_file</i> [--taskName <i>task_name</i>] [--taskLocation <i>task_location</i>][-taskType <i>task_type</i>][--deploy]</pre> <p>有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> - pathToConfig。必須。YAML 入力ファイルへのパス。 - taskName。タスク名。 - taskLocation。タスク定義を含むプロジェクトまたはプロジェクト\フォルダ。デフォルト値は「デフォルト」です。 - taskType。取り込みタスクのタイプ。アプリケーション取り込みの場合は appmi、データベース取り込みの場合は dbmi です。 タスクタイプをコマンドで指定しない場合は、YAML ファイルの先頭に含めることができます。 デフォルトは dbmi です。 - deploy。作成後にタスクをデプロイします。 <p>たとえば、次のコマンドは、sample_task.yaml ファイルを使用してタスクを作成します。</p> <pre>C:\DBMICLI>dbmicli.bat -P MyPod --user <i>user_name</i> --password <i>user_password</i> task create --pathToConfig C:\DBMI_TASKS\sample_task.yaml</pre>

コマンド	説明
<pre>task createFrom --taskName <i>task_name</i> [--taskLocation <i>task_location</i>] [--taskType <i>task_type</i>] --override <i>path_to_override_file</i> [--deploy]</pre>	<p>オーバーライドファイルのフィールドオーバーライドを使用して、指定された別のタスクからタスクを作成します。また、必要に応じてタスクをデプロイします。</p> <p>たとえば、オーバーライドファイルで、デフォルトの場所をオーバーライドする場所「project1」が指定されている場合、新しいタスク定義は project1 の場所にあります。</p> <p>構文:</p> <pre>task createFrom --taskName <i>task_name</i> [--taskLocation <i>task_location</i>][--taskType <i>task_type</i>] --override <i>path_to_override_file</i> [--deploy]</pre> <p>有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> - <code>taskName</code>。必須。タスク名。 - <code>taskLocation</code>。タスク定義を含むプロジェクトまたはプロジェクト\フォルダ。デフォルト値は「デフォルト」です。 - <code>taskType</code>。取り込みタスクのタイプ。アプリケーション取り込みの場合は <code>appmi</code>、データベース取り込みの場合は <code>dbmi</code> です。 タスクタイプをコマンドで指定しない場合は、YAML ファイルの先頭を含めることができます。 デフォルトは <code>dbmi</code> です。 - <code>override</code>。必須。元のタスクのフィールドをオーバーライドするために使用されるフィールドを含む YAML ファイルへのパス。 - <code>deploy</code>。作成後にタスクをデプロイします。

コマンド	説明
<pre>task multipleCreateFrom --taskName <i>taskName</i> --override <i>path_to_override_file</i> [--taskLocation <i>task_location</i>] [--taskType <i>task_type</i>] [--deploy] [--deployTimeout <i>wait_time_for_status</i>] [--continueOnFail]</pre>	<p>オーバーライドファイルで指定されているフィールドオーバーライドを使用して、別のタスクから複数のタスクを作成します。また、必要に応じてタスクをデプロイします。</p> <p>構文:</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>有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> - taskName。必須。他のタスクを作成するタスクの名前。 - override。必須。フィールドオーバーライドを含む YAML ファイル。 - taskLocation。新しいタスク定義を含むプロジェクトまたはプロジェクト \ フォルダ。デフォルト値は [デフォルト] です。 - taskType。取り込みタスクのタイプ。アプリケーション取り込みの場合は appmi、データベース取り込みの場合は dbmi です。 タスクタイプをコマンドで指定しない場合は、YAML ファイルの先頭を含めることができます。 デフォルトは dbmi です。 - deploy。作成後に新しいタスクをデプロイします。 - deployTimeout。一括取り込みがいずれかのデプロイステータス（デプロイ済みまたは失敗）になるまで待機してからタイムアウトするまでの時間間隔（秒単位）。このオプションは、100 個を超えるタスクを作成およびデプロイするときに使用できます。Secure Agent を実行中の状態に維持するのに役立ちます。障害が発生した場合、ジョブのステータスを示すメッセージが出力とエラーログに書き込まれます。以下に例を示します。 Failed;Mass Ingestion could not deploy the task because supplemental logging is not enabled for source table 'TEST169'. 有効な値は 0～86400 です。 多くのタスクを作成およびデプロイするときにこのオプションを指定しない場合、要求は非同期で送信され、応答を待ちません。そのため、リソース消費の増加につながる可能性があります。 - continueOnFail。タスクの作成に失敗した場合は、他のタスクの作成を続行します。たとえば、10 個のタスクを作成し、2 番目のタスクが失敗した場合、他のタスクが作成されます。

