Informatica® Cloud Data Integration

Microsoft Azure Data Lake Storage Gen2 コネクタ

Informatica Cloud Data Integration Microsoft Azure Data Lake Storage Gen2 コネクタ
2022 年 10 月

© 著作権 Informatica LLC 2019, 2022

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（電子的複写、写真複写、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 ロゴ、Informatica Cloud、および PowerCenter 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。

特許については、https://www.informatica.com/legal/patents.html を参照してください。

免責: 本文書は、一切の保証を伴わない「現状渡し」で提供されるものとし、Informatica LLC は他社の権利の非侵害、市場性および特定の目的への適合性の黙示の保証などを含めて、一切の明示的および黙示的保証の責任を負いません。Informatica LLC では、本ソフトウェアまたはドキュメントに誤りのないことを保証していません。本ソフトウェアまたはドキュメントに記載されている情報には、技術的に不正確な記述や誤植が含まれる場合があります。本ソフトウェアまたはドキュメントの情報は、予告なしに変更されることがあります。

NOTICES

この Informatica 製品（以下「ソフトウェア」）には、Progress Software Corporation（以下「DataDirect」）の事業子会社である DataDirect Technologies からの特定のドライバ（以下「DataDirect ドライバ」）が含まれています。DataDirect ドライバには、次の用語および条件が適用されます。

1. DataDirect ドライバは、特定物として現存するままの状態で提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。
2. DataDirect または第三者は、予見の有無を問わず発生した ODBC ドライバの使用に関するいかなる直接的、間接的、偶発的、特別、あるいは結果的損害に対して責任を負わないものとします。本制限事項は、すべての訴訟原因に適用されます。訴訟原因には、契約違反、保証違反、過失、厳格責任、詐称、その他の不法行為を含みますが、これらに限るものではありません。

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

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

発行日: 2022-12-05

目次

序文 5

Informatica のリソース 5

Informatica マニュアル 5

Informatica Intelligent Cloud Services Web サイト 5

Informatica Intelligent Cloud Services コミュニティ 5

Informatica Intelligent Cloud Services マーケットプレース 6

Data Integration コネクタのドキュメント 6

Informatica ナレッジベース 6

Informatica Intelligent Cloud Services Trust Center 6

Informatica グローバルカスタマサポート 6

第 1 章 : Microsoft Azure Data Lake Storage Gen2 コネクタの概要 7

Microsoft Azure Data Lake Storage Gen2 コネクタの例 7

Microsoft Azure Data Lake Storage Gen2 コネクタのアセット 8

サポートされるファイル形式 8

Microsoft Azure Data Lake Storage Gen2 コネクタの管理 9

マネージド ID 認証 10

アクセス制御リスト 10

第 2 章 : Microsoft Azure Data Lake Storage Gen2 の接続 12

Microsoft Azure Data Lake Storage Gen2 接続のプロパティ 12

プロキシサーバーの設定 14

Windows でのプロキシサーバーの設定 14

Linux でのプロキシサーバーの設定 15

プロキシサーバーのバイパス 16

第 3 章 : Microsoft Azure Data Lake Storage Gen2 のマッピング 17

マッピングでの Microsoft Azure Data Lake Storage Gen2 ソース 17

Microsoft Azure Data Lake Storage Gen2 ソースのディレクトリソース 20

ワイルドカード文字 20

サブディレクトリからのファイルの読み取り 21

ファイルの増分ロード 22

プッシュダウンの最適化 23

マッピングにおける Microsoft Azure Data Lake Storage Gen2 ターゲット 23

ターゲットの指定 26

ターゲットのタイムスタンプ 27

Microsoft Azure Data Lake Storage Gen2 ターゲットファイルのパラメータ化 28

ファイル形式オプション 28

固定長ファイル形式 30

ファイル名フィールド 31

ソースオブジェクトパスの読み取り. 32
ターゲットオブジェクトへの書き込み. 32
FileName フィールドに関するルールとガイドライン. 33
ディレクトリレベルのパーティショニング. 34
パーティションフォルダからの読み取りとパーティションフォルダへの書き込みに関するルールとガイドライン. 36
詳細モードのマッピングの例. 37
マッピングのルールおよびガイドライン. 39

第 4 章 : マッピングの移行. 41
移行されたマッピングへの同じオブジェクトパスの使用. 41
移行されたマッピングへの別のオブジェクトパスの使用. 41
移行オプション. 42
一般的なルールとガイドライン. 43

第 5 章 : データ型リファレンス 44
フラットファイルのデータ型とトランスフォーメーションデータ型. 44
Avro のデータ型とトランスフォーメーションデータ型. 45
JSON のデータ型とトランスフォーメーションデータ型. 46
ORC のデータ型とトランスフォーメーションデータ型. 47
Parquet のデータ型とトランスフォーメーションデータ型. 48

第 6 章 : トラブルシューティング. 51
マッピングのトラブルシューティング. 51
詳細モードのマッピングのトラブルシューティング. 52

索引. 53

序文

*Microsoft Azure Data Lake Storage Gen2 コネクタ*を使用して、Microsoft Azure Data Lake Storage Gen2 に対して読み書きを行う方法について説明します。また、Microsoft Azure Data Lake Storage Gen2 接続を作成する方法や、Cloud データ統合でマッピング、マッピングタスク、動的マッピングタスク、およびデータ転送タスクを開発および実行する方法も紹介します。

Informatica のリソース

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

Informatica マニュアル

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

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

Informatica Intelligent Cloud Services Web サイト

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

Informatica Intelligent Cloud Services コミュニティ

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

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

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

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

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

Informatica Intelligent Cloud Services マーケットプレース

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

https://marketplace.informatica.com/

Data Integration コネクタのドキュメント

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

Informatica ナレッジベース

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

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

Informatica Intelligent Cloud Services Trust Center

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

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

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

登録するには、https://status.informatica.com/に移動し、**［更新を購読登録］**をクリックします。その後、電子メール、SMS テキストメッセージ、Webhook、RSS フィードとして、またはこの 4 つを任意に組み合わせて送信された通知を受信することを選択ができます。

Informatica グローバルカスタマサポート

電話またはオンラインでカスタマサポートセンターに連絡できます。

オンラインサポートについては、Informatica Intelligent Cloud Services の**［サポート要求の送信］**をクリックしてください。またオンラインサポートを使用して問題を記録することもできます。オンラインサポートを利用するには、ログインが必要です。https://network.informatica.com/welcome でログイン要求できます。

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

第 1 章

Microsoft Azure Data Lake Storage Gen2 コネクタの概要

Microsoft Azure Data Lake Storage Gen2 コネクタを使用すると、Microsoft Azure Data Lake Storage Gen2 に対するデータの読み書きを安全に行うことができます。

Microsoft Azure Data Lake Storage Gen2 コネクタを使用すると、フラットファイルやフラット構造複合ファイル（Avro、JSON、ORC、Parquet など）を読み取ったり、書き込んだりできます。Microsoft Azure Data Lake Storage Gen2 オブジェクトをマッピングと mapping タスクのソースおよびターゲットとして使用することができます。

マッピングを詳細モードに切り替えて、高度な機能を有効にするためのトランスフォーメーションと関数を含めることができます。詳細クラスタは、Microsoft Azure またはセルフサービスクラスタでホストできます。

Microsoft Azure Data Lake Storage Gen2 コネクタの例

あなたは、大手金融企業でデータアナリストとして働いています。Azure Data Lake Analytics を使用してリスク管理、不正検出、およびその他の分析を行っています。分析を行うには、区切りデータを Microsoft Azure Data Lake Storage Gen2 に書き込む必要があります。

Microsoft Azure Data Lake Storage Gen2 コネクタを使用して、リレーショナルデータベースやトランザクションデータベース、Salesforce などの他のアプリケーションなどのソースからデータを読み取り、Microsoft Azure Data Lake Storage Gen2 にデータを書き込むための mapping タスクを作成できます。Microsoft Azure Data Lake Storage Gen2 でデータを利用できるようになったら、データ分析を実行できます。

Microsoft Azure Data Lake Storage Gen2 コネクタのアセット

データ統合でアセットを作成し、Microsoft Azure Data Lake Storage Gen2 コネクタを使用してデータを統合できます。

Microsoft Azure Data Lake Storage Gen2 コネクタを使用する際、次のデータ統合のアセットを含めることができます。

- Data transfer タスク
- Dynamic mapping タスク
- マッピング
- マッピングタスク

アセットとトランスフォーメーションの設定に関する詳細については、データ統合のドキュメントの「*マッピング*」、「*トランスフォーメーション*」、および「*タスク*」を参照してください。

サポートされるファイル形式

以下の表に、マッピングで使用できるファイル形式を示します。

ファイル形式	読み取り	書き込み	マッピング	詳細モードのマッピング
フラット	○	○	○	はい
Binary	○	○	○	いいえ
Avro（プリミティブデータ型と階層データ型）	はい	はい	○（プリミティブデータ型のみ）	はい
JSON（プリミティブデータ型と階層データ型）	はい	はい	○（プリミティブデータ型のみ）	はい
ORC（プリミティブデータ型と階層データ型）	はい	はい	○（プリミティブデータ型のみ）	はい
Parquet（プリミティブデータ型と階層データ型）	はい	はい	○（プリミティブデータ型のみ）	はい
インテリジェント構造モデル	はい	×	×	○

Microsoft Azure Data Lake Storage Gen2 コネクタの管理

タスクで Microsoft Azure Data Lake Storage Gen2 オブジェクトを使用する前に、管理者は以下のタスクを実行する必要があります。

- Microsoft Azure Data Lake Storage Gen2 で使用するストレージアカウントを作成し、Azure ポータルで**［階層名前空間］**を有効にします。
  ロールベースのアクセス制御またはアクセス制御リストを使用して、ユーザーがストレージアカウントのリソースにアクセスすることを許可できます。
  - **ロールベースのアクセス制御**
    ロールベースのアクセス制御を使用する場合は、［Contributor］ロールまたは［Reader］ロールをユーザーに割り当てます。

    ［Contributor］ロールにはストレージアカウント内のすべてのリソースを管理できるフルアクセスが付与されますが、ロールの割り当ては許可されません。

    ［Reader］ロールにはストレージアカウント内のすべてのリソースの閲覧権限が付与されますが、リソースの変更は許可されません。

    注: ロールの割り当てを追加または削除するには、［Owner］ロールなどの書き込みおよび削除権限が必要です。
  - **アクセス制御リスト**
    アクセス制御リストを使用する場合は、ユーザーに各ディレクトリとファイルへの読み取り、書き込み、および実行権限を与えることができます。
- Azure Active Directory にアプリケーションを登録して、Microsoft Azure Data LakeStorage Gen2 アカウントにアクセスするユーザーを認証します。
  ロールベースのアクセス制御またはアクセス制御リストを使用して、アプリケーションを許可できます。
  - **ロールベースのアクセス制御**
    ロールベースのアクセス制御を使用する場合は、Storage Blob Data Contributor または Storage Blob Data Reader のロールをアプリケーションに割り当てます。

    Storage Blob Data Contributor ロールでは、ストレージアカウント内の Azure Storage コンテナと Blob の読み取り、書き込み、および削除が可能です。

    Storage Blob Data Reader ロールでは、ストレージアカウント内の Azure Storage コンテナと Blob の読み取りおよび一覧表示のみが可能です。
  - **アクセス制御リスト**
    アクセス制御リストを使用する場合は、コンテナの各ディレクトリとファイルへの読み取り、書き込み、および実行権限を与えることができます。
- ストレージアカウントに Blob コンテナを作成します。
- Microsoft Azure Data Lake Storage Gen2 でのサービス間認証用に Azure Active Directory Web アプリケーションを作成します。

  注: コネクタを使用してアプリケーションで作成されたフォルダまたはファイルにアクセスするためのスーパーユーザー特権があることを確認します。
- 複合ファイルを読み書きするには、タイプ DTM の JVM オプションを設定して、Secure Agent のシステム構成の詳細で-Xms および-Xmx 値を増やし、Java ヒープ領域不足のエラーを回避します。推奨される-Xms 値は 512MB、-Xmx 値は 1024MB です。
- Microsoft Azure Data Lake Storage Gen2 に対して読み取りまたは書き込みを行う場合、ファイルのステージングに、エージェントマシンの中間ディレクトリが使用されます。
  中間ディレクトリがエージェントマシンに作成されていること、および中間ディレクトリに対する読み取り権と書き込み権があることを確認してください。

Microsoft Azure Data Lake Storage Gen2 接続の設定の詳細については、Informatica の How-To ライブラリの記事、Prerequisites to create a Microsoft Azure Data Lake Storage Gen2 connection を参照してください。

マネージド ID 認証

マネージド ID 認証を使用して Microsoft Azure Data Lake Storage Gen2 に接続する前に、次の前提条件を完了してください。

1. Azure 仮想マシンを作成する。
2. Secure Agent を Azure 仮想マシンにインストールする。
3. Azure 仮想マシンのシステム割り当て ID またはユーザー割り当て ID を有効にする。
   両方を有効にし、クライアント ID を指定しない場合、システムに割り当てられた ID が認証に使用されます。
4. マネージド ID を追加または削除した後、Azure 仮想マシンを再起動する。

マネージド ID 認証のルールおよびガイドライン

マネージド ID 認証については、次のルールおよびガイドラインを考慮します。

- Microsoft Azure Data Lake Storage Gen2 接続を作成するときは、ランタイム環境として設定した Azure 仮想マシンを選択します。
- システム割り当て ID を有効にする場合は、必要なロールまたは権限を Azure 仮想マシンに割り当てて、マッピングとタスクを実行します。
  ユーザー割り当て ID を有効にする場合は、ユーザー割り当て ID に必要なロールまたは権限を割り当てます。

