



Informatica® Mass Ingestion  
April 2024

一括取り込みストリーミング  
グ

Informatica Mass Ingestion 一括取り込みストーリーミング  
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](mailto: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.....   | 5  |
| 第 1 章：一括取り込みストリーミング.....                          | 6  |
| 使用例.....  | 6  |
| 一括取り込みストリーミングソース.....                             | 7  |
| Amazon Kinesis Streams ソース.....                   | 7  |
| AMQP ソース.....                                     | 8  |
| Azure Event Hubs Kafka ソース.....                   | 8  |
| フラットファイルソース.....                                  | 9  |
| Google PubSub ソース.....                            | 9  |
| JMS ソース.....                                      | 9  |
| Kafka ソース.....                                    | 10 |
| MQTT ソース.....                                     | 10 |
| OPC UA ソース.....                                   | 11 |
| REST V2 ソース.....                                  | 11 |
| 一括取り込みストリーミングターゲット.....                           | 12 |
| Amazon Kinesis Data Firehose ターゲット.....           | 12 |
| Amazon Kinesis Streams ターゲット.....                 | 13 |
| Amazon S3 ターゲット.....                              | 13 |
| Databricks Delta ターゲット.....                       | 14 |
| フラットファイルターゲット.....                                | 14 |
| Google BigQuery V2 ターゲット.....                     | 15 |
| Google Cloud Storage V2 ターゲット.....                | 15 |
| Google PubSub ターゲット.....                          | 16 |
| JDBC V2 ターゲット.....                                | 17 |
| Kafka ターゲット.....                                  | 17 |
| Microsoft Azure Data Lake Storage Gen2 ターゲット..... | 17 |
| Microsoft Azure Event Hubs ターゲット.....             | 18 |
| 一括取り込みストリーミングのトランスフォーメーション.....                   | 18 |
| データ形式.....  | 19 |
| Combiner トランスフォーメーション.....                        | 19 |
| フィルタトランスフォーメーション.....                             | 20 |
| 形式変換トランスフォーメーション.....                             | 20 |
| Java トランスフォーメーション.....                            | 20 |
| Jolt トランスフォーメーション.....                            | 22 |
| Python トランスフォーメーション.....                          | 22 |
| Splitter トランスフォーメーション.....                        | 23 |
| ストリーミング統合タスクの設定.....                              | 23 |
| 始める前に.....  | 24 |
| 基本的なタスク情報の定義.....                                 | 24 |

|   |           |
|---|-----------|
| ソースの設定 . . . . .                                | 25        |
| ターゲットの設定 . . . . .                              | 34        |
| トランスフォーメーションの設定 . . . . .                       | 43        |
| ランタイムオプションの設定 . . . . .                         | 49        |
| ストリーミング統合タスクのデプロイ . . . . .                     | 50        |
| ストリーミング統合ジョブのデプロイ解除 . . . . .                   | 50        |
| ストリーミング統合ジョブの停止と再開 . . . . .                    | 51        |
| 一括取り込みストリーミングの動作に関するよくある質問 . . . . .            | 51        |
| <b>第 2 章 : 一括取り込みストリーミング REST API . . . . .</b> | <b>53</b> |
| Dataflows リソース . . . . .                        | 53        |
| ストリーミング取り込みタスクのデプロイ . . . . .                   | 53        |
| ストリーミング取り込みタスクのデプロイ解除 . . . . .                 | 54        |
| ストリーミング取り込みタスクの開始 . . . . .                     | 54        |
| ストリーミング取り込みタスクの停止 . . . . .                     | 55        |
| CopyEntities リソース . . . . .                     | 55        |
| UpdateEntity リソース . . . . .                     | 56        |
| POST 要求 . . . . .                               | 57        |
| POST 応答 . . . . .                               | 72        |
| jobs リソース . . . . .                             | 86        |
| MIJobs リソース . . . . .                           | 87        |
| status リソース . . . . .                           | 90        |
| statistic リソース . . . . .                        | 92        |
| history リソース . . . . .                          | 94        |
| <b>索引 . . . . .</b>                             | <b>96</b> |

# 序文

一括取り込みサービスのストリーミング統合タスクの設定方法については、「一括取り込みストリーミング」をお読みください。

ストリーミング統合タスクをデプロイおよびデプロイ解除する方法を学びます。また、サポートされているソースとターゲット、およびビジネスユーザーが一括取り込みサービスのストリーミング統合タスクをどのように使用できるかについても学ぶことができます。

# 第 1 章

## 一括取り込みストリーミング

一括取り込みストリーミングは一括取り込みサービスの個別にライセンスされた取り込みタイプです。一括取り込みストリーミングは、ログ、クリックストリーム、ソーシャルメディア、IoT ソースなどのストリーミングデータソースから大規模にデータを取り込むことができます。一括取り込みストリーミングを使用して、ストリーミングソースからオンプレミスおよびクラウドストレージに大量のリアルタイムデータを取り込みます。取り込みの進行状況を追跡および監視することもできます。

ストリーミングデータから運用インテリジェンスを収集したり、リアルタイムのデータウェアハウジングを実行したりするには、データが古くなるか破損する前に、データを収集して分析する必要があります。一括取り込みストリーミングを使用して、ストリーミングソースからデータをリアルタイムで結合または分離します。データにシンプルなトランスフォーメーションを適用して、取り込んだデータを分析可能な状態にできます。

一括取り込みサービスには Informatica Intelligent Cloud Services で実行される使いやすいインターフェースがあります。一括取り込みストリーミングサービスを使用して、取り込みジョブを定義、デプロイ、デプロイ解除、および監視します。ジョブは、取り込みタスクの実行可能インスタンスです。さまざまなソースからストリーミングデータと IoT データを収集し、データにシンプルなトランスフォーメーションを適用してから、さまざまなタイプのターゲットにデータを取り込むことができます。Amazon Kinesis、イベントログ、Google PubSub、JMS、Kafka、MQTT、OPC UA、REST V2 などのソースからデータを取り込むことができます。Amazon Kinesis、Amazon S3、Azure Event Hubs、Databricks Delta、Google BigQuery v2、Google Cloud Storage、Google PubSub、Kafka、Microsoft Azure Data Lake Storage などのターゲットにデータをストリーミングできます。

## 使用例

一括取り込みストリーミングは、複数の使用要件を満たすのに役立ちます。

次のシナリオではストリーミング統合を使用することを検討してください。

- **リアルタイム分析。**リアルタイム分析のために、ストリーミングおよび IoT データを Apache Kafka や Amazon Kinesis などのメッセージングシステムに取り込みます。リアルタイム分析は、企業が利益の増加と顧客サービスの向上につながる可能性のあるビジネスチャンスと収益源を特定するのに役立ちます。
- **データ統合。**ストリーミングおよび IoT データを Amazon S3 などのクラウドデータレイクに取り込み、データをリアルタイムで統合して最新の情報を提供します。

# 一括取り込みストリーミングソース

サポートされるストリーミングソースから一括取り込みストリーミングがサポートするオンプレミスおよびクラウドターゲットに大量のリアルタイムデータを取り込むことができます。イベントまたはメッセージの形式でデータを取り込むことができます。

ストリーミング統合タスクでは次のデータソースを使用できます。

- Amazon Kinesis Streams
- AMQP
- Azure Event Hubs Kafka
- フラットファイル
- Google PubSub
- JMS
- Kafka
  - Apache Kafka
  - Confluent Kafka
  - Amazon Managed Streaming (Amazon MSK)
- MQTT
- OPC UA
- REST V2

これらのソースタイプに使用するコネクタを判断するには、「コネクタと接続」 > 「一括取り込みストリーミングコネクタ」を参照してください。

## Amazon Kinesis Streams ソース

Kinesis Streams ソースを使用して、Amazon Kinesis Stream からデータを読み取ります。Kinesis Streams ソース接続を作成するには、Kinesis 接続タイプを使用します。

Kinesis Streams は、Amazon Kinesis が AWS エコシステム内で提供するリアルタイムのデータストリーム処理サービスです。Kinesis Streams は、ストリーミングデータを処理および分析するためのカスタムアプリケーションを構築するために使用できるカスタマイズ可能なオプションです。Kinesis Streams はデータのインフローデマンドに対応するために自動的にスケーリングできないため、システムのニーズを満たすために十分な容量を手動でプロビジョニングする必要があります。

Kinesis ストリームソースを使用する前に、次のタスクを実行します。

1. Amazon Kinesis コンソールで、Amazon Kinesis ストリームを作成して設定します。
2. Amazon Web Services (AWS) Identity and Access Management (IAM) サービスで、ユーザーを作成します。
3. ユーザー作成プロセス中に生成されたアクセスキーとシークレットアクセスキーをダウンロードします。
4. Kinesis ストリームへの書き込み権限を持つグループにユーザーを関連付けます。

一括取り込みストリーミングは、プロファイルベースのクロスアカウント認証はサポートしていません。

Amazon Kinesis に使用される Amazon Web Services 資格情報には、Amazon DynamoDB サービスおよび Amazon CloudWatch サービスにアクセスするための権限が必要です。

## AMQP ソース

Advanced Message Queuing Protocol (AMQP) ソースを使用して、AMQP メッセージキューからメッセージを読み取ります。AMQP ソース接続を作成するには、AMQP 接続タイプを使用します。

AMQP は、キュー、ルーティング、信頼性、およびセキュリティ機能を備えたメッセージ指向の規格です。AMQP は、リアルタイムのメッセージストリームを渡すことでビジネストランザクションを容易にするために使用できる、ワイヤレベルのプラットフォームに依存しないプロトコルです。

AMQP コネクタを使用すると、AMQP ブローカからのメッセージの読み取り、メッセージキューの監視、およびブローカメッセージングのサブスクライブパターンの処理を行うことができます。ストリーミング統合タスクは、AMQP ブローカとして RabbitMQ を使用します。RabbitMQ は、高速かつ拡張可能で堅牢な分散型のメッセージブローカシステムです。RabbitMQ は、メッセージの安全な転送のために AMQP0-9-1 メッセージングプロトコルを使用します。

ストリーミング統合タスクでは、AMQP ソースを使用して、受信メッセージのストリームをサブスクライブできます。AMQP ブローカは、ストリーミング統合ジョブがキューからメッセージを受信するまで、メッセージをメッセージキューに保存します。ストリーミング統合ジョブがメッセージを受信すると、ジョブはメッセージの受信を確認します。応答済みメッセージは、メッセージキューから削除されます。

信頼できるバックグラウンドジョブとして実行する長期間実行タスクがある場合は、AMQP ソースを使用できます。また、Web ショップでの注文処理など、システムの一部が別の部分に通知する必要があるアプリケーション間の通信に AMQP ソースを使用することもできます。

## Azure Event Hubs Kafka ソース

Azure Event Hubs に接続するように Kafka ソースを設定できます。Azure Event Hubs Kafka ソース接続を作成するには、Kafka 接続タイプを使用します。

標準層または専用層の Event Hubs 名前空間を作成すると、名前空間の Kafka エンドポイントがデフォルトで有効になります。次に、Azure Event Hubs 対応の Kafka 接続を、ストリーミング取り込みタスクの設定中にソース接続として使用できます。トピック名として Event Hubs 名を入力します。

ストリーミング取り込みタスクの設定中に入力する Azure Event Hubs ソース情報は、通常の Kafka ソース設定の情報と同じです。Azure Event Hubs Kafka ソースプロパティの詳細については、[「Azure Event Hubs Kafka ソースのプロパティ」 \(ページ 26\)](#)を参照してください。

Administrator で Kafka 接続を作成するときに、次のプロパティを設定します。

- Kafka ブローカリスト: NAMESPACE.servicebus.windows.net:9093
- 追加の接続プロパティ: security.protocol=SASL\_SSL,sasl.mechanism=PLAIN,sasl.kerberos.service.name=Kafka
- SSL モード: 一方向
- SSL トラストストアファイルパス: ファイルシステム上の信頼されたルート証明書へのパス。例:  
<AGENT\_HOME>/jdk/jre/lib/security/cacerts
- SSL トラストストアパスワード: トラストストアパスワード。
- 追加のセキュリティプロパティ: sasl.jaas.config= org.apache.kafka.common.security.plain.PlainLoginModule required username="\$ConnectionString" password="Endpoint=sb://mynamespace.servicebus.windows.net/;SharedAccessKeyName=XXXXXX;SharedAccessKey=XXXXXX";

Azure Event Hubs Kafka ソース接続の作成の詳細については、[接続ヘルプ](#)を参照してください。

**注:** Kafka 用の Event Hubs は、標準層と専用層でのみ使用できます。基本層は、Event Hubs で Kafka をサポートしていません。



## フラットファイルソース

受信リアルタイムデータを読み取るためのソースとしてフラットファイルを使用します。同じディレクトリに保存されているフラットファイルからデータを読み取るようにフラットファイル接続を設定します。

ストリーミング統合タスクはフラットファイルソースの各行を読み取り、設定されたターゲットにデータを取り込みます。フラットファイルがリアルタイムで継続的に更新されると、ストリーミング統合タスクは、ファイル全体を再度読み取るのではなく、新しく追加されたコンテンツのみを読み取ります。

ストリーミング統合は、区切られたフラットファイルからデータを読み取ることができます。区切り文字はキャリッジリターン (\r)、ラインフィード (\n)、または両方の組み合わせである必要があります。

## Google PubSub ソース

Google PubSub ソースを使用して、設定済みの Google Cloud PubSub サブスクリプションからメッセージを読み取ります。Google PubSub ソース接続を作成するには、Google PubSub 接続タイプを使用します。

Google PubSub は、イベントを処理するサービスからイベントを生成するサービスを分離する非同期メッセージングサービスです。Google PubSub は、メッセージング指向ミドルウェアとして、またはストリーミング分析パイプラインのイベントの取り込みと配信に使用できます。Google PubSub は、高可用性と大規模で一貫したパフォーマンスを備えた、耐久性のあるメッセージストレージとリアルタイムのメッセージ配信を提供します。Google PubSub サーバーは、世界中の利用可能なすべての Google Cloud リージョンで実行できます。

Google PubSub コネクタを使用する前に、次の前提条件を満たしていることを確認する必要があります。

- 組織に Google PubSub コネクタライセンスがある。
- Google PubSub にアクセスするための Google サービスアカウントの JSON キーがある。
- Google サービスアカウントの client\_email、project\_id、private\_key の値がある。Administrator で Google PubSub 接続を作成するときに、これらの詳細が必要です。

ストリーミング統合タスクでは、Google PubSub ソースを使用して、Google PubSub トピックからのメッセージをサブスクライブできます。

## JMS ソース

JMS ソースを使用して、JMS プロバイダからデータを読み取ります。JMS ソース接続を作成するには、JMS 接続タイプを使用します。

JMS プロバイダは、メッセージベースのミドルウェアシステムであり、JMS メッセージを送信します。JMS ソースは、メッセージトピックに基づいて JMS プロバイダメッセージキューまたは JMS プロバイダから JMS メッセージを読み取ります。

JMS ソースは、次の JMS メッセージタイプを読み取ることができます。

- メッセージ。ヘッダーフィールドおよびプロパティフィールドのみが含まれます。
- TextMessage。文字列オブジェクトが含まれます。TextMessages には XML または JSON メッセージデータを含めることができます。
- BytesMessage。解釈されていないバイトのストリームです。既存のメッセージフォーマットと一致するようにメッセージ本文をエンコードするときに BytesMessage を使用します。BytesMessages には、通常、プロパティフィールドは含まれません。
- MapMessage：名前または値ペアのセットが含まれます。名前は文字列形式です。値は Java 基本データタイプです。

## JMS メッセージ配信先タイプ

次のいずれかの接続 JMS メッセージ配信先タイプを選択できます。

- **[キュー]**。JMS メッセージプロデューサは、単一のコンシューマにメッセージを配信します。キューからのメッセージをコンシュームするには、コンシューマを登録する必要があります。キューに登録されているコンシューマがない場合、キューは、コンシューマが登録されるまでメッセージを保持します。
- **[トピック]** JMS メッセージプロデューサは、トピックにサブスクライブしているアクティブなすべてのコンシューマにメッセージを配信します。複数のプロデューサがトピックの接続先にメッセージを送信でき、各メッセージを複数のサブスクライバに配信できます。トピックに登録されているコンシューマがない場合、トピックはメッセージを保持しません。サブスクリプションを共有可能、継続、またはその両方にすることができます。共有可能サブスクリプションでは、単一または複数のコンシューマが単一のサブスクリプションにアクセスできます。継続サブスクリプションは、サブスクライバがメッセージをコンシュームするか、メッセージの有効期限が切れるまで、非アクティブなサブスクライバのメッセージを保持します。

## Kafka ソース

Kafka ソースを使用して、Kafka トピックからのメッセージを読み取ります。Kafka ソース接続を作成するには、Kafka 接続タイプを使用します。

Kafka は Publish-Subscribe メッセージシステムです。これは、Kafka トピックのストリーミングデータを永続化するオープンソースの分散ストリーミングプラットフォームです。任意のトピックを、リアルタイムでデータを必要とする任意の数のシステムで読み取ることができます。Kafka は、さまざまなダウンストリームコンシューマアプリケーションが使用できるストリーミングデータの間ステージング領域として使用できます。

Kafka は、ブローカと呼ばれる 1 つまたは複数のサーバーで構成されるクラスタとして動作します。Kafka ブローカはメッセージ形式でデータをストリーミングします。このメッセージはトピックにパブリッシュされます。Kafka ソースを作成するときは、Kafka コンシューマを作成して Kafka トピックからメッセージを読み取ります。

ストリーミング統合タスクでは、Kafka ソースを使用して、受信メッセージのストリームをサブスクライブできます。Kafka トピックから読み取るように Kafka ソースを設定する場合、トピック名を指定するか、Java でサポートされている正規表現を使用して、指定したパターンに一致するすべてのトピックをサブスクライブできます。

同じ Kafka 接続を使用して、Apache Kafka 用の Amazon Managed Streaming (Amazon MSK) または Confluent Kafka ソース接続を作成できます。これで、ストリーミング統合タスクで Amazon MSK ソースまたは Confluent Kafka ソースを使用して、Apache Kafka または Confluent Kafka トピックからのメッセージを読み取ることができます。

## MQTT ソース

MQTT ソースを使用して、MQ Telemetry Transport (MQTT) ブローカからデータを読み取ります。MQTT ソースを作成するには、MQTT 接続タイプを使用します。

MQTT は Publish-Subscribe メッセージシステムです。これは、シンプルで軽量、かつ永続的なメッセージングプロトコルです。制約のあるデバイスや、低帯域幅、長い待機時間、または信頼性の低いネットワーク向けに設計されています。パブリッシャとサブスクライバはいずれも MQTT クライアントです。MQTT はパブリッシャをサブスクライバから切り離すため、ブローカがクライアント接続を管理します。

MQTT ブローカは、すべてのメッセージを受信し、メッセージをフィルタし、各メッセージをサブスクライブしたクライアントを判別してから、サブスクライブしたクライアントにメッセージを送信します。複数の MQTT ソースが 1 つの MQTT ブローカに接続する場合、各接続には一意の識別子が必要です。ストリーミング統合ジョブを実行して MQTT ソースからデータを取り込む場合、ストリーミング取り込みは、ターゲットにデータを書き込む前に、まずデータを内部キューに書き込みます。

**注:** MQTT ソースには、一意のクライアント ID が必要です。2 つの MQTT ソースが同じクライアント ID を持っている場合、MQTT ブローカはクライアントとストリーミング統合ジョブが *[実行中(警告あり)]* 状態になるのを拒否します。

一括取り込みストリーミングは MQTT Quality of Service (QoS) レベル 1 をサポートします。レベル 1 は、クライアントがメッセージをブローカに少なくとも 1 回送信し、メッセージが複数回配信される可能性があることを示します。ブローカがメッセージの受信を確認した後、クライアントは送信キューからメッセージを削除します。QoS レベルは、クライアントからブローカへの通信またはブローカからクライアントへの通信に制限されています。

## OPC UA ソース

OPC UA ソースを使用して、OPC UA アプリケーションタグからメッセージを読み取ります。OPC UA ソース接続を作成するには、OPCUA 接続タイプを使用します。

オープンプラットフォームコミュニケーション (OPC) は、インダストリー 4.0 および IIoT (Industrial Internet Of Things) の重要な通信プロトコルの 1 つです。OPC ユニファイドアーキテクチャ (OPC UA) は、産業オートメーションに使用されるマシン間通信プロトコルです。OPC UA は、企業システム、監視デバイス、および実世界のデータと相互作用するセンサー間でデータを移動するための柔軟で適応性のあるメカニズムを提供します。OPC UA を使用して、単純なダウンタイムステータスまたは大量の非常に複雑なプラント全体の情報の通信を確立できます。

OPC UA ソースは、OPC サーバーからデータを収集するクライアントです。OPC のデータポイントは、デバイスからのデータを表し、データへのリアルタイムアクセスを提供するタグです。ストリーミング統合タスクでは、OPC UA ソースを作成して、指定したタグのリストに基づいて受信データを読み取ることができます。JSON 配列形式でタグをメンションする必要があります。

## REST V2 ソース

REST V2 ソースを使用して、Web サービスアプリケーションからデータを読み取ります。REST V2 ソース接続を作成するには、REST V2 接続タイプを使用します。

REST V2 ソースコネクタは、REST API を使用するクラウドアプリケーション用の汎用コネクタです。これは Swagger 仕様バージョン 2.0 をサポートしています。Swagger 仕様ファイルには、操作 ID、パスパラメータ、クエリパラメータ、ヘッダフィールド、およびペイロードの詳細が含まれます。

Administrator でストリーミング取り込みタスクの REST V2 ソース接続を作成できたら、次のいずれかの REST 認証タイプを設定できます。

- 基本
- OAuth1.0
- OAuth2.0 クライアントの資格情報
- OAuth2.0 認証コード
- JWT ベアラートークン認証

# 一括取り込みストリーミングターゲット

サポートされているソースから、一括取り込みストリーミングがサポートするオンプレミスおよびクラウドのターゲットにストリーミングデータを取り込むことができます。

ストリーミング統合タスクでは次のターゲットを使用できます。

- Amazon Kinesis Data Firehose
- Amazon Kinesis Streams
- Amazon S3
- Databricks Delta
- フラットファイル
- Google BigQuery V2
- Google Cloud Storage
- Google PubSub
- JDBC V2
- Kafka
  - Apache Kafka
  - Confluent Kafka
  - Amazon Managed Streaming (Amazon MSK)
- Microsoft Azure Data Lake Store Gen2
- Microsoft Azure Event Hubs

これらのターゲットタイプに使用するコネクタを判断するには、「コネクタと接続」 > 「一括取り込みストリーミングコネクタ」を参照してください。

## Amazon Kinesis Data Firehose ターゲット

Kinesis ターゲットを使用してソースからデータを受信し、そのデータを Amazon Kinesis Data Firehose ターゲットに書き込みます。Kinesis ターゲットを作成するには、Amazon Kinesis 接続タイプを使用します。

Kinesis Firehose は、Amazon Kinesis が AWS エコシステム内で提供するリアルタイムのデータストリーム処理サービスです。Kinesis Firehose を使用して、データをバッチ処理、暗号化、および圧縮します。Kinesis Firehose は、システムのニーズに合わせて自動的にスケーリングできます。

Kinesis Firehose のアクセスをターゲットとして設定するには、次のタスクを実行します。

- IAM ユーザーが AWS Kinesis DataFirehose サービスを使用するために必要な IAM 権限を持つ AWS アカウントを作成します。
- Firehose 配信ストリームを定義します。ソースを Direct PUT またはその他のソースとして設定します。
- ユーザーが書き込んでいるターゲットに基づいて、IAM ユーザーの資格情報に必要な権限を付与します。権限のリストについては、<https://docs.aws.amazon.com/firehose/latest/dev/controlling-access.html#access-to-firehose> の AWS ドキュメントを参照してください。

## Amazon Kinesis Streams ターゲット

Kinesis ターゲットを使用して、ソースサービスからデータを受信し、そのデータを Amazon Kinesis Stream に書き込みます。Kinesis ターゲットを作成するには、Amazon Kinesis 接続タイプを使用します。

Kinesis Streams は、Amazon Kinesis が AWS エコシステム内で提供するリアルタイムのデータストリーム処理サービスです。Kinesis Streams は、ストリーミングデータを処理および分析するためのカスタムアプリケーションを構築するために使用できるカスタマイズ可能なオプションです。Kinesis Streams はデータのインフローデマンドに対応するために自動的にスケーリングできないため、システムのニーズを満たすために十分な容量を手動でプロビジョニングする必要があります。

Kinesis ストリームターゲットを使用する前に、次のタスクを実行します。

1. Amazon Kinesis コンソールで、Amazon Kinesis ストリームを作成して設定します。
2. Amazon Web Services (AWS) Identity and Access Management (IAM) サービスで、ユーザーを作成します。
3. ユーザー作成プロセス中に生成されたアクセスキーとシークレットアクセスキーをダウンロードします。
4. Kinesis ストリームへの書き込み権限を持つグループにユーザーを関連付けます。