コマンド	説明
<pre>task cdc --taskName=<i>taskName</i> [--taskLocation=<i>Default</i>] --scope=<i>pk all</i> [--execute] [--download] [--path=<i>path</i>] [--msRoleName=<i>role</i>] [-- msFileGroupName=<i>group</i>]</pre>	<p>データベース取り込みタスクの場合に、ソーステーブルでデータベース CDC オプションを有効にします。必要に応じて、CDC スクリプトを生成してから、スクリプトを実行するか、ダウンロードしてファイルに保存することができます。</p> <p>構文:</p> <pre>task cdc --taskName=<i>taskName</i> [--taskLocation=<i>Default</i>] --scope=<i>pk all</i> [--execute] [--download] [--path=<i>path</i>] [--msRoleName=<i>role</i>] [--msFileGroupName=<i>group</i>]</pre> <p>有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> - <i>taskName</i>。必須。タスク名。 - <i>taskLocation</i>。タスク定義を含むプロジェクトまたはプロジェクト\フォルダ。デフォルト値は [デフォルト] です。 - <i>scope</i>。必須。プライマリキー (PK) カラムまたはすべて (ALL) のカラムの CDC スクリプトを生成または実行します。 - <i>execute</i>。CDC スクリプトを実行します。 - <i>download</i>。CDC スクリプトをダウンロードします。 - <i>path</i>。CDC スクリプトの場所へのオプションのパス。ファイルまたはフォルダを指定できます。 <p>たとえば、スクリプトファイルの名前と場所を指定するには、C:\cli\cdc.txt と入力します。その後、CLI は CDC スクリプトを cli ディレクトリの cdc.txt ファイルに保存します。フォルダのみを指定するには、C:\cli\ と入力します。次に、CLI は生成された CDC スクリプトファイルを cli フォルダに保存します。</p> <p><i>path</i> 値が指定されていない場合、スクリプトファイルは CLI の場所とファイルに保存されます。</p> <pre>cliLocation\cdc_script_taskName_currentTimestamp.txt</pre> <p>次のオプションは、Microsoft SQL Server データベース統合ソースの場合にのみ有効です。</p> <ul style="list-style-type: none"> - <i>msRoleName</i>。データを変更するためのアクセスをゲートするために使用されるデータベースロールの名前。ロール名を指定しない場合、データベースはゲーティングロールを使用しません。 - <i>msFileGroupName</i>。キャブチャ用に作成される変更テーブルに使用するファイルグループの名前。ファイルグループ名を指定しない場合、デフォルトのファイルグループがデータベースであれば、変更テーブルはデフォルトのファイルグループに配置されます。
<pre>task deploy [--taskName <i>taskName</i>] [--taskLocation <i>task_location</i>]</pre>	<p>アプリケーション取り込みタスクまたはデータベース統合タスクをデプロイします。</p> <p>構文:</p> <pre>task deploy [--taskName <i>taskName</i>] [--taskLocation <i>task_location</i>]</pre> <p>有効なオプションは以下のとおりです。</p> <ul style="list-style-type: none"> - <i>taskName</i>。タスク名。 - <i>taskLocation</i>。タスク定義を含むプロジェクトまたはプロジェクト\フォルダ。

コマンド	説明
task replace --pathToConfig <i>yml_file</i> [--deploy] [--noprompt]	<p>デプロイ解除されているジョブと一度もデプロイされていないタスクのみで、タスク定義をタスクの更新バージョンに置き換えます。必要に応じて、--deploy オプションを含めて、更新されたタスクをデプロイすることもできます。</p> <p>構文:</p> <pre>task replace --pathToConfig <yml_file> [--deploy] [--noprompt]</pre> <ul style="list-style-type: none"> - pathToConfig。更新または作成した CLI YAML 構成ファイルを指します。タスクの名前や場所を更新することはできません。 - deploy。オプション。更新されたタスク定義をデプロイします。 - noprompt。オプション。置換操作に対して次の確認プロンプトを表示しないようにして、自動化されたスクリプトとタスクを続行できるようにします。 <p>The "Replace" command overwrites the current task definition with the given configuration file. Type "Y" to confirm your request:</p> <p>CLI YAML 構成ファイルがすでに存在する場合は、それを変更できます。タスクウィザードからタスクを作成し、CLI 構成ファイルが存在しない場合は、DBMI パッケージで提供されているサンプルテンプレートファイルを使用して、YAML 構成ファイルを作成できます。</p> <pre><secure_agent_location>\apps\Database_Ingestion\<version>\dbmicli\examples</pre> <p>注: コマンドの実行中に、既存のタスクと更新された構成ファイルとの検証や比較は行われません。置き換える構成ファイルが正しいことを確認してください。</p>
タスクのステータス [-f <i>field_names</i>] [--taskName <i>taskName</i>] [--taskLocation <i>task_location</i>] [--taskType <i>task_type</i>]	<p>取り込みタスクに関する情報を表示します。</p> <p>構文:</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>デフォルトでは、CLI はタスクごとに次の値の文字列を返します。</p> <pre>name;loadType;sourceConnection;targetConnection;jobName;jobStatus</pre> <p>値が返されるフィールドを変更するには、次のフィールド名を 1 つ以上指定して -f オプションを使用します。</p> <ul style="list-style-type: none"> - name。タスク名。 - location。タスク定義を含むプロジェクトまたはプロジェクト\フォルダ。 - runtimeEnvironment。タスクが実行されるランタイム環境。 - loadType。データベース取り込みタスクを実行するロード操作のタイプ。 - sourceConnection。ソースシステムの接続。 - sourceSchema。ソーステーブルまたはオブジェクトを含むソーススキーマ。 - targetConnection。ターゲットタイプの接続。 - targetSchema。ターゲットテーブルが作成されるターゲットスキーマ。 - schedule。初期ロードタスクのジョブインスタンスを実行するために使用する事前定義されたスケジュールの名前。 - jobName。ジョブ名。 - jobStatus。ジョブステータス。 <p>タスクをフィルタリングするには、次のオプションを 1 つ以上含めます。</p> <ul style="list-style-type: none"> - taskName。情報を表示するタスクの名前。 - taskLocation。タスク定義を含むプロジェクトまたはプロジェクト\フォルダ。 - taskType。取り込みタスクのタイプ。アプリケーション取り込みの場合は appmi、データベース取り込みの場合は dbmi です。このオプションを指定しない場合、両方の取り込みタイプが含まれます。