  例えば、ロールベースのアクセス制御を使用する場合は、Storage Blob Data Contributor ロールを割り当て、アクセス制御リストを使用する場合は、読み取り、書き込み、および実行権限を割り当てます。
- ファイルシステム内のコンテナのパブリックアクセスレベルを［**コンテナ**］に設定します。
- マネージド ID 認証でプロキシサーバーを使用することはできません。

アクセス制御リスト

アクセス制御リストを使用して、ディレクトリとファイルにアクセスするためのさまざまなレベルの権限を各ユーザーとサービスに付与できます。ロールベースのアクセス制御を使用してストレージアカウント内のすべてのデータへのアクセスを付与したくない場合は、アクセス制御リストを使用して、特定のディレクトリまたはファイルへの読み取り、書き込み、および実行権限を付与できます。

注: ファイルへのアクセスを付与するときは、コンテナのルートフォルダと、ファイルにつながるフォルダの階層内の各フォルダへの*実行*権限を付与する必要があります。

次の表に、アクセス制御リストを使用して Microsoft Azure Data Lake Storage Gen2 にアクセスするためのマッピングを作成するときに必要な権限を示します。

アクション	権限
接続のテスト	ソースフォルダまたはターゲットフォルダの読み取りおよび実行権限。
オブジェクトへのアクセス	オブジェクトを含むソースフォルダの読み取りおよび実行権限。
ファイル形式の選択	ソースファイルの読み取り権限。

アクション	権限
プレビュー データ	ソースファイルの読み取り権限。
ソースタイプとしてのファイルまたはディレクトリからのデータの読み取り	ソースファイルの読み取り権限。
ワイルドカード文字または再帰的ディレクトリ読み取りの使用	- ワイルドカード文字または再帰読み取りを使用するフォルダの読み取りおよび実行権限。 - ソースファイルの読み取り権限。
フォルダあまたはファイルの上書き	- 上書きするターゲットファイルまたはフォルダを含むターゲットフォルダの書き込みおよび実行権限。 - 上書きするファイルまたはフォルダの読み取り、書き込み、および実行権限。
既存のファイルへのデータの追加	ターゲットファイルの読み取りおよび書き込み権限。 マネージド ID 認証を使用する場合は、ターゲットファイルに読み取り、書き込み、および実行権限を割り当てます。

第 2 章

Microsoft Azure Data Lake Storage Gen2 の接続

Microsoft Azure Data Lake Storage Gen2 接続を作成すると、Microsoft Azure Data Lake Storage Gen2 に対するデータの読み書きを安全に行うことができます。Microsoft Azure Data Lake Storage Gen2 接続は、マッピングとマッピングタスクでソースおよびターゲットを指定するために使用できます。

Microsoft Azure Data Lake Storage Gen2 接続のプロパティ

Microsoft Azure Data Lake Storage Gen2 接続をセットアップするときは、接続プロパティを設定します。

以下の表に、Microsoft Azure Data Lake Storage Gen2 接続のプロパティを示します。

プロパティ	説明
接続名	接続の名前。 各接続名は組織内で一意である必要があります。接続名には、英数字、スペース、および次の特殊文字を含めることができます。_ .+ -, 最大長は 255 文字です。
説明	接続の説明。最大長は 4000 文字です。
タイプ	Microsoft Azure Data Lake Storage Gen2 接続タイプ。
ランタイム環境	タスクを実行するランタイム環境の名前。 Secure Agent、Hosted Agent、またはサーバーレスランタイム環境を選択します。 Hosted Agent は、詳細モードのマッピングには適用されません。
アカウント名	Microsoft Azure Data Lake Storage Gen2 のアカウント名またはサービス名。

プロパティ	説明
認証タイプ	Microsoft Azure Data Lake Storage Gen2 アカウントにアクセスするための認証タイプ。 次のいずれかのオプションを選択します。 - サービスプリンシパル認証。クライアント ID、クライアントシークレット、およびテナント ID を使用して Microsoft Azure Data Lake Storage Gen2 に接続します。 - 共有キー認証。アカウントキーを使用して、Microsoft Azure Data Lake Storage Gen2 に接続します。 - マネージド ID 認証。Azure のアプリケーションに割り当てられた ID を使用して認証し、Microsoft Azure Data Lake Storage Gen2 の Azure リソースにアクセスする場合に選択します。
クライアント ID	サービスプリンシパル認証とマネージド ID 認証に適用されます。 アプリケーションのクライアント ID。 サービスプリンシパル認証を使用するには、Azure アクティブディレクトリに登録されているアプリケーションのアプリケーション ID またはクライアント ID を指定します。 マネージド ID 認証を使用するには、ユーザー割り当てマネージド ID のクライアント ID を指定します。システム割り当てマネージド ID によって権限が提供されている場合は、フィールドを空のままにします。システム割り当て ID がなく、ユーザー割り当てマネージド ID が 1 つしかない場合は、フィールドを空のままにしておくこともできます。
クライアントシークレット	サービスプリンシパル認証に適用されます。 Azure Active Directory で OAuth 認証を完了するためのクライアントシークレットキー。
テナント ID	サービスプリンシパル認証に適用されます。 Azure Active Directory のディレクトリ ID。
アカウントキー	共有キー認証に適用されます。 Microsoft Azure Data Lake Storage Gen2 アカウントのアカウントキー。
ファイルシステム名	Microsoft Azure Data Lake Storage Gen2 アカウントのファイルシステムの名前。
ディレクトリパス	ファイルシステム名を使用していない既存のディレクトリのパス。 以下のいずれかの構文を選択できます。 - / （ルートディレクトリの場合）。 - `/dir1` - `dir1/dir2` デフォルトのディレクトリはありません。
Adls Gen2 エンドポイント	Microsoft Azure エンドポイントのタイプ。 次のいずれかのエンドポイントを選択します。 - core.windows.net。Azure エンドポイントに接続します。 - core.usgovcloudapi.net。米国政府の Microsoft Azure Data Lake Storage Gen2 エンドポイントに接続します。 - core.chinacloudapi.cn。中国地域の Microsoft Azure Data Lake Storage Gen2 エンドポイントに接続します。 デフォルトは core.windows.net です **注**: 詳細モードでは、マッピング用に Azure Government エンドポイントを設定することはできません。

プロキシサーバーの設定

組織で送信プロキシサーバーを使用してインターネットに接続する場合、Secure Agent は、そのプロキシサーバー経由で Informatica Intelligent Cloud Services に接続します。
正しいプロキシ設定については、ネットワーク管理者にお問い合わせください。

Windows でのプロキシサーバーの設定

Windows マシンで Secure Agent のプロキシサーバーを設定するには、Secure Agent を使用するか、Secure Agent の JVM オプションを使用します。

Secure Agent Manager を使用したプロキシサーバーの設定

Secure Agent Manager を使用してプロキシサーバーを設定するには、次の手順を実行します。

1. **［開始］** > **［すべてのプログラム］** > **［Informatica Cloud Secure Agent］** > **［Informatica Cloud Secure Agent］** をクリックして、Secure Agent Manager を起動します。

   Secure Agent Manager に Secure Agent のステータスが表示されます。
2. ［Secure Agent Manager］ページで **［プロキシ］** をクリックします。
3. **［プロキシサーバーを使用］** をクリックして、プロキシサーバーの設定を入力します。
4. 次のプロキシサーバーの詳細を設定します。

フィールド	説明
プロキシホスト	必須。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	必須。送信プロキシサーバーのポート番号。

5. **［OK］** をクリックします。

   Secure Agent Manager によって Secure Agent が再起動され、設定が適用されます。

JVMOptions 経由でのプロキシサーバー設定の構成

1. Informatica Intelligent Cloud Services にログインします。
2. ［管理］を開いて **［ランタイム環境］** を選択します。
3. プロキシサーバーを設定する Secure Agent を選択します。
4. ページの右上隅にある **［編集］** をクリックします。
5. **［システム構成の詳細］** セクションで、データ統合サービスの **［タイプ］** に **［DTM］** を選択します。
   - **JVMOption** フィールドに次のパラメータを追加し、各パラメータに適した値を指定します。

パラメータ	説明
-Dhttp.proxyHost=	送信 HTTP プロキシサーバーのホスト名。
-Dhttp.proxyPort=	送信 HTTP プロキシサーバーのポート番号。

パラメータ	説明
-Dhttp.proxyUser=	HTTP プロキシサーバの認証ユーザ名。このオプションは、プロキシサーバーが認証を要求した場合に必要です。
-Dhttp.proxyPassword=	認証されたユーザのパスワード。本オプションは、プロキシサーバが認証を要求した場合に必要となります。
-Dhttps.proxyHost=	送信 HTTPS プロキシサーバーのホスト名。
-Dhttps.proxyPort=	送信 HTTPS プロキシサーバーのポート番号。
-Dhttps.proxyUser=	HTTPS プロキシサーバーの認証ユーザー名。このオプションは、プロキシサーバーが認証を要求した場合に必要です。
-Dhttps.proxyPassword=	認証されたユーザのパスワード。本オプションは、プロキシサーバが認証を要求した場合に必要となります。

HTTP の例。

```
JVMOption1=-Dhttp.proxyHost=<proxy_server_hostname>
JVMOption2=-Dhttp.proxyPort=8081
JVMOption3=-Dhttp.proxyUser=<proxy_user_name>
JVMOption4=-Dhttp.proxyPassword=<proxy_password>
```

HTTPS の例。

```
JVMOption1=-Dhttps.proxyHost=<proxy_server_hostname>
JVMOption2=-Dhttps.proxyPort=8081
JVMOption3=-Dhttps.proxyUser=<proxy_user_name>
JVMOption4=-Dhttps.proxyPassword=<proxy_password>
```

6. **［保存］** をクリックします。

   Secure Agent が再起動して設定が適用されます。

Linux でのプロキシサーバーの設定

Secure Agent のインストーラにより、ブラウザで設定されている設定項目に基づいて Secure Agent のプロキシサーバー設定が設定されます。Secure Agent に定義されているプロキシサーバーの設定は、コマンドラインから更新できます。

Linux マシンで Secure Agent のプロキシサーバーを設定するには、`proxy.ini` ファイルを更新するシェルコマンドを使用します。ネットワーク管理者に問い合わせて、プロキシの設定項目を決めてください。

1. 次のディレクトリに移動します。

   `<Secure Agent のインストールディレクトリ>/apps/agentcore`

2. `proxy.ini` ファイルを更新します。

   - 認証されていないプロキシの `proxy.ini` ファイルを更新するには、次のコマンドを入力します。
     `./consoleAgentManager.sh configureProxy <proxy host> <proxy port>`

3. Secure Agent を再起動します。

プロキシサーバーのバイパス

Secure Agent に設定されているプロキシサーバーの設定は、コマンドラインからバイパスできます。

プロキシサーバーをバイパスするには、次の手順を実行します。

1. 次のディレクトリに移動します。
   `<Secure Agent のインストールディレクトリ>/apps/agentcore`
2. `proxy.ini` ファイルで次のコマンドを指定します。
   `InfaAgent.NonProxyHost=localhost|{*}`**`core.windows.net`**`|127.|[\:\:1]*`
3. Secure Agent を再起動します。

第 3 章

Microsoft Azure Data Lake Storage Gen2 のマッピング

マッピングを設定するときに、ソースからターゲットへのデータフローを記述します。

マッピングによって、マッピングタスクで使用できる再利用可能なデータフローロジックを定義します。

マッピングを作成する際は、ソーストランスフォーメーションとターゲットトランスフォーメーションを定義して、Microsoft Azure Data Lake Storage Gen2 オブジェクトを表します。データ統合の Mapping Designer を使用すると、マッピングキャンバスにソーストランスフォーメーションまたはターゲットトランスフォーメーションを追加して、Microsoft Azure Data Lake Storage Gen2 のソースおよびターゲットプロパティを設定できます。

詳細モードでは、Mapping Designer で、高度な機能を有効にするためのトランスフォーメーションと関数を含むようにマッピングキャンバスが更新されます。

Monitor を使用してジョブを監視できます。

マッピングでの Microsoft Azure Data Lake Storage Gen2 ソース

マッピングでは、単一の Microsoft Azure Data Lake Storage Gen2 オブジェクトを表すようにソーストランスフォーメーションを設定できます。

以下の表に、ソーストランスフォーメーションで設定できる Microsoft Azure Data Lake Storage Gen2 ソースプロパティを示します。

プロパティ	説明
接続	ソース接続の名前。ソース接続を選択するか、**［新しいパラメータ］**をクリックして、ソース接続の新しいパラメータを定義します。 ［実行時にパラメータのオーバーライドを許可する］オプションを選択する場合は、パラメータファイルが正しい形式で指定されていることを確認してください。 パラメータ化されていない Microsoft Azure Data Lake Storage Gen2 接続とパラメータ化された Microsoft Azure Data Lake Storage Gen2 接続を切り替えると、詳細プロパティ値が保持されます。
ソースタイプ	［単一オブジェクト］または［パラメータ］を選択します。

