



Informatica® Intelligent Cloud Services  
April 2024

Secure Agent サービス

Informatica Intelligent Cloud Services Secure Agent サービス  
April 2024

© 著作権 Informatica LLC 2021, 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 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: 2024-06-17

# 目次

<b>序文</b> .....	6
Informatica のリソース.....	6
Informatica マニュアル.....	6
Informatica Intelligent Cloud Services Web サイト.....	6
Informatica Intelligent Cloud Services コミュニティ.....	6
Informatica Intelligent Cloud Services マーケットプレイス.....	7
データ統合のコネクタのドキュメント.....	7
Informatica ナレッジベース.....	7
Informatica Intelligent Cloud Services Trust Center.....	7
Informatica グローバルカスタマサポート.....	7
<b>第 1 章 : Secure Agent サービス</b> .....	8
カスタム環境変数の設定.....	10
<b>第 2 章 : API Microgateway Service</b> .....	11
API Microgateway サービスのプロパティの編集.....	12
Secure Agent または Secure Agent グループでの API Microgateway サービスの有効化.....	13
<b>第 3 章 : CMI ストリーミングエージェント</b> .....	15
CMI ストリーミングエージェントのプロパティ.....	15
<b>第 4 章 : 共通統合コンポーネント</b> .....	18
共通統合コンポーネントプロパティ.....	18
<b>第 5 章 : データベース取り込みサービス</b> .....	21
データベース取り込みサービスのプロパティ.....	21
データベース取り込みエージェントの環境変数.....	25
<b>第 6 章 : データ統合サーバー</b> .....	26
データ統合サーバーの回復機能.....	26
データ統合サーバーのプロパティ.....	27
OSProfileUserMappingFile の作成.....	31
OSProfileScriptForTaskExecution の設定.....	32
データ統合サーバーアップグレード.....	33
<b>第 7 章 : エラスティックサーバー</b> .....	34
エラスティックサーバーのプロパティ.....	34
エラスティックサーバーの並行処理.....	36
<b>第 8 章 : ファイル統合サービス</b> .....	38

<b>第 9 章 : GitRepoConnectApp</b> .....	39
ローカルリポジトリのベースディレクトリ.....	39
GitRepoConnectApp のプロパティ.....	40
<b>第 10 章 : IDMC Data Gateway Service</b> .....	42
IDMC Data Gateway Service のプロパティ.....	42
<b>第 11 章 : 一括取り込み (ファイル)</b> .....	48
<b>第 12 章 : メタデータ基盤アプリケーション</b> .....	52
メタデータ基盤アプリケーションのプロパティ.....	52
<b>第 13 章 : メタデータプラットフォームサービス</b> .....	58
メタデータプラットフォームサービスのプロパティ.....	58
<b>第 14 章 : プロセスサーバー</b> .....	65
プロセスサーバーのプロパティ.....	65
デフォルト接続データベースのプロパティ.....	71
ログレベル.....	72
別のログデータソースの設定.....	72
プロセスサーバーのサイズ決定に関する推奨事項.....	73
Secure Agent との通信.....	75
プロセスサーバーのための Secure Agent の設定.....	76
単一の Secure Agent へのデプロイ.....	77
Secure Agent グループへのデプロイ.....	78
PostgreSQL データベースのインストールとアップグレードの前提条件.....	80
Windows での PostgreSQL データベースの管理.....	81
Windows での PostgreSQL データベースのバックアップ.....	81
Windows での PostgreSQL データベースのリストア.....	81
Windows での PostgreSQL データベースのリセット.....	82
Windows での PostgreSQL サーバーの起動.....	82
Windows での PostgreSQL サーバーの停止.....	82
Windows での PostgreSQL サーバーステータスの取得.....	82
Windows での PostgreSQL データベースのクリーンアップ.....	83
Windows での PostgreSQL データベースの再インデックス化.....	83
Windows でのトランザクションログのリセット.....	83
Linux での PostgreSQL データベースの管理.....	84
Linux での PostgreSQL データベースのバックアップ.....	84
Linux での PostgreSQL データベースのリストア.....	84
Linux での PostgreSQL データベースのリセット.....	85
Linux での PostgreSQL サーバーの起動.....	85
Linux での PostgreSQL サーバーの停止.....	85

Linux での PostgreSQL サーバーステータスの取得. . . . .	85
Linux での PostgreSQL データベースのクリーンアップ. . . . .	86
Linux での PostgreSQL データベースの再インデックス化. . . . .	86
Linux でのトランザクションログのリセット. . . . .	86
PostgreSQL データベースのアップグレード. . . . .	87
レプリケーション技術を使用した PostgreSQL データベースのアップグレード. . . . .	87
PostgreSQL 構成ファイル. . . . .	88
PostgreSQL ログローテーションの設定. . . . .	88
プロセスサーバーに対するパブリック証明書とプライベートキーの設定. . . . .	89
スループットを向上させるためのスレッドプールプロファイルの設定. . . . .	91
platform.yaml ファイル内のプロパティのオーバーライド. . . . .	92
カスタム user-platform.yaml ファイルの作成. . . . .	92
トラブルシューティング. . . . .	94
<b>第 15 章 : SecretManagerApp. . . . .</b>	<b>95</b>
<b>第 16 章 : Secure Agent サービスプロパティの設定. . . . .</b>	<b>96</b>
<b>索引. . . . .</b>	<b>98</b>