入力ファイル形式

一括取り込みコマンドラインインタフェースは、YAML 形式の入力ファイルを使用してアプリケーション取り込みタスクまたはデータベース統合タスクを作成します。YAML (.yaml) ファイルでは、必須フィールドの欠落やフィールド名の誤りなどのエラーの可能性があるかどうかを検証されます。サンプルの.yaml ファイルは、dbmicli/examples ディレクトリに、Secure Agent ファイルの一部としてあります。サンプルファイルを別の名前でコピーし、必要に応じてカスタマイズします。

ヒント： Informatica Cloud Secure Agent の下にインストールされている dbmicli ディレクトリをリモートの場所にコピーしてから、YAML ファイルをそのフォルダに追加することをお勧めします。これにより、その後の Secure Agent のアップグレードで YAML ファイルが上書きされるのを防ぐことができます。task create などのコマンドを発行するときに、使用する YAML ファイルへのパスを指定できます。

初期パラメータ

YAML ファイルの先頭、General セクションの前に、次のパラメータを入力できます。

taskType

取り込みタスクのタイプ。アプリケーション取り込みタスクの場合は「appmi」、データベース取り込みタスクの場合は「dbmi」です。

タスクタイプを入力ファイルで指定しない場合は、task create、task createFrom、または task multipleCreateFrom コマンドに含めることができます。

デフォルトは dbmi です。

general セクションパラメータ

YAML 入力ファイルの general セクションでは、タスク名、プロジェクトまたはプロジェクトフォルダの場所、ロード操作のタイプなど、タスクに関するいくつかの基本情報を指定します。

name

取り込みタスクの名前。

description

タスクのオプションの説明。

location

タスク定義を含むプロジェクトまたはプロジェクト\フォルダ。

runtimeEnvironment

タスクを実行するランタイム環境。

type

アプリケーション取り込みタスクまたはデータベース統合タスクで実行するロード操作のタイプ。有効な値は以下のとおりです。

- initial
- incremental
- combined (初期と増分)

source セクションパラメータ

YAML 入力ファイルの source セクションでは、ソースオプションを指定します。特に明記しない限り、オプションはアプリケーション取り込みとデータベース取り込みの両方に関係します。

connection

ソース: すべて

ソースシステムの接続の名前。

salesforceAPI

ソース: Salesforce、初期ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせ。

アプリケーション取り込みタスクの場合のみ、ソースデータの取得に使用する Salesforce API のタイプ。有効な値は以下のとおりです。

- **Standard (REST) API:** Base64 データ型のソースフィールドをレプリケートします。
- **Bulk API 2.0:** Base64 データ型のソースフィールドのレプリケーションを除外します。デフォルトでは、YAML ファイルで API タイプとして Bulk API 2.0 が設定されています。

schema

ソース: MongoDB を除くすべてのデータベース取り込みソース

データベース取り込みタスクの場合のみ、ソーステーブルを含むソーススキーマ。

database

ソース: MongoDB

データベース取り込みタスクの場合のみ、コレクションをソースデータとともに保存する MongoDB データベース。

journalName

ソース: Db2 for i、増分ロード

データベース取り込みタスクの場合のみ、ソーステーブルに加えられた変更を記録するジャーナルの名前。

replicationSlotName

ソース: PostgreSQL

データベース取り込みタスクの場合のみ、PostgreSQL レプリケーションスロットの一意の名前。

replicationPlugin

ソース: PostgreSQL

データベース取り込みタスクの場合のみ、PostgreSQL レプリケーションプラグイン。有効な値は以下のとおりです。

- pgoutput
- wal2json

publication

ソース: PostgreSQL

データベース取り込みタスクの場合のみ、pgoutput プラグインによって使用されるパブリケーション名。このパラメータは、replicationPlugin 値として pgoutput を指定した場合にのみ使用します。

selectionRules

ソース: すべて

ソースオブジェクトまたはテーブルのサブセットを選択するためのオプションのオブジェクトまたはテーブル選択ルール。デフォルトでは、アスタリスク (*) ワイルドカード文字のみを含む含めるルールが使用されます。このルールは、アプリケーション取り込みタスクの場合はソース上のすべてのオブジェクトを選択し、データベース統合タスクの場合はソーススキーマ内のすべてのテーブルを選択します。タスクが

処理するソースオブジェクトまたはテーブルを絞り込むには、追加の含めるルール、除外ルール、または両方のタイプのルールを定義できます。

例:

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

restartPointForIncrementalLoad

ソース: アプリケーション取り込み増分ロードジョブの SAP ECC ソース。データベース取り込み増分ロードジョブと、初期ロードジョブと増分ロードジョブの組み合わせのすべてのソース。

アプリケーション取り込みジョブまたはデータベース統合ジョブが最初に行われたときに変更レコードの読み取りを開始する、ソース変更ストリームまたはログ内の位置。有効なタイプは次のとおりです。

- earliest
- latest

例:

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

データベース統合ジョブの場合、デフォルトは latest です。

アプリケーション取り込みジョブの場合、**earliest** オプションはサポートされておらず、**latest** オプションは SAP ECC ソースにのみ適用されます。デフォルトは、restartPointForIncrementalLoadTimestamp の値です。