プロパティ	説明
オブジェクト	ソースオブジェクトの名前。 ヘッダーまたはファイルデータに特殊文字が含まれていないことを確認してください。
パラメータ	ソースオブジェクトの既存のパラメータを選択するか、**［新しいパラメータ］**をクリックしてソースオブジェクトの新しいパラメータを定義します。**［パラメータ］**プロパティは、ソースタイプとして［パラメータ］を選択した場合にのみ表示されます。 ソースオブジェクトをパラメータ化するときは、パラメータのデフォルト値にファイルシステムを含む完全なオブジェクトパスを指定します。
形式	Microsoft Azure Data Lake Storage Gen2 コネクタが Microsoft Azure Data Lake Storage Gen2 からのデータの読み取りに使用するファイル形式を指定します。 次のファイル形式タイプを選択できます。 - Flat - Avro - Parquet - JSON - ORC - 構造の検出 [2] デフォルトは**［なし］**です。形式タイプとして**［なし］**を選択した場合、Microsoft Azure Data Lake Storage Gen2 コネクタはデータを Microsoft Azure Data Lake Storage Gen2 ファイルからバイナリ形式で読み取ります。 1 GB を超える JSON ファイルを読み取ることはできません。 **注**: ソースファイルが空でないことを確認してください。 詳細については、「ファイル形式オプション」 (ページ 28)を参照してください。
インテリジェント構造モデル [2]	［構造の検出］形式タイプに適用されます。インテリジェント構造モデルを選択してください。 詳細については、「*コンポーネント*」を参照してください。
[1] 詳細モードのマッピングには適用されません。 [2] 詳細モードのマッピングにのみ適用されます。	

以下の表に、Microsoft Azure Data Lake Storage Gen2 ソース詳細プロパティを示します。

プロパティ	説明
同時スレッド [1]	Microsoft Azure Data Lake Storage Gen2 からデータを抽出するための同時接続数。サイズの大きいファイルまたはオブジェクトを読み取る場合は、複数のスレッドを生成してデータを処理できます。サイズの大きいファイルを複数の小さい部分に分割するには、**［ブロックサイズ］**を設定します。 デフォルト値は 4 です。最大値は 10 です。
ファイルシステム名のオーバーライド	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。
ソースタイプ	データの読み取り元のソースのタイプを選択します。次のソースタイプを選択することができます。 - ファイル - ディレクトリ デフォルトは［ファイル］です。

プロパティ	説明
ワイルドカード文字を許可	ディレクトリソースタイプにワイルドカード文字を使用するかどうかを指定します。 詳細については、「ワイルドカード文字」 (ページ 20)を参照してください。
ディレクトリのオーバーライド	データの読み取りに使用する Microsoft Azure Data Lake Storage Gen2 ディレクトリ。デフォルトはルートディレクトリです。実行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上書きされます。 絶対ディレクトリパスまたは相対ディレクトリパスを指定できます。 - 絶対パス - Secure Agent は、指定したファイルシステムでこのディレクトリパスを検索します。 絶対パスの例: `Dir1/Dir2` - 相対パス - Secure Agent は、オブジェクトのネイティブディレクトリパスでこのディレクトリパスを検索します。 相対パスの例: `/Dir1/Dir2` 相対パスを使用すると、インポートされたオブジェクトパスが、実行時にメタデータの取得中に使用されるファイルパスへ追加されます。 ディレクトリをオーバーライドするためにルートディレクトリ（/）を指定しないでください。
ファイル名のオーバーライド	ソースオブジェクト。データの読み取り元のファイルを選択します。実行時に指定したファイルによって、オブジェクトに指定したファイルが上書きされます。
ブロックサイズ [1]	フラットファイル形式に適用されます。大きなファイルを指定された小さなブロックサイズに分割します。サイズの大きいファイルを読み取る場合は、ファイルを小さな部分に分割し、同時接続を設定して、データを並列処理するために必要な数のスレッドを生成してください。 ブロックサイズに整数値を指定します。 バイト単位のデフォルト値は 8388608 です。
タイムアウト間隔	該当なし。
再帰的なディレクトリの読み取り	サブディレクトリに格納されているオブジェクトをマッピングで読み取るかどうかを示します。 詳細については、「サブディレクトリからのファイルの読み取り」 (ページ 21)を参照してください。
増分ファイルロード [2]	詳細モードのマッピングでソースとしてディレクトリを使用する場合に、ファイルを段階的にロードするかどうかを示します。 ファイルを段階的にロードすると、マッピングタスクは、マッピングタスクが最後に実行されてから変更されたディレクトリ内のファイルのみを読み取って処理します。 詳細については、「ファイルの増分ロード」 (ページ 22)を参照してください。
圧縮形式	圧縮データをソースから読み取ります。 次のいずれかのオプションを選択します。 - なし。Snappy 圧縮を使用する Avro ファイル、ORC ファイル、および Parquet ファイルを読み取る場合に選択します。圧縮されたファイルの拡張子は、 `.snappy` になります。 圧縮した JSON ファイルを読み取ることはできません。 - Gzip。Gzip 圧縮を使用する Avro フラットファイルおよび Parquet ファイルを読み取る場合に選択します。圧縮されたファイルの拡張子は、`.gz` になります。 圧縮したフラットファイルのデータをプレビューすることはできません。

プロパティ	説明
中間ディレクトリ [1]	オプション。フラットファイルと JSON ファイルに適用されます。 Secure Agent マシンのステージングディレクトリへのパス。 Microsoft Azure Data Lake Storage Gen2 からデータを読み取る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。 デフォルトのステージングディレクトリは/tmp です。 Hosted Agent を使用する場合、Interim ディレクトリを指定することはできません。
トレースレベル	ログファイルに表示される詳細情報の量を設定します。[簡易]、[ノーマル]、[詳細 - 初期化]、[詳細 - データ] から選択できます。デフォルトは [ノーマル] です。

[1] 詳細モードのマッピングには適用されません。
[2] 詳細モードのマッピングにのみ適用されます。

Microsoft Azure Data Lake Storage Gen2 ソースのディレクトリソース

データの読み取り元のソースのタイプを選択できます。

詳細ソースプロパティの **[ソースタイプ]** オプションから次のソースタイプを選択できます。

- ファイル
- ディレクトリ

ソースタイプとして **[ディレクトリ]** を選択するには、次のルールとガイドラインを使用します。

- ディレクトリ内のすべてのソースファイルに同じメタデータが含まれている必要があります。
- すべてのファイルに同じ形式のデータが含まれている必要があります。例えば、区切り記号、ヘッダーフィールド、およびエスケープ文字は同じである必要があります。
- 指定されたディレクトリ内のすべてのファイルが解析されます。サブディレクトリ内のファイルは解析されません。
- ディレクトリからデータを読み取るマッピングを実行する場合、エージェントはターゲットに 1 つのファイルを作成します。詳細モードのマッピングを作成すると、エージェントはターゲットに複数のファイルを作成します。

ワイルドカード文字

Avro、flat、JSON、ORC、または Parquet ファイルからデータを読み取る場合は、ワイルドカード文字を使用してソースファイル名を指定することができます。

ソースファイル名にワイルドカード文字を使用するには、ソースタイプを **[ディレクトリ]** として選択し、詳細ソースプロパティで **[ワイルドカード文字を許可]** オプションを有効にします。

Avro ファイル、JSON ファイル、ORC ファイル、Parquet ファイル、またはフラットファイルを読み取る場合は、? および*というワイルドカード文字を使用して、1 つ以上の文字を含む文字列を検索に定義できます。

以下のワイルドカード文字を使用することができます。

? （疑問符）

疑問符（?）の部分に任意の文字を 1 つ含む文字列を検索します。例えば、ソースファイル名に a?b.txt と入力すると、Secure Agent は次のような名前のファイルからデータを読み取ります。

- a1b.txt
- a2b.txt
- aab.txt
- acb.txt

*** （アスタリスク）**

アスタリスクマーク（*）の部分に文字のない文字列、または任意の文字を 1 つ以上含む文字列を検索します。ソースファイル名に a*b.txt と入力すると、Secure Agent は次のような名前のファイルからデータを読み取ります。

- aab.txt
- a1b.txt
- ab.txt
- abc11b.txt

ワイルドカード文字のルールとガイドライン

ワイルドカード文字を使用する場合は、次のルールとガイドラインを考慮してください。

マッピング

- パーティションカラムから読み取る場合、ワイルドカード文字を使用することはできません。
- マッピングで複合ファイルを読み取るときは、サブディレクトリ名またはファイル名にチルダ（~）を使用しないでください。
- ディレクトリのオーバーライドでワイルドカード文字を使用すると、Secure Agent はフォルダと名前のパターンに一致するファイルからデータを読み取ります。

詳細モードのマッピング

- フラットファイルまたは複合ファイルを読み取り、ワイルドカード文字を有効にする場合は、ディレクトリのオーバーライドまたはファイル名のオーバーライドで指定されたパスがソースのファイルパスと一致することを確認してください。
- ワイルドカード文字を使用する場合は、ファイル名の先頭の文字に特殊文字以外を使用してください。
- フラットファイルを読み取る場合は、ディレクトリのオーバーライドのディレクトリ名またはサブディレクトリ名に次の特殊文字を使用しないでください。

  [] {} " ' + ^ % * ? space

サブディレクトリからのファイルの読み取り

Microsoft Azure Data Lake Storage Gen2 のサブディレクトリに格納されているオブジェクトをマッピングで読み取ることができます。

マッピングのフラットファイルと複合ファイルには、再帰読み取りを使用できます。詳細モードのマッピングを作成する際、フラットファイルに対して再帰的な読み取り使用することはできません。

再帰読み取りを有効にするには、詳細ソースプロパティでソースタイプを **［ディレクトリ］** として選択します。サブディレクトリに格納されているオブジェクトを読み取るには、**［再帰的なディレクトリの読み取り］** 詳細ソースプロパティを有効にします。

サブディレクトリからの読み取りに関するルールとガイドライン

サブディレクトリに格納されているオブジェクトを読み取るときは、次のルールとガイドラインを考慮してください。

マッピング

- Microsoft Azure Data Lake Storage Gen2 でフラットファイルの読み取りまたは書き込みを行うときは、ディレクトリ名またはサブディレクトリ名にパーセンテージ（%）文字が含まれていないことを確認してください。そうしないと、マッピングが失敗します。
- パーティションカラムから読み取る場合、再帰読み取りを使用することはできません。
- マッピングで複合ファイルを読み取るときは、サブディレクトリ名またはファイル名にチルダ（~）を使用しないでください。
- ソースとターゲットの FileName フィールドをマッピングすると、次の形式でファイルが作成されます。
  `target_filename=source_directory_subdirectory1_sudirectory2`
- ヘッダーのみでデータがないフラットファイルを読み取って FileName フィールドをマッピングすると、FileName フィールドでは適切なディレクトリ構造が作成されません。

詳細モードのマッピング

- 複合ファイルを読み取り、再帰読み取りを有効にした場合は、ディレクトリのオーバーライドまたはファイル名のオーバーライドで指定されたパスがソースのファイルパスと一致することを確認してください。
- フラットファイルを読み取る場合は、ディレクトリのオーバーライドのディレクトリ名またはサブディレクトリ名に次の特殊文字を使用しないでください。
  `[] {} " ' + ^ % * ? space`

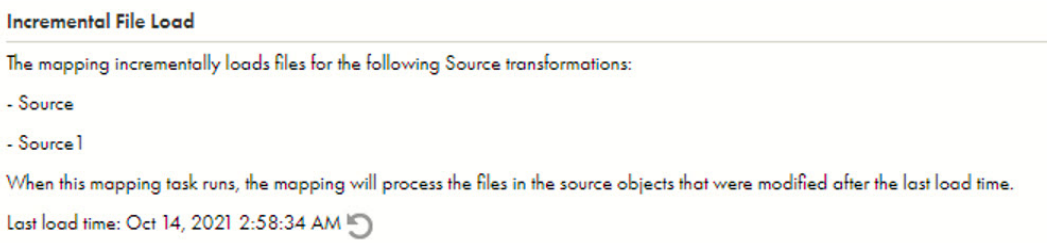
ファイルの増分ロード

ディレクトリにソースファイルを増分ロードして、マッピングタスクが最後に実行されてから変更されたファイルのみを読み取って処理することができます。

ソースファイルを増分ロードするには、Microsoft Azure Data LakeStorage Gen2 データオブジェクトの詳細読み取りプロパティの［**増分ファイルロード**］オプションを選択します。ファイルの増分ロードは、詳細モードのマッピングでのみ可能です。すべてのマッピングソースは、同じクラウド環境にある必要があります。

Microsoft Azure Data Lake Storage Gen2 からファイルを増分ロードする場合、ジョブは、最終ロード時刻からジョブの実行を開始する 5 分前に変更されたファイルをロードします。例えば、午後 2 時にジョブを実行すると、午後 1 時 55 分より前に変更されたファイルをジョブがロードします。これは、Microsoft Azure Data Lake Storage Gen2 にオブジェクトをアップロードすると、完了するまでに数分かかるためです。

マッピングタスクを設定すると、［**増分ファイルロード**］セクションには、ファイルを増分ロードするソーストランスフォーメーションと、最後のジョブがファイルのロードを完了した時刻が一覧表示されます。デフォルトでは、実行される次のジョブは、最終ロード時刻の後に変更されたファイルを確認します。次のジョブですべてのソースファイルをロードするには、最終ロード時刻をリセットします。



マッピングを実行して、Parquet や Avro などの複合ファイル形式を含むディレクトリを増分ロードするときに、前回の実行以降にソースに新しいファイルまたは変更されたファイルがない場合、マッピングは失敗します。

ソースファイルを増分ロードした後、特定の時間間隔で変更されたファイルを再処理するジョブを実行できます。ファイルを再処理するジョブを作成する場合、変更されたファイルを確認するためにジョブが使用する必要のある時間間隔を選択します。

増分ファイルロードの詳細については、データ統合のドキュメントの「*タスク*」を参照してください。

プッシュダウンの最適化

Microsoft Azure Data Lake Storage Gen2 ソースから Microsoft Azure Synapse SQL のデータウェアハウスにデータをロードする場合は、完全なプッシュダウンの最適化を有効にすることができます。データを Microsoft Azure Synapse SQL にロードするときに、データウェアハウスのモデルと要件に従ってデータを変換できます。マッピングタスクで完全プッシュダウンを有効にすると、マッピングロジックが Azure 環境にプッシュされ、Azure コマンドが利用されます。詳細については、Microsoft Azure Synapse SQL コネクタのヘルプを参照してください。