## Amazon S3 ターゲット

Amazon S3 V2 コネクタを使用して、ストリーミングデータを Amazon S3 ターゲットに書き込みます。

Amazon Simple Storage Service (Amazon S3) は、ストリーミングソースからデータをコピーして、任意のターゲットに同時にデータを移動できるストレージサービスです。Amazon S3 を使用して、構成されたソース接続のリストから Amazon S3 ターゲットにデータを転送できます。AWS Management Console の Web インタフェースを使用してこれらのタスクを実行できます。

ストリーミング取り込みタスクのターゲットとして Amazon S3 オブジェクトを使用できます。Amazon S3 ターゲットとターゲットオブジェクトの詳細プロパティを設定できます。

### データのパーティション化

ストリーミング取り込みタスクは、Amazon S3 V2 ターゲットにパーティションを作成し、データをパーティションに書き込むことができます。パーティション化を使用するには、タスクがパーティションを作成する際に従うパーティション化間隔を選択する必要があります。選択した時間間隔に基づいて、ストリーミング取り込みジョブは受信メッセージを Amazon S3 V2 バケットの/<オブジェクト名>/<年>/<月>/<日>/<時間>/<分>パーティションに保存します。ストリーミング取り込みジョブは、Amazon S3 バケットにタイムスタンプ階層フォルダを追加します。

タスクを設定するときに、次の方法でパーティション化を有効にすることができます。

- [オブジェクト名/式] フィールドでオブジェクト名に\${Timestamp}式を追加し、時間間隔を選択します。例えば、**[オブジェクト名/式]** フィールドで、/streaming/\${Timestamp}と入力し、5 分のパーティション化間隔を選択してから、2022 年 6 月 8 日の 15:20 にストリーミング取り込みタスクを実行したとします。ストリーミング取り込みジョブは、受信メッセージを Amazon S3 V2 バケットの/streaming/2022/06/08/15/20 パーティションに保存します。このジョブは、次の時間間隔の際にストリーミングされるデータを、時間フォルダ (2022/06/08/15/25) に保存します。

**注:** [オブジェクト名/式] フィールドで、オブジェクト名に\${Timestamp}式を追加し、パーティション化間隔を選択しなかった場合、ストリーミング取り込みジョブはオブジェクトを Amazon S3 バケットの 0/フォルダに保存します。

- **[オブジェクト名/式]** フィールドで正規表現を使用し、パーティション化間隔を選択します。正規表現に `{Timestamp}` 表現を追加する必要はありません。  
例えば、2022 年 6 月 18 日の 12:10 にストリーミング取り込みタスクを、正規表現 `$"SourceTable": "([.*?])"$` と、5 分のパーティション化間隔で実行したとします。受信データは `{"SourceTable": "xyz"}` です。ストリーミング取り込みジョブは、xyz オブジェクトを Amazon S3 バケットの 2022/06/18/12/10 フォルダ階層に保存します。このジョブは、次の時間間隔の際にストリーミングされるデータを、時間フォルダ (2022/06/08/15/25) に保存します。

詳細については、「[Amazon S3 ターゲットのプロパティ](#)」 (ページ 35) を参照してください。

## Databricks Delta ターゲット

ストリーミング統合タスクを使用して、Databricks Delta ターゲットにデータを書き込みます。Databricks Delta ターゲットを作成するには、Databricks Delta 接続タイプを使用します。Databricks Delta ターゲットには、Databricks クラスタバージョン 6.3 以降が必要です。

Databricks Delta は、ACID トランザクションを提供し、既存のデータレイクの上で機能するオープンソースのストレージレイヤーです。Databricks では、独自の Delta ソフトウェアを使用して保存データを管理するので、データに高速にアクセスできます。

次のストレージタイプに作成した Delta Lake テーブルにアクセスできます。

- Azure Data Lake Storage (ADLS) Gen2
- Amazon Web Services (AWS) S3

Databricks Delta ターゲットは、Databricks 上の 1 つ以上の Delta Lake テーブルにデータを書き込みます。Databricks Delta ターゲットはストリーミング統合タスクで次のユースケースに使用できます。

- すべてのストリーミングソースから Databricks Delta テーブルにバルクデータを取り込む
- すべてのストリーミングソースからの変更データキャプチャ (CDC) をマージし、Databricks Delta テーブルに書き込む

Databricks Delta 接続は、JDBC URL を使用して Databricks クラスタに接続します。ターゲットを設定するときは、クラスタへの接続に使用する JDBC URL と資格情報を指定します。また、ターゲットが Amazon S3 または Azure Data Lake Storage Gen2 のステージングの場所に接続するために使用する接続情報を定義します。

データを書き込む Delta Lake のテーブルを指定します。ターゲットは、一致する名前に基づいて、レコードフィールドからテーブルカラムにデータを書き込みます。

## フラットファイルターゲット

ストリーミング統合タスクを使用して、さまざまなソースからフラットファイルターゲットにデータを書き込みます。このタスクは、さまざまなソースからデータフローを実行する Secure Agent のファイルシステムにリアルタイムストリーミングデータを書き込みます。

ストリーミング統合タスクは、データを指定されたファイル名でステージングディレクトリに書き込みます。タスクがファイルに新しいコンテンツを追加すると、ターゲット内でそのコンテンツの前に改行文字 (`\n`) が付けられます。

フラットファイルターゲットはファイルのロールオーバーアクションを実行するため、ロールオーバープロパティを設定できます。ファイルのロールオーバープロセスでは、現在のファイルを閉じ、ファイルサイズ、イベント数、または時間に基づいて新しいファイルを作成します。ロールオーバーを設定するには、ターゲットサービスの **[ロールオーバーサイズ]**、**[ロールオーバーイベント数]**、または **[ロールオーバー時間]** プロパティを指定します。ロールオーバープロセスでは、ファイルをステージングディレクトリからターゲットに移動し、ファイルの名前を変更します。名前が変更されたファイルのファイル名形式は、元のファイル名にタイムスタンプとカウンタの情報 (yyyy\_mm\_dd-hh\_mm\_ss\_counter) が追加されたものになります。例えば、ロ



ールオーバー時に、ファイル streaming.txt の名前は streaming-2021\_08\_16-17\_17\_30\_4.txt に変更されます。

ロールオーバープロパティは組み合わせて実装できます。例えば、ロールオーバーイベント数を 1000、ロールオーバーサイズを 1 GB、ロールオーバー時間を 1 時間に設定した場合、イベントが 1000 件累積されておらず、1 時間が経過していなくても、ファイルのサイズが 1 GB に達すると、タスクはファイルをロールオーバーします。

## Google BigQuery V2 ターゲット

ストリーミング統合タスクを使用して、Google BigQuery V2 データベースターゲットにデータを書き込みます。Google BigQuery V2 ターゲットを作成するには、Google BigQuery V2 接続タイプを使用します。

Google BigQuery V2 ターゲットを使用して、BigQuery テーブルにデータを書き込むことができます。ターゲットは JSON 形式のデータをコンシュームします。フィールドがデータベースのテーブルカラムと一致しない場合、ターゲットはレコードを無視します。Google BigQuery V2 ターゲットは、単純な JSON 形式のデータ、または単純な JSON 形式の配列のデータを受け入れます。

Google BigQuery V2 コネクタを使用する前に、次の前提条件のタスクを完了する必要があります。

- Google BigQuery にアクセスするための Google サービスアカウントを持っていることを確認してください。
- サービスアカウントに client\_email、project\_id、private\_key、および regionID の値があることを確認してください。Google BigQuery V2 接続を作成するときに、対応する **【サービスアカウント ID】**、**【プロジェクト ID】**、**【サービスアカウントキー】**、および **【リージョン ID】** 接続プロパティに値を入力します。
- private\_key\_id プロパティと client\_id プロパティを **【オプションのプロパティを指定】** フィールドで指定する必要があります。これらのパラメータを指定すると、Google BigQuery V2 接続テストは失敗します。ただし、失敗したテスト接続を無視して、ストリーミング取り込みジョブを実行できます。次の形式を使用します。

```
"private_key_id": "<private key ID>" and "client_id": "<client ID>"
```

- Google BigQuery 接続のタイムアウト間隔を設定する場合は、接続プロパティの **【オプションのプロパティを指定】** フィールドでタイムアウト間隔プロパティを指定します。次の形式を使用します。

```
"timeout": "<timeout_interval_in_seconds>"
```
- ストリーミング取り込みタスクをデプロイする前に、データを書き込むためのテーブルが存在している必要があります。
- ターゲットテーブルを含む Google BigQuery データセットへの読み取りおよび書き込みアクセス権が必要です。

## Google Cloud Storage V2 ターゲット

ストリーミング統合タスクを使用して、Google Cloud Storage ターゲットにデータを書き込みます。Google Cloud Storage ターゲットを作成するには、Google Cloud Storage V2 接続タイプを使用します。

Google Cloud Storage を使用して、マルチメディアをストリーミングしたり、カスタムデータ分析パイプラインを保存したり、直接ダウンロードして大きなデータオブジェクトをユーザーに配信したりできます。データをバックアップするために、Google Cloud Storage にデータを書き込むことができます。データベースに障害が発生した場合は、Google Cloud Storage からデータを読み取り、データベースにリストアできます。

Google Cloud Storage は、データの可用性、待ち時間、価格などの要因に基づいて、さまざまなストレージクラスを提供します。Google Cloud Storage には次のコンポーネントがあります。

- **プロジェクト**。Google Cloud Storage では、すべてのリソースがプロジェクト内に格納されます。プロジェクトは、請求の詳細とユーザーの詳細を格納する最上位のコンテナです。プロジェクトは複数作成できます。プロジェクトには、一意のプロジェクト名、プロジェクト ID、およびプロジェクト番号があります。

- **バケット**。各バケットは、データを格納するコンテナのように機能します。バケットを使用して、データを整理し、データにアクセスできます。複数のバケットを作成できますが、バケットをネストすることはできません。バケット内に複数のフォルダを作成でき、フォルダをネストすることもできます。アクセス制御リストを定義して、オブジェクトとバケットを管理できます。アクセス制御リストは、権限と範囲のエントリで構成されています。権限は、読み取りまたは書き込み操作を実行するためのアクセスを定義します。範囲は、操作を実行できるユーザーまたはグループを定義します。
- **オブジェクト**。オブジェクトは、Google Cloud Storage にアップロードするデータを構成します。バケット内にオブジェクトを作成できます。オブジェクトは、オブジェクトデータとオブジェクトメタデータコンポーネントで構成されます。オブジェクトデータは、Google Cloud Storage に保存するファイルです。オブジェクトメタデータは、オブジェクトの特性を説明する名前と値のペアのコレクションです。

Google Cloud Storage V2 コネクタを使用する前に、次の前提条件のタスクを完了する必要があります。

1. Google Cloud Storage にアクセスするための Google サービスアカウントを持っていることを確認してください。
2. サービスアカウントの `client_email`、`project_id`、`private_key` の値があることを確認してください。Administrator で Google Cloud Storage 接続を作成するときに、これらの詳細を入力する必要があります。
3. サービスアカウントで Google Cloud Storage JSON API が有効になっていることを確認してください。Google Cloud Storage V2 コネクタは、Google API を使用して Google Cloud Storage と統合します。
4. ターゲットファイルを含む Google Cloud Storage バケットへの書き込みアクセスがあることを確認してください。
5. 組織で Cloudera CDH または Hortonworks HDP パッケージを使用するためのライセンスが有効になっていることを確認してください。

ストリーミング統合タスクをデプロイすると、Secure Agent は Google Cloud Storage API を使用して、指定された操作を実行し、Google Cloud Storage ファイルにデータを書き込みます。Google Cloud Storage ターゲットにデータを書き込むことができます。Google Cloud Storage ターゲットで更新、更新/挿入、または削除操作を実行することはできません。

## Google PubSub ターゲット

ストリーミング統合タスクを使用して、Google PubSub トピックにデータを書き込みます。Google PubSub ターゲットを作成するには、Google PubSub 接続タイプを使用します。

Google PubSub は、イベントを処理するサービスからイベントを生成するサービスを分離する非同期メッセージングサービスです。Google PubSub は、メッセージング指向ミドルウェアとして、またはストリーミング分析パイプラインのイベントの取り込みと配信に使用できます。Google PubSub は、高可用性と大規模で一貫したパフォーマンスを備えた、耐久性のあるメッセージストレージとリアルタイムのメッセージ配信を提供します。Google PubSub サーバーは、世界中の利用可能なすべての Google Cloud リージョンで実行できます。

Google PubSub コネクタを使用する前に、次の前提条件を満たしていることを確認する必要があります。

- 組織に Google PubSub コネクタライセンスがある。
- Google PubSub にアクセスするための Google サービスアカウントの JSON キーがある。
- Google サービスアカウントの `client_email`、`project_id`、`private_key` の値がある。Administrator で Google PubSub 接続を作成するときに、これらの詳細が必要です。

ストリーミング統合タスクでは、Google PubSub ターゲットを使用して、Google PubSub トピックにメッセージをパブリッシュできます。



## JDBC V2 ターゲット

ストリーミング統合タスクを使用して、データベースターゲットにデータを書き込みます。JDBC V2 ターゲットを作成するには、JDBC V2 接続タイプを使用します。

JDBC V2 ターゲットを使用して、データベーステーブルにデータを書き込むことができます。ターゲットは JSON 形式のデータをコンシュームします。ターゲットは、データベースのテーブルカラムにマッピングされていないフィールドを無視します。

JDBC V2 をターゲットとして構成する前に、以下の前提条件を考慮してください。

- ストリーミング取り込みタスクをデプロイする前に、データを書き込むためのテーブルが存在している必要があります。
- データベースドライバファイルを次のディレクトリにコピーします。  
    <Secure Agent インストールディレクトリ>/apps/Streaming\_Ingestion\_Agent/ext

**注:** JDBC V2 ターゲットは、単純な JSON または単純な JSON 形式の配列のデータのみを受け入れます。

## Kafka ターゲット

ストリーミング統合タスクを使用して、Kafka ターゲットにデータを書き込みます。Kafka ターゲットを作成するには、Kafka 接続タイプを使用します。

Kafka は Publish-Subscribe メッセージシステムです。オープンソースの分散ストリーミングプラットフォームです。このプラットフォームにより、データを生成するシステムは、Kafka トピックでデータをリアルタイムで永続化できます。任意のトピックを、リアルタイムでデータを必要とする任意の数のシステムで読み取ることができます。Kafka は、さまざまなダウンストリームコンシューマアプリケーションが使用できるストリーミングデータの間ステージング領域として使用できます。

Kafka は、1 つ以上の Kafka ブローカで構成されるクラスターとして実行されます。Kafka ブローカは、データをメッセージの形式でストリーミングし、メッセージをトピックにパブリッシュし、メッセージをトピックからサブスクライブしてから、メッセージを Kafka ターゲットに書き込みます。

Kafka ターゲットを作成するときは、Kafka プロデューサを作成して Kafka メッセージを書き込みます。ストリーミング Kafka メッセージを書き込むストリーミング取り込みジョブで各 Kafka ターゲットを使用できます。Kafka ターゲットを設定するときは、メッセージをパブリッシュするトピックと、Kafka ブローカが実行される IP アドレスとポートを指定します。Kafka トピックがターゲットに存在しない場合、トピックを手動で作成する代わりに、存在しないトピックがパブリッシュされたときにトピックを自動作成するようにブローカを設定することもできます。

同じ Kafka 接続を使用して、Apache Kafka 用の Amazon Managed Streaming (Amazon MSK) または Confluent Kafka 接続を作成できます。これで、ストリーミング統合タスクで Amazon MSK または Confluent Kafka ターゲットを使用して、Apache Kafka または Confluent Kafka ターゲットにメッセージを書き込むことができます。

## Microsoft Azure Data Lake Storage Gen2 ターゲット

ストリーミング統合タスクを使用して、Microsoft Azure Data Lake Storage Gen2 ターゲットにデータを書き込みます。Microsoft Azure Data Lake Storage Gen2 ターゲットを作成するには、Microsoft Azure Data Lake Storage Gen2 接続タイプを使用します。

Microsoft Azure Data Lake Storage Gen2 は、ビッグデータ分析のための次世代のデータレイクソリューションです。データをディレクトリおよびサブディレクトリの形で保存し、データのアクセスと操作を効率化することができます。任意のサイズ、構造、形式のデータを保存できます。大量のデータを処理して、より迅速なビジネス成果を達成できます。データサイエンティストやデータアナリストは、データレイクのデータを使用して特定のパターンがないか調べたうえで、分析したデータをデータウェアハウスに移動できます。Microsoft Azure Blob ストレージ上で使用可能なビッグデータ分析を実行できます。

ストリーミング統合タスクは、指定された条件に基づいて Microsoft Azure Data Lake Storage Gen2 にデータを書き込みます。

Microsoft Azure Data Lake Storage Gen2 の詳細については、Microsoft Azure Data Lake Storage Gen 2 のドキュメントを参照してください。

## Microsoft Azure Event Hubs ターゲット

ストリーミング統合タスクを使用して、Azure Event Hubs ターゲットにデータを書き込みます。Azure Event Hubs ターゲットを作成するには、Azure Event Hubs 接続タイプを使用します。

Azure Event Hubs は、イベントを受信して処理する、拡張性の高いデータストリーミングプラットフォームおよびイベント取り込みサービスです。Azure Event Hubs は、短い待機時間と高い信頼性で大量のイベントを取り込んで処理できます。これは、接続されたさまざまなデバイスやシステムからのメッセージストリームを処理できるマネージドサービスです。

イベントハブにデータを送信するエンティティはすべて、イベントパブリッシャです。イベントパブリッシャは、HTTPS または Kafka 1.0 以降を使用してイベントをパブリッシュできます。イベントパブリッシャは、共有アクセス署名（SAS）トークンを使用して、イベントハブに対して自分自身を識別し、一意の ID を持つか、共通の SAS トークンを使用できます。

Event Hubs の詳細については、Microsoft Azure Event Hubs のドキュメントを参照してください。

## 一括取り込みストリーミングのトランスフォーメーション

トランスフォーメーションはストリーミング統合ジョブの一部です。トランスフォーメーションは、ストリーミングデータを取り込むときに実行する操作を表します。

各トランスフォーメーションは特定の関数を実行します。たとえば、フィルタトランスフォーメーションは、指定された条件に基づいて、取り込んだデータからデータをフィルタします。

ストリーミング統合タスクを作成する場合、トランスフォーメーションの追加はオプションです。各トランスフォーメーションタイプには、設定可能な固有のオプションのセットがあります。

ストリーミング統合タスクでは次のトランスフォーメーションを使用できます。

- Combiner
- フィルタ
- 形式変換
- Java
- Jolt
- Python
- Splitter

複数のトランスフォーメーションをストリーミング統合タスクに追加できます。このような場合、ソースデータは指定された順序で各トランスフォーメーションを通過するため、トランスフォーメーションの順序は重要です。1 つのトランスフォーメーションの出力は、タスクフローの次のトランスフォーメーションへの入力になります。

ストリーミング統合タスクでは、Combiner トランスフォーメーションと形式変換トランスフォーメーションをそれぞれ 1 つのみ追加できます。形式変換トランスフォーメーションは、タスクフローの最後のトランスフ

オーメーションである必要があります。タスクに Combiner トランスフォーマーと形式変換トランスフォーマーの両方が含まれている場合、形式変換トランスフォーマーは、タスクフローの最後のトランスフォーマーである必要があります、その前が Combiner トランスフォーマーである必要があります。

## データ形式

各トランスフォーマータイプは、特定の形式の受信ストリーミングデータを処理します。

ストリーミング統合トランスフォーマーでは、ストリーミングデータを次の形式で処理できます。

- バイナリ。任意のタイプの構造化データと非構造化データ。
- JSON。データを構造化するための読み取り可能な形式。
- XML。構造化テキストデータ。

タスクにトランスフォーマーが含まれていない場合、受信データは元の形式で使用されます。

## Combiner トランスフォーマー

Combiner トランスフォーマーは、指定された条件に基づいて、ストリーミングソースからの複数のイベントを単一のイベントに結合します。

Combiner トランスフォーマーは、バイナリデータと JSON データを処理します。JSON メッセージ形式の場合、Combiner トランスフォーマーは、受信データをデータの配列に結合し、JSON 配列オブジェクトを出力として返します。バイナリメッセージ形式の場合、指定された条件に基づいて受信データを結合します。

ストリーミング統合タスクでは、Combiner トランスフォーマーを 1 つだけ追加できます。タスクに Combiner トランスフォーマーと形式変換トランスフォーマーの両方が含まれている場合、形式変換トランスフォーマーは、タスクフローの最後のトランスフォーマーである必要があります、その前が Combiner トランスフォーマーである必要があります。タスクに形式変換トランスフォーマーが含まれていない場合、Combiner トランスフォーマーがタスクフローの最後のトランスフォーマーである必要があります。

Combiner トランスフォーマーには、次のいずれかの条件を使用できます。

- イベントの最小数
- 最大集計サイズ
- 制限時間

たとえば、次のようなイベントがあるとします。

- Record created
- Record published

区切り文字としてカンマ (,) を使用すると、Combiner トランスフォーマーは次の結合イベントを返します。

Record created,Record Published

**注:** Combiner トランスフォーマーを使用してバイナリデータを処理する場合、正規表現を区切り文字として使用することはできません。

## フィルタトランスフォーメーション

フィルタトランスフォーメーションは、指定されたフィルタ条件に基づいて受信ストリーミングイベントのデータをフィルタします。

データは 1 つ以上の条件に基づいてフィルタできます。たとえば、ある日付範囲内のデータを使用する場合は、指定された日付に基づいてデータを削除する条件を作成できます。

## 形式変換トランスフォーメーション

形式変換トランスフォーメーションは、XML および JSON 受信メッセージのデータ形式を、指定された条件に基づいて Parquet 形式に変換してから、データレイクにストリーミングします。

ストリーミング取り込みタスクに追加できる形式変換トランスフォーメーションは 1 つだけです。形式変換トランスフォーメーションは、タスクフローの最後のトランスフォーメーションである必要があります。

受信データの日付、時刻、およびタイムスタンプ形式を指定できます。形式が指定されていない場合は、エポック（1970 年 1 月 1 日午前 0 時 GMT）からのミリ秒単位と見なされます。

## Java トランスフォーメーション

Java トランスフォーメーションは、Java コードを実行して受信メッセージを処理し、処理したデータを別のトランスフォーメーションまたはターゲットに送信します。

Java トランスフォーメーションを使用して、単純なトランスフォーメーション機能またはやや複雑なトランスフォーメーション機能を定義できます。Java トランスフォーメーションは、バイナリ、JSON、および XML データを処理できます。

Java コードをスニペットとしてインポートできるため、Java プログラム全体を作成する必要はありません。サンプル Java コードをインポートし、Java トランスフォーメーションを作成してコンパイルできます。

非標準の Java パッケージをインポートする場合、各 JAR ファイルのクラスパス、または Java パッケージに関連付けられたクラスファイルディレクトリのクラスパスを設定する必要があります。組み込みの Java パッケージの場合、クラスパスを設定する必要はありません。たとえば `java.io` はビルトイン Java パッケージです。`java.io` をインポートした場合、`java.io` 用にクラスパスを設定する必要はありません。

Java トランスフォーメーションは `inputData` 変数と `outputData` 変数を使用して、受信データと送信データを格納します。

次の表は、データ型間のマッピングを示しています。

| 受信データ | Java データ型 |
|-------|-----------|
| JSON  | String    |
| XML   | String    |
| バイナリ  | Byte[]    |

### JSON 用のサンプル Java スクリプト

ClassPath: /<Secue Agent Location>/apps/Streaming\_Ingestion\_Agent/ext/json-simple-1.1.1.jar

```
#####/* Import Code */#####
import org.json.simple.JSONObject;
import org.json.simple.parser.JSONParser;
```

```
import org.json.simple.parser.ParseException;

#####/* Main code */#####
JSONParser parser = new JSONParser();
try {
    JSONObject object = (JSONObject) parser.parse(inputData);
    object.put("age", 23);
    outputData = object.toJSONString();
} catch (ParseException e) {
    throw new RuntimeException();
}

#####/* inputData and outputData */#####
inputData: {"name":"test"}
outputData: {"name":"test","age":23}
```

## バイナリ用のサンプル Java スクリプト

ClassPath:/<Secure Agent Location>/apps/Streaming\_Ingestion\_Agent/ext/binary-2.3.0.jar

```
#####/* Import Code */#####
import java.io.*;

#####/* Main code */#####
String temp = new String(inputData);
outputData = (temp+"-text").getBytes();

#####/* inputData and outputData */#####
inputData: Sample
outputData: Sample-text
```