# 序文

「Secure Agent サービス」を使用して、Informatica Intelligent Cloud Services<sup>SM</sup> Secure Agent がデータ処理に使用するマイクロサービスについて確認します。サービスプロパティを設定する方法を確認します。

## 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](mailto:infa_documentation@informatica.com)) までご連絡ください。

### Informatica Intelligent Cloud Services Web サイト

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

### Informatica Intelligent Cloud Services コミュニティ

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

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

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

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

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

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

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

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

## データ統合のコネクタのドキュメント

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

## Informatica ナレッジベース

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

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

## Informatica Intelligent Cloud Services Trust Center

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

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

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

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

## Informatica グローバルカスタマサポート

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

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

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

# 第 1 章

## Secure Agent サービス

Secure Agent サービスは、Secure Agent がデータ処理に使用するプラグブルマイクロサービスです。例えば、Secure Agent はデータ統合サーバーを使用してデータ統合ジョブを実行し、プロセスサーバーを使用してアプリケーション統合を実行してオーケストレーションジョブを処理します。各 Secure Agent サービスは、エージェントで実行されている他のサービスとは独立して実行されます。

独立したサービスアーキテクチャには、次の利点があります。

- コネクタまたはパッケージを追加したときに、Secure Agent が再起動しない。
- サービスは、別のサービスの再起動時に影響を受けない。例えば、データ統合サーバーを再起動しても、プロセスオーケストレーションジョブは引き続き実行されます。
- アップグレード中のダウンタイムは最小化されます。アップグレードプロセスにより、Secure Agent の新しいバージョンをインストールし、コネクタパッケージを更新して、データ統合サーバーとデータベース取り込みエージェントサービスの設定の変更を適用します。ダウンタイムを最小化するために、古いエージェントは引き続き使用可能なままで、アップグレード中にデータ統合ジョブを実行し続けます。Secure Agent の新しいバージョンは、アップグレードプロセスの完了後に開始されるジョブを実行します。

Secure Agent で実行されるサービスは、Secure Agent グループで有効なライセンスと Informatica Intelligent Cloud Services によって異なります。

以下の表に、エージェントで実行される Secure Agent サービスと、それらのサービスを使用する Informatica Intelligent Cloud Services を示します。

Secure Agent サービス	説明	次により使用
API Microgateway サービス	Secure Agent で実行されるアプリケーションの統合プロセスを管理します。	アプリケーションの統合、API マネージャ
B2B プロセッサ	B2B Gateway のインバウンドおよびアウトバウンドプロセスフローを実行します。 注: Informatica グローバルカスタマサポートから指示された場合を除き、このサービスの値は変更しないでください。	B2B Gateway
CIH プロセッサ	プライベートパブリケーションリポジトリを使用する組織のために、Cloud Integration Hub のパブリケーションおよびサブスクリプションを実行します。	Cloud Integration Hub
CMI ストリーミングエージェント	一括取り込みサービスでストリーミング取り込みジョブを実行します。	一括取り込みサービス
共通統合コンポーネント	シェルスクリプトまたはバッチコマンドをタスクフローのコマンドタスクステップで実行します。	データ統合



Secure Agent サービス	説明	次により使用
データベース取り込み	一括取り込みサービスでアプリケーション取り込みジョブとデータベース取り込みジョブを実行します。	一括取り込みサービス
データ統合サーバー	マッピング、タスク、およびタスクフローインスタンスなどのデータ統合ジョブを実行します。	B2B Gateway Cloud Integration Hub、Azure のデータアクセラレータ、データ統合、データプロファイリング
ドメイン管理アプリ	オンプレミスの CDI-PC ドメインを Informatica Intelligent Cloud Services に接続し、バックエンドのドメイン更新タスクを処理します。	PowerCenter 用クラウドデータ統合 (CDI-PC)
EDC 検索エージェント	Azure のデータアクセラレータのため、およびデータ統合でのデータカタログ検出のために、Enterprise Data Catalog データアセットを検出します。 <b>注:</b> Informatica グローバルカスタマサポートから指示された場合を除き、このサービスの値は変更しないでください。	Azure のデータアクセラレータ、データ統合
エラスティックサーバー	詳細クラスタおよびそのクラスタで実行するジョブを管理します。	データ統合
ファイル統合サービス	リモートサーバーとのファイルの送信または受信、あるいはその両方に、HTTPS、AS2 および SFTP などのファイル転送プロトコルを使用します。	B2B Gateway、データ統合
GitRepoConnectApp	組織がオンプレミスのソース管理リポジトリを使用している場合、Informatica Intelligent Cloud Services とソース管理リポジトリ間の通信を管理します。	ソース管理を使用するすべての Informatica Intelligent Cloud Services
IDMC Data Gateway Service	CLAIRE GPT でデータ探索タスクを実行できます。ソースデータを探索し、データセットのサンプルをプレビューして、サンプルデータの取得に使用された SQL コードを確認し、後で参照できるようにサンプルデータを CSV ファイルに保存することができます。	CLAIRE GPT
一括取り込み	ファイル取り込みタスクおよびファイルリスナジョブを実行します。 <b>注:</b> Informatica グローバルカスタマサポートから指示された場合を除き、このサービスの値は変更しないでください。	一括取り込みサービス
メタデータ基盤アプリケーション	組織内で設定されているソースシステムからメタデータを抽出し、抽出したメタデータを Secure Agent を介してメタデータコマンドセンターにアップロードします。	データガバナンスとカタログ、メタデータコマンドセンター
メタデータプラットフォームサービス	メタデータコマンドセンターで実行しているジョブに対してプロファイルアクティビティを実行します。 <b>注:</b> メタデータコマンドセンターのランタイム環境にアクティブなメタデータプラットフォームサービスが存在しない場合、プロファイリングは失敗します。	データガバナンスとカタログ、メタデータコマンドセンター