サポートされている他のクラウドデータウェアハウスにデータをロードする必要がある場合は、該当するクラウドデータウェアハウスのコネクタのヘルプを参照してください。

マッピングにおける Microsoft Azure Data Lake Storage Gen2 ターゲット

マッピングでは、Microsoft Azure Data Lake Storage Gen2 オブジェクトをターゲットとして使用できます。

Microsoft Azure Data Lake Storage Gen2 ターゲットオブジェクトを使用する場合は、Microsoft Azure Data Lake Storage Gen2 コレクションをターゲットとして選択できます。マッピングウィザードの［ターゲット］ページで、Microsoft Azure Data Lake Storage Gen2 ターゲットプロパティを設定できます。Microsoft Azure Data Lake Storage Gen2 にデータを書き込む場合は、［ターゲットの作成］フィールドを使用して、実行時にターゲットを作成できます。ソースに基づいて新しいターゲットを作成する場合は、フィールドマッピングからすべてのバイナリフィールドを削除する必要があります。

以下の表に、ターゲットトランスフォーメーションで設定できる Microsoft Azure Data Lake Storage Gen2 ターゲットプロパティを示します。

プロパティ	説明
接続	ターゲット接続の名前。ターゲット接続を選択するか、**［新しいパラメータ］**をクリックして、ターゲット接続の新しいパラメータを定義します。 パラメータ化されていない Microsoft Azure Data Lake Storage Gen2 接続とパラメータ化された Microsoft Azure Data Lake Storage Gen2 接続を切り替えると、詳細プロパティ値が保持されます。
ターゲットタイプ	［単一オブジェクト］または［パラメータ］を選択します。

プロパティ	説明
オブジェクト	ターゲットオブジェクトの名前。既存のオブジェクトを選択するか、実行時に新規ターゲットを作成できます。 **[実行時に新規作成]** を選択した場合は、ターゲットオブジェクトの名前を入力し、使用するソースフィールドを選択します。デフォルトでは、すべてのソースフィールドが使用されます。 ターゲット名に英数字を含めることはできません。ファイル名では、ピリオド（.）、アンダースコア（_）、アットマーク（@）、ドル記号（$）、パーセント記号（%）のみを特殊文字として使用できます。 ヘッダーまたはファイルデータに特殊文字が含まれていないことを確認してください。 ターゲット名では、パラメータファイルに定義したパラメータを使用できます。[ターゲットの作成] オプションを選択した場合、実行時にターゲットをパラメータ化することはできません。 **注**: 実行時に作成したフラットファイルにデータを書き込んだ場合、ターゲットフラットファイルのファイルの末尾に空白行が含まれます。
パラメータ	ターゲットオブジェクトの既存のパラメータを選択するか、**[新しいパラメータ]** をクリックしてターゲットオブジェクトの新しいパラメータを定義します。 **[パラメータ]** プロパティは、ターゲットタイプとして [パラメータ] を選択した場合にのみ表示されます。 ターゲットオブジェクトをパラメータ化するときは、パラメータのデフォルト値にファイルシステムを含む完全なオブジェクトパスを指定します。
形式	Microsoft Azure Data Lake Storage Gen2 コネクタが Microsoft Azure Data Lake Storage Gen2 へのデータの書き込みに使用するファイル形式を指定します。 次のファイル形式タイプを選択できます。 - Flat - Avro - Parquet - JSON - ORC デフォルトは **[なし]** です。 形式タイプとして **[なし]** を選択した場合、Microsoft Azure Data Lake Storage Gen2 コネクタはデータを Microsoft Azure Data Lake Storage Gen2 ファイルにバイナリ形式で書き込みます。 詳細については、「ファイル形式オプション」 (ページ 28)を参照してください。
操作	ターゲット操作。[挿入] を選択します。Microsoft Azure Data Lake Storage Gen2 ターゲットにデータを挿入することのみが可能です。

[1] 詳細モードのマッピングには適用されません。
[2] 詳細モードのマッピングにのみ適用されます。

注: [ターゲットの作成] オプションを使用し、**[形式オプション]** の [入力形式] と一致しない拡張子のオブジェクト名を指定すると、Secure Agent は、**[形式オプション]** で指定した入力形式を無視します。

例えば、入力形式として [Parquet] を選択し、**[ターゲットオブジェクト]** ダイアログボックスのオブジェクト名で `customer.avro` を指定すると、Secure Agent は Parquet を無視して Avro ターゲットファイルを作成します。

以下の表に、Microsoft Azure Data Lake Storage Gen2 の詳細ターゲットプロパティを示します。

詳細ターゲットプロパティ	説明
同時スレッド [1]	Microsoft Azure Data Lake Storage Gen2 からデータをロードするための同時接続数。サイズの大きいファイルを書き込む場合は、複数のスレッドを生成してデータを処理できます。サイズの大きいファイルを複数の小さい部分に分割するには、**［ブロックサイズ］** を設定します。 デフォルト値は 4 です。最大値は 10 です。
ファイルシステム名のオーバーライド	デフォルトのファイル名をオーバーライドします。
ディレクトリのオーバーライド	データの書き込みに使用する Microsoft Azure Data Lake Storage Gen2 ディレクトリ。デフォルトはルートディレクトリです。ディレクトリが存在しない場合には、Secure Agent によって作成されます。実行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上書きされます。 絶対ディレクトリパスまたは相対ディレクトリパスを指定できます。 - 絶対パス - Secure Agent は、指定したファイルシステムでこのディレクトリパスを検索します。 絶対パスの例: `Dir1/Dir2` - 相対パス - Secure Agent は、オブジェクトのネイティブディレクトリパスでこのディレクトリパスを検索します。 相対パスの例: `/Dir1/Dir2` 相対パスを使用すると、インポートされたオブジェクトパスが、実行時にメタデータの取得中に使用されるファイルパスへ追加されます。 ディレクトリをオーバーライドするためにルートディレクトリ（/）を指定しないでください。
ファイル名のオーバーライド	ターゲットオブジェクト。データの書き込み元のファイルを選択します。実行時に指定したファイルによって、オブジェクトに指定したファイルが上書きされます。
書き込みストラテジ	マッピングのフラットファイルに適用されます。 詳細モードのマッピングを作成する際は、フラットファイルと複合ファイルの両方に書き込みストラテジを使用できます。 Microsoft Azure Data Lake Storage Gen2 にファイルが存在する場合は、既存のファイルを上書きするか、追加するかを選択できます。 詳細モードのマッピングにデータを追加すると、データは既存のターゲットディレクトリに新しいパートファイルとして追加されます。
ブロックサイズ [1]	フラット、Avro、および Parquet ファイル形式に適用されます。大きなファイルを指定された小さなブロックサイズに分割します。サイズの大きいファイルを書き込む場合は、ファイルを小さな部分に分割し、同時接続を設定して、データを並列処理するために必要な数のスレッドを生成してください。 ブロックサイズに整数値を指定します。 バイト単位のデフォルト値は 8388608 です。

詳細ターゲットプロパティ	説明
圧縮形式	指定した形式に基づいて、データを圧縮してターゲットに書き込みます。 次のいずれかのオプションを選択します。 - **なし**。Snappy 圧縮を使用する Avro ファイル、ORC ファイル、および Parquet ファイルを書き込む場合に選択します。 圧縮した JSON ファイルを書き込むことはできません。 - Gzip。Gzip 圧縮を使用する Avro フラットファイルおよび Parquet ファイルを書き込む場合に選択します。 タスクが実行されると、ファイル拡張子の.gz また.snappy はターゲットオブジェクト名には表示されません。
タイムアウト間隔	該当なし。
中間ディレクトリ [1]	オプション。フラットファイルと JSON ファイルに適用されます。 Secure Agent マシンのステージングディレクトリへのパス。 Microsoft Azure Data Lake Storage 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。 デフォルトのステージングディレクトリは/tmp です。 詳細モードのマッピングに、Interim ディレクトリを指定することはできません。 Hosted Agent を使用する場合、Interim ディレクトリを指定することはできません。
拒否された行の転送 [1]	拒否された行を次のトランスフォーメーションに渡すか、または削除するように、トランスフォーメーションを設定できます。
[1] 詳細モードのマッピングには適用されません。	

ターゲットの指定

既存のターゲットを使用するか、ターゲットを作成して、マッピングの結果を格納するためにことができます。ターゲットを作成することを選択した場合、エージェントはタスクの実行時にターゲットを作成します。

ターゲットのプロパティを指定するには、次の手順に従います。

1. マッピングでターゲットトランスフォーメーションを選択します。
2. **[受信フィールド]** タブで、ターゲットに含めるフィールドを指定するフィールドルールを設定します。
3. ターゲットを指定するには、**[ターゲット]** タブをクリックします。
4. ターゲット接続を選択します。
5. ターゲットタイプとして **[単一オブジェクト]** または **[パラメータ]** を選択します。
6. ターゲットオブジェクトまたはパラメータを指定します。
   - 実行時にターゲットファイルを作成するには、Accounts.csv のように、拡張子を含めてターゲットファイルの名前を入力します。

     注: フラットファイルから読み取る場合、ファイルにヘッダーだけでなくデータが含まれていることを確認します。ファイルにヘッダーのみが含まれる場合、ヘッダーはターゲットに書き込まれません。
   - ファイル名にタイムスタンプを含める場合は、**[特殊文字の処理]** をクリックして、ファイル名に特殊文字を追加します。例えば、すべてのタイムスタンプ情報を含める場合は、Accounts_%d%m%y%T.csv のように指定します。

注: ［特殊文字の処理］オプションは、詳細モードのマッピングには適用されません。

7. ファイルの形式オプションを設定する場合は、**［形式オプション］**をクリックし、**［OK］**をクリックします。
8. **［選択］**をクリックしてターゲットオブジェクトを選択します。既存のターゲットオブジェクトを選択するか、実行時に新しいターゲットオブジェクトを作成し、オブジェクト名を指定することができます。
9. 必要に応じて、ターゲットの詳細プロパティを指定します。

ターゲットのタイムスタンプ

マッピングで実行時にターゲットを作成する場合、タイムスタンプ情報をファイル名に追加して、ファイルがいつ作成されたのかを示すことができます。

ターゲットファイルのファイル名を指定する場合は、LinuxSTRFTIME 関数は、mapping タスクでファイル名にタイムスタンプ情報を含めるために使用される形式です。タイムスタンプは、組織のタイムゾーンに基づいています。

詳細モードのマッピングでは、ファイル名にタイムスタンプ情報を追加することはできません。

次の表に、マッピングまたは mapping タスクで使用される一般的な STRFTIME 関数の形式を示します。

特殊文字	説明
%d	2 桁の 10 進数で表される日（01～31）。
%m	2 桁の 10 進数で表される月（01～12）。
%y	2 桁の 10 進数で表される、世紀を含まない年（00～99）。
%Y	世紀を含む年（2015 など）。
%T	フラットファイルにのみ適用されます。24 時間表記の時刻（%H:%:M:%S に相当）。
%H	24 時間表記の時間（00～24）。
%l	12 時間表記の時間（01～12）。
%M	10 進数で表される分（00～59）。
%S	10 進数で表される秒（00～60）。
%p	AM または PM。

注: 複雑なファイルの場合、%T の代わりに同等の%H_%M_%S を使用できます。

Microsoft Azure Data Lake Storage Gen2 ターゲットファイルのパラメータ化

Microsoft Azure Data Lake Storage Gen2 ターゲットオブジェクトのファイル名とターゲットフォルダの場所をパラメータ化して、実行時にファイル名とフォルダの場所を渡すことができます。フォルダが存在しない場合は、Secure Agent によってフォルダ構造が動的に作成されます。

パラメータファイルによる Microsoft Azure Data Lake Storage Gen2 ターゲットファイルのパラメータ化

パラメータファイルを使用して、Microsoft Azure Data Lake Storage Gen2 ターゲットファイルの［**ディレクトリのオーバーライド**］および［**ファイル名のオーバーライド**］の詳細ターゲットプロパティをパラメータ化します。

パラメータファイルを使用して Microsoft Azure Data Lake Storage Gen2 ターゲットファイルをパラメータ化するには、Microsoft Azure Data Lake Storage Gen2 ターゲットオブジェクトを作成し、ターゲットオブジェクト名とターゲットオブジェクトパスにパラメータを追加します。ターゲットオブジェクトに追加したパラメータをパラメータファイルで定義し、そのパラメータファイルを次の場所に配置してマッピングタスクを実行します。

```
<Informatica Cloud Secure Agent\apps\Data_Integration_Server\data\userparameters>
```

パラメータファイルを Microsoft Azure Data Lake Storage Gen2 のクラウドでホストされているディレクトリに保存することもできます。

詳細モードのマッピングでは、パラメータファイルを使用することはできません。

ファイル形式オプション

Microsoft Azure Data Lake Storage Gen2 ファイルの形式を選択し、形式オプションを設定します。

次の表に、Avro、Parquet、JSON、ORC、および区切りフラットファイルの形式オプションを示します。

プロパティ	説明
スキーマソース	ソースファイルまたはターゲットファイルのこのスキーマ。 以下のいずれかのオプションを選択して、スキーマを指定します。 - データファイルから読み取る。Microsoft Azure Data Lake Storage Gen2 のファイルからスキーマをインポートします。 - スキーマファイルからインポートする。エージェントマシンのスキーマ定義ファイルからスキーマをインポートします。
スキーマファイル	スキーマ定義ファイルをアップロードします。実行時にターゲットを作成するときに、スキーマファイルをアップロードすることはできません。