このパラメータを指定する場合、restartPointForIncrementalLoadPosition パラメータまたは restartPointForIncrementalLoadTimestamp パラメータと一緒に指定しないでください。

restartPointForIncrementalLoadPosition

ソース: データベース取り込み増分ロードジョブと、初期ロードジョブと増分ロードジョブの組み合わせのすべてのソース。

データベース統合ジョブが最初に行われたときに変更レコードの読み取りを開始する、ソース変更ストリームまたはログ内の RBA 位置。例:

```
restartPointForIncrementalLoadPosition: 0
```

このパラメータは、アプリケーション取り込みジョブではサポートされません。

このパラメータを指定する場合、restartPointForIncrementalLoad パラメータまたは restartPointForIncrementalLoadTimestamp パラメータと一緒に指定しないでください。

restartPointForIncrementalLoadTimestamp

ソース: アプリケーション増分ロードジョブ、データベース取り込み増分ロードジョブ、および初期ロードジョブと増分ロードジョブの組み合わせのすべてのソース。

アプリケーション取り込みジョブまたはデータベース統合ジョブが最初に行われたときに変更レコードの読み取りを開始する、ソース変更ストリームまたはログ内の日付と時刻 (AM または PM を含む)。例:

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

アプリケーション取り込みジョブの場合、このパラメータはデフォルトの動作を決定します。

このパラメータを指定する場合、restartPointForIncrementalLoad パラメータまたは restartPointForIncrementalLoadPosition パラメータと一緒に指定しないでください。

cdcinterval

ソース: Salesforce、増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせ

アプリケーション取り込みタスクの場合のみ、アプリケーション取り込みジョブを実行して増分ロードの変更レコードを取得する時間間隔。cdcIntervalDays、cdcIntervalHours、および cdcIntervalMins パラメータを設定して間隔を指定できます。

fetchSize

ソース: Salesforce、初期ロードタスク、および増分ロードタスク

アプリケーション取り込みタスクの場合のみ、アプリケーション取り込みジョブがソースから一度に読み取るレコード数。初期ロードタスクのデフォルト値は 50000 で、増分ロードタスクのデフォルト値は 2000 です。salesforceAPI パラメータに Standard (REST) API を指定した場合は、fetchSize 値を 2000 に変更する必要があります。

fetchSizeForInitialLoad

ソース: Salesforce、初期ロードタスクと増分ロードタスクの組み合わせ

アプリケーション取り込みタスクの場合のみ、アプリケーション取り込みジョブがデータの初期ロード中に一度に読み取るソースレコード数。デフォルト値は 50000 です。salesforceAPI パラメータに Standard (REST) API を指定した場合は、fetchSize 値を 2000 に変更する必要があります。

fetchSizeForIncrementalLoad

ソース: Salesforce、初期ロードタスクと増分ロードタスクの組み合わせ

アプリケーション取り込みタスクの場合のみ、アプリケーション取り込みジョブが変更データの増分ロード中に一度に読み取るソースレコード数。デフォルト値は 2000 です。初期ロードタスク、または初期ロードタスクと増分ロードタスクの組み合わせで、salesforceAPI パラメータに Standard (REST) API を指定した場合は、fetchSize 値を 2000 に変更する必要があります。

includeArchivedAndDeletedRows

ソース: Salesforce、初期ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせ

アプリケーション取り込みタスクの場合のみ、データの初期ロード中に、アーカイブされて論理削除された行をソースから取り込みます。デフォルトの値は false です。

includeBase64Fields

ソース: Salesforce

アプリケーション取り込みタスクの場合のみ、Base64 データ型のソースフィールドを取り込みます。デフォルトの値は false です。

注: このパラメータは、salesforceAPI パラメータを Standard (REST) API に設定した場合にのみ設定します。

maximumBase64BodySize

ソース: Salesforce

アプリケーション取り込みタスクの場合のみ、Base64 エンコードデータの本文の最大サイズ。デフォルト値は 7MB です。

注: このパラメータは、includeBase64Fields パラメータを yes に設定した場合にのみ設定します。

includeViews

ソース: Microsoft SQL Server、Oracle

データベース取り込みタスクの場合のみ、データベース統合タスクのテーブル数とテーブル名のリストにビューを含めるかどうかを示します。有効な値は以下のとおりです。

- true。ビューを含めます。
- false。ビューを含めません。

customProperties

ソース: すべて

特別な要件を満たすために Informatica が提供するカスタムプロパティ。これらのプロパティを指定する場合は、Informatica グローバルカスタマサポートにお問い合わせください。

例:

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

target セクションパラメータ

YAML 入力ファイルの target セクションでは、ターゲットオプションを指定します。

connection

ターゲット: すべて

ターゲットシステムの接続の名前。

schema

ターゲット: Amazon Redshift、Databricks Delta、Google BigQuery、Microsoft Azure Synapse Analytics、Oracle、Snowflake

一括取り込みがターゲットオブジェクトまたはテーブルを作成するターゲットスキーマ。

bucket

ターゲット: Amazon Redshift、Google Cloud Storage、Google BigQuery

ターゲットにロードするデータオブジェクトへのアクセスを保存、整理、制御する既存のバケットコンテナの名前を指定します。

directory

ターゲット: Databricks Delta、Amazon Redshift、Google Cloud Storage、Google BigQuery

データを含むターゲットオブジェクトの仮想ディレクトリを指定します。

useTableNameAsTopicName

ターゲット: Apache Kafka

一括取り込みがソースデータを含むメッセージを、ソーステーブルまたはオブジェクトごとに 1 つずつ個別のトピックに書き込むか、すべてのメッセージを 1 つのトピックに書き込むかを示します。有効な値は以下のとおりです。