Secure Agent サービス	説明	次により使用
OI データコレクタ	PowerCenter、Data Engineering Integration、および Data Quality から運用データとドメイン関連のメタデータを収集するオペレーションインサイトデータコレクタを実行します。 注: Informatica グローバルカスタマサポートから指示された場合を除き、このサービスの値は変更しないでください。	オペレーションインサイト
プロセスサーバー	アプリケーション統合プロセス、コネクタ、および接続を実行します。	アプリケーションの統合、アプリケーション統合コンソール
SecretManagerApp	組織が AWS Secrets Manager や Azure Key Vault などの外部シークレットマネージャを使用している場合に、Informatica Intelligent Cloud Services とシークレットマネージャ間の通信を管理します。	データ統合

それぞれの Secure Agent サービスには、Tomcat 設定や Tomcat JRE 設定などの一意的な設定プロパティセットがあります。パフォーマンスを最適化するため、または Informatica グローバルカスタマサポートから指示された場合は、サービスを設定したり、サービスのプロパティを変更しなければならないことがあります。Secure Agent サービスは、エージェントで実行されている他のサービスとは独立して実行されます。

## カスタム環境変数の設定

custom\_env\_settings.sh スクリプトにスクリプトまたは変数を追加することで、環境をカスタマイズできます。

いずれかのサービスにカスタム環境変数を設定する必要がある場合は、それらの変数を custom\_env\_settings.sh という名前のスクリプトファイルに追加し、このファイルを Secure Agent がインストールされているディレクトリに配置します。

例えば、次のスクリプトにより、エージェントのアップグレード後にカスタム変数が自動的に設定されるようになります。

```
$ pwd
/apps/cloudagent/apps/Data_Integration_Server/65.0.3.1/.lcm
$ ls lcm-env.sh
lcm-env.sh
```

## 第 2 章

# API Microgateway Service

API Microgateway Service は、組織のオンプレミスの Secure Agent 上で実行されるアプリケーションの統合プロセスを管理します。API Microgateway サービスを使用して、管理対象 API を API Microgateway プロキシとして公開します。

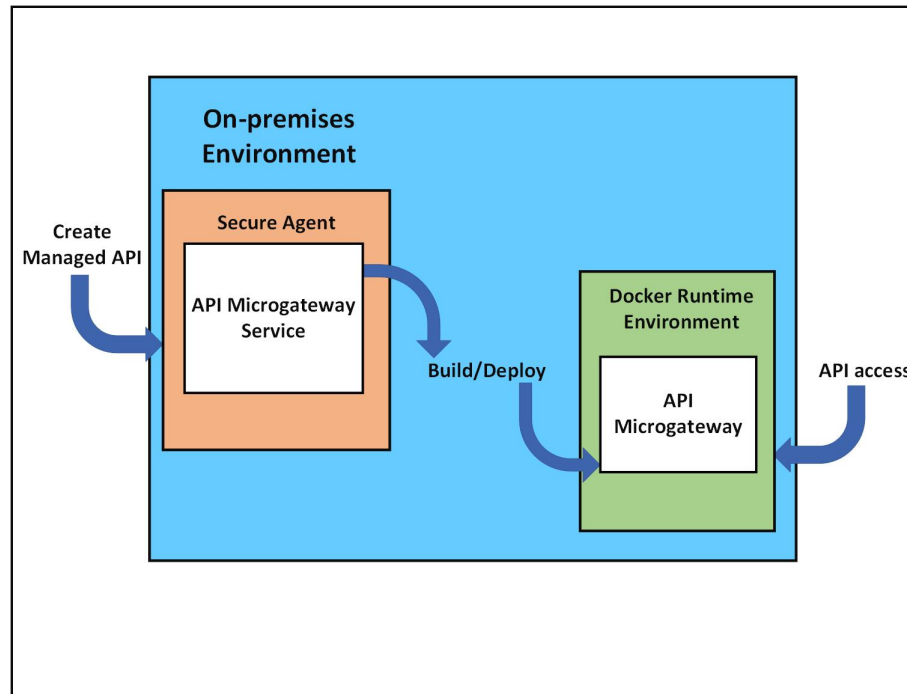
次の方法を使用して、API Microgateway サービスでパブリッシュする管理対象 API へのアクセスを制御できます。

- IP フィルタリングポリシー
- レート制限ポリシー
- 基本認証または OAuth 2.0 認証

API Microgateway サービスは、API Microgateway プロキシを作成およびデプロイするための REST API を提供します。API コンシューマは、組織のオンプレミス環境に API Microgateway プロキシとしてデプロイされた管理対象 API にアクセスします。アプリケーションの統合プロセスは、REST サービスの URL および SOAP サービスの URL のエンドポイントを公開します。

API Microgateway サービスを使用して、管理する API エンドポイントへの API Microgateway プロキシを構築します。API Microgateway Service は、組織の Secure Agent マシン上に不変の Docker イメージとして API Microgateway を構築します。次に、API Microgateway サービスを使用して、API アクセス用の Secure Agent Docker ランタイム環境のコンテナに Docker イメージをデプロイします。API Microgateway は、リクエストをアプリケーションの統合エンドポイントに転送する前に設定した API アクセスポリシーを適用します。