次の表に、フラットファイルの形式オプションを示します。

プロパティ	説明
フラットファイルタイプ	フラットファイルのタイプ。 次のいずれかのオプションを選択します。 - 区切り。カラム区切り文字を含むフラットファイルを読み取ります。 - 固定長。固定長のフィールドを含むフラットファイルを読み取ります。 **［固定長ファイル形式］**オプションのファイル形式を選択する必要があります。固定長ファイル形式がない場合、**［新規］** > **［コンポーネント］** > **［固定長ファイル形式］**をクリックして作成します。
区切り文字	区切りフラットファイルのデータのカラムを分けるのに使用される文字。カンマ、タブ、コロン、セミコロンなどの値を設定できます。 1 文字または複数文字の区切り文字を使用できます。 **［区切り文字］**フィールドで区切り文字としてタブを直接設定することはできません。タブを区切り文字として設定するには、任意のテキストエディタでタブ文字を入力する必要があります。その後、そのタブ文字をコピーして**［区切り文字］**フィールドに貼り付けます。
EscapeChar	引用符で囲まれていない文字列におけるカラム区切り文字の直前の文字、または区切りフラットファイルでの、引用符で囲まれた文字列データ内における引用符の直前の文字。 Microsoft Azure Data Lake Storage Gen2 にデータを書き込んで修飾子を指定すると、デフォルトでは、修飾子はエスケープ文字と見なされます。それ以外の場合は、エスケープ文字として指定された文字が考慮されます。
修飾子	区切りフラットファイル内にある、データの境界を定義する引用符。修飾子は、一重引用符または二重引用符で設定できます。
修飾子モード	区切りフラットファイルにデータを書き込むときの修飾子の動作を指定します。 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。 - 最小。デフォルトモード。区切り文字の値または特殊文字で囲まれているデータに修飾子を適用します。 - すべて。すべてのデータに修飾子を適用します。 - Non_Numeric。該当なし。 - All_Non_Null。該当なし。
コードページ	Secure Agent が、区切りフラットファイルに対するデータの読み取りまたは書き込みで使用する必要があるコードページを選択します。 マッピングに UTF-8 を選択します。 詳細モードのマッピングについて、次のいずれかのオプションを選択します。 - UTF-8 - MS Windows Latin 1 - Shift-JIS - ISO 8859-15 Latin 9（Western European） - ISO 8859-3 Southeast European - ISO 8859-5 Cyrillic - ISO 8859-9 Latin 5（Turkish） - IBM EBCDIC International Latin-1
ヘッダー行番号	区切りフラットファイルからのデータの読み取り時にヘッダーとして使用する行番号を指定します。 値を 0 または 1 として指定します。 ヘッダーのないファイルからデータを読み取るには、値を 0 に指定します。

プロパティ	説明
最初のデータ行 [1]	Secure Agent が、区切りフラットファイルのデータの読み取りを開始する行番号を指定します。1 以上の値を入力する必要があります。 ヘッダーからデータを読み取るには、**[ヘッダー行番号]** と **[最初のデータ行]** のフィールドに同じ値を指定する必要があります。デフォルトは 1 です。
ターゲットヘッダー	区切りフラットファイルでヘッダーを含むターゲットとヘッダーを含まないターゲットのどちらにデータを書き込むかを指定します。**[ヘッダーあり]** オプションまたは **[ヘッダーなし]** オプションを選択できます。 このプロパティは、Microsoft Azure Data Lake Storage Gen2 ソースからデータを読み取る場合には適用されません。
分散カラム	該当なし。
エスケープ文字データを保持	該当なし。
プレビューする最大行数	該当なし。
行区切り文字	データの列を区切るために使用する文字。値には、`\r`、`\n`、`\r\n` を設定できます。 このプロパティは、Microsoft Azure Data Lake Storage Gen2 ソースからデータを読み取る場合には適用されません。
[1] 詳細モードのマッピングには適用されません。	

次の表に、JSON ファイルの形式オプションを示します。

プロパティ	説明
サンプリングするデータ要素 [1]	メタデータを取り込むための最良の一致を見つけるために読み取る行数を指定します。
データの処理に使用可能なメモリ [1]	パーサーが JSON サンプルスキーマを読み取って処理するために使用するメモリ。 デフォルト値は 2 MB です。 ファイルサイズが 2 MB を超える場合、エラーが発生する可能性があります。読み取るファイルサイズに値を設定します。
複数行の JSON ファイルの読み取り	該当なし。
[1] 詳細モードのマッピングにのみ適用されます。	

固定長ファイル形式

固定長フラットファイルは、マッピングとマッピングタスクのソースまたはターゲットとして使用できます。

ソーストランスフォーメーションまたはターゲットトランスフォーメーションを設定し、固定長フラットファイルタイプを選択する場合は、固定長フラットファイルのデータに基づいて、使用する最も適切な固定長ファ

イル形式を選択する必要があります。サンプルフラットファイルが UTF-8 文字セットエンコーディングのみを使用していることを確認してください。

固定長フラットファイルについては、次のルールとガイドラインを考慮してください。

- 固定長フラットファイルを、詳細モードのマッピングおよびデータ転送タスクのソースまたはターゲットとして使用することはできません。
- 実行時にフラットファイルターゲットを作成し、固定長ファイル形式を選択すると、Secure Agent は、固定長フラットファイル形式に指定した固定長カラムの境界を無視し、新しいターゲットオブジェクトに追加の固定長属性を適用します。
- 固定長フラットファイルソースから、同じ固定長ファイル形式をソースファイルとして使用する空の固定長フラットファイルターゲットに数値データ型のカラムを書き込むと、Secure Agent は、ターゲットの数値カラムの値に Null 文字を追加します。
- 固定長ファイル形式を作成する場合は、Secure Agent がインストールされているオペレーティングシステムに応じ、サンプルファイルで次の文字が改行記号として使用されていることを確認します。

  - Linux の場合は、\n 文字を使用します。

  - Windows の場合は、\r\n 文字を使用します。

  ソースファイルでも、サンプルファイルで定義されているのと同じ文字が使用されている必要があります。
- 固定長フラットファイルをソースまたはターゲットとして使用する場合、フィールドのメタデータは編集できません。
- date データ型のデータを読み取る場合、ミリ秒までの日付しか読み取ることができません。
- 固定長ファイルから double データ型のデータを読み取り、そのデータを Parquet ファイルまたは Avro ファイルに書き込むと、double データ型はターゲットで decimal データ型にマップされます。したがって、データは正しく書き込まれません。
  データをターゲットに正しく書き込むには、ターゲットトランスフォーメーションのメタデータを編集し、decimal データ型を double データ型に変更します。
- ヘッダー付きの固定長ファイルから読み取り、固定長ファイルに書き込んで FileName フィールドをマップすると、ヘッダーはデータの一部と見なされます。ヘッダーをスキップするには、固定長ファイル形式を作成するときに、**[スキップする行数]** フィールドの値を 1 に設定します。
- 固定長ファイルにデータを書き込む際、既存のファイルにデータを追加することはできません。

ファイル名フィールド

ファイル名フィールドは、ファイルのソースパスを含む文字列フィールドです。ファイル名フィールドのデフォルトの精度は、フラットファイルの場合は 255 文字で複合ファイルの場合は 1024 文字です。

ファイル名フィールドを設定することはできません。ファイル名のデータを読み書きしない場合は、ファイル名フィールドを削除できます。フラットファイルの場合は 255 文字を超え、複合ファイルの場合は 1024 文字を超えるフォルダ名は作成できません。

ファイル名フィールドは、次のファイル形式に適用されます。

- フラットファイル
- Avro
- Parquet
- ORC

ソースオブジェクトパスの読み取り

ソースオブジェクトをインポートすると、Secure Agent によって、インポートされたソースオブジェクトに FileName フィールドが追加されます。ファイル名フィールドには、Secure Agent が実行時にデータを読み取るソースファイルの絶対パスが保存されます。

例えば、ディレクトリに多数のファイルが含まれ、各ファイルには読み取りを行う複数のレコードが含まれているとします。Microsoft Azure Data Lake Storage Gen2 ソースの詳細プロパティで、ソースタイプとしてディレクトリを選択します。マッピングを実行すると、Secure Agent は各レコードを読み取って、それぞれのソースファイルの絶対パスが FileName フィールドに保存されます。

ソースオブジェクトで FileName フィールドを使用する場合、Secure Agent は、詳細モードのマッピングについては異なる方法でファイル名を読み取ります。

入力形式	構文	例
複合ファイル	abfss://<filesystem_name>@<account name>.<endpoint>/<directory>/<source_file_name>	abfss://adapterqa-source@adlsgen2qa.dfs.core.windows.net/parquet/reader/customer.parquet
フラットファイル	<directory>/<source_file_name>	csv/customer.csv

ターゲットオブジェクトへの書き込み

ターゲットオブジェクトをインポートすると、Secure Agent により、インポートされたターゲットオブジェクトに FileName フィールドが追加されます。ターゲットオブジェクトのファイル名フィールドを受信フィールドにマッピングすると、Secure Agent によって、ファイル名フィールドに基づいてフォルダ構造とターゲットファイルが作成されます。

以下の表に、Secure Agent がマッピングのファイル名をどのように読み取るかを示します。

入力形式	構文	例
複合ファイル	<directory>/<target_file_name>/<target_file_name>=<values of filename field>/part_file	parquet/writer/customer.parquet/customer.parquet=1
フラットファイル	<directory>/<target_file_name>/<target_file_name>=<values of filename field>/part_file この構文はマッピングのみに適用されます。 詳細モードのマッピングを作成しても、Secure Agent はディレクトリ構造を作成しません。	csv/customer.csv/customer.csv=1

以下の表に、マッピングの FileName フィールドシナリオの構文と例を示します。

説明	構文	例
ターゲットがない場合、新しいターゲットファイルが作成されます。	<target_file_name>=<source_file_name>	customer_tgt.csv=customer_src.csv
ソースのファイル名フィールドを既存のターゲットのファイル名フィールドにマッピングした場合、新しいターゲットファイルが作成され、既存のターゲットは影響を受けません。	<target_file_name>=<source_field_value>	customer_tgt.csv=customer_src.csv
ファイル名フィールド以外のソースフィールドを既存のターゲットのファイル名フィールドにマッピングした場合、ソースフィールドの一意の値ごとに個別のファイルが作成されます。	<target_file_name>=<source_field_value>	customer_tgt.csv=a, customer_tgt.csv=b... customer_tgt.csv=n この例では、ターゲットファイルが存在するディレクトリに n 個のファイルが挿入されます。ここで、n はソースフィールドの一意の値の数です。

FileName フィールドに関するルールとガイドライン

マッピングで FileName フィールドを使用する場合は、次のガイドラインを考慮します。

- FileName は予約済みのキーワードです。ソースデータのカラム名として FileName を使用しないでください。この名前は、大文字と小文字が区別されます。
- 複合ファイルの場合、ソースオブジェクトの FileName フィールドを、ターゲットオブジェクトの FileName フィールドにマッピングしないでください。ターゲットオブジェクトの FileName フィールドを受信フィールドにマッピングした場合、Secure Agent によるディレクトリ構造の作成が予期したとおりに実行されません。
- ターゲットオブジェクトに FileName フィールドを使用すると、Secure Agent は、詳細モードのマッピングの NULL 値に対し、別の名前でフォルダを作成します。

  - マッピングの場合、ターゲットファイル名に_EMPTY_が追加されます。

  - 詳細モードのマッピングの場合、ターゲットファイル名に_HIVE_DEFAULT_PARTITION_が付きます。
- ターゲットオブジェクトのファイル名フィールドに日付タイプの受信フィールドをマッピングした場合、Secure Agent によって、ネストされたフォルダ構造がターゲットオブジェクトの受信日付の値に基づいて作成されます。
- 受信フィールドをターゲットオブジェクトのファイル名フィールドにマッピングし、ターゲットで **[特殊文字の処理]** オプションが有効になっている場合、またはターゲットファイル名に特殊文字が含まれている場合、Windows マシンでは次のエラーが発生してマッピングが失敗します。
  [エラー] java.io.IOException: ファイル名、ディレクトリ名、またはボリュームラベルの構文が正しくありません

- ターゲットオブジェクトのファイル名フィールドに受信フィールドをマッピングすると、初回のマッピングは正常に実行されます。それ以降の実行では、マッピングが次のエラーで失敗します。
  オブジェクトが見つかりません。

  マッピングの再実行が正常に行われるようにするには、設計時にダミーのターゲットファイルを使用して、詳細ターゲットプロパティでダミーのターゲットファイルを上書きします。

ディレクトリレベルのパーティショニング

詳細モードのマッピングを作成する際は、パーティションカラムに対する読み取りと書き込みが可能です。

テーブルまたはデータセットをパーティション化してまとめ、カラムやパーティションキーに基づいて同じタイプのデータをグループ化できます。テーブルまたはデータセットの 1 つ以上のパーティションカラムを選択できます。

パーティションカラムから読み取るには、パーティションディレクトリを選択して、パーティションカラムを特定します。パーティションカラムに書き込むには、フィールドのリストからパーティションカラムを追加し、必要に応じてパーティションの順序を変更します。

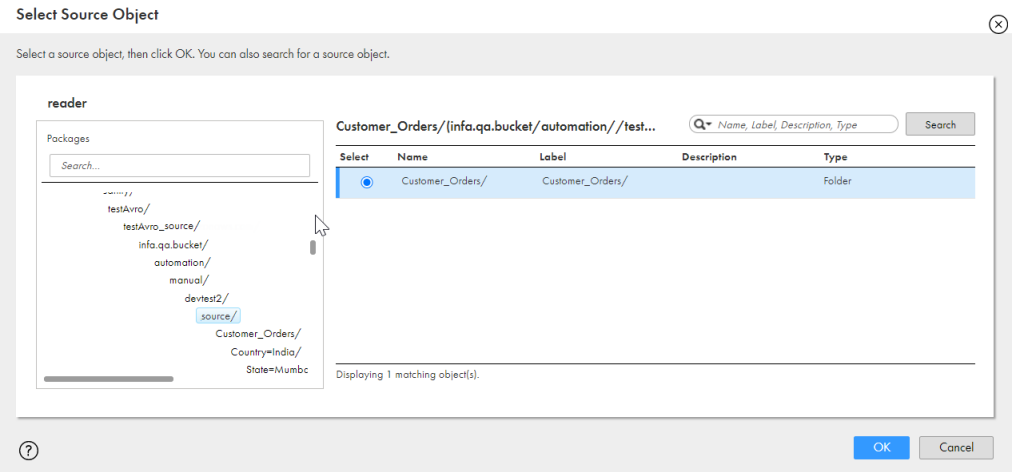
次のファイル形式で、パーティションカラムに対してデータを読み書きすることができます。