## XML 用のサンプル Java スクリプト

/<Secure Agent Location>/apps/Streaming\_Ingestion\_Agent/ext/dom-0.9.4.jar

```
#####/* Import Code */#####
import java.io.*;
import javax.xml.parsers.*;
import javax.xml.transform.*;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
import org.w3c.dom.*;

#####/* Main code */#####
try {
    DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
    DocumentBuilder builder = null;
    builder = factory.newDocumentBuilder();
    StringBuilder xmlStringBuilder = new StringBuilder();
    xmlStringBuilder.append(inputData);
    ByteArrayInputStream input = new ByteArrayInputStream(xmlStringBuilder.toString().getBytes("UTF-8"));
    Document doc = builder.parse(input);
    Node enterprise = doc.getFirstChild();
    Node employee = doc.getElementsByTagName("employee").item(0);
    Element job = doc.createElement("job");
    job.appendChild(doc.createTextNode("Commercial"));
    employee.appendChild(job);
    DOMSource domSource = new DOMSource(doc);
    StringWriter writer = new StringWriter();
    StreamResult result = new StreamResult(writer);
    TransformerFactory tf = TransformerFactory.newInstance();
    Transformer transformer = tf.newTransformer();
    transformer.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION, "yes");
    transformer.transform(domSource, result);
    outputData = writer.toString();
}
```

```
} catch (Exception e) {}
```

```
#####/* inputData and outputData */#####
inputData: < entreprise > < employee id = "1" > < name > Alex < /name><age>25</age > < address > San Francisco
< /address></employee > < /entreprise>
outputData: < entreprise > < employee id = "1" > < name > Alex < /name><age>25</age > < address > San
Francisco < /address><job>Commercial</job > < /employee></entreprise >
```

## Jolt トランスフォーメーション

Jolt トランスフォーメーションを使用して、複雑な JSON データを単純な JSON データに変換します。

Jolt トランスフォーメーションは、JSON から JSON へのデータ変換を実行する一連の操作を提供します。複数の Jolt 仕様を順番で（チェーン）単純な仕様の配列に追加して、全体的な JSON から JSON へのトランスフォーメーションを形成できます。仕様に基づいて、Jolt トランスフォーメーションは複雑な入力構造を単純な JSON 構造に変換します。

## Python トランスフォーメーション

Python トランスフォーメーションは、Python スクリプトを実行して、ストリーミングソースからの受信データを変換します。

Python トランスフォーメーションは、バイナリデータ、JSON と XML データを処理します。Python トランスフォーメーションは、受信データと送信データを保存するために *inputData* と *outputData* の 2 つの変数を使用します。

XML または JSON メッセージ形式の受信データは、*inputData* 変数に文字列として格納され、XML または JSON メッセージ形式は *outputData* に文字列として格納されます。バイナリメッセージ形式の受信データは、*inputData* 変数に `numpy.ndarray` として格納されます。バイナリメッセージ形式の送信データは、*outputData* 変数に `bytearray` として格納されます。

*inputData* 変数のバイナリデータは ASCII 文字としてエンコードされます。それに応じてデータをデコードする必要があります。また、Python トランスフォーメーションスクリプトが *inputData* 変数に存在する非 ASCII 文字を処理することを確認してください。

Python トランスフォーメーションを使用する前に、Python をインストールするためのディレクトリ（Python ホーム）を作成します。Python を Python ホームディレクトリにインストールした後、サードパーティのライブラリである NumPy と Jep（Java Embedded Python）を Python ホームと同じディレクトリにインストールしてください。

1 つの Secure Agent では、2 つの異なるバージョンの Python を使用して同じ Python トランスフォーメーションを実行することはできません。

### JSON 用のサンプル Python スクリプト

```
import json
temp=json.loads(inputData)
temp["name"]="Mr "+temp["name"]
outputData=json.dumps(temp)
#####
inputData: { "name":"John", "age":30, "city":"New York"}
outputData: { "name":"Mr John", "age":30, "city":"New York"}
```

### バイナリ用のサンプル Python スクリプト

```
temp = ''.join(str(chr(c)) for c in inputData)
temp += " - this is edited again text"
outputData = bytearray(temp, 'utf-8')
#####
inputData: Sample text
outputData: Sample text - this is edited again text
```

## XML 用のサンプル Python スクリプト

```
import xml.etree.ElementTree as ET
myroot = ET.fromstring(inputData)
for x in myroot:
    if x.tag=="body":
        x.tag="Msg"
xmlstr = ET.tostring(myroot)
outputData=xmlstr.decode('utf-8')
#####
inputData: <note><to>You</to><from>Me</from><heading>Message</heading><body>Happy Coding</body></note>
outputData: <note><to>You</to><from>Me</from><heading>Message</heading><Msg>Happy Coding</Msg></note>
```

## Splitter トランスフォーメーション

Splitter トランスフォーメーションは、複数行のメッセージまたはメッセージ配列を、ターゲットに取り込む前に指定した条件に基づいて個別のメッセージに分割します。

Splitter トランスフォーメーションは、指定した条件に基づいてバイナリ、JSON、および XML メッセージを分割し、分割されたメッセージを新しいファイルに渡してターゲットに取り込みます。Splitter トランスフォーメーションを使用して、複雑なメッセージを論理コンポーネントに分割します。たとえば、メッセージにコンマで区切られたエラーコードとエラーメッセージが含まれている場合、コンマを使用してコードとメッセージを異なるファイルに分けることができます。

### バイナリメッセージ

バイナリメッセージ形式では、Splitter トランスフォーメーションは、行の境界またはバイトシーケンスに基づいてメッセージを分割します。最大行数によって行の境界が決まります。各出力分割ファイルには、設定された行数またはバイト数しか含まれていません。行の境界のデフォルト値は 1 です。バイトシーケンスのデフォルトは「,」です。

### JSON メッセージ

JSON メッセージ形式では、Splitter トランスフォーメーションは、JSONPath 式で指定された配列要素に基づいて JSON ファイルを個別のファイルに分割します。生成された各ファイルは、指定された配列の要素で構成されます。生成されたファイルは、タスクのダウンストリームターゲットまたはトランスフォーメーションに転送されます。指定された JSONPath が見つからないか、配列要素に評価されない場合、元のファイルは *failure* にルーティングされ、ファイルは生成されません。デフォルトの JSONPath 式は '\$' です。

### XML メッセージ

XML メッセージ形式では、Splitter トランスフォーメーションは、入力深度のレベルに基づいて XML メッセージを多くのファイルに分割します。これらの各ファイルには、元のファイルの子または子孫が含まれています。

## ストリーミング統合タスクの設定

ストリーミング取り込みタスクを作成する場合は、ソースとターゲットを定義します。オプションで、データを変換するトランスフォーメーションを定義できます。ストリーミング統合タスクを設定するには、タスクウィザードを使用します。

ストリーミング統合タスクを設定するには、次の手順を実行します。

1. タスクの基本情報を定義する。
2. ソースを設定する。



3. ターゲットを設定する。
4. 必要に応じて、1つまたは複数のトランスフォーメーションを追加します。
5. 必要に応じて、ランタイムオプションを設定する。

タスクウィザードを使用するときは、ターゲットを設定した後にいつでも【保存】をクリックして作業を保存できます。ウィザードが完了したら、【保存】をクリックしてタスクを保存します。

開始する前に、[「始める前に」](#) (ページ 24)に記載されている条件を満たしていることを確認します。

## 始める前に

ストリーミング統合タスクを作成する前に、次の条件を満たしていることを確認します。

- 組織に一括取り込みストリーミングおよびストリーミング統合パッケージのライセンスがあることを確認してください。
- 一括取り込みストリーミングが Secure Agent で実行されている。
- ソース接続とターゲット接続が存在する。

## 基本的なタスク情報の定義

ストリーミング取り込みタスクウィザードの【定義】タブを使用して、タスクの詳細を定義します。

1. 【新規】 > 【ストリーミング取り込みタスク】をクリックします。  
タスクウィザードの【定義】ページが表示されます。
2. 以下のプロパティを設定します。

| プロパティ   | 説明  |
|---------|---|
| 名前      | ストリーミング統合タスクの名前。<br>ストリーミング統合タスクの名前は組織内で一意である必要があります。タスク名には、英数字、スペース、およびアンダースコアを含めることができます。名前の先頭には英字またはアンダースコアを指定する必要があります。<br>タスク名は大文字と小文字を区別しません。 |
| 場所      | タスクを保存するプロジェクトフォルダ。   |
| ランタイム環境 | Secure Agent を含むランタイム環境。Secure Agent がタスクを実行します。  |
| 説明      | オプション。タスクの説明。最大長は 4,000 文字です。   |

3. 【次へ】をクリックします。



## ソースの設定

ソースを設定するには、ストリーミングデータを取り込むソース接続を選択してから、ソースプロパティを設定します。ソースを設定する前に、ソースへの接続が Administrator サービスで作成されていることを確認してください。

1. **【ソース】** ページで接続を選択します。

ストリーミング統合タスクは次のソースをサポートします。

- Amazon Kinesis Streams
- AMQP
- Apache Kafka
- フラットファイル
- Google PubSub
- JMS
- MQTT
- OPC UA
- REST V2

接続タイプは、選択した接続名に基づいて自動的に入力されます。

2. 選択したソースに基づいて、必要な詳細を入力します。

タスクウィザードの **【ソース】** タブに表示されるオプションは、選択するソースの種類によって異なります。

3. **詳細プロパティ**で、必要な情報を入力します。

4. **【次へ】** をクリックします。

**【ターゲット】** タブが表示されます。

### Amazon Kinesis Streams ソースのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに **【ソース】** タブで設定できる Amazon Kinesis Streams ソースのプロパティについて説明します。

| プロパティ | 説明  |
|-------|---|
| 接続    | Amazon KinesisStream ソース接続の名前。                          |
| 接続タイプ | Amazon Kinesis 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。 |
| ストリーム | データの読み取り元の Kinesis Stream の名前。                          |

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで Amazon Kinesis Streams ソースに設定できる詳細プロパティについて説明します。

| プロパティ                       | 説明  |
|-----------------------------|---|
| GUID を<br>DynamoDB テーブル名に追加 | Amazon DynamoDB テーブル名にサフィックスとして GUID を追加するかどうかを指定します。無効になっている場合は、Amazon DynamoDB テーブル名を入力する必要があります。デフォルトでは有効になっています。  |
| Amazon<br>DynamoDB          | Kinesis ソースデータのチェックポイントの詳細を保存する Amazon DynamoDB テーブル名。<br>Amazon DynamoDB テーブル名は自動的に生成されます。ただし、選択した名前を入力すると、ストリーミング取り込みタスクは、自動生成された名前の前に指定された名前をプレフィックスとして付けます。 |

Kinesis Streams の詳細については、Amazon Web Services のドキュメントを参照してください。

## AMQP ソースのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで設定できる Advanced Message Queuing Protocol (AMQP) ソースのプロパティについて説明します。

| プロパティ     | 説明  |
|-----------|---|
| 接続        | AMQP ソース接続の名前。  |
| 接続タイプ     | AMQP 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。                           |
| キュー       | ストリーミング取り込みタスクがメッセージを読み取る既存の AMQP キューの名前。このキューは、AMQP 管理者によって事前定義されています。 |
| 自動確認メッセージ | True または False を選択できます。True を選択すると、AMQP ブローカは受信したメッセージを自動的に確認します。       |
| バッチサイズ    | 1 回のセッションでプルする必要のあるメッセージの最大数。<br>デフォルトは 10 メッセージです。                     |

## Azure Event Hubs Kafka ソースのプロパティ

Azure Event Hubs 名前空間を使用して Kafka 接続を作成できます。

標準層または専用層の Event Hubs 名前空間を作成すると、名前空間の Kafka エンドポイントがデフォルトで有効になります。次に、Azure Event Hubs 対応の Kafka 接続を、ストリーミング取り込みタスクの作成中にソース接続として使用できます。トピック名として Event Hubs 名を入力します。

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで設定できる Kafka ソースのプロパティについて説明します。

| プロパティ | 説明   |
|-------|--|
| 接続    | Kafka ソース接続の名前。  |
| 接続タイプ | Kafka 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。   |
| トピック  | イベントの読み取り元の Event Hubs の名前。<br>トピック名を手動で入力するか、Kafka 対応の Event Hubs 接続の作成済みメタデータを取得できます。<br>1. <b>【選択】</b> をクリックします。<br>使用可能なすべてのトピックを示す <b>【ソースオブジェクトの選択】</b> ダイアログボックスが表示されます。<br>2. 必要なトピックを選択して <b>【OK】</b> をクリックします。 |

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで Kafka ソースに設定できる詳細プロパティについて説明します。

| プロパティ         | 説明  |
|---------------|---|
| コンシューマ設定プロパティ | コンシューマが Kafka に接続するための設定プロパティのカンマ区切りリスト。キーと値のペアとして値を指定します。例: key1=value1, key2=value2<br>Kafka コンシューマの group.id プロパティは自動生成されます。このプロパティは上書きできます。 |

**注:** Kafka 用の Event Hubs は、標準層と専用層でのみ使用できます。基本層は、Event Hubs で Kafka をサポートしていません。

## フラットファイルソースのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで設定できるフラットファイルソースのプロパティについて説明します。

| プロパティ  | 説明  |
|--------|---|
| 接続     | フラットファイルソース接続の名前。   |
| 接続タイプ  | フラットファイル接続タイプ。<br>接続タイプは、選択した接続に基づいて自動的に表示されます。   |
| 初期開始位置 | テールするファイル内でデータを読み取る際の開始位置。読み取りを開始する次のいずれかの位置を選択できます。 <ul style="list-style-type: none"><li>- ファイルの先頭。テールするファイルの先頭から読み取ります。すでにロールオーバーされているデータは取り込まないでください。</li><li>- 現在の時刻。テールするファイルの最後に更新された部分から読み取ります。すでにロールオーバーされているデータや、テールするファイル内のすでに書き込まれているデータを取り込まないでください。</li></ul> |

| プロパティ    | 説明   |
|----------|--|
| テーリングモード | <p>ロギングパターンに基づいて、1 つまたは複数のファイルをテールします。</p> <p>次のいずれかのモードを選択します。</p> <ul style="list-style-type: none"> <li>- 単一ファイル。1 つのファイルのみをテールします。</li> <li>- 複数ファイル。ベースディレクトリに示されているすべてのファイルをテールします。このモードでは、正規表現を入力して、テールするファイルを示すことができます。</li> </ul> |
| ファイル     | <p>読み取るファイルの名前の絶対パス。</p> <p>テールするファイルの名前、またはテールするファイルを見つけるための正規表現。複数ファイルモードのベースディレクトリを入力します。</p>   |

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブでフラットファイルソースに設定できる詳細プロパティについて説明します。

| 接続プロパティ        | 説明  |
|----------------|---|
| ローリングファイル名パターン | <p>ロールオーバーするファイルの名前パターン。</p> <p>テールするファイルがロールオーバーする場合、ファイル名パターンは、ロールオーバーされたファイルを特定するために使用されます。基盤となるストリーミング取り込み Secure Agent はこのファイルパターンを認識します。Secure Agent が再起動し、ファイルがロールオーバーされると、中断したところから再開します。</p> <p>アスタリスク (*) と疑問符 (?) をワイルドカード文字として使用し、ファイルが同じディレクトリにロールオーバーされることを示すことができます。例: <code>\${filename}.log.*</code>。ここで、アスタリスク (*) は、ファイル名に追加される連続するバージョン番号を表します。</p> |

## Google PubSub ソースのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで設定できる Google PubSub ソースのプロパティについて説明します。

| プロパティ     | 説明   |
|-----------|--|
| 接続        | Google PubSub ソース接続の名前。  |
| 接続タイプ     | <p>Google PubSub 接続タイプ。</p> <p>接続タイプは、選択した接続名に基づいて自動的に入力されます。</p>  |
| サブスクリプション | <p>メッセージのプル元となる Google PubSub サービスのサブスクリプションの名前。</p> <p>Google PubSub 接続は、サブスクリプションのプル配信タイプのみをサポートします。</p> |
| バッチサイズ    | <p>Cloud サービスがバッチでバンドルするメッセージの最大数。</p> <p>デフォルトは 1 です。</p>   |

## JMS ソースのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで設定できる JMS ソースのプロパティについて説明します。

| プロパティ       | 説明  |
|-------------|---|
| 接続          | JMS ソース接続の名前。   |
| 接続タイプ       | JMS 接続タイプ。接続タイプは、選択した接続名に基づいて自動的に入力されます。  |
| 宛先タイプ       | <p>ソースサービスが JMS メッセージを送信する接続先のタイプ。次のいずれかの接続先タイプを選択できます。</p> <ul style="list-style-type: none"> <li>- キュー。JMS プロバイダは、キューに登録されている単一のコンシューマにメッセージを配信します。</li> <li>- [トピック]。JMS プロバイダは、トピックにサブスクライブしているすべてのアクティブなコンシューマにメッセージを配信します。この接続先タイプを使用すると、複数のコンシューマがメッセージを読み取ることができます。</li> </ul> <p>デフォルトは【キュー】です。</p>  |
| 共有サブスクリプション | <p>複数のコンシューマが単一のサブスクリプションにアクセスできるようにします。トピックの接続先タイプに適用されます。</p> <p>デフォルトは false です。</p>   |
| 継続サブスクリプション | <p>非アクティブなサブスクライバがメッセージを保持し、その保持したメッセージを、サブスクライバが再接続したときに配信できるようにします。トピックの接続先タイプに適用されます。</p> <p>デフォルトは false です。</p>  |
| サブスクリプション名  | <p>サブスクリプションの名前。トピックサブスクリプションが共有可能、継続、またはその両方である場合に、トピックの接続先タイプに適用されます。値が指定されていない場合、取り込みサービスは一意的なサブスクリプション名を生成します。</p>  |
| JMS 接続先     | <p>JMS プロバイダがメッセージを配信するキューまたはトピックの名前。</p> <p><b>注:</b> JMS Weblogic サーバーとの JMS 接続が作成された場合、キューまたはトピックの JMS 接続先は、ピリオドで始まってその後にスラッシュが続きます(./)。以下に例を示します。。</p> <p>。/&lt;JMS Server module name&gt;! &lt;Queue or topic name&gt;</p> <p>一括取り込みストリーミングを Oracle Weblogic JMS サーバーに接続する方法の詳細については、Informatica ナレッジベース Informatica Knowledge Base の記事「<a href="#">000186952</a>」を参照してください。</p> |

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで JMS ソースに設定できる詳細プロパティについて説明します。

| プロパティ     | 説明   |
|-----------|--|
| クライアント ID | <p>オプション。JMS 接続を識別する一意の識別子。</p> <p>非共有の継続サブスクリプションに値が指定されていない場合、ストリーミング取り込みタスクは一意的なクライアント ID を生成します。</p> |

## Kafka ソースのプロパティ

Kafka をストリーミング取り込みタスクのソースとして定義している場合、**【ソース】** タブで Kafka ソースの必須プロパティを設定する必要があります。必要に応じて、コンシューマ設定のプロパティのカンマ区切りリストを指定します。

次の表に、Kafka ソースの必須プロパティを示します。

| プロパティ | 説明  |
|-------|---|
| 接続    | Kafka ソース接続の名前。   |
| 接続タイプ | Kafka 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。  |
| トピック  | イベントの読み取り元となる Kafka ソーストピック名、または Java でサポートされている Kafka ソーストピック名パターンの正規表現。<br>トピック名を手動で入力するか、Kafka 接続のメタデータを取得できます。Kafka 接続のメタデータを選択するには、次のアクションを実行します。<br>1. <b>【選択】</b> をクリックします。<br><b>【ソースオブジェクトの選択】</b> ダイアログボックスが表示され、Kafka ブローカで使用可能なすべてのトピックまたはトピックパターンが表示されます。<br>2. トピックを選択して <b>【OK】</b> をクリックします。<br><b>注:</b> 新しい Kafka ソーストピックを <b>【稼働中】</b> 状態のストリーミング取り込みジョブに追加するとき、新しいトピックからのデータ損失を回避するために、ジョブをすぐに再デプロイするようにしてください。 |

## コンシューマ設定プロパティ

**【ソース】** タブの **【詳細プロパティ】** セクションにある **【コンシューマ設定プロパティ】** で、オプションのコンシューマ設定プロパティのカンマ区切りリストを指定できます。キーと値のペアとして値を指定します。

以下の表に、Kafka ソースで設定できるコンシューマ設定プロパティを示します。

| プロパティ             | 説明  |
|-------------------|---|
| group.id          | Kafka コンシューマが属するコンシューマグループの名前を指定します。Kafka コンシューマのコンストラクト時に group.id が存在しない場合、タスクはコンシューマグループを自動的に作成します。このプロパティは自動生成されます。このプロパティは上書きできます。デフォルトは、key1=value1, key2=value2 です。  |
| auto.offset.reset | コミットされた位置がない場合、またはオフセットが範囲外の場合のコンシューマの動作を指定します。<br>次のタイプの自動オフセットリセットを使用できます。<br>- 最も古い。オフセット位置をトピックの先頭にリセットします。<br>- 最新。オフセット位置をトピックの最新の位置にリセットします。<br>- なし。<br>Kafka トピックからデータを読み取るか、トピックパターンを使用するとき、メッセージリカバリ中に前回のチェックポイントのオフセットが削除された場合、次に使用可能なオフセットからメッセージをリカバリするには、次のプロパティを指定します。<br>auto.offset.reset=earliest<br>それ以外の場合は、ストリーミング取り込みタスクによって、使用可能な最新のオフセットからデータが読み取られます。 |

| プロパティ              | 説明   |
|--------------------|--|
| message-demarcator | <p>Kafka ソースはメッセージをバッチで受信します。特定のトピックとパーティションのすべての Kafka メッセージを 1 つのバッチに含めることができます。このプロパティを使用すると、複数の Kafka メッセージの境界として使用する文字列を指定できます。値を指定しない場合、各 Kafka メッセージは単一のイベントとしてトリガされず。</p> <p>次の区切り文字を境界として使用できます。</p> <ul style="list-style-type: none"> <li>- 改行。新しい改行で新しいコンテンツを区切ります。新しい改行をメッセージの境界として使用するには、次の値を入力します。<br/>message-demarcator=\${literal('&amp;#10;'):unescapeXml()}</li> <li>- カンマ。カンマで新しいコンテンツを区切ります。カンマをメッセージの境界として使用するには、次の値を入力します。<br/>message-demarcator=\${literal('&amp;#44;'):unescapeXml()}</li> <li>- セミコロン。セミコロンで新しいコンテンツを区切ります。セミコロンをメッセージの境界として使用するには、次の値を入力します。<br/>message-demarcator=\${literal('&amp;#59;'):unescapeXml()}</li> <li>- タブ。タブで新しいコンテンツを区切ります。タブをメッセージの境界として使用するには、次の値を入力します。<br/>message-demarcator=\${literal('&amp;#09;'):unescapeXml()}</li> </ul> |
| max.poll.records   | <p>ポーリングの 1 回の呼び出しで返されるレコードの最大数を指定します。</p> <p>例: max.poll.records=100000</p>   |

## MQTT ソースのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで設定できる MQTT ソースのプロパティについて説明します。

| プロパティ | 説明  |
|-------|---|
| 接続    | MQTT ソース接続の名前。                                |
| 接続タイプ | MQTT 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。 |
| トピック  | MQTT トピックの名前。                                 |

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで MQTT ソースに設定できる詳細プロパティについて説明します。

| 接続プロパティ   | 説明   |
|-----------|--|
| クライアント ID | オプション。MQTT ソースと MQTT ブローカ間の接続を識別する一意の識別子。クライアント ID は、MQTT ソースがメッセージの処理中にメッセージを格納するために使用するファイルベースの永続ストアです。<br>クライアント ID を指定しない場合、ストリーミング取り込みタスクは、MQTT 接続で提供されたクライアント ID を使用します。ただし、MQTT 接続でもクライアント ID を指定していない場合は、ストリーミング取り込みタスクは一意のクライアント ID を生成します。 |
| 最大キューサイズ  | オプション。プロセッサが同時にメモリに保存できるメッセージの最大数。<br>デフォルト値は 1024 バイトです。  |

## OPC UA ソースのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで設定できる OPC UA ソースのプロパティについて説明します。

| プロパティ       | 説明   |
|-------------|--|
| 接続          | OPC UA ソース接続の名前。   |
| 接続タイプ       | OPC UA 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。  |
| タグリストの指定形式  | タグのリストを指定する形式。<br>次のいずれかの形式を選択します。<br><ul style="list-style-type: none"> <li>- <b>タグのリスト</b>。JSON 配列として指定された、OPC クライアントによって読み取られるタグのリスト。</li> <li>- <b>タグファイルのパス</b>。JSON 配列として指定された、OPC クライアントによって読み取られるタグのリストを含むファイル。</li> </ul> |
| タグまたはファイルパス | JSON 配列として指定される、タグのリスト、または読み取るタグのリストを含むファイルへのパス。<br>タグまたはファイルパスのリストは 2048 文字を超えることはできません。  |
| 最小パブリッシュ間隔  | サブスクリプション通知メッセージの最小パブリッシュ間隔（ミリ秒単位）。<br>データの急激な変化を検出するには、このプロパティを低い値に設定します。<br>デフォルトは 1,000 ミリ秒です。  |