次の図は、オンプレミス環境で管理対象 API を公開する API Microgateway サービスおよび API Microgateway コンポーネントを示しています。



Secure Agent Docker ランタイム環境は、Blue-Green デプロイメントストラテジを使用して Docker イメージをホストし、API Microgateway コンポーネントの更新中のダウンタイムをゼロにします。

## API Microgateway サービスのプロパティの編集

API Microgateway サービスのプロパティは Administrator で編集します。

次の図は、**【システム構成の詳細】** 領域で編集できる API Microgateway サービスのプロパティを示しています。

▼ System Configuration Details

Service:	<input type="text" value="API Microgateway Service"/>	
Type:	<input type="text" value="All Types"/>	
Type	Name	Value
AGENT_RUNTIME_SETTINGS	project-name	'project1'
AGENT_RUNTIME_SETTINGS	docker-registry-name	'info.agent.apimgw'
DOCKER_CONTAINER_SETTINGS	blue	<a href="#">http-port: '16090'</a> <a href="#">https-port: '16095'</a>
DOCKER_CONTAINER_SETTINGS	green	<a href="#">http-port: '17090'</a> <a href="#">https-port: '17095'</a>
DOCKER_CONTAINER_SETTINGS	haproxy	<a href="#">http-port: '6090'</a> <a href="#">https-port: '6095'</a>

次の表に、API Microgateway サービスの設定を示します。

タイプ	名前	説明
AGENT_RUNTIME_SETTINGS	project-name	API 設定を保存するプロジェクトの名前。新しいプロジェクトを作成する場合など、必要に応じて名前を変更できません。 注: プロジェクト名には、/または/0 という文字を含めることはできません。プロジェクト名に制限された文字が含まれている場合、プロジェクトの作成は失敗します。
AGENT_RUNTIME_SETTINGS	docker-registry-name	Secure Agent マシン上の名前付きおよびタグ付きの API Microgateway Docker イメージをすべて含むローカルの Docker レジストリの名前。 注: Docker イメージとタグの名前には、次の文字を含めることはできません: - , _ , . Docker イメージまたはタグ名に制限された文字が含まれている場合、イメージの構築は失敗します。
DOCKER_CONTAINER_SETTINGS	blue	Secure Agent マシンに最初にデプロイされる Docker イメージコンテナ (green と交互になります)。 blue コンテナの次のポートを変更することができます。 - http-port。デフォルト値は 16090 です。 - https-port。デフォルト値は 16095 です。
DOCKER_CONTAINER_SETTINGS	green	Secure Agent マシンに 2 番目にデプロイされる Docker イメージコンテナ (blue と交互になります)。 green コンテナの次のポートを変更することができます。 - http-port。デフォルト値は 17090 です。 - https-port。デフォルト値は 17095 です。
DOCKER_CONTAINER_SETTINGS	haproxy	Secure Agent マシン上の Docker イメージコンテナのルーター。blue と green のコンテナ間でトラフィックを切り替えます。 haproxy コンテナの次のポートを変更することができます。 - http-port。デフォルト値は 6090 です。 - https-port。デフォルト値は 6095 です。

注: API Microgateway を停止するには、3 つの Docker イメージコンテナをすべて停止します。

## Secure Agent または Secure Agent グループでの API Microgateway サービスの有効化

API Microgateway サービスを実行する Secure Agent を変更する場合は、管理者で Secure Agent または Secure Agent グループに対して API Microgateway サービスを有効にします。Secure Agent グループに対して API Microgateway サービスを有効にすると、このサービスは、グループ内のすべての Secure Agent と、後でグループに追加するすべての Secure Agent に対して有効になります。

1. **【ランタイム環境】** ページに移動し、Secure Agent または Secure Agent グループの **【アクション】** メニューから **【サービスの有効化または無効化、コネクタ】** を選択します。  
**【エージェントのコンポーネントの有効化/無効化】** ウィンドウが表示されます。

2. サービスのリストから **【API Microgateway】** を選択し、**【保存】** をクリックします。  
API Microgateway サービスが Secure Agent または Secure Agent グループに対して有効になります。

## 第 3 章

# CMI ストリーミングエージェント

CMI ストリーミングエージェントを使用して、ストリーミング取り込みタスクを定義し、展開します。ストリーミング取り込みタスクを一括取り込みサービスで設定します。

CMI ストリーミングエージェントは、オンプレミスシステムで実行され、一括取り込みストリーミングサービスと連携して動作します。オンプレミスシステムで、CMI ストリーミングエージェントは一括取り込みストリーミングで展開されたジョブを実行します。エージェントは各ジョブのステータスおよび統計情報を更新します。

Linux でエージェントのインストールディレクトリ名にスペースが含まれている場合、CMI ストリーミングエージェントが起動しません。エージェントは接続タイムアウトステータスを返します。再起動を数回試行した後に、エージェントはエラー状態になります。

**注:** Informatica Intelligent Cloud Services 一括取り込みサービスの 2020 年 4 月のリリースよりも前、CMI ストリーミングエージェントは、ストリーミング取り込みエージェントという名前でした。

## CMI ストリーミングエージェントのプロパティ

CMI ストリーミングエージェントの動作を変更または最適化するには、ランタイム環境でエージェントプロパティを設定します。CMI ストリーミングエージェントのプロパティは、Secure Agent の編集時に **【システム構成の詳細】** 領域で設定します。