- Avro
- Parquet
- ORC
- JSON

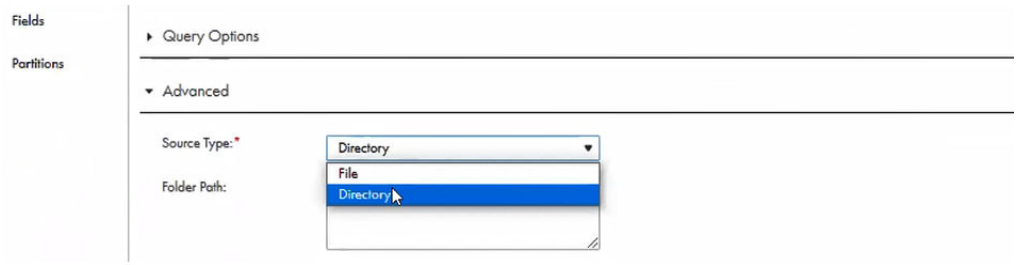
パーティションカラムからの読み取り

パーティションカラムからデータを読み取るには、次の手順を実行します。

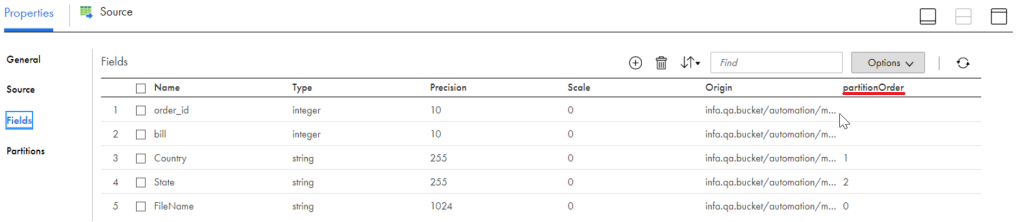
1. ソースオブジェクトのリストからディレクトリを選択します。



2. 詳細ソースプロパティで［**ディレクトリ**］としてソースタイプを選択します。



3. ［フィールド］タブで、パーティションの数を確認できます。
   [partitionOrder] カラムで、カラムをパーティション化するかどうか、およびパーティション化の際にフィールドが選択される順序を指定します。

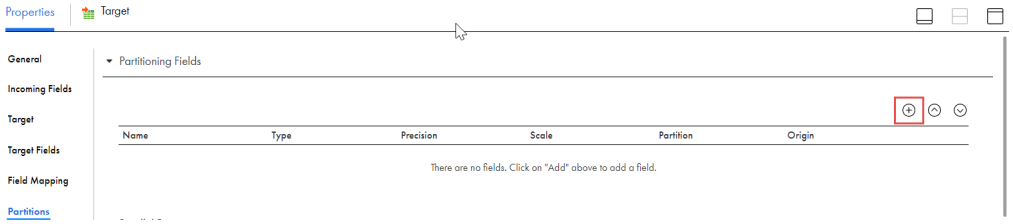


   上の画像では、2 つのパーティションカラムが存在しています。パーティション順序の値 1 と 2 は、パーティション化の際に［国］フィールドと［州］フィールドが選択される順序を示しています。［FileName］フィールドのパーティション順序は 0 です。

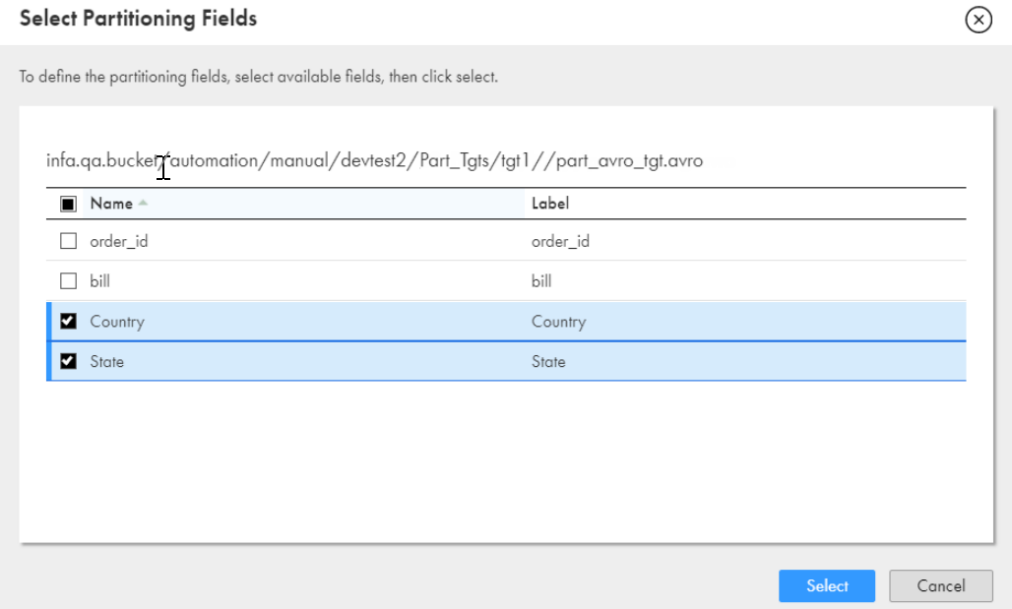
パーティションカラムへの書き込み

パーティションカラムに書き込むには、次の手順を実行します。

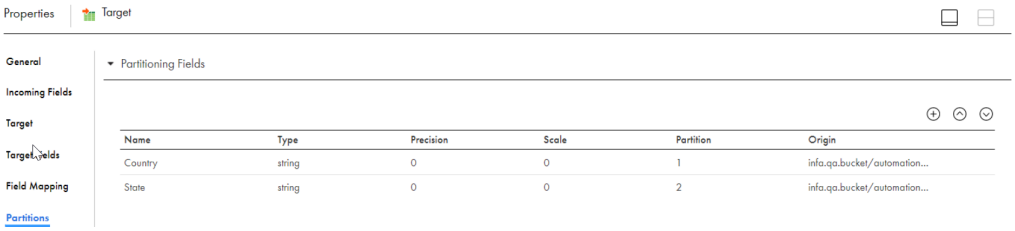
1. **［パーティション］** タブで、**［追加］** をクリックして、ターゲットにパーティションカラムを追加します。



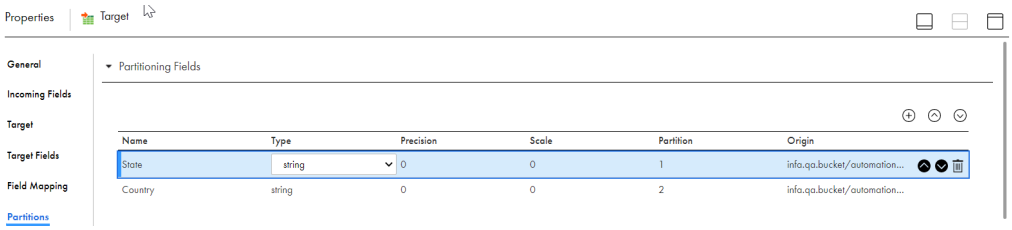
2. 使用可能なフィールドのリストからパーティション化フィールドを選択します。



3. **［選択］**をクリックします。
   ［パーティション］タブに、選択したパーティションカラムが表示されます。



次の画像に示すように、上矢印と下矢印を使用してパーティション順序を変更できます。



パーティションフォルダからの読み取りとパーティションフォルダへの書き込みに関するルールとガイドライン

詳細モードのマッピングのパーティションフォルダに対し、読み取りや書き込みを行う場合は、次のルールとガイドラインを考慮してください。

- パーティションフォルダのみを含むディレクトリをインポートし、詳細ソースプロパティでソースタイプを**［ディレクトリ］**として選択する必要があります。
- データのないパーティションディレクトリをインポートすると、検証エラーが発生します。
- パーティションフォルダはあってもパーティションフォルダにファイルがないパーティションディレクトリをインポートすると、検証エラーが発生します。
- ［FileName］フィールドのパーティション順序は 0 です。

- 選択したパーティションディレクトリには、「FileName」という名前のパーティションカラムを含めることはできません。この名前は、大文字と小文字が区別されません。
- 既存のターゲットオブジェクトをインポートした場合、またはパーティションディレクトリを持つ新規ターゲットオブジェクトを作成した場合、［ファイル名］フィールドはターゲット側には追加されません。［ファイル名］フィールドはソース側にのみ追加されます。
- パーティションカラムにタイムスタンプ値を渡すと、値がエンコードされます。例えば、「03:26:01」は「03%3A26%3A01」としてエンコードされます。
- パーティションカラムに特殊文字を渡した場合、値はエンコードされます。例えば、「#"」は「%23%22」としてエンコードされます。
- パーティションフォルダのあるディレクトリをインポートすると、パーティションカラムのデータ型が文字列としてインポートされます。
- パーティションカラムのデータ型を編集することはできません。
- パーティションカラムで［**メタデータの編集**］オプションを使用することはできません。
- ソース側とターゲット側のパーティションディレクトリに［**スキーマの表示**］オプションを使用することはできません。
- スキーマファイルにはパーティションカラムの情報がないため、ソースのパーティションディレクトリに［**スキーマファイルからインポートする**］オプションを使用することはできません。
- パーティションカラムで［**データプレビュー**］オプションを使用することはできません。
- ターゲットオブジェクトがパラメータ化されている場合、マッピングタスクでパーティションカラムを選択することはできません。
- ［**ターゲットの作成**］では、パーティションフィールドを追加し、パーティションカラムを順番に配置できます。
- ムが含まれる Microsoft Azure Data Lake Storage Gen2 オブジェクトをインポートした場合、パーティションフィールドがリストの最後に表示されます。
- パーティションカラムに 255 文字を超えるデータが含まれている場合、データは切り捨てられ、255 文字のみがパーティションカラムに書き込まれます。
- パーティションカラム名に 74 文字を超える文字が含まれている場合、名前は切り捨てられ、パーティションカラム名には 74 文字のみが書き込まれます。
- パーティションカラム名パーティションディレクトリ内のターゲットファイルの組み合わせを使用して形成されるパーティションディレクトリファイルパスの値は、1024 文字を超えないようにする必要があります。それ以外の場合、マッピングは失敗します。
- パーティションカラムで［**ファイル名のオーバーライド**］オプションを使用することはできません。
- JSON ファイルを読み書きする場合、パーティションカラムで［**サンプリングするデータ要素**］および［**データの処理に使用可能なメモリ**］詳細プロパティを使用することはできません。

詳細モードのマッピングの例

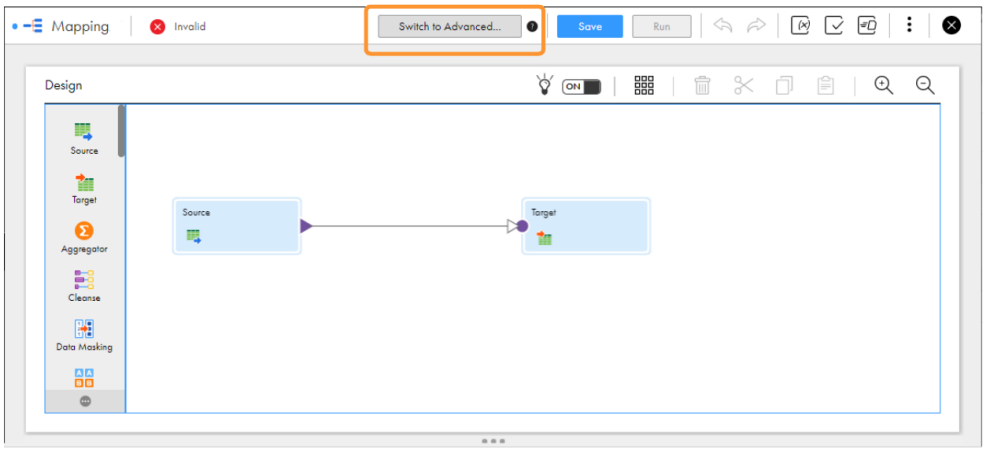
運用中の学生データベースで数百万件のレコードを保持している大規模なコミュニティカレッジで働いているとします。このカレッジには、世界中に 45 か所のキャンパスと 700 か所の拠点があり、10,000 人以上の教職員が働いています。非常に大規模な IT インフラストラクチャを保有していて、毎日インターネットからおよそ 15 TB の情報がダウンロードされています。

パフォーマンスと拡張性の問題を回避し、高額なコストが発生しないようにするために、このカレッジではデータ全体をオペレーショナルデータストアから Microsoft Azure Data Lake Storage Gen2 に短期間で移行する

よう計画しています。詳細モードのマッピングを設定することで、オペレーショナルデータストアからデータを読み取ったり、Microsoft Azure Data Lake Storage Gen2 ターゲットにデータを書き込む際のパフォーマンスを高速化することができます。

1. データ統合で、**［新規］** > **［マッピング］** > **［マッピング］** をクリックします。
2. Mapping Designer で、**［詳細に切り替え］** をクリックします。