- true。メッセージをテーブル固有の別のトピックに書き込みます。
- false。topicName パラメータで指定された名前の単一のトピックにすべてのメッセージを書き込みます。

includeSchemaName

ターゲット: Apache Kafka

useTableNameAsTopicName を true に設定している場合、この設定により、テーブル固有のトピック名にソーススキーマ名が追加されます。トピック名の形式は次のとおりです。スキーマ名_テーブル名。

有効な値は以下のとおりです。

- true。テーブル固有のトピック名にソーススキーマ名が追加されます。
- false。テーブル固有のトピック名にソーススキーマ名が追加されません。

tablePrefix

ターゲット: Apache Kafka

useTableNameAsTopicName を true に設定している場合、このパラメータは、テーブル固有のトピック名に追加するオプションのプレフィックスを指定します。例えば、myprefix_ を指定すると、トピック名の形式は「myprefix_ テーブル名」になります。プレフィックスの後のアンダースコア (_) を省略すると、プレフィックスがテーブル名の前に追加されます。

tableSuffix

ターゲット: Apache Kafka

useTableNameAsTopicName を true に設定している場合、このパラメータは、テーブル固有のトピック名に追加するオプションのサフィックスを指定します。例えば、_mysuffix を指定すると、トピック名の形式は「テーブル名_mysuffix」になります。サフィックスの前のアンダースコア (_) を省略すると、サフィックスがテーブル名に追加されます。

topicName

ターゲット: Apache Kafka

ソースデータを含むすべてのメッセージが書き込まれる単一の Kafka トピックの名前。
useTableNameAsTopicName が false に設定されている場合は、このパラメータを使用します。

stage: ステージ

ターゲット: Snowflake

データがターゲットテーブルに書き込まれる前にソースから読み取られたデータを保持する内部ステージング領域の名前。この名前にスペースを含めることはできません。指定されたステージング領域が存在しない場合、自動的に作成されます。

outputFormat

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage、Kafka

出力ファイルの形式。有効な値は以下のとおりです。

- Avro
- CSV
- PARQUET

デフォルトの値は [CSV] です。

注: CSV 形式の出力ファイルでは、各フィールドの区切り文字として二重引用符 (") が使用されます。

parquetFormat

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage、Kafka

outputFormat パラメータの出力形式として AVRO を指定している場合、このパラメータを true に設定して、非圧縮の Parquet 形式でデータを書き込むようにします。または、outputFormat パラメータを PARQUET に設定し、このパラメータを含めないようにすることもできます。

重要: このオプションを true に設定した場合、Secure Agent が実行されているコンピュータに Visual Studio2013 の Visual C++再配布可能パッケージをインストールする必要があります。

有効な値は以下のとおりです。

- true. 非圧縮の Parquet 形式でデータを書き込みます。
- false. AVRO 形式でデータを書き込みます。

avroFormat

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage、Kafka

出力形式として AVRO を指定している場合、ソーステーブルまたはオブジェクトごとに作成される Avro スキーマの形式を指定します。有効な値は以下のとおりです。

- Avro-Flat この Avro スキーマ形式は、すべての Avro フィールドを 1 つのレコードにリストします。
- Avro-Generic この Avro スキーマ形式は、ソーステーブルまたはオブジェクトのすべてのカラムまたはフィールドを Avro フィールドの単一配列にリストします。
- Avro-Nested この Avro スキーマ形式は、各タイプの情報を個別のレコードに編成します。

デフォルト値は [Avro-Flat] です。

avroSerializationFormat

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage、Kafka

出力形式として AVRO を指定している場合、Avro 出力ファイルのシリアル化形式を指定します。有効な値は以下のとおりです。

- バイナリ
- JSON

デフォルト値は [Binary] です。

avroSchemaDirectory

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage、Kafka

出力形式として AVRO を指定している場合は、一括取り込みが各ソーステーブルまたはオブジェクトの Avro スキーマ定義を格納しているローカルディレクトリを指定します。スキーマ定義ファイルの命名パターンは、スキーマ名_テーブル名.txt です。

注: このディレクトリが指定されていない場合、Avro スキーマ定義ファイルは作成されません。

fixedDirectoryForEachTable

ターゲット: Amazon S3、Google Cloud Storage、または Microsoft Azure Data Lake Storage Gen2

初期ロードタスクの場合、すべてのジョブ実行で、一括取り込みがソースデータを含むフラットファイルを書き込む先のディレクトリ名として、ソーステーブルまたはオブジェクト名を使用するかどうかを指定します。

有効な値は以下のとおりです。

- true。各ジョブ実行のディレクトリ名として、ソーステーブルまたはオブジェクト名を使用します。
- false。テーブル名_タイムスタンプという命名パターンを使用して、ジョブ実行ごとに新しいディレクトリのセットを作成します。

fileCompressionType

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage

出力形式が outputFormat パラメータで AVRO または CSV 形式に設定されている場合は、必要に応じてファイル圧縮タイプを指定します。有効な値は以下のとおりです。