## REST V2 ソースのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ソース】タブで設定できる REST V2 ソースのプロパティについて説明します。

| プロパティ              | 説明   |
|--------------------|--|
| 接続                 | REST V2 ソース接続の名前。  |
| 接続タイプ              | REST V2 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。   |
| REST エンドポイント       | 入力 Swagger ファイルで指定された REST エンドポイントのリスト。<br>これらのエンドポイントは、選択した REST 接続に基づいて表示されます。   |
| スキーム               | Swagger 定義で指定されたスキームのリスト。<br>選択したスキームを使用して URL を作成します。   |
| ポーリング間隔            | 2 つの連続する REST 呼び出しの間隔。<br>デフォルトは 10 秒です。   |
| 失敗した応答コードに対するアクション | 失敗した REST 呼び出しに必要なアクション。<br>以下のいずれかのオプションを選択することができます。 <ul style="list-style-type: none"><li>- アラートを上げる</li><li>- ダウンストリームにルーティングする。応答をダウンストリームプロセッサにルーティングします。</li><li>- 拒否ディレクトリへのルーティング: [ランタイムオプション] ページで設定された拒否ディレクトリに応答をルーティングします。</li></ul> |

選択した **REST エンドポイント** プロパティで定義された操作 ID に基づき、**パス**、**クエリ**や**ペイロード**などの動的プロパティが REST V2 ソースページの下部に表示されます。

- **ヘッダー**。REST 呼び出しにヘッダーを追加します。
- **パス**。Swagger 定義で指定されている複数のパスパラメータで構成されます。**パスキー**を編集することはできません。パスキーに対応する値のみを入力できます。
- **クエリ**。クエリパラメータで構成されます。クエリパラメータはパスパラメータに似ています。
- **ペイロード**。
  - **サンプルペイロード**。PUT、POST、または PATCH 要求の要求本文のスキーマを示す読み取り専用のテキストボックス。例: { "name" : "string", "salary" : "string", "age" : "string" }。
  - **本文**。PUT、POST、または PATCH 要求の場合に送信される要求本文。サンプルペイロードからサンプル要求本文をコピーして、必要に応じて値を置き換えることができます。

これらのプロパティはいずれも、Swagger 仕様ファイルで必須として定義できます。次に、ストリーミング取り込み REST V2 ソース設定中に同じプロパティが必須と見なされます。REST エンドポイントを Swagger 仕様ファイルで定義しない場合、対応するセクションはストリーミング取り込み REST V2 ページには表示されません。

**注:** ストリーミング取り込みタスクを設定するときは、指定する Swagger 仕様ファイルの絶対パスがランタイム環境で使用可能であることが必要です。

## ターゲットの設定

ターゲットを設定するには、ストリーミングデータを転送するターゲット接続を選択してから、ターゲットプロパティを設定します。ターゲットを設定する前に、ターゲットへの接続が Administrator サービスで作成されていることを確認してください。

1. **【ターゲット】** ページで接続を選択します。

ストリーミング統合タスクは次のターゲットをサポートします。

- Amazon Kinesis Data Firehose
- Amazon Kinesis Streams
- Amazon S3 V2
- Apache Kafka
- Databricks Delta
- フラットファイル
- Google BigQuery V2
- Google PubSub
- Google Cloud Storage V2
- JDBC V2
- Microsoft Azure Data Lake Storage Gen2
- Microsoft Azure Event Hubs

2. 選択したターゲットに基づいて、必要な詳細を入力します。

タスクウィザードの **【ターゲット】** タブに表示されるオプションは、選択するターゲットの種類によって異なります。

3. **詳細プロパティ**で、必要な情報を入力します。

4. 次のいずれかの手順に従います。

- トランスフォーメーションを追加するには、**【次へ】** をクリックします。  
**【トランスフォーメーション】** タブが表示されます。
- タスクを保存するには、**【保存】** をクリックします。  
これで、ストリーミング統合タスクをデプロイできます。ストリーミング統合タスクのデプロイの詳細については、[「ストリーミング統合タスクのデプロイ」 \(ページ 50\)](#)を参照してください。

## Amazon Kinesis Data Firehose ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Amazon Kinesis Data Firehose ターゲットのプロパティについて説明します。

| プロパティ    | 説明  |
|----------|---|
| 接続       | Amazon Kinesis Data Firehose ターゲット接続の名前。  |
| 接続タイプ    | Amazon Kinesis 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。   |
| ストリーム名/式 | Kinesis ストリーム名または Kinesis ストリーム名パターンの正規表現。<br>正規表現には\$expression\$形式を使用します。\$expression\$はデータを評価し、一致するデータをキャプチャグループ 1 に送信します。 |

Kinesis Data Firehose の詳細については、Amazon Web Services のドキュメントを参照してください。

## Amazon Kinesis Streams ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Amazon Kinesis Streams ターゲットのプロパティについて説明します。

| プロパティ    | 説明  |
|----------|---|
| 接続       | Amazon KinesisStream ターゲット接続の名前。  |
| 接続タイプ    | Amazon Kinesis 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。   |
| ストリーム名/式 | Kinesis ストリーム名または Kinesis ストリーム名パターンの正規表現。<br>正規表現には\$expression\$形式を使用します。\$expression\$はデータを評価し、一致するデータをキャプチャグループ 1 に送信します。 |

Kinesis Streams の詳細については、Amazon Web Services のドキュメントを参照してください。

## Amazon S3 ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Amazon S3 ターゲットのプロパティについて説明します。

| プロパティ     | 説明  |
|-----------|---|
| 接続        | Amazon S3 ターゲット接続の名前。   |
| 接続タイプ     | Amazon S3 V2 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。   |
| オブジェクト名/式 | Amazon S3 ファイル名または Amazon S3 ファイル名パターンの正規表現。<br>正規表現には\$expression\$形式を使用します。\$expression\$はデータを評価し、一致するデータをキャプチャグループ 1 に送信します。 |

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Amazon S3 ターゲット詳細プロパティについて説明します。

| プロパティ            | 説明   |
|------------------|--|
| パーティション化間隔       | オプション。ストリーミング取り込みタスクが Amazon S3 バケットにパーティションを作成する際に従う時間間隔。このオプションを使用するには、【オブジェクト名/式】フィールドで、オブジェクト名に\${Timestamp}式を追加する必要があります。<br>デフォルトは「なし」です。<br>詳細については、「 <a href="#">Amazon S3 ターゲット</a> 」(ページ 13)を参照してください。 |
| 最小アップロードパートサイズ   | オプション。複数の独立したパートセットとしてサイズの大きなファイルをアップロードする場合の最小アップロードパートサイズ (MB 単位)。このプロパティを使用して、ファイルのロードを Amazon S3 に合わせます。<br>デフォルト値は 5120 MB です。  |
| マルチパートアップロードしきい値 | オプション。複数のパートにオブジェクトを同時にアップロードするタイミングを決定するマルチパートダウンロードの最小しきい値。<br>デフォルト値は 5120 MB です。   |

## Azure Event Hubs ターゲットプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Azure Event Hubs ターゲットのプロパティについて説明します。

| プロパティ     | 説明  |
|-----------|---|
| 接続        | Azure Event Hubs ターゲット接続の名前。                              |
| 接続タイプ     | Azure Event Hub 接続のタイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。 |
| Event Hub | Azure Event Hubs の名前。                                     |

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Azure Event Hubs ターゲット詳細プロパティについて説明します。

| プロパティ              | 説明   |
|--------------------|--|
| 共有アクセスポリシー名        | オプション。Event Hub 名前空間共有アクセスポリシーの名前。<br>このポリシーは、この接続に関連付けられたすべてのデータオブジェクトに適用される必要があります。<br>Event Hub から読み取るには、リスン権限が必要です。Event Hub に書き込むには、ポリシーに送信権限が必要です。 |
| 共有アクセスポリシーのプライマリキー | オプション。Event Hub 名前空間共有アクセスポリシーのプライマリキー。  |

## Databricks Delta ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Databricks Delta ターゲットのプロパティについて説明します。

| プロパティ      | 説明  |
|------------|---|
| 接続         | Databricks Delta ターゲット接続の名前。  |
| 接続タイプ      | Databricks Delta 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。   |
| ステージングの場所  | ステージングファイルを保存するための相対ディレクトリパス。 <ul style="list-style-type: none"><li>- Databricks クラスタが AWS にデプロイされている場合は、Amazon S3 ステージングバケットの相対パスを使用します。</li><li>- Databricks クラスタが Azure にデプロイされている場合は、Azure Data Lake Store Gen2 ステージングファイルシステム名の相対パスを使用します。</li></ul> |
| ターゲットテーブル名 | 追加する Databricks Delta テーブルの名前。  |

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Databricks Delta ターゲットの詳細プロパティについて説明します。

| プロパティ        | 説明   |
|--------------|--|
| ターゲットデータベース名 | Administrator の Databricks Delta 接続で指定されたデータベース名を上書きします。 |

Databricks Delta ターゲットの場合、ソースメッセージは JSON 形式とする必要があります。

**注:** Databricks Delta ターゲットを使用したストリーミング取り込みジョブで、ソーススキーマを変更して追加のデータカラムを含める場合は、ジョブを再デプロイして変更データキャプチャを含めることをお勧めします。

Databricks Delta ターゲットを使用したストリーミング取り込みタスクでフィルタトランスフォーメーションを使用する場合は、取り込んだデータが有効な JSON データ形式に準拠していることを確認してください。JSONPath フィルタタイプを使用したフィルタトランスフォーメーションは、受信データを検証します。受信データが有効な JSON データ形式に準拠していない場合、ストリーミング取り込みタスクはデータを拒否します。次に、拒否されたデータは、設定された拒否ディレクトリに移動します。拒否ディレクトリがまだ設定されていない場合、拒否されたデータは失われます。

Informatica では、Databricks Delta ターゲットを含むストリーミング取り込みタスクで Combiner トランスフォーメーションを使用することをお勧めします。ターゲットに書き込む前に、Combiner トランスフォーメーションを追加します。ストリーミング取り込みタスクは、Databricks Delta ターゲットに書き込む前に、ステージングされたすべてのデータを結合します。パフォーマンスを最適化するには、100 件以上のレコードをバッチ処理します。

## フラットファイルターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できるフラットファイルターゲットのプロパティについて説明します。

| プロパティ   | 説明  |
|---|---|
| 接続  | フラットファイルターゲット接続の名前。   |
| 接続タイプ   | フラットファイル接続タイプ。<br>接続タイプは、選択した接続に基づいて表示されます。   |
| ステージングディレクトリの場所   | Secure Agent 上のステージングディレクトリへのパス。<br>フラットファイルターゲットにデータを書き込むときに、ファイルをステージングするステージングディレクトリを指定します。ディレクトリに十分なスペースがあり、ディレクトリへの書き込み権限があることを確認します。  |
| ロールオーバーサイズ*   | タスクがファイルをステージングディレクトリからターゲットに移動する際のファイルサイズ (KB)。<br>例えば、ロールオーバーサイズを 1 MB に設定し、ファイルに target.log という名前を付けます。ソースサービスが 5 MB をターゲットに送信する場合、ストリーミング取り込みタスクは最初に target.log.<タイムスタンプ>ファイルを作成します。target.log.<タイムスタンプ>のサイズが 1 MB に達すると、タスクはファイルをロールオーバーします。 |
| ロールオーバーイベント数*   | ファイルのロールオーバーのために蓄積するイベントまたはメッセージの数。<br>例えば、ロールオーバーイベント数を 1000 に設定した場合、ファイルのイベントが 1000 件蓄積されると、タスクはファイルをロールオーバーします。  |
| ロールオーバー時間*  | ターゲットファイルがロールオーバーする時間の長さ (ミリ秒単位)。この時間が経過すると、ターゲットファイルはロールオーバーします。<br>例えば、ロールオーバー時間を 1 時間に設定した場合、ファイルの時間が 1 時間に達すると、タスクはファイルをロールオーバーします。   |
| ファイル名   | タスクがターゲット上に作成するファイルの名前。   |
| *ターゲットファイルのロールオーバーを実行するには、少なくとも 1 つのロールオーバーオプションの値を指定します。 |   |

## Google BigQuery V2 ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Google BigQuery V2 ターゲットのプロパティについて説明します。

| プロパティ   | 説明  |
|---------|---|
| 接続      | Google BigQuery V2 ターゲット接続の名前。                                      |
| 接続タイプ   | Google BigQuery V2 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。         |
| データセット名 | Google BigQuery データセットの名前。データセットは Google Cloud Platform になければなりません。 |
| テーブル名   | JSON 形式でデータを挿入する Google BigQuery テーブルの名前。                           |

## Google Cloud Storage ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Google Cloud Storage ターゲットのプロパティについて説明します。

| プロパティ | 説明  |
|-------|---|
| 接続    | Google Cloud Storage ターゲット接続の名前。  |
| 接続タイプ | Google Cloud Storage の接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。            |
| 再試行回数 | ストリーミング取り込みタスクが、Google Cloud Storage ターゲットへの書き込みを再試行する回数。<br>デフォルトは 6 です。 |
| バケット  | Google Cloud Storage にアップロードするオブジェクトを保存、整理、アクセスするためのコンテナ。                 |
| キー    | Google Cloud Storage ターゲット接続の名前。  |

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Google Cloud Storage ターゲット詳細プロパティについて説明します。

| プロパティ      | 説明  |
|------------|---|
| プロキシホスト    | Secure Agent が使用する送信プロキシサーバーのホスト名。  |
| プロキシポート    | 送信プロキシサーバーのポート番号。   |
| コンテンツタイプ   | ファイルのコンテンツタイプ。<br>application/json、multipart、text、html などの任意の MIME タイプを指定できます。これらの値では大文字と小文字は区別されません。<br>デフォルトは text です。  |
| オブジェクト ACL | アップロードされたオブジェクトに関連付けられたアクセス制御。<br>次のいずれかの認証のタイプを選択します。 <ul style="list-style-type: none"> <li>- <b>認証された読み取り。</b> バケットまたはオブジェクトの所有者に FULL_CONTROL 権限を付与し、認証されたすべての Google アカウント所有者に READ 権限を付与します。</li> <li>- <b>バケット所有者フルコントロール。</b> バケットまたはオブジェクトの所有者にフルコントロールのアクセス許可を付与し、認証されたすべての Google アカウント所有者に読み取り権限を付与します。</li> <li>- <b>バケット所有者読み取り専用。</b> オブジェクトの所有者にフルコントロールのアクセス許可を付与し、バケット所有者に読み取り権限を付与します。このタイプはオブジェクトでのみ使用します。</li> <li>- <b>非公開。</b> バケットまたはオブジェクトの所有者に、バケットまたはオブジェクトに対する FULL_CONTROL 権限を付与します。</li> <li>- <b>プロジェクト非公開。</b> ロールに基づいてプロジェクトチームに権限を与えます。チームの一員である人は誰でも READ 権限を持ち、プロジェクト所有者とプロジェクト編集者は FULL_CONTROL 権限を持ちます。これは、新しく作成されたバケットのデフォルトの ACL です。</li> <li>- <b>公開読み取り専用。</b> バケット所有者に FULL_CONTROL 権限を付与し、匿名のすべてのユーザーに READ および WRITE 権限を付与します。この ACL はバケットのみに適用されます。これをバケットに適用すると、インターネット上の誰でも認証せずにオブジェクトを一覧表示、作成、上書き、削除できます。</li> </ul> |

| プロパティ        | 説明   |
|--------------|--|
| サーバーサイド暗号化キー | Google Cloud Storage バケットのサーバーサイドの暗号化キー。Google Cloud Storage バケットが SSE-KMS で暗号化されている場合に必要です。 |
| コンテンツ破棄タイプ   | オブジェクトに添付される RFC-6266 コンテンツ廃棄のタイプ。 <b>インライン</b> または <b>添付</b> のどちらかを選択します。                   |

## Google PubSub ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Google PubSub ターゲットのプロパティについて説明します。

| プロパティ  | 説明   |
|--------|--|
| 接続     | Google PubSub ターゲット接続の名前。                              |
| 接続タイプ  | Google PubSub 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。 |
| トピック   | ターゲットの Google PubSub トピックの名前。                          |
| バッチサイズ | Cloud サービスがバッチでバンドルするメッセージの最大数。<br>デフォルトは 1 です。        |

## JDBC V2 ターゲットプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる JDBC V2 ターゲットのプロパティについて説明します。

| プロパティ | 説明   |
|-------|--|
| 接続    | JDBC V2 ターゲット接続の名前。                              |
| 接続タイプ | JDBC V2 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。 |
| テーブル名 | JSON 形式でデータを挿入するテーブルの名前。                         |



## Kafka ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Kafka ターゲットのプロパティについて説明します。

| プロパティ   | 説明  |
|---------|---|
| 接続      | Kafka ターゲット接続の名前。   |
| 接続タイプ   | Kafka 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。  |
| トピック名/式 | Kafka トピック名または Java がサポートする Kafka トピック名パターンの正規表現。<br>正規表現には\$expression\$形式を使用します。\$expression\$はデータを評価し、一致するデータをキャプチャグループ 1 に送信します。<br>トピック名を手動で入力するか、Kafka 接続の作成済みメタデータを取得できます。<br>1. <b>【選択】</b> をクリックします。<br><b>【ターゲットオブジェクトの選択】</b> ダイアログボックスが表示され、Kafka ブローカで使用可能なすべてのトピックが示されます。ただし、Kafka トピック名のパターンはリストに表示されません。<br>2. 必要なトピックを選択して <b>【OK】</b> をクリックします。 |

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Kafka ターゲット詳細プロパティについて説明します。

| プロパティ                | 説明   |
|----------------------|--|
| プロデューサ設定プロパティ        | プロデューサの設定プロパティ。                              |
| メタデータフェッチタイムアウト（ミリ秒） | それ以降にメタデータが取得されなくなる時間。                       |
| バッチフラッシュサイズ（バイト）     | ストリーミング統合タスクがそれ以降なターゲットにデータを書き込むイベントのバッチサイズ。 |

## Microsoft Azure Data Lake Storage Gen2 ターゲットのプロパティ

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Microsoft Azure Data Lake Storage Gen2（ADLS Gen2）ターゲットのプロパティについて説明します。

| プロパティ | 説明   |
|-------|--|
| 接続    | Microsoft Azure Data Lake Storage Gen2 ターゲット接続の名前。 |
| 接続タイプ | ADLS Gen2 接続タイプ。<br>接続タイプは、選択した接続名に基づいて自動的に入力されます。 |

| プロパティ        | 説明  |
|--------------|---|
| 書き込みストラテジ    | <p>ADLS Gen2 ファイルにデータを書き込む操作タイプ。<br/>ファイルが ADLS Gen2 ストレージに存在する場合は、既存のファイルの上書き、付加、失敗、またはロールオーバーを選択できます。<br/>デフォルトは <b>【付加】</b> です。</p> <ul style="list-style-type: none"> <li>- <b>付加</b>。ディレクトリ内の既存のファイルにデータを追加します。</li> <li>- <b>上書き</b>。既存のファイル内の既存のデータを削除し、新しく読み取ったデータを挿入します。</li> <li>- <b>失敗</b>。既存のファイルへのデータの書き込みは失敗します。</li> <li>- <b>ロールオーバー</b>。データが書き込まれている現在のファイルを閉じ、設定されたロールオーバー値に基づいて新しいファイルを作成します。</li> </ul> <p><b>注:</b> ロールオーバーストラテジを使用するターゲットを含むストリーミング取り込みジョブを編集または再デプロイすると、定義されたロールオーバー条件を満たしていない場合でも、ステージングディレクトリ内のすべてのファイルがターゲットディレクトリに移動されます。</p> |
| 中間ディレクトリ     | <p>ADLS Gen2 のステージングディレクトリへのパス。<br/>ADLS Gen2 にデータを書き込むときに、ファイルをステージングするステージングディレクトリを指定します。ディレクトリに十分なスペースがあり、ディレクトリへの書き込み権限があることを確認します。</p> <p><b>【書き込みストラテジ】</b> として <b>【ロールオーバー】</b> を選択した場合に適用されます。<br/>ストリーミング取り込みジョブで ADLS Gen2 ターゲットを設定しているときに、ロールオーバープロパティの値を指定しないと、ファイルは中間ディレクトリに残ります。ストリーミング取り込みジョブを停止またはデプロイ解除すると、中間ディレクトリ内のこれらのファイルは、デフォルトでターゲットの場所に移動されます。</p>  |
| ロールオーバーサイズ   | <p>ロールオーバーをトリガする際のターゲットファイルサイズ (KB)。<br/><b>【書き込みストラテジ】</b> として <b>【ロールオーバー】</b> を選択した場合に適用されます。</p>  |
| ロールオーバーイベント数 | <p>ロールオーバーのために蓄積するイベントまたはメッセージの数。<br/><b>【書き込みストラテジ】</b> として <b>【ロールオーバー】</b> を選択した場合に適用されます。</p>   |
| ロールオーバー時間    | <p>ターゲットファイルがロールオーバーする時間の長さ (ミリ秒単位)。この時間が経過すると、ターゲットファイルはロールオーバーします。<br/><b>【書き込みストラテジ】</b> として <b>【ロールオーバー】</b> を選択した場合に適用されます。</p>  |
| ファイル名/式      | <p>ファイル名またはファイル名パターンの正規表現。<br/>正規表現には \$expression\$ 形式を使用します。\$expression\$ はデータを評価し、一致するデータをキャプチャグループ 1 に送信します。</p>   |

次の表では、ストリーミング取り込みタスクを定義するときに【ターゲット】タブで設定できる Microsoft Azure Data Lake Storage Gen2 (ADLS Gen2) ターゲット詳細プロパティについて説明します。

| プロパティ             | 説明  |
|-------------------|---|
| ファイルシステム名のオーバーライド | 接続で提供されたデフォルトのファイルシステム名をオーバーライドします。このファイルシステム名は、実行時にファイルに書き込むために使用されます。   |
| ディレクトリのオーバーライド    | デフォルトのディレクトリパスをオーバーライドします。<br>データの書き込みに使用する ADLS Gen2 ディレクトリ。<br>デフォルトはルートディレクトリです。<br>ターゲットの作成時に指定されたディレクトリパスで、接続の作成時に指定されたパスがオーバーライドされます。   |
| 圧縮形式              | オプション。ストリーミング取り込みタスクがターゲットファイルにデータを書き込む前に使用する圧縮形式。<br>次のいずれかの形式を使用します。<br><ul style="list-style-type: none"> <li>- なし</li> <li>- gzip</li> <li>- Bzip2</li> <li>- Zlib</li> <li>- Deflate</li> </ul> デフォルトは【なし】です。<br>データレイクストレージから圧縮ファイルを読み取るには、その圧縮ファイルに特定の拡張子が付いている必要があります。圧縮ファイルの読み取りに使用された拡張子が有効でない場合、Secure Agent はそのファイルを処理しません。 |

## トランスフォーメーションの設定

ストリーミングデータのデータ形式を指定できます。データ形式に基づいて、トランスフォーメーションを設定できます。

1. 【トランスフォーメーション】ページで、ストリーミングデータの形式を選択します。  
ストリーミング統合トランスフォーメーションは次のデータ形式をサポートします。
  - Binary
  - JSON
  - XML
2. 選択したデータ形式に基づいて、サポートされているトランスフォーメーションの 1 つを選択し、設定します。
3. 複数のトランスフォーメーションを追加するには、【トランスフォーメーションの追加】をクリックします。
  - a. 【トランスフォーメーション】タブで【トランスフォーメーションの追加】をクリックします。  
【新しいトランスフォーメーション】ダイアログボックスが表示されます。
  - b. 選択したトランスフォーメーションのタイプに基づいて、必要なプロパティを入力します。
  - c. 【保存】をクリックします。  
保存されたトランスフォーメーションは【トランスフォーメーションの詳細】ウィザードの【トランスフォーメーション】の下に表示されます。
4. 次のいずれかの手順に従います。

- タスクのランタイムオプションを設定するには、**[次へ]** をクリックします。  
**[ランタイムオプション]** タブが表示されます。
- タスクを保存するには、**[保存]** をクリックします。  
これで、ストリーミング取り込みタスクをデプロイできます。ストリーミング取り込みタスクのデプロイの詳細については、[「ストリーミング統合タスクのデプロイ」 \(ページ 50\)](#)を参照してください。

## トランスフォーメーションの追加

1. **[トランスフォーメーション]** タブで **[+]** をクリックしてトランスフォーメーションを追加します。

Transformation Details

Incoming Message Format: Binary

Transformations +

| Name               | Type |
|--------------------|------|
| No data to display |      |

**[新しいトランスフォーメーション]** ダイアログボックスが表示されます。

2. 選択したトランスフォーメーションのタイプに基づいて、必要なプロパティを入力します。
3. **[保存]** をクリックします。  
保存されたトランスフォーメーションは **[トランスフォーメーションの詳細]** ウィザードの **[トランスフォーメーション]** の下に表示されます。