   次の画像は、Mapping Designer の **［詳細に切り替え］** ボタンを示しています。



3. **［詳細に切り替え］** ダイアログボックスで、**［詳細に切り替え］** をクリックします。

   Mapping Designer は、詳細モードで使用できるトランスフォーメーションや関数が表示されるようにマッピングキャンバスを更新します。
4. マッピングの名前、場所、説明を入力します。
5. ソーストランスフォーメーションの全般プロパティで、名前と説明を指定します。
6. **［ソース］** タブで、次の手順を実行して、ソースからデータを読み込むためのソース詳細を入力します。
   a. **［接続］** フィールドで、必要なソース接続を選択します。
   b. **［ソースタイプ］** フィールドで、ソースのタイプを選択します。
   c. **［オブジェクト］** フィールドで、必要なオブジェクトを選択します。
   d. **［詳細プロパティ］** セクションで、適切な値を入力します。

   注: フラットファイルを読み取る場合は、ファイルにヘッダー行があることを確認してください。ファイルにヘッダー行がない場合、Secure Agent による最初の行の読み取りが失敗します。
7. **［フィールド］** タブで、ソースフィールドをターゲットフィールドにマッピングします。
8. ターゲットトランスフォーメーションの全般プロパティで、名前と説明を指定します。
9. **［ターゲット］** タブで、次の手順を実行して、Microsoft Azure Data Lake Storage Gen2 ターゲットにデータを書き込むためのターゲット詳細を入力します。
   a. **［接続］** フィールドで、Microsoft Azure Data Lake Storage Gen2 ターゲット接続を選択します。
   b. **［ターゲットタイプ］** フィールドで、ターゲットのタイプを選択します。
   c. **［オブジェクト］** フィールドで、必要なオブジェクトを選択します。
   d. **［操作］** フィールドで、必要な操作を選択します。
   e. **［詳細プロパティ］** セクションで、詳細プロパティに適切な値を指定します。
10. ソースおよびターゲットをマッピングします。
11. **［保存］** > **［実行］** をクリックして、マッピングを検証します。

    ［監視］で、タスクの実行後にログのステータスを監視できます。

マッピングのルールおよびガイドライン

マッピングについてのルールとガイドラインは次のとおりです。

マッピング

- マッピングを設定する際、ディレクトリ名、ファイル名、またはオブジェクトパスでは、Microsoft Azure Data Lake Storage Gen2 で許可されている特殊文字のみを使用するようにしてください。Microsoft Azure Data Lake Storage Gen2 で許可されている特殊文字の詳細については、Microsoft Azure のドキュメントを参照してください。
- マッピング内のネストされた、またはインデントされた複数行の JSON ファイルのプリミティブなデータ型の読み取りおよび書き込みを行うことはできません。

詳細モードのマッピング

- Microsoft Azure Data Lake Storage Gen2 との間でデータの読み取りと書き込みを行う際は、ソース接続とターゲット接続の両方に同じストレージアカウントを使用します。
  異なるストレージアカウントを使用する場合は、一方のアカウントに共有認証を使用し、もう一方のアカウントにサービスプリンシパル認証を使用します。両方のストレージアカウントに同じ認証タイプを使用することはできません。
- Microsoft Azure Data Lake Storage Gen2 との間でデータの読み取りと書き込みを行い、共有キー認証を使用する場合は、ソース接続とターゲット接続の両方に同じアクセスキーを使用するようにします。
  例えば、ソース接続のアクセスキーとしてキー 1 を使用し、ターゲット接続のアクセスキーとしてキー 2 を使用する場合、マッピングは失敗します。
- マネージド ID 認証を使用する場合、システム割り当て ID は使用できません。
- マネージド ID 認証を使用する場合は、接続で指定されたストレージアカウントが、Azure クラスタのステージングの場所とログの場所で指定されたストレージアカウントと同じではないことを確認してください。
- Avro、JSON、および Parquet のファイルについては、階層データ型の読み取りと書き込みが可能です。ORC ファイルについては、階層データ型の読み取りも可能です。
- 修飾子モードを［最小］に設定してエスケープ文字を使用すると、ターゲットではエスケープ文字はエスケープされず、引用符で囲まれます。この問題を解決するには、修飾子モードを［すべて］に設定します。
- 修飾子モードを［すべて］に設定し、修飾子の値を指定しない場合、\00（Null）が修飾子と見なされます。
- マッピングで複数のパイプラインを追加することはできません。
- サイズが 128 MB 以上の複合ファイルソースから読み取る場合、Secure Agent により誤ったデータが書き込まれ、既存のターゲットが上書きされずに複数のターゲットファイルが作成されます。
- ［実行時にパラメータのオーバーライドを許可する］オプションを使用して、パラメーターファイルで接続またはオブジェクトをオーバーライドすることはできません。
- マッピングの詳細プロパティに入出力パラメータを使用することはできません。
- ソースのスキーマファイルをアップロードし、実行時に Microsoft Azure Data Lake Storage Gen2 ターゲットを作成するときは、ソースファイルが空でないことを確認してください。
- 既存のターゲットにデータを追加するときは、詳細ターゲットプロパティでオーバーライドを設定する必要があります。設定しないと、マッピングが失敗します。
- 実行時に作成されたターゲットにデータを追加する場合、同じ名前のファイルがターゲットディレクトリに存在すると、マッピングは失敗します。
  この場合、最初に既存のファイルを上書きしてから、データを追加する必要があります。

- JSON ファイルに空の構造データを含むフィールドがある場合、Secure Agent はそのフィールドを無視し、メタデータの読み取り中に残りのフィールドを読み取ります。
  例えば、JSON ファイルの最初の行にデータ`{"id":123,"address":{}}`がある場合、`address` フィールドは無視され、**[フィールド]** タブには表示されません。JSON ファイルで連続する行の `address` フィールドに値がある場合、**[サンプリングするデータ要素]** プロパティを使用してこのフィールドを取得できます。

第 4 章

マッピングの移行

ある環境で接続とマッピングを設定した後で、別の環境に移行してマッピングを実行することができます。

詳細モードで設定したマッピングを移行することもできます。移行後は、Administrator サービスから接続プロパティを変更できますが、マッピングを変更する必要はありません。データ統合は、以前の環境で設定されたランタイム属性を使用して、新しい環境でマッピングを正常に実行します。

開発組織（組織 1）でマッピングを開発し、次にプロダクション組織（組織 2）でマッピングを移行して実行するシナリオについて考えてみます。移行後に、組織 2 では同じまたは異なる接続エンドポイントあるいはオブジェクトパスを使用したい場合があるでしょう。要件に応じて、移行を計画する前に、この節のガイドラインを考慮してください。

移行されたマッピングへの同じオブジェクトパスの使用

組織 2 に移行されたマッピングで組織 1 と同じオブジェクトパスを使用する場合は、組織 2 の Microsoft Azure Data Lake Storage Gen2 アカウントで同じファイルシステム、ディレクトリ、およびファイル名を使用する必要があります。

例えば、組織 1 に使用されるアカウント 1 と組織 2 に使用されるアカウント 2 という 2 つの異なるアカウントがある場合、ファイルシステム、ディレクトリ、およびファイル名のオブジェクトパスは、両方のアカウントで同じである必要があります。

アカウント 1: FileSystem1/Directory1/FileName1

アカウント 2: FileSystem1/Directory1/FileName1

このシナリオでは、詳細プロパティのファイルシステム、ディレクトリ、およびファイル名をオーバーライドする必要はなく、マッピングは正常に実行されます。

移行されたマッピングへの別のオブジェクトパスの使用

マッピングを移行した後に、別のオブジェクトパスを使用して、新しい環境からマッピングを実行できます。

このシナリオでは、マッピングを移行する前に、オブジェクトメタデータ、ランタイム属性、または接続属性を変更して、移行された環境にオブジェクトパスを反映させることができます。新しい環境でマッピングを編集または更新する必要はありません。

原則として、詳細プロパティ、接続、またはオブジェクトのプロパティでファイルシステム、ディレクトリ、およびファイル名を指定すると、データ統合は次の優先順位で属性を使用します。

1. **ランタイムの詳細属性**。マッピングのソースまたはターゲットトランスフォーメーションのファイルシステム、ディレクトリ、ファイル名などの詳細プロパティ。
2. **接続属性**。接続プロパティで設定されたファイルシステムやディレクトリなどの属性。
3. **オブジェクトメタデータ**。マッピングのソースまたはターゲットトランスフォーメーションで選択されたオブジェクト。

移行オプション

移行を行うときは、次のオプションのいずれかを選択して、オブジェクトパスを更新できます。

オプション 1: 接続プロパティを更新して、新しいオブジェクトを参照する

マッピングを組織 2 にインポートすると、**[接続の確認]** セクションで既存の接続を変更して、指定された組織 2 のファイルシステム、ディレクトリ、およびファイル名にアクセス可能な接続にマッピングできます。

オプション 2: 詳細プロパティからプロパティをオーバーライドする

移行の前に、組織 1 のマッピングの詳細プロパティで、組織 2 のオブジェクトに必要なファイルシステム、ディレクトリ、およびファイル名を指定します。

移行後にマッピングを実行すると、Secure Agent は設定された詳細パラメータを使用して、指定されたオブジェクトを組織 1 からインポートされたマッピングでオーバーライドします。

オプション 3: 詳細プロパティをパラメータ化する

移行前に、ファイルシステム、ディレクトリ、ファイル名などの詳細属性をパラメータ化するという選択ができます。マッピングでは、入力パラメータ、入出力パラメータ、およびパラメータファイルを設定できます。マッピングを移行した後は、マッピングを編集または更新しないでください。ファイルシステム、ディレクトリ、ファイル名などの詳細属性に入出力パラメータを使用した場合は、パラメータファイルからこれらのパラメータを更新できます。

マッピング内のオブジェクトではなく、詳細プロパティのみをパラメータ化する

詳細プロパティのみをパラメータ化し、実行時にそれらのプロパティを使用する場合は、マッピングのオブジェクトプロパティでプレースホルダオブジェクトを選択してから、詳細プロパティからこのプレースホルダオブジェクトへのオーバーライドを指定します。プレースホルダオブジェクトに、オーバーライドとして指定した対応するテーブルと同じメタデータが含まれていることを確認してください。マッピングを実行すると、詳細プロパティで指定した値でプレースホルダオブジェクトがオーバーライドされます。

オブジェクトと詳細プロパティをパラメータ化する

Microsoft Azure Data Lake Storage Gen2 オブジェクトタイプと詳細フィールドをパラメータ化したままにする場合は、パラメータの追加時に入力パラメータウィンドウで **[実行時にパラメータのオーバーライドを許可]** オプションが選択されていない状態で、必要なオブジェクトをタスクレベルで選択します。タスクを実行すると、詳細プロパティで指定した値が優先されます。

パラメータ化ルール

オブジェクトと詳細プロパティをパラメータ化する場合は、次のルールを考慮してください。

- パラメータ化は、**[ターゲットの作成]** オプションを使用するマッピングには適用されません。
- パラメータ化は、移行する場合の詳細モードのマッピングには適用されません。
- パラメータファイルが Microsoft AzureData Lake Storage Gen2 のクラウドでホストされているディレクトリに保存されている場合は、移行後にディレクトリパスまたはパラメータファイル名を変更しないでください。

- マッピングに複数のパイプラインが設定されている場合は、Microsoft Azure Data Lake Storage Gen2 オブジェクトをパラメータ化しないでください。移行する前に、マッピングの作成時にプレースホルダオブジェクトを選択する必要があります。

一般的なルールとガイドライン

マッピングを移行するときは、次のルールとガイドラインを考慮してください。

- マッピングの移行時に、インポートしたオブジェクトのメタデータがオーバーライドで取得されたメタデータと一致することを確認してください。
- マッピングの移行時に、［ディレクトリのオーバーライド］フィールドに相対ディレクトリを指定しないでください。マッピングを移行して実行すると、設計時にオブジェクトをインポートするために使用したパスが、実行時にメタデータを取得するときに指定したファイルパスに追加されます。
- 実行時にターゲットを作成し、動的スキーマ処理オプションを有効にするようにマッピングを設定して、移行時にデータベースで［ファイル名］フィールドを削除したマッピングのソース接続を変更すると、削除済みであっても［ファイル名］フィールドが引き続きターゲットに書き込まれます。
  この問題を解決するには、ターゲットトランスフォーメーションから［ファイル名］フィールドを除外する必要があります。
- 移行されたマッピングに別のオブジェクトパスを使用すると、データ統合は、次の場合に実行時に取得されたメタデータの代わりに、インポートされたオブジェクトメタデータを使用します。

  - 事前にソースプロパティで Gzip 圧縮が有効になっている。

  - ファイル名フィールドがマッピングされている。
- ［再帰的ディレクトリ読み取り］オプションを有効にし、第 1 レベルのディレクトリに複数のフォルダが含まれているもののファイルが含まれていない場合、マッピングを移行して実行した後、データ統合は実行時に取得されたメタデータの代わりに、インポートされたオブジェクトメタデータを使用します。

第 5 章

データ型リファレンス

Cloud データ統合は、Microsoft Azure Data Lake Storage Gen2 のマッピングとマッピングタスクで次のデータ型を使用します。

- フィールドのメタデータを編集しようとすると、Microsoft Azure Data Lake Storage Gen2 のネイティブなデータ型がソーストランスフォーメーションとターゲットトランスフォーメーションに表示されます。
- トランスフォーメーションデータ型。トランスフォーメーションで表示されるデータ型のセットです。ANSI SQL-92 汎用データ型に基づく内部データ型で、プラットフォーム間でデータを移動するとき Secure Agent によって使用されます。マッピング内のすべてのトランスフォーメーションでこのデータ型が表示されます。