CMI ストリーミングエージェントのエンジン、エージェント、およびスクリプトのプロパティを設定できます。

次の図は、CMI ストリーミングエージェントのプロパティの一部を示しています。

▼ System Configuration Details

Service:	CMI Streaming Agent	▼
Type:	All Types	▼
Type	Name	Value
Engine	MaxLogFileSize	'5MB'
Engine	LogLevel	'DEBUG'
Agent	DataflowPullInterval	60
Agent	JVM	'-Xms256M -Xmx256M'
Agent	LogLevel	'DEBUG'
Agent	MaxLogFileSize	'10MB'
Agent	MaxNumberOfBackups	5
Scripts	LogLevel	'DEBUG'
Scripts	MaxFileSize	'5MB'
Scripts	MaxBackupIndex	5

CMI ストリーミングエージェントの次のプロパティを設定できます。

タイプ	プロパティ名	説明
エンジン	MaxLogFileSize	エンジンが作成可能なログファイルの最大サイズ。デフォルトは 5 MB です。
エンジン	LogLevel	エンジンのログレベル。
エージェント	DataflowPullInterval	エージェントがタスクで更新を確認するまでの間隔。デフォルトは 60 秒です。
エージェント	JVM	エージェントの JVM プロパティのリスト。例：[-Xms256M -Xmx256M]
エージェント	LogLevel	エージェントのログレベル。
エージェント	MaxLogFileSize	エージェントが作成可能なログファイルの最大サイズ。デフォルトは 10MB です。



タイプ	プロパティ名	説明
エージェント	MaxNumberOfBackups	エージェントのバックアップログファイルの最大数。 デフォルトは 5 です。
スクリプト	LogLevel	スクリプトのログレベル。
スクリプト	MaxFileSize	最大ファイルサイズ。この最大ファイルサイズに達した後、ログは ロールオーバーされ、新しいファイルが作成されます。 デフォルトは 10MB です。
スクリプト	MaxBackupIndex	ロールオーバー後に保持するバックアップファイルの最大数。 デフォルトは 5 です。

## 第 4 章

# 共通統合コンポーネント

共通統合コンポーネントサービスは、タスクフローのコマンドタスクステップ内で指定されたコマンドを実行する Secure Agent サービスです。

共通統合コンポーネントサービスを表示または設定するには、共通統合コンポーネントサービスとコマンド実行者パッケージのライセンスを所有している必要があります。

いくつかのサービスプロパティを設定して、共通統合コンポーネントサービスのパフォーマンスを最適化できます。サービスプロパティは、Secure Agent の編集時に変更できます。

共通統合コンポーネントサービスが処理するすべての要求は、次のディレクトリに記録されます。

<Secure Agent インストールディレクトリ>\apps\Common\_Integration\_Components\logs\<バージョン>

各コマンドタスクのログファイルは、次のディレクトリ内で参照できます。

<Secure Agent インストールディレクトリ>\apps\Common\_Integration\_Components\logs\command\<コマンドジョブ ID>

サーバーレスランタイム環境では、Secure Agent は各コマンドタスクのログファイルを Amazon S3 にプッシュします。

## 共通統合コンポーネントプロパティ

共通統合コンポーネントサービスの動作を変更する、または最適化するには、Secure Agent の編集時にプロパティを **【システム構成の詳細】** セクションで設定します。

次の画像は、共通統合コンポーネントサービスのプロパティを示しています。

### ▼ System Configuration Details

Service:

Type:

Type	Name
Tomcat	NetworkTimeoutPeriod
Tomcat	JRE_OPTS
Platform	LCM_JRE_OPTS
SYSTEM_CFG	HTTP_CONNECTION_TIMEOUT_SECONDS
SYSTEM_CFG	HTTP_SOCKET_TIMEOUT_SECONDS
COMMAND_CFG	MaximumConcurrentJobs

以下の共通統合コンポーネントサービスのプロパティを設定できます。

タイプ	名前	説明
Tomcat	JRE_OPTS	Apache Tomcat プロセスの JRE VM オプション。
プラットフォーム	LCM_JRE_OPTS	Apache Tomcat プロセスを開始する、停止する、またはステータスを取得するための JRE オプション。 <b>注:</b> Informatica グローバルカスタマサポートから指示されない限り、このプロパティ値は変更しないでください。
SYSTEM_CFG	HTTP_CONNECTION_TIMEOUT_SECONDS	Secure Agent が Informatica Intelligent Cloud Services と通信するための HTTP 接続を設定するために待機する秒単位での最大時間。 デフォルトは 60 です。 <b>注:</b> Informatica グローバルカスタマサポートから指示されない限り、このプロパティ値は変更しないでください。
SYSTEM_CFG	HTTP_SOCKET_TIMEOUT_SECONDS	Secure Agent と Informatica Intelligent Cloud Services との間の HTTP 接続でデータパケットの転送中の秒単位での最大アイドル時間。 デフォルトは 60 です。 <b>注:</b> Informatica グローバルカスタマサポートから指示されない限り、このプロパティ値は変更しないでください。

タイプ	名前	説明
COMMAND_CFG	MaximumConcurrentJobs	<p>単一の Secure Agent によって実行できる同時コマンドタスクの最大数。</p> <p>1 つの Secure Agent グループ内の各 Secure Agent のデフォルト値は 10 です。</p> <p>例えば、1 つの Secure Agent グループ内に 3 つの Secure Agent がある場合、このサービスが処理できる同時コマンドタスクの最大数は 30 です。</p> <p>最大制限を超えたすべてのコマンド実行要求はキュー入れられ、Secure Agent が使用可能になると実行されます。</p>
<p><b>注:</b> Informatica グローバルカスタマサポートから指示されない限り、共通統合コンポーネントサービスの他のプロパティ値は変更しないでください。</p>		