## Combiner トランスフォーメーションのプロパティ

次の表では、Combiner トランスフォーメーションに設定できるプロパティについて説明します。

| プロパティ            | 説明   |
|------------------|--|
| トランスフォーメーションのタイプ | [Combiner] を選択します。   |
| トランスフォーメーション名    | Combiner トランスフォーメーションの名前。  |
| イベントの最小数         | トランスフォーメーション前に収集するイベントの最小数は、イベントを 1 つのイベントに結合します。<br>デフォルトは 1 です。                                |
| 最大集計サイズ          | 結合されたイベントの最大サイズ (メガバイト単位)。<br>指定されていない場合、このトランスフォーメーションは、イベントを結合する前に、他の 2 つの条件のいずれかが満たされるのを待ちます。 |
| 制限時間             | イベントを結合するまでの最大待ち時間。<br>指定されていない場合、このトランスフォーメーションはイベントを結合する前に他の条件を待機するか、永久に待機します。                 |

| プロパティ                   | 説明   |
|-------------------------|--|
| 区切り文字                   | 変換されたデータのデータ文字列間の分割を指定するために使用される記号。<br>バイナリデータ形式にのみ適用されます。   |
| 各バッチの最終レコードの末尾に区切り文字を追加 | イベントまたはレコードを含むバッチが多数ある場合は、各バッチの最後のレコードの末尾に区切り文字を使用するかどうかを選択できます。これにより、区切り文字が各バッチ間の境界として機能できるようになります。 |

## Filter トランスフォーメーションのプロパティ

次の表では、フィルタトランスフォーメーションに設定できるプロパティについて説明します。

| プロパティ            | 説明  |
|------------------|---|
| トランスフォーメーションのタイプ | [フィルタ] を選択します。  |
| トランスフォーメーション名    | フィルタトランスフォーメーションの名前。  |
| フィルタタイプ          | 受信データを評価するフィルタのタイプ。<br>次のいずれかのフィルタタイプを使用します。 <ul style="list-style-type: none"> <li>- JSON パス。JSON プロパティのシーケンスで構成される式。</li> <li>- 正規表現。値の範囲またはパターン。</li> <li>- XPath。XML ドキュメント内のノードまたはノードセットを選択する式。</li> </ul> |
| 式                | 選択したフィルタタイプの式。  |

## 形式変換トランスフォーメーションのプロパティ

ストリーミング取り込みタスクを定義して、形式変換トランスフォーメーションを追加するときに、タスクウィザードの【新しいトランスフォーメーション】ページでトランスフォーメーションプロパティの値を指定します。

次の表では、形式変換トランスフォーメーションに設定できるプロパティについて説明します。

| プロパティ            | 説明   |
|------------------|--|
| トランスフォーメーションのタイプ | [形式変換] を選択します。   |
| トランスフォーメーション名    | 形式変換トランスフォーメーションの名前。   |
| 形式に変換            | ストリーミング取り込みタスクは、受信データを選択した形式に変換します。現在、形式変換トランスフォーメーションは、受信データを Parquet 形式のみに変換します。 |
| 日付形式*            | 入力フィールドに日付の形式を入力します。例: MM/dd/yyyy。   |
| 時間形式*            | 入力フィールドに時間の形式を入力します。例: HH/mm/ss。   |

| プロパティ  | 説明   |
|--|--|
| タイムスタンプの形式*  | 入力フィールドにタイムスタンプの形式を入力します。<br>例えば、10/11/2021 12:04:41 GMT (MM/dd/yyyy HH:mm:ss) のエポックタイムスタンプは 1633953881 であり、ミリ秒単位のタイムスタンプは 1633953881000 です。 |
| レコードを配列と想定   | 単一のレコードとレコードの配列のどちらを想定するかを決定します。各レコードで配列を想定するには、このプロパティを選択します。XML 受信メッセージにのみ適用されます。デフォルトでは、このプロパティの選択は解除されています。                              |
| *形式が指定されていない場合は、エポック（1970 年 1 月 1 日午前 0 時 GMT）からのミリ秒単位と見なされます。 |  |

## Java トランスフォーメーションプロパティ

次の表では、Java トランスフォーメーションに設定できるプロパティについて説明します。

| プロパティ            | 説明  |
|------------------|---|
| トランスフォーメーションのタイプ | Java を選択します。  |
| トランスフォーメーション名    | Java トランスフォーメーション名。   |
| クラスパス            | Java コードの実行に使用される JAR ファイル。区切り文字を使用して、複数の JAR ファイルを含めることができます。UNIX の場合、複数のクラスパスの各項目を区切るにはコロンを使用します。Windows の場合、複数のクラスパスの各項目を区切るにはセミコロンを使用します。<br>例: /home/user/commons-text-1.9.jar; /home/user/json-simple-1.1.1.jar   |
| コードをインポート        | サードパーティ、組み込み、およびカスタムの Java パッケージをインポートします。複数のパッケージをインポートできます。複数のパッケージを区切るにはセミコロン (:) を使用します。次の構文を使用して、パッケージをインポートできます。<br>import <package name><br>例: import java.io.*;   |
| メインコード           | トランスフォーメーションロジックを提供する Java コード。<br>例:<br><pre>JSONParser parser = new JSONParser();     try {         JSONObject object = (JSONObject) parser.parse(inputData);         object.put("age", 23);         outputData=object.toJSONString();     } catch (ParseException e) {         throw new RuntimeException();     }</pre> |

## Jolt トランスフォーメーションプロパティ

次の表では、Jolt トランスフォーメーションに設定する必要があるプロパティについて説明します。

| プロパティ            | 説明  |
|------------------|---|
| トランスフォーメーションのタイプ | Jolt を選択します。  |
| トランスフォーメーション名    | Jolt トランスフォーメーション名。   |
| Jolt の仕様         | <p>JSON 構造を入力し、次の操作を使用して複数の操作のチェーンを追加します。</p> <ul style="list-style-type: none"> <li>- shift。入力 JSON の値を読み取り、その値を出力 JSON の指定された場所に追加します。<br/>例:<br/>[ { "operation": "shift", "spec": { "breadbox": "counterTop" } } ]</li> <li>- default。値または値の配列を出力 JSON に追加します。<br/>例:<br/>[ { "operation": "default", "spec": { "counterTop": { "loaf1": { "slices": [ "slice1", "slice2", "slice3", "slice4 ] } } } } ]</li> <li>- cardinality。入力 JSON の要素を出力 JSON の単一の値または配列（リスト）に変換します。<br/>例:<br/>[ { "operation": "cardinality", "spec": { "counterTop": { "loaf1": { "slices": "ONE" } } } } ]</li> <li>- remove。入力 JSON で見つかった場合、要素を削除します。<br/>例:<br/>[ { "operation": "remove", "spec": { "counterTop": { "loaf2": "", "jar1": "" } } } ]</li> <li>- modify。計算された値を出力 JSON の要素に書き込みます。<br/>例:<br/>[ { "operation": "modify-overwrite-beta", "spec": { "counterTop": { "jar2": { "contents": "=toUpper" } } } } ]</li> <li>- sort。すべての配列とマップを入力 JSON から出力 JSON に並べ替えます。<br/>例:<br/>[ { "operation": "sort" } ]</li> </ul> <p>以下は、複数の操作を含む Jolt の仕様の例です。</p> <p>Input: { "name": "test" }</p> <pre>[{   "operation": "shift",   "spec": {     "name": "testname"   }, {     "operation": "default",     "spec": {       "city": ["Anantapur", "Bangalore", "Hyderabad"]     }   }, {     "operation": "cardinality",     "spec": {       "city": "ONE"     }   } }]</pre> |

| プロパティ | 説明   |
|-------|--|
|       | <pre>       }, {         "operation": "remove",         "spec": {           "age": ""         }       }, {         "operation": "modify-overwrite-beta",         "spec": {           "city": "=toUpper"         }       }, {         "operation": "sort"       }     ] </pre> <p>注: 入力レコードが Jolt の仕様と一致しない場合、トランスフォーメーションは null レコードをターゲットに書き込みます。</p> |

## Python トランスフォーメーションのプロパティ

次の表では、Python トランスフォーメーションに設定できるプロパティについて説明します。

| プロパティ            | 説明  |
|------------------|---|
| トランスフォーメーションのタイプ | [Python] を選択します。  |
| トランスフォーメーション名    | Python トランスフォーメーションの名前。   |
| スクリプトの入力タイプ      | Python スクリプトの入力タイプ。Python スクリプトを <b>[スクリプト本文]</b> に入力するか、 <b>[スクリプトパス]</b> で利用可能な Python スクリプトへのパスを指定します。 |
| Python のパス       | Python パスライブラリへのディレクトリ。   |

## Splitter トランスフォーメーションのプロパティ

次の表では、バイナリメッセージ形式の Splitter トランスフォーメーションに設定できるプロパティについて説明します。

| プロパティ            | 説明   |
|------------------|--|
| トランスフォーメーションのタイプ | [Splitter] を選択します。   |
| トランスフォーメーション名    | Splitter トランスフォーメーションの名前。  |
| 分割タイプ            | 条件を分割して、受信データを評価します。<br>次のいずれかの分割タイプを使用します。 <ul style="list-style-type: none"> <li>- 行の分割。</li> <li>- コンテンツの分割。</li> </ul> |



| プロパティ    | 説明                           |
|----------|------------------------------|
| 行の分割数    | ヘッダー行を除く、各出力分割ファイルに含まれる最大行数。 |
| バイトシーケンス | コンテンツを分割するバイトの指定されたシーケンス。    |

次の表では、JSON メッセージ形式の Splitter トランスフォーマーに設定できるプロパティについて説明します。

| プロパティ      | 説明   |
|------------|--|
| 分割式        | 条件を分割して、受信データを評価します。<br>次のいずれかの分割タイプを使用します。<br>- 配列の分割。<br>- JSONPath 式。 |
| JSONPath 式 | JSON またはスカラーフラグメントに分割する配列要素を指定する JSONPath 式。<br>デフォルトの JSONpath 式は\$です。  |

次の表では、XML メッセージ形式の Splitter トランスフォーマーに設定できるプロパティについて説明します。

| プロパティ | 説明  |
|-------|---|
| 分割の深さ | XML フラグメントの分割を開始する XML のネストの深さ。<br>デフォルトの分割の深さは 1 です。 |

## ランタイムオプションの設定

ストリーミング統合タスクの追加のランタイムオプションを設定できます。ランタイムオプションには、拒否イベントを管理する設定と、エラーについてユーザーに通知する設定が含まれます。


1. **【ランタイムオプション】** ページの **【通知管理】** で、通知を無効にするか、エラーが発生した場合に通知を受信するまでの時間制限を設定するかを選択します。  
時間を分単位または時間単位で設定できます。デフォルトは 10 分です。
2. タスクがエラーありで実行された場合に通知を送信する電子メールアドレスのリストを入力します。  
電子メールアドレスのリストを区切るには、カンマを使用します。組織用に設定された電子メール通知オプションはここでは使用されません。
3. **【エージェントパラメータ】** で、次のタスクを実行します。
  - a. 拒否されたイベントを保存するディレクトリを指定します。  
拒否されたイベントはデフォルトでは保存されません。  
正規表現でフィルタされたイベントは、拒否ディレクトリには移動されません。
  - b. ログファイルを消去するには、ログファイル消去後のログファイルの最大ファイルサイズを指定します。  
ファイルサイズはメガバイトまたはギガバイトで設定できます。デフォルトは 10MB です。
  - c. ログに記録するイベントの重要度を選択します。

デフォルトのログレベルは【情報】です。

サポートされているログレベルは次のとおりです。

- デバッグ。追加のデバッグレベルメッセージとともにすべてのメッセージを記録します。
- エラー。エラーメッセージのみを記録します。
- 情報。すべてのエラー、警告、および重要な情報メッセージを記録します。
- 警告。すべてのエラーと警告メッセージを記録します。

4. 【詳細パラメータ】で、次のタスクを実行して、ストリーミング統合タスクのパフォーマンスを向上させます。

- a. アイコン （【詳細パラメータ】の横にある）をクリックします。

【キー】フィールドと【値】フィールドが表示されます。

- b. 有効なパラメータプロパティとその値を指定します。

5. タスクを保存するには、【保存】をクリックします。

これで、ストリーミング統合タスクをデプロイできます。ストリーミング統合タスクのデプロイの詳細については、[「ストリーミング統合タスクのデプロイ」](#)（ページ 50）を参照してください。

## ストリーミング統合タスクのデプロイ

ストリーミング統合タスクを作成した後、タスクをジョブとして実行するには、Secure Agent にデプロイする必要があります。タスクをデプロイする前に、タスク用に設定された Secure Agent が実行されていることを確認してください。

- ▶ オブジェクトをデプロイするには、次のいずれかのアクションを実行します。

- タスクを保存したら、【デプロイ】をクリックします。
- 【参照】ページで、タスクを含むプロジェクトを開き、タスクの【アクション】メニューから【デプロイ】を選択します。

タスクが正常にデプロイされたことを示すメッセージが表示されます。これで、ジョブは実行のためのキューに入れられました。

ジョブを編集またはデプロイ解除しても、ジョブを再度デプロイしても、同じ Secure Agent または Secure Agent グループで実行されている他のジョブには影響しません。

ロールオーバーストラテジを使用するターゲットを含むストリーミング統合ジョブを編集または再デプロイすると、定義されたロールオーバー条件を満たしていない場合でも、ステージングディレクトリ内のすべてのファイルがターゲットディレクトリに移動されます。

## ストリーミング統合ジョブのデプロイ解除

Secure Agent からストリーミング統合ジョブをデプロイ解除できます。

- ▶ 【参照】ページで、ジョブを含むプロジェクトを開き、ジョブの【アクション】メニューから【デプロイ解除】を選択します。

または、オペレーションインサイトの一括取り込みページのジョブ行の【アクション】メニューからストリーミング統合ジョブをデプロイ解除することもできます。

ストリーミング統合ジョブをデプロイ解除すると、Secure Agent はジョブの以前の状態履歴を保存しません。したがって、ストリーミング統合タスクを再度デプロイすると、タスクは完全に新しいジョブとして実行されます。

## ストリーミング統合ジョブの停止と再開

ストリーミング統合ジョブは停止および再開できます。一括取り込みの【マイジョブ】ページまたはオペレーションインサイトの一括取り込みページの【すべてのジョブ】ページでジョブを停止または再開します。

### ジョブを停止します。

稼働中、実行中（エラーあり）、または実行中（警告あり）のストリーミング統合ジョブを停止することができます。

ジョブを停止するには、【マイジョブ】ページを開き、ジョブの【アクション】メニューから【停止】を選択します。

### ジョブを再開します。

停止されているストリーミング統合ジョブを再開できます。

ジョブを再開するには、【マイジョブ】ページを開き、ジョブの【アクション】メニューから【再開】を選択します。

または、オペレーションインサイトの一括取り込みページのジョブ行の【アクション】メニューからストリーミング統合ジョブを停止または再開することもできます。

ストリーミング統合ジョブを停止すると、Secure Agent はジョブの以前の状態履歴を保存します。停止されたストリーミング統合ジョブを再開すると、ジョブは最後に保存された状態から実行を開始します。

## 一括取り込みストリーミングの動作に関するよくある質問

以下のよくある質問を確認して、製品の動作を理解してください。

### チェックポイントは、Secure Agent、クラウドリポジトリ、ソース、またはターゲットに保存されていますか。

チェックポイントはソースとターゲットに応じて、Secure Agent、クラウドリポジトリ、ソース、またはターゲットに保存されます。チェックポイントの保存先は、ソースとターゲットによって異なります。

例えば、Amazon Kinesis Streams ソースの場合、ストリーミング取り込みによって Amazon S3 などのデータウェアハウスに DynamoDB チェックポイントテーブルが作成されます。

### タスクをデプロイまたはデプロイ解除した後にチェックポイントを設定できますか。

いいえ、タスクをデプロイまたはデプロイ解除した後にチェックポイントを設定することはできません。

Kafka ソースと Kafka ターゲットを使用するストリーミング取り込みタスクでは、Kafka に保存されているオフセットを再起動チェックポイントとして使用できますか。

はい、Kafka に保存されているオフセットを再起動チェックポイントとして扱うことができます。

Kafka ソースと Kafka ターゲットを使用したストリーミング取り込みタスクでは、group.id プロパティは、タスクがデプロイ解除されてデプロイされた後にチェックポイントを再起動するのに役立ちますか。

group.id は、チェックポイントのリカバリ役立つ Kafka の共有プロパティです。グループ内の Kafka ソースがメッセージを正常に処理すると、トピックパーティション内のオフセットが更新されます。このオフセットは、group.id とともに保存されます。Kafka データフローがデプロイ解除からデプロイ済みステータスに移行し、稼働状態になると、同じ group.id と保存されたオフセットを使用して、データフローが停止した場所から処理を再開します。

**タスクの失敗後にターゲットで重複している可能性を許容する [少なくとも 1 回は配信] を保証できますか。**

はい、[少なくとも 1 回は配信] は保証されています。

**ソースコネクタとターゲットコネクタによって配信に違いはありますか。**

各コネクタには固有の実装がありますが、配信ではすべて [少なくとも 1 回は配信] を保証するように設計されています。

**障害後のリカバリはどのように行われますか。**

Secure Agent が障害から復旧すると、ランタイムエンジンが再起動し、調整プロセスが実行されます。

**一括取り込みストリーミングは、複数の Secure Agent が含まれる Secure Agent グループをサポートしますか。**

一括取り込みストリーミングは、1 つのエージェントグループで複数の Secure Agent をサポートします。ただし、データフローを複数の Secure Agent で同時に実行することはできません。

**一括取り込みストリーミングは共有 Secure Agent グループをサポートしていますか。**

一括取り込みストリーミングは、共有 Secure Agent グループをサポートしていません。

## 第 2 章

# 一括取り込みストリーミング REST API

ストリーミング取り込みタスクのデプロイ、デプロイ解除、開始、停止、コピー、更新、およびストリーミング取り込みジョブの監視を行うには、ストリーミング取り込みリソースを使用します。

ストリーミング取り込みリソースを使用する場合は、次の要求ヘッダー形式を使用してください。

```
<METHOD><base URL>  
Content-Type: application/json  
Accept: application/json  
IDS-SESSION-ID: <SessionId>
```

## Dataflows リソース

Dataflows リソースを使用して、ストリーミング統合タスクをデプロイ、デプロイ解除、開始、および停止します。

次のベース URL を使用する。

```
<server URI>/sisvc/api/v1/Dataflows('<dataflow ID>')/OData.SI.<API name>
```

**注:** HTTP バージョンを自動的に含める Postman などのツールを使用する場合は、URL に HTTP バージョンを入力しないでください。HTTP バージョンが URL に 2 回出現すると、要求が失敗します。

## ストリーミング取り込みタスクのデプロイ

POST 要求を使用してストリーミング統合タスクをデプロイします。

### POST 要求

ストリーミング統合タスクをデプロイするには、次の URL を使用します。

```
<server URI>/sisvc/api/v1/Dataflows('<dataflow ID>')/OData.SI.Deploy
```

URL はデータフロー ID を渡すため、要求本文は必要ありません。

### POST 要求の例

ストリーミング統合タスクをデプロイするには、次の例のような要求を送信します。

```
POST <serverUrl>/sisvc/api/v1/Dataflows('50077311-d4a4-437c-9218-c3596d1f182f')/OData.SI.Deploy  
Content-Type: application/json
```

```
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

### POST 応答の例

要求が成功した場合、次の例のような応答を受信します。

```
{
  "@odata.context": "$metadata#OData.SI.DeploymentResult",
  "successful": true,
  "code": null,
  "errorMessage": null
}
```

## ストリーミング取り込みタスクのデプロイ解除

POST 要求を使用してストリーミング統合タスクをデプロイ解除します。

### POST 要求

ストリーミング統合タスクをデプロイ解除するには、次の URL を使用します。

<server URI>/sisvc/api/v1/Dataflows('<dataflow ID>')/OData.SI.Undeploy

URL はデータフロー ID を渡すため、要求本文は必要ありません。

### POST 要求の例

ストリーミング統合タスクをデプロイ解除するには、次の例のような要求を送信します。

```
POST <serverUrl>/sisvc/api/v1/Dataflows('50077311-d4a4-437c-9218-c3596d1f182f')/OData.SI.Undeploy
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

### POST 応答の例

要求が成功した場合、次の例のような応答を受信します。

```
{
  "@odata.context": "$metadata#OData.SI.DeploymentResult",
  "successful": true,
  "code": null,
  "errorMessage": null
}
```

## ストリーミング取り込みタスクの開始

POST 要求を使用してストリーミング統合タスクを開始します。

### POST 要求

ストリーミング統合タスクを開始するには、次の URL を使用します。

<server URI>/sisvc/api/v1/Dataflows('<dataflow ID>')/OData.SI.Start

URL はデータフロー ID を渡すため、要求本文は必要ありません。

### POST 要求の例

ストリーミング統合タスクを開始するには、次の例のような要求を送信します。

```
POST <serverUrl>/sisvc/api/v1/Dataflows('50077311-d4a4-437c-9218-c3596d1f182f')/OData.SI.Start
Content-Type: application/json
Accept:application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

## POST 応答の例

要求が成功した場合、次の例のような応答を受信します。

```
{
  "@odata.context": "$metadata#OData.SI.DeploymentResult",
  "successful": true,
  "code": null,
  "errorMessage": null
}
```

## ストリーミング取り込みタスクの停止

POST 要求を使用してストリーミング統合タスクを停止します。

### POST 要求

ストリーミング統合タスクを停止するには、次の URL を使用します。

<server URI>/sisvc/api/v1/Dataflows('<dataflowID>')/OData.SI.Stop

URL はデータフロー ID を渡すため、要求本文は必要ありません。

### POST 要求の例

ストリーミング統合タスクを停止するには、次の例のような要求を送信します。

```
POST <serverUrl>/sisvc/api/v1/Dataflows('d7572789-dc4c-4c56-bbeb-3772736d61aa')/OData.SI.Stop
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: 2l0oeVx22Rujiej7yTokmT
```

### POST 応答の例

要求が成功した場合、次の例のような応答を受信します。

```
{
  "@odata.context": "$metadata#OData.SI.DeploymentResult",
  "successful": true,
  "code": null,
  "errorMessage": null
}
```

## CopyEntities リソース

ストリーミング統合タスクをコピーするには、CopyEntities リソースを使用します。

### POST 要求

ストリーミング統合タスクをコピーするには、次の URL を使用します。

<server URI>/sisvc/restapi/v1/CopyEntities

要求に以下のフィールドを含めることができます。

| フィールド            | タイプ    | 必須 | 説明                          |
|------------------|--------|----|-----------------------------|
| targetLocationID | String | ○  | オブジェクトのコピー先となるターゲットの場所の ID。 |
| sourceEntities   | Array  | ○  | ソース接続とターゲット接続の構成。           |

| フィールド             | タイプ    | 必須 | 説明              |
|-------------------|--------|----|-----------------|
| sourceId          | String | ○  | ソースオブジェクトの ID。  |
| targetName        | String | ○  | ターゲットオブジェクトの名前。 |
| targetDescription | String | -  | ターゲットオブジェクトの説明。 |

## POST 要求の例

2 つのストリーミング統合タスクをコピーするには、次の例のような要求を送信できます。

```
POST <serverUrl>/sisvc/restapi/v1/CopyEntities
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: 2l0oeVx22Rujiej7yTokmT
{
  "targetLocationID": "2RGmVdwN15PbfnQP5PSoSB",
  "sourceEntities": [
    {
      "sourceId": "5Ff6jeaSh2UfAqiV01ldKD",
      "targetName": "Test_Copy_A",
      "targetDescription": "Description_1"
    },
    {
      "sourceId": "fZnCSqcwT0QkJ0r8VCWZQE",
      "targetName": "Test_Copy_B",
      "targetDescription": "Description_2"
    }
  ]
}
```

要求が失敗すると、応答にエラーの理由が含まれます。

## POST 応答

POST リクエストを使用してストリーミング統合タスクをコピーする場合、成功すると成功応答を返し、エラーが発生するとエラーオブジェクトを返します。

## POST 応答の例

要求が成功した場合、次の例のような応答を受信します。