- Deflate

- gzip
- Snappy

注: このパラメータを指定しない場合、出力ファイルは圧縮されません。

avroCompressionType

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Kafka、Microsoft Azure Data Lake Storage

出力形式が `outputFormat` パラメータで AVRO に設定されている場合は、必要に応じて Avro 圧縮タイプを指定します。有効な値は以下のとおりです。

- なし
- Bzip2
- Deflate
- Snappy

デフォルト値は **None** です。これは圧縮が使用されないことを意味します。

parquetCompressionType

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Kafka、Microsoft Azure Data Lake Storage

`outputFormat` パラメータが PARQUET に設定されている場合、または `outputFormat` パラメータが AVRO に設定されており、`parquetFormat` パラメータが `true` に設定されている場合は、必要に応じて Parquet 圧縮タイプを指定します。有効な値は以下のとおりです。

- なし
- gzip
- Snappy

deflateCompressionLevel

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Kafka、Microsoft Azure Data Lake Storage

`avroCompressionType` パラメータで Deflate を指定している場合、0 から 9 までの圧縮レベルを指定します。デフォルトは 0 です。

addDirectoryTags

ターゲット: Amazon S3、Google Cloud Storage、Microsoft Azure Data Lake Storage

Hive パーティショニングの命名規則と互換性を持たせるために、一括取り込みが適用サイクルディレクトリの名前に「dt=」プレフィックスを追加するかどうかを示します。有効な値は以下のとおりです。

- `true`。適用サイクルディレクトリの名前に「dt=」プレフィックスを追加します。
- `false`。適用サイクルディレクトリの名前は変更しません。

デフォルトの値は `false` です。

directoryTags

ターゲット: Amazon S3、Google Cloud Storage、Microsoft Azure Data Lake Storage

このフィールドは、`addDirectoryTags` フィールドに依存しています。

例:

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


renamingRules

ターゲット: すべて

選択したソーステーブルまたはオブジェクトに対応するターゲットテーブルの名前を変更するためのオプションのルール。

例:

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

dataTypeRules

ターゲット: Amazon Redshift、Databricks Delta、Google BigQuery、Oracle、Snowflake、Synapse Analytics

ソースデータ型からターゲットデータ型へのデフォルトのマッピングをオーバーライドするオプションのデータ型マッピングルール。デフォルトのマッピングについては、「デフォルトのデータ型マッピング」で説明しています。

例:

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

cdcCompatibleFormat

ターゲット: Amazon S3、フラットファイル、Google Cloud Storage、Kafka、Microsoft Azure Data Lake Storage

ジョブがターゲットに書き込む出力に、一括取り込みが UNDO データを含めるかどうかを示します。デフォルトの値は false です。

詳細パラメータ

addOperationType

ターゲット: Amazon S3、Apache Kafka、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage

増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせの場合は、このパラメータを true に設定して、ジョブがターゲットにプロパゲートする出力にソース SQL 操作タイプを含むメタデータカラムを追加します。増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせの場合、デフォルトは true です。初期ロードタスクの場合、デフォルトは false です。

addOperationTime

ターゲット: Amazon S3、Apache Kafka、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage

増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせの場合は、このパラメータを true に設定して、ジョブがターゲットにプロパゲートする出力にソース SQL 操作時間を含むメタデータカラムを追加します。初期ロードの場合、ジョブは常に現在の日付と時刻を書き込みます。デフォルトは、すべてのロードタイプで false です。

addOperationOwner

ターゲット: Amazon S3、Apache Kafka、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage

増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせの場合、このパラメータを true に設定して、ジョブがターゲットにプロパゲートする出力にソース SQL 操作の所有者を含むメタデータカラムを追加します。初期ロードの場合、ジョブは常に所有者として「INFA」を書き込みます。デフォルトは false です。アプリケーション取り込みジョブには使用されません。

addOperationTransactionId

ターゲット: Amazon S3、Apache Kafka、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage

増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせの場合、このパラメータを true に設定して、ジョブが SQL 操作のターゲットにプロパゲートする出力にソーストランザクション ID を含むメタデータカラムを追加します。デフォルトは false です。アプリケーション取り込みジョブには使用されません。

addBeforeImages

ターゲット: Amazon S3、Apache Kafka、フラットファイル、Google Cloud Storage、Microsoft Azure Data Lake Storage

増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせの場合、このパラメータを true に設定して、増分ロードジョブがターゲットに書き込む出力に UNDO データを含めます。初期ロードの場合、ジョブは null を書き込みます。デフォルトは false です。

asyncWrite

ターゲット: Kafka

Kafka へのメッセージの同期配信を使用するかどうかを制御します。有効な値は以下のとおりです。

- true。非同期配信を使用します。一括取り込みは、ソースから変更が取得された順序に関係なく、できるだけ早くメッセージを送信します。
- false。同期配信を使用します。Kafka は、一括取り込みが次のメッセージを送信する前に、各メッセージの受信を確認する必要があります。このモードでは、Kafka が重複メッセージを受信する可能性はほとんどありません。ただし、パフォーマンスが低下する可能性があります。

デフォルト値は true です。

producerConfigurationProperties

ターゲット: Kafka

Apache Kafka、Confluent Kafka、または Kafka 対応の Event Hubs ターゲットの Kafka プロデューサプロパティを入力するための、*key=value* ペアのカンマ区切りリスト。

Confluent Schema Registry を使用してスキーマを格納する Confluent ターゲットがある場合は、次のプロパティを指定する必要があります。