## 第 5 章

# データベース取り込みサービス

一括取り込みアプリケーションと一括取り込みデータベースはデータベース取り込みエージェントサービスを使用して、取り込みジョブを実行します。

Secure Agent をランタイム環境にダウンロードしてデータベース取り込みサービスを有効にすると、Secure Agent が実行されているオンプレミスのシステムにデータベース取り込みパッケージがプッシュされます。その後、必要に応じて Secure Agent で実行されるデータベース取り込みサービスのプロパティを設定できます。

## データベース取り込みサービスのプロパティ

Secure Agent グループが使用するデータベース取り込みサービスの動作を変更または最適化するには、ランタイム環境にデータベース取り込みプロパティを設定します。

プロパティを設定するには、ランタイム環境を開き、**[編集]** をクリックします。**[システム構成の詳細]** で、**[データベース取り込み]** サービスと **[DBMI\_AGENT\_CONFIG]** タイプを選択します。

次の表に、データベース取り込みエージェントサービスのプロパティを示します。

プロパティ	説明
maxTaskUnits	<p>Secure Agent が実行されているオンプレミスマシンで同時に実行できるアプリケーション取り込みタスクユニットとデータベース統合タスクユニットの最大数。</p> <p>タスクユニットは、ハードウェアまたはソフトウェアの容量や可用性とは関係がありません。maxTaskUnits を設定すると、CPU 使用率を正確に制御できます。有効な値は 1-2000000000 (20 億) です。</p> <p>Secure Agent マシンの適切なタスクユニット数を計算するには、コア数を 3 または 4 で割ることをお勧めします。例えば、8 コアのマシンを使用している場合は、このプロパティを 2 に設定できます。その後、CPU 使用率を監視し、必要に応じてプロパティ値を調整してパフォーマンスチューニングを行います。</p> <p>初期ロード処理中、このプロパティは同時にアンロードできるテーブルの数を決定します。残りのテーブルはキューに入れられ、リソースが使用可能になるとアンロード処理を開始します。</p> <p><b>注:</b> 1 つのジョブで多くのテーブルを処理できます。処理できるテーブルの総数の制限となるのは、使用可能なメモリのみです。1KB の行サイズに基づく初期ロードタスクには、平均してテーブルごとに 25MB の RAM が必要です。</p> <p>増分ロード処理中、このプロパティは同時に実行できるアプリケーション取り込みジョブとデータベース統合ジョブの数を決定します。</p> <p>このプロパティを Secure Agent マシンのコア数よりも大きい値に設定すると、タスク実行の並列処理が増える可能性があります。タスク実行時にパフォーマンスのボトルネックが発生する可能性もあります。</p>
serviceLogRetentionPeriod	<p>最終更新がファイルに書き込まれた後に、各内部データベース取り込みサービスログファイルが保持される日数。この保持期間が経過すると、ログファイルは削除されます。デフォルト値は 7 日です。</p> <p>サービスログは、それらが作成された Secure Agent ホスト (&lt;infaagent&gt;/apps/Database_Ingestion/logs) に保持されます。</p> <p><b>注:</b> このプロパティは一括取り込みアプリケーションと一括取り込みデータベースの両方に適用されます。</p>
taskLogRetentionPeriod	<p>最終更新がジョブログファイルに書き込まれた後、各ジョブログファイルを保存する日数。この保持期間が経過すると、ログファイルは削除されます。デフォルト値は 7 日です。</p>

プロパティ	説明
ociPath	<p>Oracle ソースおよびターゲットの場合、oci.dll または libclntsh.so ファイルが含まれる Oracle Call Interface (OCI) ディレクトリへのパス。デフォルトでは、Oracle は、\$ORACLE_HOME/lib (Linux の場合) または %ORACLE_HOME%\bin (Windows の場合) を使用します。OCI ライブラリは、データベース取り込み CDC タスクによって Oracle に接続するために使用されます。</p> <p>実行中の DBMI エージェントの場合、この値は Windows の PATH 環境変数の値、または Linux の LD_LIBRARY_PATH 環境変数の値に付加されます。PATH または LD_LIBRARY_PATH 環境変数にすでに OCI パスが含まれている場合、このプロパティは必要ありません。</p> <p><b>注:</b> このプロパティは一括取り込みデータベースにのみ適用されます。</p>
serviceUrl	<p>データベース取り込みサービスが Informatica Intelligent Cloud Services クラウドへの接続に使用する URL。</p> <p><b>注:</b> このプロパティは一括取り込みアプリケーションと一括取り込みデータベースの両方に適用されます。</p>
logLevel	<p>データベース取り込みサービスが生成するログに含める詳細レベル。次のオプションがあります。</p> <ul style="list-style-type: none"> <li>- トレース</li> <li>- デバッグ</li> <li>- 情報</li> <li>- 警告</li> <li>- エラー</li> </ul> <p>デフォルト値はトレースです。</p> <p><b>注:</b> このプロパティは一括取り込みアプリケーションと一括取り込みデータベースの両方に適用されます。</p>
taskExecutionHeapSize	<p>タスク実行サービスの最大ヒープサイズ (ギガバイト単位)。この値は、maxTaskUnits プロパティとともに、Secure Agent で実行可能な同時アプリケーション取り込みタスクと同時データベース統合タスクの数に影響します。より多くのタスクを同時に実行するには、ヒープサイズを増やすことをお勧めします。この値に続けてギガバイトの場合は「g」と入力します (例: 「9g」)。デフォルト値は「8g」です。</p> <p><b>注:</b> このプロパティは一括取り込みアプリケーションと一括取り込みデータベースの両方に適用されます。</p>