```
{
  "Status Message": "Operation succeeded on 2 artifacts.",
  "Success": {
    "Test_Copy_A": "ideNJw6l54gizxofF53HQH",
    "Test_Copy_B": "c0Q3gcWKSyIkzVqqg6IOok"
  }
}
```

# UpdateEntity リソース

ストリーミング統合タスクを更新するには、UpdateEntity リソースを使用します。Amazon Kinesis、Amazon S3 V2、Microsoft Azure Event Hub、Microsoft Azure Data Lake Storage Gen2、フラットファイル、JDBC V2、JMS、Kafka、または MQTT のコネクタを使用するストリーミング統合タスクを更新できます。



## POST 要求

POST 要求を使用してストリーミング統合タスクを更新します。

ストリーミング統合タスクを更新するには、次の URL を使用します。

`<server URI>/sisvc/restapi/v1/UpdateEntity/Documents('<document ID>')`

要求に以下のフィールドを含めることができます。

| フィールド          | タイプ    | 必須 | 説明                    |
|----------------|--------|----|-----------------------|
| name           | String | ○  | タスクの名前。               |
| description    | String | -  | タスクの説明。               |
| runtimeId      | String | ○  | ランタイム環境の ID。          |
| currentVersion | String | ○  | 最新のデータフローオブジェクトバージョン。 |
| nodes          | Array  | ○  | タスクのソース接続とターゲット接続の詳細。 |

### nodes 配列のフィールド

配列内のフィールドは、接続の名前、タイプ、および接続 ID を指定します。これには、編集可能なキーと値のペアであるソース接続とターゲット接続の構成が含まれます。nodes 配列には以下のフィールドを含めることができます。

| フィールド              | タイプ    | 必須 | 説明                 |
|--------------------|--------|----|--------------------|
| name               | String | ○  | 接続の名前。             |
| type               | String | ○  | 接続タイプ、ソースまたはターゲット。 |
| connectionId       | String | ○  | 接続の ID。            |
| transformationType | String | -  | 該当なし。              |
| config             | Array  | ○  | ソース接続とターゲット接続の構成。  |

## MQTT をソースとするタスクの接続構成

タスクソースのソース接続が MQTT の場合、ソース接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー           | タイプ     | 必須 | 説明  |
|--------------|---------|----|---|
| ClientID     | String  | -  | MQTT ソースと MQTT ブローカ間の接続を識別する一意の識別子。クライアント ID は、MQTT ソースがメッセージの処理中にメッセージを格納するために使用するファイルベースの永続ストアです。<br>255 文字までの文字列を入力できます。 |
| MaxQueueSize | Integer | -  | プロセッサがメモリに保存できるメッセージの最大数。<br>1 から 2147483647 までの数値を入力できます。  |
| Topic        | String  | ○  | MQTT トピックの名前。   |

### POST 要求の例

MQTT ソースとフラットファイルターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```
{
  "name": "mqtt to flatfile",
  "description": "mqtt to flatfile",
  "runtimeId": "010000250000000000003",
  "locationId": "5sJ0JDyJyWlrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "mqtt to flatfile_source",
      "type": "source",
      "connectionId": "012MGS0B0000000000010",
      "transformationType": "",
      "config": [
        {
          "key": "ClientID",
          "value": "test"
        },
        {
          "key": "MaxQueueSize",
          "value": 1024
        },
        {
          "key": "Topic",
          "value": "test"
        }
      ]
    },
    {
      "name": "mqtt to flatfile_target",
      "type": "target",
      "connectionId": "012MGS0B000000000002N",
      "transformationType": "",
      "config": [
        {
          "key": "interimDirectory",
          "value": "/home/agent/test"
        },
        {
          "key": "rolloverSize",
          "value": 1024
        }
      ]
    }
  ]
}
```

```

        "key": "rolloverEvents",
        "value": 100
    },
    {
        "key": "rolloverTime",
        "value": 300000
    },
    {
        "key": "File Name",
        "value": "test"
    }
]
}
},
"edges": [
{
    "from": "mqtt to flatfile_source",
    "to": "mqtt to flatfile_target"
}
]
}
}

```

## JMS をソースとするタスクの接続構成

タスクソースのソース接続が JMS の場合、ソース接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー                  | タイプ    | 必須 | 説明  |
|---------------------|--------|----|---|
| destinationType     | String | ○  | ソースサービスが JMS メッセージを送信する接続先のタイプ。次のいずれかの値を入力します。<br>- [キュー]。JMS プロバイダは、キューに登録されている単一のコンシューマにメッセージを配信します。<br>- [トピック]。JMS プロバイダは、トピックにサブスクライブしているすべてのアクティブなコンシューマにメッセージを配信します。 |
| clientId            | String | ○  | JMS 接続の一意の ID。255 文字までの文字列を入力できます。  |
| sharedSubscription  | String | ○  | 複数のコンシューマが単一のサブスクリプションにアクセスできるようにします。トピックの接続先タイプに適用されます。次のいずれかの値を入力します。<br>- True<br>- False  |
| durableSubscription | String | ○  | True に設定した場合、JMS ソースサービスは、非アクティブなサブスクライバがメッセージを保持し、サブスクライバが再接続したときにそれらのメッセージを配信できるようにします。トピックの接続先タイプに適用されます。次のいずれかの値を入力します。<br>- True<br>- False                            |
| subscriptionName    | String | ○  | サブスクリプションの名前。トピックサブスクリプションタイプが共有、継続、またはその両方である場合に、トピックの接続先タイプに適用されます。   |
| JMS Destination     | String | ○  | JMS プロバイダがメッセージを配信するキューまたはトピックの名前。  |

## POST 要求の例

JMS ソースとフラットファイルターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```
{
  "name": "crud",
  "description": "JMS to FileToFile",
  "runtimeId": "010000250000000000003",
  "locationId": "5sJ0JDyJyWllrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "crud_source",
      "type": "source",
      "connectionId": "012MGS0B0000000000003",
      "transformationType": "",
      "config": [
        {
          "key": "destinationType",
          "value": "QUEUE"
        },
        {
          "key": "clientId",
          "value": ""
        },
        {
          "key": "JMS Destination",
          "value": "test"
        }
      ]
    },
    {
      "name": "crud_target",
      "type": "target",
      "connectionId": "012MGS0B0000000000000H",
      "transformationType": "",
      "config": [
        {
          "key": "interimDirectory",
          "value": "/home/agent/test"
        },
        {
          "key": "rolloverSize",
          "value": 1024
        },
        {
          "key": "rolloverEvents",
          "value": 100
        },
        {
          "key": "rolloverTime",
          "value": 300000
        },
        {
          "key": "File Name",
          "value": "test"
        }
      ]
    }
  ],
  "edges": [
    {
      "from": "crud_source",
      "to": "crud_target"
    }
  ]
}
```

## Microsoft Azure Data Lake Storage Gen2（ADLS Gen2）をターゲットとするタスクの接続構成

タスクターゲットのターゲット接続が ADLS Gen2 の場合、ターゲット接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー                                | タイプ     | 必須 | 説明  |
|-----------------------------------|---------|----|---|
| writeStrategy                     | String  | ○  | ADLS Gen2 ストレージに同じ名前のファイルが存在する場合に実行するアクション。<br>次のいずれかの値を入力します。<br>- 付加。既存のファイルにデータを追加します。<br>- 上書き。既存のファイルを新しいファイルに置き換えます。<br>- 失敗。要求を失敗させます。<br>- ロールオーバー。現在のファイルを閉じて、構成されたロールオーバー値に基づいて新しいファイルを作成します。 |
| rolloverSize *                    | Integer | -  | ロールオーバーをトリガする際のターゲットファイルサイズ (KB)。ロールオーバー書き込みストラテジに適用されます。<br>1 から 2147483647 までの数値を入力できます。  |
| rolloverEvents *                  | Integer | -  | ロールオーバーの前に蓄積するイベントまたはメッセージの数。ロールオーバー書き込みストラテジに適用されます。<br>1 から 2147483647 までの数値を入力できます。  |
| rolloverTime *                    | Integer | -  | ロールオーバーをトリガするまでの時間（ミリ秒単位）。ロールオーバー書き込みストラテジに適用されます。<br>1 から 2147483647 までの数値を入力できます。   |
| filesystemNameOverride            | String  | -  | 接続で提供されたデフォルトのファイルシステム名をオーバーライドします。このファイルシステム名は、実行時にファイルに書き込むために使用されます。<br>1,280 文字までの文字列を入力できます。   |
| directoryOverride                 | String  | -  | デフォルトのディレクトリパスをオーバーライドします。データの書き込み先の ADLS Gen2 ディレクトリパス。空白の場合は、デフォルトのディレクトリパスが使用されます。<br>1,280 文字までの文字列を入力できます。   |
| compressionFormat                 | String  | -  | ストリーミング取り込みタスクがターゲットファイルにデータを書き込む前に使用する圧縮形式。<br>次のいずれかの値を入力します。<br>- なし<br>- GZIP<br>- BZIP2<br>- DEFAULT1<br>Zlib 形式を使用するには、この値を入力します。<br>- DEFAULT2<br>Deflate 形式を使用するには、この値を入力します。                   |
| File Name/Expression              | String  | ○  | ADLS Gen2 ファイル名または正規表現。<br>249 文字までの文字列を入力できます。   |
| * これらのフィールドの少なくとも 1 つに値を入力してください。 |         |    |   |

## POST 要求の例

フラットファイルソースと ADLS Gen2 ターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```
{
  "name": "flatfile to adls",
  "description": "flatfile to adls",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWllrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "flatfile to adls_source",
      "type": "source",
      "connectionId": "012MGS0B000000000002N",
      "transformationType": "",
      "config": [
        {
          "key": "File",
          "value": "logfile"
        },
        {
          "key": "initialPosition",
          "value": "Current Time"
        },
        {
          "key": "rolloverPattern",
          "value": "test"
        },
        {
          "key": "tailingMode",
          "value": "Single file"
        }
      ]
    },
    {
      "name": "flatfile to adls_target",
      "type": "target",
      "connectionId": "012MGS0B000000000003D",
      "transformationType": "",
      "config": [
        {
          "key": "writeStrategy",
          "value": "Rollover"
        },
        {
          "key": "filesystemNameOverride",
          "value": "test"
        },
        {
          "key": "File Name/Expression",
          "value": "test"
        },
        {
          "key": "compressionFormat",
          "value": "NONE"
        },
        {
          "key": "directoryOverride",
          "value": "/test"
        },
        {
          "key": "interimDirectory",
          "value": "/home/agent/test"
        },
        {
          "key": "rolloverSize",
          "value": 1024
        }
      ]
    }
  ]
}
```

```

    {
      "key": "rolloverEvents",
      "value": 100
    },
    {
      "key": "rolloverTime",
      "value": 300000
    }
  ]
}

```

## Amazon S3 をターゲットとするタスクの接続構成

タスクターゲットのターゲット接続が Amazon S3 の場合、ターゲット接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー                       | タイプ     | 必須 | 説明  |
|--------------------------|---------|----|---|
| partitionTime            | String  | -  | ストリーミング取り込みタスクが Amazon S3 バケットにパーティションを作成する際に従う時間間隔。<br>次のいずれかの値を入力します。<br><ul style="list-style-type: none"> <li>- なし</li> <li>- 5 分</li> <li>- 10 分</li> <li>- 15 分</li> <li>- 20 分</li> <li>- 30 分</li> <li>- 1 時間</li> <li>- 1 日</li> </ul> |
| minUploadPartSize        | Integer | -  | 複数の独立したパートセットとしてサイズの大きなファイルをアップロードする場合の最小パートサイズ (MB 単位)。このプロパティを使用して、ファイルのロードを Amazon S3 に合わせます。<br>50 から 5120 までの数値を入力できます。  |
| multipartUploadThreshold | Integer | -  | オブジェクトを複数のパートで並行してアップロードする場合のマルチパートしきい値。<br>50 から 5120 までの数値を入力できます。  |
| Object Name/Expression   | String  | ○  | Amazon S3 ターゲットファイル名、または Amazon S3 ファイル名パターンの正規表現。  |

## POST 要求の例

フラットファイルソースと Amazon S3 ターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```

{
  "name": "flatfile to amazon S3",
  "description": "flatfile to amazon S3",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLLrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "flatfile to amazon S3_source",

```

```

    "type": "source",
    "connectionId": "012MGS0B00000000002N",
    "transformationType": "",
    "config": [
      {
        "key": "File",
        "value": "logfile"
      },
      {
        "key": "initialPosition",
        "value": "Current Time"
      },
      {
        "key": "rolloverPattern",
        "value": "test"
      },
      {
        "key": "tailingMode",
        "value": "Single file"
      }
    ]
  },
  {
    "name": "flatfile to amazon S3_target",
    "type": "target",
    "connectionId": "012MGS0B00000000000I7",
    "transformationType": "",
    "config": [
      {
        "key": "partitionTime",
        "value": "None"
      },
      {
        "key": "minUploadPartSize",
        "value": 5120
      },
      {
        "key": "multipartUploadThreshold",
        "value": 5120
      },
      {
        "key": "Object Name/Expression",
        "value": "test"
      }
    ]
  }
],
"edges": [
  {
    "from": "flatfile to amazon S3_source",
    "to": "flatfile to amazon S3_target"
  }
]
}

```



## Azure Event Hubs をターゲットとするタスクの接続構成

タスクターゲットのターゲット接続が Azure Event Hubs の場合、ターゲット接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー                  | タイプ    | 必須 | 説明  |
|---------------------|--------|----|---|
| sasPolicyName       | String | -  | Event Hub 名前空間共有アクセスポリシーの名前。<br>255 文字までの文字列を入力できます。  |
| sasPolicyPrimaryKey | String | -  | Event Hub 名前空間共有アクセスポリシーのプライマリキー。<br>255 文字までの文字列を入力できます。                                   |
| Event Hub           | String | ○  | Azure Event Hubs の名前。<br>255 文字までの文字列を入力できます。名前には、小文字、大文字、数字、および特殊文字（「_」および「-」）を含めることができます。 |

### POST 要求の例

フラットファイルソースと Azure Event Hubs ターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```
{
  "name": "flatfile to azure event hub",
  "description": "flatfile to azure event hub",
  "runtimeId": "010000250000000000003",
  "locationId": "5sJ0JDyJyWLLrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "flatfile to azure event hub_source",
      "type": "source",
      "connectionId": "012MGS0B000000000002N",
      "transformationType": "",
      "config": [
        {
          "key": "File",
          "value": "logfile"
        },
        {
          "key": "initialPosition",
          "value": "Current Time"
        },
        {
          "key": "rolloverPattern",
          "value": "test"
        },
        {
          "key": "tailingMode",
          "value": "Single file"
        }
      ]
    },
    {
      "name": "flatfile to azure event hub_target",
      "type": "target",
      "connectionId": "012MGS0B0000000000001S",
      "transformationType": "",
      "config": [
        {
          "key": "sasPolicyName",
          "value": "test"
        }
      ]
    }
  ]
}
```

```

    {
      "key": "sasPolicyPrimaryKey",
      "value": "test"
    },
    {
      "key": "Event Hub",
      "value": "test"
    }
  ]
},
"edges": [
  {
    "from": "flatfile to azure event hub_source",
    "to": "flatfile to azure event hub_target"
  }
]
}

```

## JDBC V2 をターゲットとするタスクの接続構成

タスクターゲットのターゲット接続が JDBC V2 の場合、ターゲット接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー         | タイプ    | 必須 | 説明  |
|------------|--------|----|---|
| Table Name | String | ○  | JSON 形式でデータを挿入するテーブルの名前。<br>988 文字までの文字列を入力します。 |

## POST 要求の例

フラットファイルソースと JDBC V2 ターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```

{
  "name": "FileFile to jdbc",
  "description": "FileToFile to jdbc_target",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLLrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "flatfile to jdbc_source",
      "type": "source",
      "connectionId": "012MGS0B000000000002N",
      "transformationType": "",
      "config": [
        {
          "key": "initialPosition",
          "value": "Current Time"
        },
        {
          "key": "tailingMode",
          "value": "Single file"
        },
        {
          "key": "rolloverPattern",
          "value": "test"
        },
        {
          "key": "File",
          "value": "logfile"
        }
      ]
    }
  ]
}

```

```

    },
    {
      "name": "flatfile to jdbc_target",
      "type": "target",
      "connectionId": "012MGS0B0000000000KF",
      "transformationType": "",
      "config": [
        {
          "key": "Table Name",
          "value": "table"
        }
      ]
    }
  ],
  "edges": [
    {
      "from": "flatfile to jdbc_source",
      "to": "flatfile to jdbc_target"
    }
  ]
}

```

## Amazon Kinesis Streams をソースおよびターゲットとするタスクの接続構成

タスクのソース接続とターゲット接続が Amazon Kinesis Streams の場合、ソース接続とターゲット接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー                         | タイプ     | 必須 | 説明  |
|----------------------------|---------|----|---|
| appendGUID                 | Boolean |    | Amazon DynamoDB テーブル名にサフィックスとして GUID を追加するかどうかを指定します。<br>次のいずれかの値を入力します。<br>- true<br>- false |
| DynamoDB                   | String  |    | Kinesis ソースデータのチェックポイントの詳細を保存する Amazon DynamoDB テーブル名。<br>128 文字までの文字列を入力できます。                |
| Stream                     | String  | ○  | データの読み取り元の Kinesis Stream の名前。<br>128 文字までの文字列を入力します。<br>ソースノードに表示されます。                       |
| Stream Name/<br>Expression | String  | ○  | データの書き込み先の Kinesis Stream 名または正規表現。<br>128 文字までの文字列を入力します。<br>ターゲットノードに表示されます。                |

## POST 要求の例

Amazon Kinesis Streams ソースと Amazon Kinesis Streams ターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```

{
  "name": "kinesis to kinesis",
  "description": "kinesis to kinesis",
  "runtimeId": "010000250000000000003",
  "locationId": "5sJQJDyJyWLLrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {

```

```

    "name": "kinesis to kinesis_source",
    "type": "source",
    "connectionId": "012MGS0B00000000000F",
    "transformationType": "",
    "config": [
      {
        "key": "appendGUID",
        "value": true
      },
      {
        "key": "dynamoDB",
        "value": "table"
      },
      {
        "key": "Stream",
        "value": "test"
      }
    ]
  },
  {
    "name": "kinesis to kinesis_target",
    "type": "target",
    "connectionId": "012MGS0B00000000000F",
    "transformationType": "",
    "config": [
      {
        "key": "Stream Name/Expression",
        "value": "trgt"
      }
    ]
  }
],
"edges": [
  {
    "from": "kinesis to kinesis_source",
    "to": "kinesis to kinesis_target"
  }
]
}

```

## フラットファイルをソースおよびターゲットとするタスクの接続構成

タスクのソース接続とターゲット接続がフラットファイルの場合、ソース接続とターゲット接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー              | タイプ    | 必須 | 説明  |
|-----------------|--------|----|---|
| File            | String | ○  | ソースファイルの絶対パスおよび名前。複数ファイルモードのベースディレクトリを入力します。  |
| initialPosition | String | ○  | テールするファイルからデータを読み取る際の開始位置。次のいずれかの値を入力します。 <ul style="list-style-type: none"> <li>- ファイルの先頭。ファイルの先頭から読み取ります。すでにロールオーバーされているデータは取り込みません。</li> <li>- 現在の時刻。ファイルの最後に更新された部分から読み取ります。ロールオーバーされたデータまたは書き込まれたファイル内のデータは取り込みません。</li> </ul> |

| キー               | タイプ     | 必須 | 説明  |
|------------------|---------|----|---|
| rolloverPattern  | String  | -  | <p>ロールオーバーするファイルのファイル名パターン。</p> <p>テールするファイルがロールオーバーした場合、Secure Agent はファイル名パターンを使用して、ロールオーバーしたファイルを識別します。ファイルのロールオーバー中に Secure Agent が停止した場合、再起動時に、中断された箇所からファイルを取得します。</p> <p>アスタリスク (*) と疑問符 (?) をワイルドカード文字として使用し、ファイルが同じディレクトリにロールオーバーされることを示すことができます。例: \${filename}.log.*を入力します。ここで、アスタリスク (*) は、ファイル名に追加される連続するバージョン番号を表します。</p> |
| tailingMode      | String  | ○  | <p>ロギングパターンに基づいて、1 つまたは複数のファイルをテールします。次のいずれかの値を入力します。</p> <ul style="list-style-type: none"> <li>- 単一ファイル。1 つのファイルをテールします。</li> <li>- 複数ファイル。ベースディレクトリに示されているすべてのファイルをテールします。正規表現を入力して、テールするファイルを示すことができます。</li> </ul>  |
| File Name        | String  | ○  | ターゲットファイルの名前。   |
| interimDirectory | String  | ○  | Secure Agent 上のステージングディレクトリへのパス。  |
| rolloverSize     | Integer | ○  | <p>タスクがファイルをステージングディレクトリからターゲットに移動する際のファイルサイズ (KB)。</p> <p>1 から 2147483647 までの数値を入力できます。</p>  |
| rolloverEvents   | Integer | ○  | <p>ファイルのロールオーバーの前に蓄積するイベントまたはメッセージの数。</p> <p>1 から 2147483647 までの数値を入力できます。</p>  |
| rolloverTime     | Integer | -  | <p>ターゲットファイルがロールオーバーするまでの時間 (ミリ秒単位)。</p> <p>1 から 2147483647 までの数値を入力できます。</p>   |
| edges            | Array   | -  | データフロー実行のシーケンス。   |

## POST 要求の例

フラットファイルソースとフラットファイルターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```
{
  "name": "FileToFile",
  "description": "FileToFile_V2",
  "runtimeId": "01000025000000000003",
  "locationId": "5sJ0JDyJyWLLrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "FileToFile_source",
      "type": "source",
      "connectionId": "0100000B00000000000002",
      "transformationType": "",
      "config": [
        {
          "key": "File",
          "value": "siagent.log"
        }
      ]
    }
  ],
}
```

```

    {
      "key": "initialPosition",
      "value": "Current Time"
    },
    {
      "key": "rolloverPattern",
      "value": ""
    },
    {
      "key": "tailingMode",
      "value": "Single file"
    }
  ]
},
{
  "name": "FileToFile_target",
  "type": "target",
  "connectionId": "0100000B0000000000002",
  "transformationType": "",
  "config": [
    {
      "key": "File Name",
      "value": "testing.log"
    },
    {
      "key": "interimDirectory",
      "value": "/home/agent/infa/test_file_target"
    },
    {
      "key": "rolloverSize",
      "value": 100
    },
    {
      "key": "rolloverEvents",
      "value": 100
    },
    {
      "key": "rolloverTime",
      "value": 100
    }
  ]
}
],
"edges": [
  {
    "from": "FileToFile_source",
    "to": "FileToFile_target"
  }
],
"runtimeOptions": {
  "maxLogSize": {
    "value": 10,
    "unit": "MB"
  },
  "logLevel": "INFO"
}
}

```

## Kafka をソースおよびターゲットとするタスクの接続構成

タスクのソース接続とターゲット接続が Kafka の場合、ソース接続とターゲット接続の構成配列に次のフィールドおよびキーと値のペアを含めることができます。

| キー                        | タイプ     | 必須 | 説明   |
|---------------------------|---------|----|--|
| Topic                     | String  | ○  | イベントの読み取り元となる Kafka ソーストピック名、または Java でサポートされている Kafka ソーストピック名パターンの正規表現。<br>249 文字までの文字列を入力します。           |
| consumerProperties        | String  | -  | オプションのコンシューマ設定プロパティのカンマ区切りリストを指定します。キーと値のペアとして値を指定します。例: key1=value1, key2=value2<br>4000 文字までの文字列を入力できます。 |
| producerProperties        | String  | -  | プロデューサの設定プロパティ。<br>カンマ区切りリストを指定し、値をキーと値のペアとして指定します。<br>4000 文字までの文字列を入力できます。                               |
| mdFetchTimeout            | Integer | -  | それ以降にメタデータが取得されなくなる時間。<br>1〜2147483647 の範囲で値を入力してください。   |
| batchSize                 | Integer | -  | ストリーミング取り込みタスクがそれ以降なターゲットにデータを書き込むイベントのバッチサイズ。<br>1〜2147483647 の範囲で値を入力してください。                             |
| Topic Name/<br>Expression | String  | ○  | Kafka トピック名または Java がサポートする Kafka トピック名パターンの正規表現。<br>249 文字までの文字列を入力できます。                                  |

### POST 要求の例

Kafka ソースと Kafka ターゲットを使用するストリーミング統合タスクを更新するには、次の例のような要求を送信できます。

```
{
  "name": "kafka to kafka",
  "description": "kafka to kafka",
  "runtimeId": "0100002500000000000003",
  "locationId": "5sJ0JDyJyWLLrosS5qJjsQ",
  "currentVersion": "2",
  "messageFormat": "binary",
  "nodes": [
    {
      "name": "kafka to kafka_source",
      "type": "source",
      "connectionId": "012MGS0B0000000000002",
      "transformationType": "",
      "config": [
        {
          "key": "consumerProperties",
          "value": "key=value"
        },
        {
          "key": "Topic",
          "value": "test"
        }
      ]
    }
  ]
}
```