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

Kafka プロデューサプロパティは、このパラメータまたは Kafka 接続の **【追加の接続プロパティ】** フィールドのいずれかに設定します。

このパラメータでプロデューサプロパティを指定すると、プロパティはこのタスクにのみ関連付けられた取り込みジョブに関係します。接続のプロデューサプロパティを入力する場合、**【プロデューサ設定のプロパティ】** フィールドのプロパティを指定して特定のタスクの接続レベルのプロパティをオーバーライドしない限り、プロパティは接続定義を使用するすべてのタスクのジョブに関係します。

Kafka プロデューサのプロパティについては、Apache Kafka のドキュメントまたは Confluent Kafka のドキュメントを参照してください。

カスタムパラメータ

customProperties

ターゲット: すべて

特別な要件を満たすために Informatica が提供するカスタムプロパティ。これらのプロパティを指定する場合は、Informatica グローバルカスタマサポートにお問い合わせください。

例:

```
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 セクションパラメータ

YAML 入力ファイルの runtimeOptions セクションは、任意のロードタイプのジョブのランタイムオプションを設定します。

schemaDriftOptions

ソース: Microsoft SQL Server または Oracle、増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせ。Synapse ターゲットのサポートは、カラム名変更 DDL 操作の値を無視します。Microsoft SQL Server、Oracle または PostgreSQL、増分ロードタスク、および初期ロードタスクと増分ロードタスクの組み合わせもサポートされています。

サポートされている各タイプの DDL 操作に使用するスキーマドリフトオプション:

- addColumn
- modifyColumn
- dropColumn
- renameColumn

有効な値は以下のとおりです。

- ignore
- replicate
- stop job
- stop table

例:

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

numberOfRowsInOutputFile

ターゲット: Amazon Redshift、Amazon S3、Google Big Query、Google Cloud Storage、Microsoft Azure Data Lake Storage、Microsoft Azure Synapse Analytics、Oracle、Snowflake

アプリケーション取り込みタスクまたはデータベース統合タスクがターゲットの出力データファイルに書き込む最小行数を指定します。

fileExtensionBasedOnFileType

ターゲット: フラットファイル、Amazon S3、Google Cloud Storage、Microsoft Azure Data Lake Storage

出力データファイルの拡張子を.dat にするかどうかを示します。

- true。出力ファイルには、ファイルタイプに基づいたファイル名拡張子が付きます。
- false。出力ファイルの拡張子は.dat です。

デフォルト値は true です。

詳細パラメータ

applyCycleChangeLimit

ターゲット: Amazon S3、Google Cloud Storage、または Microsoft Azure Data Lake Storage Gen2

Amazon S3 または Microsoft Azure Data Lake Storage Gen2 ターゲットを持つアプリケーション取り込み増分ロードジョブまたはデータベース取り込み増分ロードジョブの場合は、ジョブが適用サイクルを終了する前に処理する必要があるレコードの数。このレコード制限に達すると、ジョブは適用サイクルを終了し、変更データをターゲットに書き込みます。デフォルトは 10000 です。

注: applyCycleChangeLimit パラメータまたは applyCycleInterval*time_unit* パラメータにはゼロ以外の値が必要です。

applyCycleIntervalDays

ターゲット: Amazon S3、Google Cloud Storage、または Microsoft Azure Data Lake Storage Gen2

Amazon S3 または Microsoft Azure Data Lake Storage Gen2 ターゲットを持つアプリケーション取り込み増分ロードジョブまたはデータベース取り込み増分ロードジョブの場合は、アプリケーション取り込みジョブまたはデータベース取り込みジョブが適用サイクルを終了するまでに経過する必要がある日数。デフォルトは 0 です。applyCycleIntervalHours、applyCycleIntervalMins、および applyCycleIntervalSecs でこのパラメータを指定するか、これらのパラメータのサブセットを指定できます。

applyCycleIntervalHours

ターゲット: Amazon S3、Google Cloud Storage、または Microsoft Azure Data Lake Storage Gen2

Amazon S3 または Microsoft Azure Data Lake Storage Gen2 ターゲットを持つアプリケーション取り込み増分ロードジョブまたはデータベース取り込み増分ロードジョブの場合は、アプリケーション取り込みジョブまたはデータベース取り込みジョブが適用サイクルを終了するまでに経過する必要がある時間数。デフォルトは 0 です。

applyCycleIntervalMins

ターゲット: Amazon S3、Google Cloud Storage、または Microsoft Azure Data Lake Storage Gen2

Amazon S3 または Microsoft Azure Data Lake Storage Gen2 ターゲットを持つアプリケーション取り込み増分ロードジョブまたはデータベース取り込み増分ロードジョブの場合は、アプリケーション取り込みジョブまたはデータベース取り込みジョブが適用サイクルを終了するまでに経過する必要がある分数。デフォルトは 15 です。

applyCycleIntervalSecs

ターゲット: Amazon S3、Google Cloud Storage、または Microsoft Azure Data Lake Storage Gen2

Amazon S3 または Microsoft Azure Data Lake Storage Gen2 ターゲットを持つアプリケーション取り込み増分ロードジョブまたはデータベース取り込み増分ロードジョブの場合は、アプリケーション取り込みジョブまたはデータベース取り込みジョブが適用サイクルを終了するまでに経過する必要がある秒数。デフォルトは 0 です。

lowActivityFlushHours

ターゲット: Amazon S3、Google Cloud Storage、または Microsoft Azure Data Lake Storage Gen2

アプリケーション取り込みジョブまたはデータベース取り込みジョブが適用サイクルを終了する前に、ソースで変更アクティビティがない期間中に経過する必要がある期間（時間単位）。このパラメータを lowActivityFlushMins と組み合わせて使用することも、いずれか 1 つだけを指定することもできます。この時間制限に達すると、取り込みジョブは適用サイクルを終了し、変更データをターゲットに書き込みます。デフォルトは 0 です。

lowActivityFlushHours または lowActivityFlushMins の値を指定しない場合、アプリケーション取り込みジョブは、**applyCycleChangeLimit** またはサイクル間隔の制限時間に達した後でのみ適用サイクルを終了します。

lowActivityFlushMins

ターゲット: Amazon S3、Google Cloud Storage、または Microsoft Azure Data Lake Storage Gen2

アプリケーション取り込みジョブまたはデータベース取り込みジョブが適用サイクルを終了する前に、ソースで変更アクティビティがない期間中に経過する必要がある期間（分単位）。このパラメータを lowActivityFlushHours と組み合わせて使用することも、いずれか 1 つだけを指定することもできます。全体の時間制限に達すると、取り込みジョブは適用サイクルを終了し、変更データをターゲットに書き込みます。デフォルトは 0 です。

lowActivityFlushHours または lowActivityFlushMins の値を指定しない場合、アプリケーション取り込みジョブまたはデータベース取り込みジョブは、**applyCycleChangeLimit** またはサイクル間隔の制限時間に達した後でのみ適用サイクルを終了します。

チェックポイントパラメータ

checkpointAllRows

ターゲット: Apache Kafka

Apache Kafka ターゲットを持つアプリケーション取り込み増分ロードジョブおよびデータベース取り込み増分ロードジョブの場合、Kafka ターゲットに送信されるすべてのメッセージについて取り込みジョブがチェックポイント処理を実行するかどうかを示します。このパラメータが true に設定されている場合、checkpointEveryCommit、checkpointRowCount、および checkpointFrequencySecs パラメータは無視されます。デフォルトは true です。

checkpointEveryCommit

ターゲット: Apache Kafka

Apache Kafka ターゲットを持つアプリケーション取り込み増分ジョブおよびデータベース取り込み増分ロードジョブの場合、ソースで発生するすべてのコミットについて取り込みジョブがチェックポイント処理を実行するかどうかを示します。デフォルトは false です。このパラメータを true に設定する場合、checkpointAllRows を一緒に指定しないでください。

checkpointRowCount

ターゲット: Apache Kafka

Apache Kafka ターゲットを持つアプリケーション取り込み増分ロードジョブおよびデータベース取り込み増分ロードジョブの場合は、チェックポイントを追加する前に取り込みジョブがターゲットに送信するメッセージの最大数。このオプションを 0 に設定すると、取り込みジョブはメッセージの数に基づいてチェックポイント処理を実行しません。このオプションを 1 に設定すると、取り込みジョブは各メッセージにチェックポイントを追加します。デフォルトは 0 です。このパラメータをゼロ以外の数に設定する場合、checkpointAllRows を一緒に指定しないでください。

checkpointFrequencySecs

ターゲット: Apache Kafka

Apache Kafka ターゲットを持つアプリケーション取り込み増分ロードジョブおよびデータベース取り込み増分ロードジョブの場合、取り込みジョブがチェックポイントを追加するまでに経過する必要がある最大秒数。このオプションを 0 に設定すると、取り込みジョブは経過時間に基づいてチェックポイント処理を実行しません。デフォルトは 0 です。このパラメータをゼロ以外の数に設定する場合、`checkpointAllRows` を一緒に指定しないでください。

カスタムパラメータ

customProperties

特別な要件を満たすために Informatica が提供するカスタムプロパティ。これらのプロパティを指定する場合は、Informatica グローバルカスタマサポートにお問い合わせください。

例:

```
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 セクションパラメータ

YAML 入力ファイルの `schedule` セクションは、タスクスケジュールを指定する単一のパラメータで構成されます。スケジュールされたジョブの数が多いと、リソース消費が増加する可能性があります。

schedule

監視インタフェースのいずれかからジョブがデプロイされた後に、ジョブを手動で開始する代わりに、初期ロードタスクの実行ジョブインスタンスを実行するために使用する事前定義されたスケジュールの名前。

デフォルト値

`defaults.yml` ファイルは `/ymls` フォルダにあります。このファイルには、`location`、`salesforceAPI`、`selectionRules`、`cdcIntervalMins`、`fetchSize`、および `restartPointForIncrementalLoad` のデフォルト値が含まれています。必要に応じてこのファイルをカスタマイズできます。

例えば、カスタマイズしたタスク設定値を使用して `yml` ファイルの `selectionRules` を指定しなかった場合、`defaults.yml` ファイルの値が適用されます。

索引

C

Cloud アプリケーション統合コミュニティ
URL [4](#)
Cloud 開発者コミュニティ
URL [4](#)

I

Informatica Intelligent Cloud Services
Web サイト [4](#)
Informatica グローバルカスタマサポート
連絡先情報 [5](#)

W

Web サイト [4](#)

あ

アップグレード通知 [5](#)

し

システムステータス [5](#)

す

ステータス
Informatica Intelligent Cloud Services [5](#)

め

メンテナンスの停止 [5](#)