Secure Agent は、ソースデータを読み取るときに、ネイティブデータ型を対応するトランスフォーメーションデータ型に変換してから、データのトランスフォームを実行します。Secure Agent は、ターゲットに書き込むときに、トランスフォーメーションデータ型を対応するネイティブデータ型に変換します。

次の表は、Data Integration でサポートされる Microsoft Azure Data Lake Storage Gen2 データ型と、対応するトランスフォーメーションデータ型を示しています。

Microsoft Azure Data Lake Storage Gen2 **ネイティブデータ型**	**トランスフォーメーションデータ型**	**説明**
String	String	1〜104,857,600 文字

フラットファイルのデータ型とトランスフォーメーションデータ型

フラットファイルのデータ型は、Secure Agent がプラットフォーム間でデータを移動するために使用するトランスフォーメーションデータ型にマッピングされます。

次の表に、Secure Agent がサポートしているフラットファイルのデータ型と、対応するトランスフォーメーションデータ型の一覧を示します。

フラットファイルのデータ型	トランスフォーメーションデータ型（マッピング）	トランスフォーメーションデータ型（詳細モードのマッピング）	範囲と内容
BigInt	該当なし	Long	-9,223,372,036,854,775,808～9,223,372,036,854,775,807 文字。精度 19、スケール 0
Nstring*	テキスト	Text	1～104,857,600 文字
Number*	Decimal	Double	精度 1～28 桁、スケール 0～28 桁
String*	String	String	1～104,857,600 文字。精度 256。
*Number、String、または Nstring データ型のデータを読み取るには、**［スキーマソース］**を**［スキーマファイルからのインポート］**として選択する必要があります。			

Avro のデータ型とトランスフォーメーションデータ型

Avro ファイルのデータ型は、Secure Agent がプラットフォーム間でデータを移動するために使用するトランスフォーメーションデータ型にマップされます。

次の表に、Secure Agent がサポートしている Avro ファイルのデータ型と、対応するトランスフォーメーションデータ型の一覧を示します。

Avro データ型	トランスフォーメーションデータ型	範囲と内容
Array[1]	配列	文字数に制限はありません
Boolean	Integer	1 または 0 True は Integer の 1 に相当し、False は Integer の 0 に相当します。
Byte	Binary	精度 4000
Double	Double	精度 15
Float	Double	精度 15
Int	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0

Avro データ型	トランスフォーメーションデータ型	範囲と内容
Long	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 精度 19、位取り 0
Map[1]	マップ	文字数に制限はありません
Null	Integer	-2,147,483,648～2,147,483,647 精度 10、位取り 0
Record[1]	構造体	文字数に制限はありません
String	String	1～104,857,600 文字 精度 4000
Union[1]	["primitive_type|complex_type", "null"]または["null", "primitive_type|complex_type"]の共有体で表される対応するデータ型。	プリミティブデータ型または複合データ型に依存します。
[1] 詳細モードのマッピングにのみ適用されます。		

JSON のデータ型とトランスフォーメーションデータ型

JSON ファイルデータ型は、Secure Agent がプラットフォーム間でデータを移動するために使用するトランスフォーメーションデータ型にマッピングされます。

次の表に、Secure Agent がサポートしている JSON ファイルデータ型と、対応するトランスフォーメーションデータ型の一覧を示します。

JSON データ型	トランスフォーメーションデータ型	範囲と内容
Array[1]	配列	文字数に制限はありません
ブール	整数	Boolean のデフォルトのトランスフォーメーション型は Integer です。True および False の値を持つ String データ型を指定することもできます。True は Integer の 1 に相当し、False は Integer の 0 に相当します。
Number（double）	double	-1.79769313486231570E+308～1.79769313486231570E+308。精度 15。
Number（float）	double	-1.79769313486231570E+308～1.79769313486231570E+308。精度 15。
Number（int）	integer	-2,147,483,648～2,147,483,647。精度 10、位取り 0。

JSON データ型	トランスフォーメーションデータ型	範囲と内容
Number（long）	bigint	-9,223,372,036,854,775,808～9,223,372,036,854,775,807。精度 19、位取り 0。
Object[1]	構造体	文字数に制限はありません
string	string	1～104,857,600 文字。 精度 4000
[1] 詳細モードのマッピングにのみ適用されます。		

ORC のデータ型とトランスフォーメーションデータ型

ORC ファイルのデータ型は、Secure Agent がプラットフォーム間でデータを移動するために使用するトランスフォーメーションデータ型にマップされます。

次の表に、Secure Agent がサポートしている ORC ファイルのデータ型と、対応するトランスフォーメーションデータ型の一覧を示します。

ORC ファイルのデータ型	トランスフォーメーションデータ型	範囲と内容
BigInt	BigInt	-9223372036854775808～9,223,372,036,854,775,807
Boolean	Integer	1 または 0 True は Integer の 1 に相当し、False は Integer の 0 に相当します。
Char	String	1～104,857,600 文字
Date	日付/時刻	西暦 1753 年 1 月 1 日～西暦 4712 年 12 月 31 日 （精度はマイクロ秒まで）
Double	ダブル	精度 15 桁
Float	ダブル	精度 15 桁
Integer	Integer	-2,147,483,648～2,147,483,647
SmallInt	Integer	-32,768～32,767

ORC ファイルのデータ型	トランスフォーメーションデータ型	範囲と内容
String	String	1〜104,857,600 文字 精度 4000
タイムスタンプ	Date/Time	1〜19 文字 精度 19〜26、位取り 0〜6
TinyInt	Integer	-128〜127
Varchar	String	1〜104,857,600 文字

Parquet のデータ型とトランスフォーメーションデータ型

Parquet ファイルのデータ型は、Secure Agent がプラットフォーム間でデータを移動するために使用するトランスフォーメーションデータ型にマップされます。

次の表に、Secure Agent がサポートしている Parquet ファイルのデータ型と、対応するトランスフォーメーションデータ型の一覧を示します。

Parquet データ型	トランスフォーメーションデータ型	範囲と内容
Boolean	Integer	1 または 0 True は Integer の 1 に相当し、False は Integer の 0 に相当します。
Byte_Array	Binary	任意の長バイト配列
Date	日付/時刻	0001 年 1 月 1 日〜9999 年 12 月 31 日。
Decimal	Decimal	精度 1〜28 桁、位取り 0〜28。 28 を超える精度を持つ 10 進数値は使用できません。
Double	Double	精度 15
Float	Double	精度 15
group(LIST)[1]	Array	文字数に制限はありません。
Int32	Integer	-2,147,483,648〜+2,147,483,647

Parquet データ型	トランスフォーメーションデータ型	範囲と内容
Int64	Bigint	-9,223,372,036,854,775,808 〜 +9,223,372,036,854,775,807 8 バイト符号付き整数
Int96	Binary	12 バイト符号付き整数 詳細モードのマッピングを設定すると、データプレビューに Secure Agent マシンのタイムゾーンが表示されます。
Map[1]	Map	文字数に制限はありません。
String	String	1〜104,857,600 文字 精度 4000
Struct[1]	Struct	文字数に制限はありません。
Time	Date/Time	1 日のうちの時間。精度はマイクロ秒まで。
タイムスタンプ	Date/Time	0001 年 1 月 1 日 00:00:00〜9999 年 12 月 31 日 23:59:59.997。精度はマイクロ秒まで。 精度をナノ秒に設定することはできません。
[1] 詳細モードのマッピングにのみ適用されます。		

Parquet ファイルの読み取りまたは書き込みを行うために指定する Parquet スキーマは小文字である必要があります。Parquet は大文字と小文字を区別するスキーマではありません。

Parquet の Timestamp データ型のサポート

次の表に、Parquet ファイル形式でサポートされている Timestamp データ型の一覧を示します。

タイムスタンプデータ型	マッピング	詳細モードのマッピング
Timestamp_micros	はい	いいえ
Timestamp_millis	はい	いいえ
Time_millis	はい	いいえ
Time_micros	はい	いいえ
int96	○	はい

Secure Agent でサポートされていない Parquet のデータ型は以下のとおりです。

- Timestamp_nanos

- Time_nanos
- Timestamp_tz

第 6 章

トラブルシューティング

マッピングでのエラーのトラブルシューティングには、以下のセクションを使用します。

マッピングのトラブルシューティング

Java **ヒープ領域エラーが発生してマッピングが失敗する**

大きなデータセットを MicrosoftAzure Data Lake Storage Gen2 から読み取ったり、Microsoft Azure Data Lake Storage Gen2 に書き込んだりすると、特定のマッピングが次のエラーで失敗する場合があります。

```
[ERROR] java.lang.OutOfMemoryError: Java heap space
```

マッピングを正常に実行するには、ヒープサイズを増やす必要があります。推奨されるヒープサイズは 1GB です。

以下の手順を実行して Secure Agent の JVM オプションを設定し、Java ヒープサイズのメモリを増やします。

1. **［Administrator］** > **［ランタイム環境］** を選択します。
2. **［ランタイム環境］** ページで、利用可能な Secure Agent のリストからメモリを増加する Secure Agent を選択します。
3. 右上隅の **［編集］** をクリックします。
4. **［システム構成の詳細］** セクションで、サービスとして **［データ統合サーバー］** を選択し、タイプとして **［DTM］** を選択します。
5. ［JVMOption］プロパティを編集し、「*-Xmx1024m*」と入力します。

   注: 推奨されるヒープサイズは 1GB です。処理するデータに基づいて、ヒープサイズを増やすことができます。
6. **［保存］** をクリックします。

Unicode **文字列がターゲットに正しく書き込まれない**

Linux マシンで Microsoft Azure Data Lake Storage Gen2 マッピングを実行し、ソース接続とオブジェクトをパラメータ化し、既存のターゲットにデータを追加すると、データ内の Unicode 文字列がターゲットに正しく書き込まれません。

この問題を解決するには、環境変数 `LC_ALL="en_US.UTF-8"`を Linux マシンの Secure Agent で設定し、Secure Agent を再起動します。

ディレクトリ名またはサブディレクトリ名に Unicode **文字が含まれている場合にマッピングが失敗する**

Microsoft Azure Data Lake Storage Gen2 でフラットファイルの読み取りまたは書き込みを行った場合、ディレクトリ名またはサブディレクトリ名に Unicode 文字が含まれていると、マッピングが失敗します。

この問題を解決するには、環境変数 `LC_ALL = "en_US.UTF-8"`を Secure Agent で設定し、Secure Agent を再起動します。

詳細モードのマッピングのトラブルシューティング

Avro ファイルまたは Parquet ファイルの Date および Int96 データ型を読み書きするように設定されたマッピングが失敗する

Avro ファイルまたは Parquet ファイルとの間で読み書きするように設定されたマッピングは、次の場合に失敗します。

- データが Date データ型で、日付が 1582-10-15 より前の場合。
- データが Int96 データ型で、タイムスタンプが 1900-01-01T00:00:00Z より前の場合。

この問題を解決するには、マッピングタスクまたは Secure Agent のカスタムプロパティファイルで、次の Spark セッションプロパティを指定します。

- `spark.sql.legacy.timeParserPolicy=LEGACY`
- `spark.sql.parquet.int96RebaseModeInWrite=LEGACY`
- `spark.sql.parquet.datetimeRebaseModeInWrite=LEGACY`
- `spark.sql.parquet.int96RebaseModeInRead=LEGACY`
- `spark.sql.parquet.datetimeRebaseModeInRead=LEGACY`
- `spark.sql.avro.datetimeRebaseModeInWrite=LEGACY`
- `spark.sql.avro.datetimeRebaseModeInRead=LEGACY`

索引

A

Azure Data Lake Storage Gen2
　接続プロパティ 12

C

Cloud Application Integration コミュニティ
　URL 5
Cloud 開発者コミュニティ
　URL 5

I

Informatica Intelligent Cloud Services
　Web サイト 5
Informatica グローバルカスタマサポート
　連絡先情報 6

J

JSON ファイルデータ型
　トランスフォーメーションデータタイプ 46

L

Linux
　プロキシの設定 15

M

Microsoft Azure Data Lake Storage Gen2
　ソーストランスフォーメーション 17
　ターゲットトランスフォーメーション 23
　マッピングでのソース 17
　マッピングのターゲット 23
Microsoft Azure Data Lake Storage Gen2 コネクタ
　概要 7
　例 7
Microsoft Azure Data Lake Storage Gen2 接続
　概要 12

O

ORC ファイルのデータ型
　トランスフォーメーションデータタイプ 47

W

Web サイト 5
Windows
　プロキシの設定 14

あ

アップグレード通知 6

し

システムステータス 6

す

ステータス
　Informatica Intelligent Cloud Services 6

そ

ソース
　マッピングでの Microsoft Azure Data Lake Storage Gen2 17
ソーストランスフォーメーション
　Microsoft Azure Data Lake Storage Gen2 プロパティ 17

た

ターゲット
　マッピングでの Microsoft Azure Data Lake Storage Gen2 23
ターゲットトランスフォーメーション
　Microsoft Azure Data Lake Storage Gen2 プロパティ 23
ターゲットの作成
　ターゲットファイルのパラメータ化 27, 28
　タイムスタンプの追加 27
ターゲットの指定 26

て

ディレクトリソース
　Microsoft Azure Blob Storage ソース 20
データ型
　Avro 45
　Parquet 48
データ型リファレンス
　概要 44

は

パラメータファイルを使用したパラメータ化 28

ふ

ファイルの増分ロード
　概要 22
プロキシ設定
　JVMOptions 14
　Linux での設定 15
　Windows 上での設定 14

ま

マッピング
　Microsoft Azure Data Lake Storage Gen2 ソースプロパティ 17
　Microsoft Azure Data Lake Storage Gen2 ターゲットプロパティ 23

め

メンテナンスの停止 6

わ

ワイルドカード文字
　概要 20