プロパティ	説明
useProxy	<p>このプロパティを true に設定すると、ターゲットへの接続時およびターゲットへのデータの書き込み時に DBMI エージェントがプロキシを通過できるようになります。次に、DBMI エージェントは、Secure Agent プロキシ構成のプロキシ設定を使用します。デフォルトでは、プロキシ設定は使用されません。</p> <p><b>注:</b> このプロパティは一括取り込みアプリケーションと一括取り込みデータベースの両方に適用されます。</p>
intermediateStorageDirectory	<p>増分ロードジョブおよび初期ロードと増分ロードの組み合わせジョブの場合、関連するタスク定義で <b>【永続ストレージの有効化】</b> オプションが選択されているときに、データを含む中間ファイルが保存されるローカルルートディレクトリです。</p> <p><b>注:</b> このプロパティは一括取り込みデータベースにのみ適用されます。</p>
storageBackupDirectory	<p>増分ロードジョブおよび初期ロードと増分ロードの組み合わせジョブの場合、関連するタスク定義で <b>【永続ストレージの有効化】</b> オプションが選択されているときに、バックアップファイルが保存されるディレクトリへのパスです。</p> <p><b>注:</b> このプロパティは一括取り込みデータベースにのみ適用されます。</p>
storageProperties	<p>増分ロードジョブおよび初期ロードと増分ロードの組み合わせジョブの場合、関連するタスク定義で <b>【永続ストレージの有効化】</b> オプションが選択されているときに使用されるキー=値のペアのカンマ区切りリストです。このプロパティを指定する場合は、Informatica グローバルカスタマサポートにお問い合わせください。</p> <p><b>注:</b> このプロパティは一括取り込みデータベースにのみ適用されます。</p>
task_container.jvm.allowExceptionForInvalidEncodedData	<p>無効なエンコーディングを UTF-8 に報告する文字変換エラーを受け取った場合にソースデータを修復または修正しないようにするには、ソースからデータをアンロードしようとしたときにデータベース取り込みジョブが失敗しないようにこのプロパティを false に設定します。この設定を使用するとデータベース取り込みサービスによって同等の Java プロパティが DataDirect JDBC ドライバに渡されるため、例外が発生しなくなります。このプロパティを設定した後に、データベース取り込みサービスを再開する必要があります。</p> <p><b>注:</b> このプロパティは一括取り込みデータベースにのみ適用されます。</p>



# データベース取り込みエージェントの環境変数

データベース取り込みエージェントの動作を変更または最適化するには、次の環境変数を定義します。

環境変数	説明
DBMI_REPLACE_UNSUPPORTED_CHARS	<p>Microsoft Azure Synapse Analytics ターゲットの場合に、アプリケーション取り込みジョブまたはデータベース取り込みジョブが、ターゲットが正しく処理できない文字データ内の文字を置き換えるかどうかを指定します。文字の置き換えを有効にするには、この環境変数を true に設定します。</p> <p>DBMI_REPLACE_UNSUPPORTED_CHARS=true</p> <p>設定後、一括取り込みアプリケーションまたは一括取り込みデータベースは、DBMI_UNSUPPORTED_CHARS_REPLACEMENT 環境変数に指定されている文字を使用して、サポートされていない文字を置き換えます。</p>
DBMI_UNSUPPORTED_CHARS_REPLACEMENT	<p>DBMI_REPLACE_UNSUPPORTED_CHARS 環境変数が true に設定されている場合に、Microsoft Azure Synapse Analytics ターゲットが正しく処理できないソースデータ内の文字を置き換える文字を指定します。</p> <p>デフォルト値: ? (疑問符)</p> <p>注: この環境変数は一括取り込みデータベースに対してのみ定義します。</p>
DBMI_WRITER_CONN_POOL_SIZE	<p>アプリケーション取り込みジョブまたはデータベース統合ジョブが変更データをターゲットにプロパゲートするために使用する接続の数を示します。デフォルト値は 8 です。有効な値は 4~8 です。</p>
DBMI_WRITER_RETRIES_MAX_COUNT	<p>データベース統合ジョブがソースデータを Amazon S3 または Microsoft Azure Data Lake Storage Gen2 ターゲットにロードしている最中にネットワークの問題が発生した場合に、データベース統合ジョブで初期ロードまたは増分ロードを続行する要求を再試行する最大回数を指定します。再試行がすべて失敗した場合、ジョブは失敗となります。</p> <p>デフォルト値は 5 です。</p>
DBMI_WRITER_RETRIES_INTERVAL_IN_MILLIS	<p>ネットワークで問題が発生した場合に、データベース統合ジョブが Amazon S3 または Microsoft Azure Data Lake Storage Gen2 ターゲットへの初期ロードまたは増分ロードを続行する要求を再試行する前に待機する間隔（ミリ秒単位）を指定します。</p> <p>デフォルト値は 1000 です。</p>