```

    ],
    {
      "name": "kafka to kafka_target",
      "type": "target",
      "connectionId": "012MGS0B0000000000002",
      "transformationType": "",
      "config": [
        {
          "key": "producerProperties",
          "value": "key=value"
        },
        {
          "key": "mdFetchTimeout",
          "value": 5000
        },
        {
          "key": "batchSize",
          "value": 1048576
        },
        {
          "key": "Topic Name/Expression",
          "value": "test"
        }
      ]
    }
  ],
  "edges": [
    {
      "from": "kafka to kafka_source",
      "to": "kafka to kafka_target"
    }
  ]
}

```

## POST 応答

REST API はアクションの実行に成功すると、200 または 201 成功応答が返されます。REST API でエラーが発生すると、適切なエラーコードが返されます。

要求が成功した場合、応答は次のフィールドを返します。

| フィールド          | タイプ    | 説明                    |
|----------------|--------|-----------------------|
| name           | String | タスクの名前。               |
| description    | String | ある場合は、タスクの説明。         |
| runtimeId      | String | ランタイム環境の ID。          |
| currentVersion | String | 最新のデータフローオブジェクトバージョン。 |
| nodes          | Array  | タスクのソース接続とターゲット接続の詳細。 |



## nodes 配列のフィールド

応答には、nodes 配列の以下のフィールドを含めることができます。

| フィールド              | タイプ    | 説明   |
|--------------------|--------|--|
| name               | String | 接続の名前。   |
| type               | String | 接続タイプ。   |
| connectionId       | String | 接続の ID。  |
| TRANSFORMATIONTYPE | String | トランスフォーメーションのタイプ。  |
| config             | String | キーと値のペアであるソース接続とターゲット接続の構成。配列内のキーは、ソース接続とターゲット接続のタイプによって異なります。 |

要求が失敗すると、応答にエラーの理由が含まれます。

## MQTT をソースとする場合の構成配列内の構成情報

要求が成功した場合、応答は次のフィールドを返します。

| キー           | タイプ     | 説明  |
|--------------|---------|---|
| ClientID     | String  | MQTT ソースと MQTT ブローカ間の接続を識別する一意の識別子。クライアント ID は、MQTT ソースがメッセージの処理中にメッセージを格納するために使用するファイルベースの永続ストアです。 |
| MaxQueueSize | Integer | プロセッサがメモリに保存できるメッセージの最大数。   |
| Topic        | String  | MQTT トピックの名前。   |

要求が失敗すると、応答にエラーの理由が含まれます。

## POST 応答の例

要求が成功した場合、Success ノードで次の例のような応答を受信することがあります。

```
{
  "Success": {
    "name": "mqtt to flatfile",
    "description": "mqtt to flatfile",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWllrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "mqtt to flatfile_source",
        "type": "source",
        "connectionId": "012MGS0B000000000010",
        "transformationType": "",
        "config": [
          {
            "key": "ClientID",
            "value": "test"
          },
          {
            "key": "MaxQueueSize",
            "value": 1024
          }
        ]
      }
    ]
  }
}
```

```

    },
    {
      "key": "Topic",
      "value": "test"
    }
  ]
},
{
  "name": "mqtt to flatfile_target",
  "type": "target",
  "connectionId": "012MGS0B000000000002N",
  "transformationType": "",
  "config": [
    {
      "key": "interimDirectory",
      "value": "/home/agent/test"
    },
    {
      "key": "rolloverSize",
      "value": 1024
    },
    {
      "key": "rolloverEvents",
      "value": 100
    },
    {
      "key": "rolloverTime",
      "value": 300000
    },
    {
      "key": "File Name",
      "value": "test"
    }
  ]
}
],
"edges": [
  {
    "from": "mqtt to flatfile_source",
    "to": "mqtt to flatfile_target"
  }
]
}
}

```

## JMS をソースとする場合の構成配列内の構成情報

応答は、要求で入力したフィールドのみを返します。

要求が成功した場合、応答は次のフィールドを返します。

| キー                  | タイプ    | 説明  |
|---------------------|--------|---|
| destinationType     | String | ソースサービスが JMS メッセージを送信する接続先のタイプ。   |
| clientId            | String | JMS 接続の一意の ID。  |
| sharedSubscription  | String | 複数のコンシューマが単一のサブスクリプションにアクセスできるようにします。トピックの接続先タイプに適用されます。  |
| durableSubscription | String | JMS ソースサービスは、非アクティブなサブスクライバがメッセージを保持し、サブスクライバが再接続したときにそれらのメッセージを配信できるようにします。トピックの接続先タイプに適用されます。 |

| キー               | タイプ    | 説明  |
|------------------|--------|---|
| subscriptionName | String | サブスクリプションの名前。トピックサブスクリプションタイプが共有、継続、またはその両方である場合に、トピックの接続先タイプに適用されます。 |
| JMS Destination  | String | JMS プロバイダがメッセージを配信するキューまたはトピックの名前。                                    |

要求が失敗すると、応答にエラーの理由が含まれます。

## POST 応答の例

要求が成功した場合、Success ノードで次の例のような応答を受信することがあります。

```
{
  "Success": {
    "name": "crud",
    "description": "JMS to FileToFile",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJQJDyJyWllrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "crud_source",
        "type": "source",
        "connectionId": "012MGSQB0000000000003",
        "transformationType": "",
        "config": [
          {
            "key": "destinationType",
            "value": "QUEUE"
          },
          {
            "key": "clientId",
            "value": ""
          },
          {
            "key": "JMS Destination",
            "value": "test"
          }
        ]
      },
      {
        "name": "crud_target",
        "type": "target",
        "connectionId": "012MGSQB0000000000000H",
        "transformationType": "",
        "config": [
          {
            "key": "interimDirectory",
            "value": "/home/agent/test"
          },
          {
            "key": "rolloverSize",
            "value": 1024
          },
          {
            "key": "rolloverEvents",
            "value": 100
          },
          {
            "key": "rolloverTime",
            "value": 300000
          },
          {
            "key": "File Name",

```

```

    "value": "test"
  }
]
},
"edges": [
  {
    "from": "crud_source",
    "to": "crud_target"
  }
]
}
}
}

```

## ADLS Gen2 をターゲットとする場合の構成配列内の構成情報

応答は、要求で入力したフィールドのみを返します。

要求が成功した場合、応答は次のフィールドを返します。

| キー                                | タイプ     | 説明  |
|-----------------------------------|---------|---|
| writeStrategy                     | String  | ADLS Gen2 ストレージにファイルが存在する場合に実行するアクション。  |
| rolloverSize *                    | Integer | ロールオーバーをトリガする際のターゲットファイルサイズ (KB)。ロールオーバー書き込みストラテジに適用されます。                             |
| rolloverEvents *                  | Integer | ロールオーバーの前に蓄積するイベントまたはメッセージの数。ロールオーバー書き込みストラテジに適用されます。                                 |
| rolloverTime *                    | Integer | ロールオーバーをトリガするまでの時間 (ミリ秒単位)。ロールオーバー書き込みストラテジに適用されます。                                   |
| filesystemNameOverride            | String  | 接続で提供されたデフォルトのファイルシステム名をオーバーライドします。このファイルシステム名は、実行時にファイルに書き込むために使用されます。               |
| directoryOverride                 | String  | デフォルトのディレクトリパスをオーバーライドします。データの書き込み先の ADLS Gen2 ディレクトリパス。空白の場合は、デフォルトのディレクトリパスが使用されます。 |
| compressionFormat                 | String  | ストリーミング取り込みタスクがターゲットファイルにデータを書き込む前に使用する圧縮形式。  |
| File Name/Expression              | String  | ADLS Gen2 ファイル名または正規表現。   |
| * これらのフィールドの少なくとも 1 つに値を入力してください。 |         |   |

要求が失敗すると、応答にエラーの理由が含まれます。

## POST 応答の例

要求が成功した場合、Success ノードで次の例のような応答を受信することがあります。

```

{
  "Success": {
    "name": "flatfile to adls",
    "description": "flatfile to adls",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJQJdyJyWLLrosS5qJjsQ",
    "currentVersion": "2",
  }
}

```

```

"messageFormat": "binary",
"nodes": [
  {
    "name": "flatfile to adls_source",
    "type": "source",
    "connectionId": "012MGSQB00000000002N",
    "transformationType": "",
    "config": [
      {
        "key": "File",
        "value": "logfile"
      },
      {
        "key": "initialPosition",
        "value": "Current Time"
      },
      {
        "key": "rolloverPattern",
        "value": "test"
      },
      {
        "key": "tailingMode",
        "value": "Single file"
      }
    ]
  },
  {
    "name": "flatfile to adls_target",
    "type": "target",
    "connectionId": "012MGSQB00000000003D",
    "transformationType": "",
    "config": [
      {
        "key": "writeStrategy",
        "value": "Rollover"
      },
      {
        "key": "filesystemNameOverride",
        "value": "test"
      },
      {
        "key": "File Name/Expression",
        "value": "test"
      },
      {
        "key": "compressionFormat",
        "value": "NONE"
      },
      {
        "key": "directoryOverride",
        "value": "/test"
      },
      {
        "key": "interimDirectory",
        "value": "/home/agent/test"
      },
      {
        "key": "rolloverSize",
        "value": 1024
      },
      {
        "key": "rolloverEvents",
        "value": 100
      },
      {
        "key": "rolloverTime",
        "value": 300000
      }
    ]
  }
]
}
]

```

```
}  
}
```

## Amazon S3 をターゲットとする場合の構成配列内の構成情報

応答は、要求で入力したフィールドのみを返します。

要求が成功した場合、応答は次のフィールドを返します。

| キー                       | タイプ     | 説明   |
|--------------------------|---------|--|
| partitionTime            | String  | ストリーミング取り込みタスクが Amazon S3 バケットにパーティションを作成する際に従う時間間隔。   |
| minUploadPartSize        | Integer | 複数の独立したパートセットとしてサイズの大きなファイルをアップロードする場合の最小パートサイズ (MB 単位)。このプロパティを使用して、ファイルのロードを Amazon S3 に合わせます。 |
| multipartUploadThreshold | Integer | オブジェクトを複数のパートで並行してアップロードする場合のマルチパートしきい値。   |
| Object Name/Expression   | String  | Amazon S3 ターゲットファイル名、または Amazon S3 ファイル名パターンの正規表現。   |

要求が失敗すると、応答にエラーの理由が含まれます。

### POST 応答の例

要求が成功した場合、Success ノードで次の例のような応答を受信することがあります。

```
{  
  "Success": {  
    "name": "flatfile to amazon S3",  
    "description": "flatfile to amazon S3",  
    "runtimeId": "01000025000000000003",  
    "locationId": "5sJ0JDyJyWllrosS5qJjsQ",  
    "currentVersion": "2",  
    "messageFormat": "binary",  
    "nodes": [  
      {  
        "name": "flatfile to amazon S3_source",  
        "type": "source",  
        "connectionId": "012MGS0B00000000002N",  
        "transformationType": "",  
        "config": [  
          {  
            "key": "File",  
            "value": "logfile"  
          },  
          {  
            "key": "initialPosition",  
            "value": "Current Time"  
          },  
          {  
            "key": "rolloverPattern",  
            "value": "test"  
          },  
          {  
            "key": "tailingMode",  
            "value": "Single file"  
          }  
        ]  
      },  
      {  
        "name": "flatfile to amazon S3_target",
```

```

    "type": "target",
    "connectionId": "012MGS0B0000000000I7",
    "transformationType": "",
    "config": [
      {
        "key": "partitionTime",
        "value": "None"
      },
      {
        "key": "minUploadPartSize",
        "value": 5120
      },
      {
        "key": "multipartUploadThreshold",
        "value": 5120
      },
      {
        "key": "Object_Name/Expression",
        "value": "test"
      }
    ]
  },
  "edges": [
    {
      "from": "flatfile to amazon S3_source",
      "to": "flatfile to amazon S3_target"
    }
  ]
}

```

## Azure Event Hub をターゲットとする場合の構成配列内の構成情報

応答は、要求で入力したフィールドのみを返します。

要求が成功した場合、応答は次のフィールドを返します。

| キー                  | タイプ    | 説明                                |
|---------------------|--------|-----------------------------------|
| sasPolicyName       | String | Event Hub 名前空間共有アクセスポリシーの名前。      |
| sasPolicyPrimaryKey | String | Event Hub 名前空間共有アクセスポリシーのプライマリキー。 |
| Event Hub           | String | Azure Event Hubs の名前。             |

要求が失敗すると、応答にエラーの理由が含まれます。

## POST 応答の例

要求が成功した場合、Success ノードで次の例のような応答を受信することがあります。

```

{
  "Success": {
    "name": "flatfile to azure event hub",
    "description": "flatfile to azure event hub",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWLLrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "flatfile to azure event hub_source",
        "type": "source",
        "connectionId": "012MGS0B000000000002N",
        "transformationType": "",

```

```

    "config": [
      {
        "key": "File",
        "value": "logfile"
      },
      {
        "key": "initialPosition",
        "value": "Current Time"
      },
      {
        "key": "rolloverPattern",
        "value": "test"
      },
      {
        "key": "tailingMode",
        "value": "Single file"
      }
    ]
  },
  {
    "name": "flatfile to azure event hub_target",
    "type": "target",
    "connectionId": "012MGS0B000000000001S",
    "transformationType": "",
    "config": [
      {
        "key": "sasPolicyName",
        "value": "test"
      },
      {
        "key": "sasPolicyPrimaryKey",
        "value": "test"
      },
      {
        "key": "Event Hub",
        "value": "test"
      }
    ]
  }
],
"edges": [
  {
    "from": "flatfile to azure event hub_source",
    "to": "flatfile to azure event hub_target"
  }
]
}

```

## ターゲットとしての JDBC をターゲットとする場合の構成配列内の構成情報

応答は、要求で入力したフィールドのみを返します。

リクエスト要求が成功した場合、応答は次のフィールドを返します。

| キー         | タイプ    | 説明                       |
|------------|--------|--------------------------|
| Table Name | String | JSON 形式でデータを挿入するテーブルの名前。 |

要求が失敗すると、応答にエラーの理由が含まれます。

### POST 応答の例

要求が成功した場合、Success ノードで次の例のような応答を受信することがあります。

```

{
  "Success": {

```



```

"name": "FileFile to jdbc",
"description": "FileToFile to jdbc_target",
"runtimeId": "01000025000000000003",
"locationId": "5sJQJDyJyWLLrosS5qJjsQ",
"currentVersion": "2",
"messageFormat": "binary",
"nodes": [
  {
    "name": "flatfile to jdbc_source",
    "type": "source",
    "connectionId": "012MGSQB00000000002N",
    "transformationType": "",
    "config": [
      {
        "key": "initialPosition",
        "value": "Current Time"
      },
      {
        "key": "tailingMode",
        "value": "Single file"
      },
      {
        "key": "rolloverPattern",
        "value": "test"
      },
      {
        "key": "File",
        "value": "logfile"
      }
    ]
  },
  {
    "name": "flatfile to jdbc_target",
    "type": "target",
    "connectionId": "012MGSQB00000000000KF",
    "transformationType": "",
    "config": [
      {
        "key": "Table Name",
        "value": "table"
      }
    ]
  }
],
"edges": [
  {
    "from": "flatfile to jdbc_source",
    "to": "flatfile to jdbc_target"
  }
]
}

```

## Amazon Kinesis Streams をソースおよびターゲットとする場合の構成配列内の構成情報

応答は、要求で入力したフィールドのみを返します。

要求が成功した場合、応答は次のフィールドを返します。

| キー                         | タイプ     | 説明   |
|----------------------------|---------|--|
| appendGUID                 | Boolean | Amazon DynamoDB テーブル名にサフィックスとして GUID を追加するかどうかを指定します。  |
| DynamoDB                   | String  | Kinesis ソースデータのチェックポイントの詳細を保存する Amazon DynamoDB テーブル名。                                       |
| Stream                     | String  | データの読み取り元の Kinesis Stream の名前。<br>Amazon Kinesis Streams をソースとして使用する場合に適用されます。               |
| Stream Name/<br>Expression | String  | Kinesis ストリーム名または Kinesis ストリーム名パターンの正規表現。<br>Amazon Kinesis Streams をターゲットとして使用する場合に適用されます。 |

要求が失敗すると、応答にエラーの理由が含まれます。

## POST 応答の例

要求が成功した場合、Success ノードで次の例のような応答を受信することがあります。

```
{
  "Success": {
    "name": "kinesis to kinesis",
    "description": "kinesis to kinesis",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWlrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "kinesis to kinesis_source",
        "type": "source",
        "connectionId": "012MGS0B000000000000F",
        "transformationType": "",
        "config": [
          {
            "key": "appendGUID",
            "value": true
          },
          {
            "key": "dynamoDB",
            "value": "table"
          },
          {
            "key": "Stream",
            "value": "test"
          }
        ]
      },
      {
        "name": "kinesis to kinesis_target",
        "type": "target",
        "connectionId": "012MGS0B000000000000F",
        "transformationType": "",
        "config": [
          {
            "key": "Stream Name/Expression",
            "value": "trgt"
          }
        ]
      }
    ],
    "edges": [
```

```

    {
      "from": "kinesis to kinesis_source",
      "to": "kinesis to kinesis_target"
    }
  ]
}
}

```

## フラットファイルをソースおよびターゲットとする場合の構成配列内の構成情報

応答は、要求で入力したフィールドのみを返します。

要求が成功した場合、応答は次のフィールドを返します。

| キー               | タイプ     | 必須 | 説明   |
|------------------|---------|----|--|
| File             | String  | ○  | 読み取るソースファイルの絶対パスと名前。                             |
| initialPosition  | String  | ○  | テールするファイルからデータを読み取る際の開始位置。                       |
| rolloverPattern  | String  | -  | ロールオーバーするファイルのファイル名パターン。                         |
| tailingMode      | String  | ○  | ロギングパターンに基づいて、1 つまたは複数のファイルをテールします。              |
| File Name        | String  | ○  | ターゲットファイルの名前。                                    |
| interimDirectory | String  | ○  | Secure Agent 上のステージングディレクトリへのパス。                 |
| rolloverSize     | Integer | ○  | タスクがファイルをステージングディレクトリからターゲットに移動する際のファイルサイズ (KB)。 |
| rolloverEvents   | Integer | ○  | ファイルのロールオーバーの前に蓄積するイベントまたはメッセージの数。               |
| rolloverTime     | Integer | -  | ターゲットファイルがロールオーバーするまでの時間 (ミリ秒単位)。                |

要求が失敗すると、応答にエラーの理由が含まれます。

### POST 応答の例

要求が成功した場合、次の例のような応答を受信します。

```

{
  "Success": {
    "name": "FileToFile",
    "description": "FileToFile_V2",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJQJDyJyWllrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "FileToFile_source",
        "type": "source",
        "connectionId": "0100000B000000000002",
        "transformationType": "",
        "config": [
          {
            "key": "File",
            "value": "siagent.log"
          }
        ]
      }
    ]
  }
}

```

```

    },
    {
      "key": "initialPosition",
      "value": "Current Time"
    },
    {
      "key": "rolloverPattern",
      "value": ""
    },
    {
      "key": "tailingMode",
      "value": "Single file"
    }
  ]
},
{
  "name": "FileToFile_target",
  "type": "target",
  "connectionId": "0100000B00000000000002",
  "transformationType": "",
  "config": [
    {
      "key": "File Name",
      "value": "testing.log"
    },
    {
      "key": "interimDirectory",
      "value": "/home/agent/infa/test_file_target"
    },
    {
      "key": "rolloverSize",
      "value": 100
    },
    {
      "key": "rolloverEvents",
      "value": 100
    },
    {
      "key": "rolloverTime",
      "value": 100
    }
  ]
}
],
"edges": [
  {
    "from": "FileToFile_source",
    "to": "FileToFile_target"
  }
],
"runtimeOptions": {
  "maxLogSize": {
    "value": 10,
    "unit": "MB"
  },
  "logLevel": "INFO"
}
}
}

```

## Kafka をソースおよびターゲットとする場合の構成配列内の構成情報

応答は、要求で入力したフィールドのみを返します。

要求が成功した場合、応答は次のフィールドを返します。

| キー                    | タイプ     | 説明  |
|-----------------------|---------|---|
| Topic                 | String  | イベントの読み取り元となる Kafka ソーストピック名、または Java でサポートされている Kafka ソーストピック名パターンの正規表現。 |
| consumerProperties    | String  | オプションのコンシューマ設定プロパティのカンマ区切りリスト。  |
| producerProperties    | String  | プロデューサの設定プロパティ。   |
| mdFetchTimeout        | Integer | それ以降にメタデータが取得されなくなる時間。  |
| batchSize             | Integer | ストリーミング取り込みタスクがそれ以降なターゲットにデータを書き込むイベントのバッチサイズ。                            |
| Topic Name/Expression | String  | Kafka トピック名または Java がサポートする Kafka トピック名パターンの正規表現。                         |

要求が失敗すると、応答にエラーの理由が含まれます。

## POST 応答の例

要求が成功した場合、Success ノードで次の例のような応答を受信することがあります。

```
{
  "Success": {
    "name": "kafka to kafka",
    "description": "kafka to kafka",
    "runtimeId": "01000025000000000003",
    "locationId": "5sJ0JDyJyWllrosS5qJjsQ",
    "currentVersion": "2",
    "messageFormat": "binary",
    "nodes": [
      {
        "name": "kafka to kafka_source",
        "type": "source",
        "connectionId": "012MGS0B0000000000002",
        "transformationType": "",
        "config": [
          {
            "key": "consumerProperties",
            "value": "key=value"
          },
          {
            "key": "Topic",
            "value": "test"
          }
        ]
      },
      {
        "name": "kafka to kafka_target",
        "type": "target",
        "connectionId": "012MGS0B0000000000002",
        "transformationType": "",
        "config": [
          {
            "key": "producerProperties",
            "value": "key=value"
          },
          {
            "key": "mdFetchTimeout",
            "value": 5000
          },
          {
            "key": "batchSize",

```

```

        "value": 1048576
      },
      {
        "key": "Topic Name/Expression",
        "value": "test"
      }
    ]
  },
  "edges": [
    {
      "from": "kafka to kafka_source",
      "to": "kafka to kafka_target"
    }
  ]
}

```

## jobs リソース

jobs リソースを使用して、ストリーミング統合ジョブの詳細を取得します。

### GET 要求

ストリーミング統合ジョブの詳細を要求するには、次の URL を使用します。

<server URI>/sisvc/monitor/v1/jobs/<dataflow ID>/<run ID of the job>

### GET 要求の例

ストリーミング統合ジョブの詳細を取得するには、次の例のような要求を送信します。

```

POST https://usw1-ing.dm2-us.informaticacloud.com/sisvc/monitor/v1/jobs/1948938e-3923-4602-aba8-f122e3d66faf/42559
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: 2l0oeVx22Rujiej7yTokmT

```

### GET 応答

成功した場合はジョブオブジェクトが返され、エラーが発生した場合はエラーオブジェクトが返されます。

成功した場合、応答には、ストリーミング統合ジョブに関する次の情報が含まれます。

| パラメータ     | タイプ     | 説明  |
|-----------|---------|---|
| assetId   | String  | ストリーミング統合ジョブの ID。                                 |
| assetName | String  | ストリーミング統合ジョブの名前。                                  |
| duration  | Integer | ジョブのデプロイに要した時間。                                   |
| endTime   | Integer | ジョブのデプロイの終了時刻（UTC 時間）。                            |
| startTime | Integer | ジョブのデプロイの開始時刻（UTC 時間）。                            |
| extraData | String  | タスク ID、デプロイされたバージョン、Secure Agent グループ ID などの追加情報。 |

| パラメータ      | タイプ     | 説明   |
|------------|---------|--|
| runId      | Integer | ストリーミング統合ジョブの実行 ID。ID はデプロイのたびに変わります。  |
| orgId      | String  | ログインユーザーが属する組織の ID。  |
| runtimeEnv | String  | ストリーミング統合ジョブをデプロイした Secure Agent の ID。   |
| startedBy  | String  | ストリーミング統合タスクを作成したユーザーの名前。  |
| status     | String  | ストリーミング統合ジョブのステータス。ジョブは、次のいずれかのステータスになります。 <ul style="list-style-type: none"> <li>- デプロイ中。ジョブはデプロイ中です。</li> <li>- 稼働中。ジョブは実行中です。</li> <li>- 実行中（警告あり）。ジョブは実行中ですが警告があります。</li> <li>- 実行中（エラーあり）。ジョブは実行中ですがエラーがあります。</li> <li>- アンデプロイ済み。ジョブはアンデプロイされています。</li> <li>- 停止。ジョブが意図的に停止されました。</li> </ul> |

## GET 応答の例

ストリーミング統合ジョブの詳細を取得する要求が成功した場合、次の例のような応答を受け取ります。

```
{
  "assetId": "1948938e-3923-4602-aba8-f122e3d66faf",
  "assetName": "testmonitor",
  "assetType": "SI_DATAFLOW",
  "correlationId": null,
  "duration": 1543,
  "endTime": "2022-02-14T04:04:13.000+0000",
  "extraData": "{\"id\":\"ORwiUUb9bVwjL67dWOKjoI\",\"version\":1,\"agentGroupId\":null}",
  "location": "Default",
  "runId": 42559,
  "orgId": "2lFy0UUNnlnbjhaoT3TSqw",
  "runtimeEnv": "011ZFB2500000000000N",
  "startedBy": "siqa_new",
  "status": "Undeployed",
  "startTime": "2022-02-14T03:38:30.000+0000",
  "deployedVersion": 1
}
```

# MIJobs リソース

MIJobs リソースを使用して、使用可能なストリーミング統合ジョブのリストを取得します。

## GET 要求

使用可能なストリーミング統合ジョブのリストを要求するには、次の URL を使用します。

<server URI>/mijobmonitor/api/v1/MIJobs

URI で以下のクエリパラメータを含めることができます。

| パラメータ     | タイプ     | 必須 | 説明  |
|-----------|---------|----|---|
| \$count   | Boolean | ×  | データベース内の取り込みジョブの数を表示します。  |
| \$filter  | String  | ×  | 入力に基づいてジョブをフィルタリングします。次のいずれかのフィールドを使用してフィルタできます。 <ul style="list-style-type: none"><li>- assetName</li><li>- assetType</li><li>- startedBy</li><li>- status</li></ul> 単一または複数のフィールドを使用してジョブをフィルタリングできます。                  |
| \$orderby | String  | ×  | ジョブの順序を並べ替えます。次のフィールドを使用してジョブを並べ替えることができます。 <ul style="list-style-type: none"><li>- assetName</li><li>- assetType</li><li>- status</li><li>- runtimeEnv</li><li>- startTime</li></ul> 単一または複数のフィールドを使用してジョブを並べ替えることができます。 |
| \$skip    | Integer | ×  | 指定したストリーミング取り込みジョブの数をスキップします。例えば、最初の 5 つのストリーミング取り込みジョブをスキップしたい場合などです。指定している場合は、\$filter および \$orderby パラメータの値を検討してください。  |
| \$top     | Integer | ×  | 指定した数の上位ストリーミング取り込みジョブを表示します。例えば、上位 10 個のストリーミング取り込みジョブを確認したい場合などです。指定している場合は、\$filter および \$orderby パラメータの値を検討してください。  |

## GET 要求の例

使用可能なストリーミング統合ジョブのリストを取得するには、次の例のような要求を送信します。

```
POST https://usw1-ing.dm2-us.informaticacloud.com/mijobmonitor/api/v1/MIJobs?$count=true&$filter=(startedBy eq 'siqa_new')&$orderby=deployTime desc&$skip=0&$top=25
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID:2l0oeVx22Rujiej7yTokmT
```

## GET 応答

成功した場合は MIJobs オブジェクトが返され、エラーが発生した場合はエラーオブジェクトが返されます。

成功した場合、応答には、ストリーミング統合ジョブに関する次の情報が含まれます。

| パラメータ     | タイプ     | 説明                                    |
|-----------|---------|---------------------------------------|
| assetName | String  | ストリーミング統合ジョブの名前。                      |
| runId     | Integer | ストリーミング統合ジョブの実行 ID。ID はデプロイのたびに変わります。 |



| パラメータ        | タイプ     | 説明   |
|--------------|---------|--|
| orgId        | String  | ログインユーザーが属する組織の ID。  |
| runtimeEnv   | String  | ストリーミング統合ジョブをデプロイした Secure Agent の ID。   |
| startTime    | Integer | ジョブのデプロイの開始日時 (UTC 時間)。  |
| endTime      | Integer | ジョブのデプロイの終了日時 (UTC 時間)。  |
| deployTime   | Integer | ジョブのデプロイ日時 (UTC 時間)。   |
| undeployTime | Integer | ジョブのデプロイ解除日時 (UTC 時間)。   |
| startedBy    | Integer | ストリーミング統合タスクを作成したユーザーの名前。  |
| status       | String  | ストリーミング統合ジョブのステータス。ジョブは、次のいずれかのステータスになります。 <ul style="list-style-type: none"> <li>- デプロイ中。ジョブはデプロイ中です。</li> <li>- 稼働中。ジョブは実行中です。</li> <li>- 実行中 (警告あり)。ジョブは実行中ですが警告があります。</li> <li>- 実行中 (エラーあり)。ジョブは実行中ですがエラーがあります。</li> <li>- アンデプロイ済み。ジョブはアンデプロイされています。</li> <li>- 停止。ジョブが意図的に停止されました。</li> </ul> |
| extraData    | String  | タスク ID、ストリーミング取り込みジョブの場所、Secure Agent ID などの追加情報。  |

## GET 応答の例

使用可能なストリーミング統合ジョブのリストを取得する要求が成功した場合、次の例のような応答を受け取ります。

```
{
  "@odata.context": "$metadata#Collection(OData.MI.JobMonitor.MIJob)",
  "@odata.count": 421,
  "value": [
    {
      "assetId": "7ce6bbc7-f0e2-4278-bd6d-d1187f4a1420",
      "assetName": "SIdeployJms",
      "assetType": "SI_DATAFLOW",
      "runId": 33015,
      "duration": 300000,
      "orgId": "1Pm6cSfPcAqfgeV57Fn3u4",
      "runtimeEnv": "011U5M08000000000003",
      "startTime": "2021-04-29T13:09:48.000+0000",
      "endTime": "2021-04-29T13:14:48.000+0000",
      "deployTime": "2021-04-29T13:09:48.000+0000",
      "undeployTime": "2021-04-29T13:14:48.000+0000",
      "startedBy": "siga_new",
      "status": "Undeployed",
      "outOfSync": true,
      "extraData": "{\\taskId\\:\\7Z4ZZjXc9QViT4t2okiHuz\\\",\\runtimeEnv\\:\\011U5M25000000000002\\\",\\location\\:\\RestAutomation\\\"}",
      "deployedVersion": 1,
      "replace": null,
      "lastUpdateTime": 0
    },
    {
      "assetId": "a03b9aa1-4a4a-47ee-808d-ddc0ee7b3a4a",
      "assetName": "kafka to kafka test",
      "assetType": "SI_DATAFLOW",
      "runId": 33527,

```

```

        "duration": 204988000,
        "orgId": "1Pm6cSfPcAqfgeV57Fn3u4",
        "runtimeEnv": "011U5M08000000000002",
        "startTime": "2021-05-04T05:41:39.000+0000",
        "endTime": "2021-05-06T14:38:07.000+0000",
        "deployTime": "2021-05-04T05:41:39.000+0000",
        "undeployTime": "2021-05-06T14:38:07.000+0000",
        "startedBy": "siga_new",
        "status": "Undeployed",
        "outOfSync": true,
        "extraData": "{\"taskId\":\"8V21nib7Sgjiw3QoDRi5uK\",\"runtimeEnv\": \"011U5M250000000000002\", \"location\":\"Default\"}",
        "deployedVersion": 1,
        "replace": null,
        "lastUpdateTime": 0
    }
}

```

## status リソース

status リソースを使用して、ストリーミング統合ジョブのステータスを取得します。

### GET 要求

ストリーミング統合ジョブのステータスを要求するには、次の URL を使用します。

<server URI>/sisvc/monitor/v1/status/dataflows/<dataflow ID>

### GET 要求の例

ストリーミング統合ジョブのステータスを取得するには、次の例のような要求を送信します。

```

POST https://usw1-ing.dm2-us.informaticacloud.com/sisvc/monitor/v1/status/dataflows/1948938e-3923-4602-aba8-f122e3d66faf
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: 2l0oeVx22Rujiej7yTokmT

```

### GET 応答

成功した場合はジョブステータスオブジェクトが返され、エラーが発生した場合はエラーオブジェクトが返されます。

成功した場合、応答には、ストリーミング統合ジョブのステータスに関する次の情報が含まれます。

| パラメータ        | タイプ     | 説明                |
|--------------|---------|-------------------|
| dataflowName | String  | ストリーミング統合ジョブの名前。  |
| dataflowId   | Integer | ストリーミング統合ジョブの ID。 |

| パラメータ     | タイプ     | 説明   |
|-----------|---------|--|
| status    | String  | ストリーミング統合ジョブのステータス。ジョブは、次のいずれかのステータスになります。 <ul style="list-style-type: none"> <li>- デプロイ中。ジョブはデプロイ中です。</li> <li>- 稼働中。ジョブは実行中です。</li> <li>- 実行中（警告あり）。ジョブは実行中ですが警告があります。</li> <li>- 実行中（エラーあり）。ジョブは実行中ですがエラーがあります。</li> <li>- アンデプロイ済み。ジョブはアンデプロイされています。</li> <li>- 停止。ジョブが意図的に停止されました。</li> </ul> |
| timestamp | Integer | Secure Agent がストリーミング統合ジョブのステータスを記録する時間（ミリ秒単位）。  |
| reports   | Array   | 各ノードのステータスの詳細。   |
| graph     | String  | ジョブのソースとターゲットのスループット情報。  |
| runId     | Integer | ストリーミング統合ジョブの実行 ID。ID はデプロイのたびに変わります。  |

## GET 応答の例

ストリーミング統合ジョブのステータスを取得する要求が成功した場合、次の例のような応答を受け取ります。

```
{
  "dataflowName": "testmonitor",
  "dataflowId": "1948938e-3923-4602-aba8-f122e3d66faf",
  "status": "Running",
  "timestamp": 1644839755000,
  "reports": [
    {
      "name": "testmonitor_testmonitor_source",
      "id": "a5684428-f41f-4d24-b73f-33c232314a91",
      "status": "Running",
      "timestamp": 1644839756000,
      "message": null
    },
    {
      "name": "testmonitor_testmonitor_target",
      "id": "4f59b5fb-b5b2-4b83-994b-0d3e56f67e22",
      "status": "Running",
      "timestamp": 1644839756000,
      "message": null
    }
  ],
  "graph": "{\\\"agentId\\\":\\\"011ZFB080000000000N\\\",\\\"nodes\\\":[{\\\"id\\\":\\\"a5684428-f41f-4d24-b73f-33c232314a91\\\",\\\"name\\\":\\\"testmonitor_source\\\",\\\"serviceType\\\":\\\"source\\\",\\\"config\\\":[{\\\"key\\\":\\\"nativeName\\\",\\\"value\\\":\\\"src\\\",\\\"key\\\":\\\"consumerProperties\\\",\\\"value\\\":null}],\\\"connectionId\\\":\\\"011ZFB08000000000000KJ\\\",\\\"type\\\":\\\"\\\",\\\"metaMetadata\\\":\\\"\\\",{\\\"id\\\":\\\"4f59b5fb-b5b2-4b83-994b-0d3e56f67e22\\\",\\\"name\\\":\\\"testmonitor_target\\\",\\\"serviceType\\\":\\\"target\\\",\\\"config\\\":[{\\\"key\\\":\\\"nativeName\\\",\\\"value\\\":\\\"trgt\\\",{\\\"key\\\":\\\"batchSize\\\",\\\"value\\\":\\\"1048576\\\"},{\\\"key\\\":\\\"mdFetchTimeout\\\",\\\"value\\\":\\\"5000\\\"},{\\\"key\\\":\\\"producerProperties\\\",\\\"value\\\":null}],\\\"connectionId\\\":\\\"011ZFB08000000000000KJ\\\",\\\"type\\\":\\\"\\\",\\\"metaMetadata\\\":\\\"\\\"}],\\\"edges\\\":[{\\\"id\\\":\\\"6ae185ea-7e6e-4bf6-bd9e-0be5ef3a8e78\\\",\\\"name\\\":\\\"testmonitor_source_testmonitor_target\\\",\\\"from\\\":\\\"testmonitor_source\\\",\\\"to\\\":\\\"testmonitor_target\\\",\\\"type\\\":\\\"success\\\",\\\"config\\\":[],\\\"metaMetadata\\\":\\\"\\\"}],\\\"runtimeOptions\\\":null}\\\",\\\"version\\\":1,\\\"runId\\\":42563}
```

## statistic リソース

statistics リソースを使用して、ストリーミング統合ジョブの統計を取得します。

統計を表示するには、ストリーミング取り込みジョブが次のいずれかのステータスになっている必要があります。

- デプロイ中
- 稼働中
- 実行中(警告あり)
- 実行中(エラーあり)
- 停止

### GET 要求

ストリーミング統合ジョブの統計を要求するには、次の URL を使用します。

```
<server URI>/sisvc/monitor/v1/statistics/dataflows/<dataflow ID>
```

URI で以下のクエリパラメータを含めることができます。

| パラメータ     | タイプ     | 必須 | 説明  |
|-----------|---------|----|---|
| intervals | Integer | ○  | ストリーミング取り込みジョブの統計を表示する時間（秒単位）。例えば、30 秒を指定すると、応答は過去 30 秒間のジョブ統計を表示します。 |
| overall   | Boolean | ×  | ジョブがデプロイされた時点からの統計を表示します。   |

### GET 要求の例

ストリーミング統合ジョブの統計を要求するには、次の例のような要求を送信します。

```
POST https://usw1-ing.dm2-us.informaticacloud.com/sisvc/monitor/v1/statistics/dataflows/7f1daca9-3983-4677-930f-a9529802c56b?intervals=30&overall=true
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: 2l0oeVx22Rujiej7yTokmT
```

### GET 応答

成功した場合は統計オブジェクトが返され、エラーが発生した場合はエラーオブジェクトが返されます。

成功した場合、応答には、ストリーミング統合ジョブの統計に関する次の情報が含まれます。

| パラメータ         | タイプ     | 説明  |
|---------------|---------|---|
| dataflowId    | String  | ストリーミング統合ジョブの ID。   |
| dataflowRunId | Integer | ストリーミング統合ジョブの実行 ID。   |
| startTime     | Integer | ストリーミング取り込みジョブの開始時刻（ミリ秒単位）。   |
| stopTime      | Integer | ストリーミング取り込みジョブの停止時刻（ミリ秒単位）。   |
| inMessages    | Integer | ノードに到着するメッセージの数。ノードは、ストリーミング取り込みタスクで使用されるソース、トランスフォーメーション、またはターゲットです。ソースノードの値はゼロです。 |

| パラメータ       | タイプ     | 説明  |
|-------------|---------|---|
| outMessages | Integer | ノードから転送されるメッセージの数。<br>ターゲットノードの値はゼロです。    |
| inBytes     | Integer | 受信メッセージの合計サイズ（バイト単位）。<br>ソースノードの値はゼロです。   |
| outBytes    | Integer | 送信メッセージの合計サイズ（バイト単位）。<br>ターゲットノードの値はゼロです。 |
| nodes       | Array   | タスクで使用されるソースとターゲットのストリーミングデータに関する情報。      |
| intervals   | Integer | リクエストで指定した時間間隔でのジョブの統計。間隔を設定するときに適用されます。  |

## GET 応答の例

ストリーミング統合ジョブの統計を取得する要求が成功した場合、次の例のような応答を受け取ります。

```
{
  "dataflowId": "7f1daca9-3983-4677-930f-a9529802c56b",
  "dataflowName": "newnew",
  "dataflowVersion": 1,
  "dataflowRunId": 54231,
  "snapshotCount": 171,
  "overall": {
    "dataflowId": "7f1daca9-3983-4677-930f-a9529802c56b",
    "dataflowName": "newnew",
    "dataflowVersion": 1,
    "dataflowRunId": 54231,
    "traits": {},
    "interval": 6007,
    "startTime": 1646649995000,
    "stopTime": 1646656000000,
    "nodes": [
      {
        "name": "newnew_newnew_source",
        "id": "17a51cdf-1f27-481e-81b8-d2e8ff60ec28",
        "inMessages": 0,
        "outMessages": 0,
        "inBytes": 0,
        "outBytes": 0,
        "nodeType": "Unknown"
      },
      {
        "name": "newnew_newnew_target",
        "id": "c30d6db4-6a3b-40d3-adfb-88779a972098",
        "inMessages": 0,
        "outMessages": 0,
        "inBytes": 0,
        "outBytes": 0,
        "nodeType": "Unknown"
      }
    ]
  },
  "intervals": {
    "30": {
      "dataflowId": "7f1daca9-3983-4677-930f-a9529802c56b",
      "dataflowName": "newnew",
      "dataflowVersion": null,
      "dataflowRunId": 54231,
      "traits": {},
      "interval": 30,
      "startTime": 1646655972683,

```

```

        "stopTime": 1646656002683,
        "nodes": []
      }
    }
  }
}

```

## history リソース

history リソースを使用して、ストリーミング統合ジョブの履歴を取得します。

### GET 要求

ストリーミング統合ジョブの履歴を要求するには、次の URL を使用します。

<server URI>/sisvc/monitor/v1/history/dataflows/<dataflow ID>

### GET 要求の例

ストリーミング統合ジョブの履歴を取得するには、次の例のような要求を送信します。

```

POST https://usw1-ing.dm2-us.informaticacloud.com/siscv/monitor/v1/history/dataflows/1948938e-3923-4602-aba8-f122e3d66faf
Content-Type: application/json
Accept: application/json
IDS-SESSION-ID: 2l0oeVx22Rujiej7yTokmT

```

### GET 応答

成功した場合はジョブ履歴オブジェクトが返され、エラーが発生した場合はエラーオブジェクトが返されます。

成功した場合、応答には、ストリーミング統合ジョブの履歴に関する次の情報が含まれます。

| パラメータ        | タイプ     | 説明                                    |
|--------------|---------|---------------------------------------|
| dataflowName | String  | ストリーミング統合ジョブの名前。                      |
| dataflowId   | Integer | ストリーミング統合ジョブの ID。                     |
| deployedAt   | Integer | ジョブのデプロイの開始時刻（UTC 時間）。                |
| undeployedAt | Integer | ジョブのデプロイ解除の終了時刻（UTC）。                 |
| runId        | Integer | ストリーミング統合ジョブの実行 ID。ID はデプロイのたびに変わります。 |

### GET 応答の例

ストリーミング統合ジョブの履歴を取得する要求が成功した場合、次の例のような応答を受け取ります。

```

[
  {
    "dataflowId": "1948938e-3923-4602-aba8-f122e3d66faf",
    "dataflowName": "testmonitor",
    "deployedAt": 1644809910000,
    "undeployedAt": 1644811453000,
    "dataflowVersion": 1,
    "runId": 42559,
    "overall": null,
    "intervals": {},
    "graph": null
  },
  {}
]

```

```
    "dataflowId": "1948938e-3923-4602-aba8-f122e3d66faf",  
    "dataflowName": "testmonitor",  
    "deployedAt": 1644811513000,  
    "undeployedAt": 1644838813000,  
    "dataflowVersion": 1,  
    "runId": 42561,  
    "overall": null,  
    "intervals": {},  
    "graph": null  
  }  
]
```

# 索引

## A

Amazon MSK  
  ターゲット [17](#)  
Apache Kafka  
  ソースのプロパティ [30](#)  
Azure Event Hub  
  ターゲット [18](#)  
Azure Event Hubs Kafka  
  ソース [8](#)

## C

Confluent Kafka  
  ソース [10](#)  
  ターゲット [17](#)

## D

Databricks Delta  
  ターゲット [14](#)  
  ターゲットプロパティ [37](#)

## G

Google Cloud Storage  
  ターゲット [15](#)  
  ターゲットのプロパティ [39](#)  
Google PubSub  
  ソース [9](#)  
  ソースのプロパティ [28](#)  
  ターゲット [16](#)  
  ターゲットプロパティ [40](#)

## J

JMS  
  ソース [9](#)

## K

Kafka  
  ソース [10](#)  
  ターゲット [17](#)

## M

Microsoft Azure Data Lake Storage Gen2  
  ターゲット [17](#)

## O

OPC UA  
  ソース [11](#)  
  ソースのプロパティ [32](#)

## R

REST API  
  history [94](#)  
  MIJobs [86](#), [87](#)  
  statistics [92](#)  
  status [90](#)  
  ストリーミング取り込みジョブのステータス [90](#)  
  ストリーミング取り込みジョブの詳細 [86](#)  
  ストリーミング取り込みジョブの統計 [92](#)  
  ストリーミング取り込みジョブの履歴 [94](#)  
  ストリーミング取り込みタスクのコピー [55](#)  
  ストリーミング取り込みタスクのデプロイ [53](#)  
  ストリーミング取り込みタスクのデプロイ解除 [54](#)  
  ストリーミング取り込みタスクの開始 [54](#)  
  ストリーミング取り込みタスクの更新 [56-59](#), [61](#), [63](#), [65-68](#), [71](#), [73](#),  
    [74](#), [76](#), [78-81](#), [83](#), [84](#)  
  ストリーミング取り込みタスクの停止 [55](#), [72](#)  
  使用可能なストリーミング取り込みジョブのリスト [87](#)

## す

ストリーミングの一括取り込み  
  使用例 [6](#)  
ストリーミングの一括取り込みタスク  
  AMQP [7](#)  
  Azure Event Hubs Kafka [7](#)  
  Google PubSub [7](#)  
  Kinesis [7](#)  
  OPC UA [7](#)  
  REST V2 [7](#)  
  ソースタイプ  
    JMS [7](#)  
    Kafka [7](#)  
    MQTT [7](#)  
    フラットファイル [7](#)  
ストリーミング取り込み  
  ターゲット  
    Amazon S3 [13](#)  
    Amazon S3 プロパティ [35](#)  
    Azure SQL データベース [17](#)  
    Google BigQuery V2 データベース [15](#)  
    JDBC V2 [17](#)  
    Kinesis Firehose [12](#)  
    フラットファイル [14](#)  
  トランスフォーメーション  
    Java トランスフォーメーション [20](#)  
    Jolt トランスフォーメーション [22](#)



ストリーミング取り込み (続く)  
 トランスフォーメーション (続く)  
 Python トランスフォーメーション [22](#)  
 再開 [51](#)  
 停止 [51](#)  
ストリーミング取り込みジョブ  
 REST API [87](#), [90](#), [92](#), [94](#)  
ストリーミング取り込みタスク  
 Microsoft Azure Data Lake Storage Gen2 ターゲットプロパティ [41](#)  
 REST API [53-59](#), [61](#), [63](#), [65-68](#), [71-74](#), [76](#), [78-81](#), [83](#), [84](#)  
 エージェントパラメータ [49](#)  
 ソース設定 [25](#)  
 ターゲットの設定 [34](#)  
 デプロイ [50](#)  
 デプロイ解除 [50](#)  
 トランスフォーメーションの設定 [43](#)  
 トランスフォーメーションの追加 [43](#), [44](#)  
 ランタイムオプション [49](#)  
 ロールオーバー [50](#)  
 ログレベル [49](#)  
 拒否ディレクトリ [49](#)  
 再デプロイ [50](#)  
 消去 [49](#)  
 前提条件 [24](#)  
 定義 [24](#)  
 電子メールアドレス [49](#)  
ストリーミング統合  
 ソース  
 Amazon MSK [10](#)  
 Azure Event Hubs Kafka [8](#)  
 Confluent Kafka [10](#)  
 Google PubSub [9](#)  
 Google PubSub のプロパティ [28](#)  
 JMS [9](#)  
 Kafka  
 Azure Event Hub  
 Azure Event Hubs ソース [26](#), [30](#)  
 名前空間 [26](#), [30](#)  
 Kafka Azure Event Hub のプロパティ [26](#)  
 Kafka のプロパティ [30](#)  
 MQTT [10](#)  
 OPC UA [11](#)  
 OPC UA のプロパティ [32](#)  
 フラットファイル [9](#)  
 ターゲット  
 Amazon MSK [17](#)  
 Azure Event Hub [18](#)  
 Confluent Kafka [17](#)  
 Databricks Delta [14](#)  
 Databricks Delta のプロパティ [37](#)  
 Google Cloud Storage [15](#)

ストリーミング統合 (続く)  
 ターゲット (続く)  
 Google Cloud Storage のプロパティ [39](#)  
 Google PubSub [16](#)  
 Google PubSub のプロパティ [40](#)  
 Kafka [17](#)  
 Microsoft Azure Data Lake Storage Gen2 [17](#)  
 データ形式  
 binary [19](#)  
 JSON [19](#)  
 XML [19](#)  
 トランスフォーメーション  
 Combiner トランスフォーメーション [19](#)  
 Splitter トランスフォーメーション [23](#)  
 フィルタトランスフォーメーション [20](#)

## そ

ソース  
 Amazon Kinesis Streams [7](#)  
 AMQP [8](#)  
ソースのプロパティ  
 MQTT  
 クライアント ID [31](#)  
 最大キューサイズ [31](#)  
 フラットファイル  
 ローリングファイル名パターン [27](#)

## た

ターゲット  
 Amazon Kinesis Streams [13](#)  
ターゲットプロパティ  
 フラットファイル  
 ローリングファイル名パターン [38](#)

## と

トランスフォーメーション  
 Combiner トランスフォーメーション [19](#)  
 Java トランスフォーメーション [20](#)  
 Jolt トランスフォーメーション [22](#)  
 Python トランスフォーメーション [22](#)  
 Splitter トランスフォーメーション [23](#)  
 フィルタトランスフォーメーション [20](#)  
 形式変換トランスフォーメーション [20](#)