注: 環境変数を定義または変更したら、データベース取り込みエージェントを再起動して、変更を有効にします。

## 第 6 章

# データ統合サーバー

データ統合サーバーは、マッピング、タスク、およびタスクフローインスタンスなどのデータ統合ジョブを実行する Secure Agent サービスです。

詳細クラスタが、詳細モードのマッピングのデータロジックを処理する場合、データ統合サーバーは詳細クラスタのサブタスクをエラスティックサーバーに委ねます。

いくつかのサービスプロパティを設定して、データ統合サーバーのパフォーマンスを最適化できます。例えば、ネットワークの回復機能設定または Secure Agent の接続タイムアウト期間を変更できます。サービスプロパティは、Secure Agent の編集時に変更できます。

## データ統合サーバーの回復機能

ネットワークの一時的な問題が発生している際、Secure Agent が接続の再確立を試みている間、データ統合タスクを続行できます。データ統合サーバーのネットワークの回復機能プロパティを設定できます。

Secure Agent が接続の再確立を試みる方法は、次のデータ統合サーバーのプロパティで決定されます。

### NetworkTimeoutPeriod

Secure Agent で Informatica Intelligent Cloud Services との通信の再確立を試行する時間の長さを決定します。期間の終わりに通信が確立されていない場合、実行されていた進行中のデータ統合タスクは停止します。デフォルト値は 300 秒です。

### NetworkRetryInterval

Secure Agent が指定されたタイムアウト期間内に Informatica Intelligent Cloud Services への接続を試行する頻度を決定します。デフォルト値は 5 秒です。

例えば、デフォルト設定の場合、ネットワークが停止すると、Secure Agent は Informatica Intelligent Cloud Services との通信の再確立を 300 秒間試行します。その 300 秒の期間中、Secure Agent は 5 秒ごとに Informatica Intelligent Cloud Services への接続を試行します。300 秒の期間内に Secure Agent が通信を再確立すれば、進行中のデータ統合タスクは影響を受けません。Secure Agent は、300 秒の期間内に通信を再確立できない場合、進行中のデータ統合タスクを停止します。

# データ統合サーバーのプロパティ

データ統合サーバーの動作を変更または最適化するには、データ統合サーバーのプロパティを設定します。データ統合サーバーのプロパティは、Secure Agent の編集時に **【システム構成の詳細】** 領域で設定します。

次の画像は、データ統合サーバーのプロパティを示しています。

▼ System Configuration Details

Service:	Data Integration Server	
Type:	All Types	
Type	Name	Value
Tomcat	NetworkTimeoutPeriod	300
Tomcat	NetworkRetryInterval	5
Tomcat JRE	INFA_SSL	
Tomcat JRE	INFA_MEMORY	'-Xms32m -Xmx512m -XX:MaxPermSize=128m'
Tomcat JRE	JRE_OPTS	'-Xrs'
Tomcat JRE	JAVA_LIBS	
Tomcat Log4j	log4j_rootLogger	'INFO, tomcatLog'
Tomcat Log4j	log4j_appender_tomcatLog	'org.apache.log4j.FileAppender'
Tomcat Log4j	log4j_appender_tomcatLog_layout	'org.apache.log4j.PatternLayout'
Tomcat Log4j	log4j_appender_tomcatLog_layout_ConversionPattern	'%d %d{z} %p [%c] - %m%n'

設定可能なデータ統合サーバーのプロパティを次に示します。

タイプ	名前	説明
Tomcat	NetworkTimeoutPeriod	Secure Agent が Informatica Intelligent Cloud Services との通信の再確立を試行するまでの時間 (秒)。デフォルトは 300 です。
Tomcat	NetworkRetryInterval	Secure Agent が、指定したタイムアウト期間内に Informatica Intelligent Cloud Services への接続を試行する頻度 (秒)。デフォルトは 5 です。
Tomcat	INFA_DTM_STAGING_ENABLED_CONNECTORS	特定のクラウドデータウェアハウスコネクタに適用されます。 データをターゲットにロードする前に、データ統合サーバーがローカルのフラットファイル内のステージングターゲットデータを最適化できるようにします。 ステージングを最適化するには、このプロパティをコネクタのプラグイン ID に設定します。 詳細については、該当するコネクタのヘルプを参照してください。

タイプ	名前	説明
Tomcat	INFA_DTM_RDR_STAGING_ENABLED_CONNECTORS	<p>特定のクラウドデータウェアハウスコネクタに適用されます。</p> <p>ソースからデータを読み取った後に、データ統合サーバーがローカルのフラットファイル内のソースデータのステージングを最適化できるようにします。</p> <p>ステージングを最適化するには、このプロパティをコネクタのプラグイン ID に設定します。</p> <p>詳細については、該当するコネクタのヘルプを参照してください。</p>
Tomcat	INFA_DTM_LKP_STAGING_ENABLED_CONNECTORS	<p>特定のクラウドデータウェアハウスコネクタに適用されます。</p> <p>ルックアップオブジェクトからデータを読み取った後に、データ統合サーバーがローカルのフラットファイル内のルックアップデータのステージングを最適化できるようにします。</p> <p>ステージングを最適化するには、このプロパティをコネクタのプラグイン ID に設定します。</p> <p>詳細については、該当するコネクタのヘルプを参照してください。</p>
Tomcat JRE	JRE_OPTS	Apache Tomcat プロセスの JRE VM オプション。
Tomcat JRE	INFA_MEMORY	Apache Tomcat プロセスの仮想マシンメモリに対して設定される JRE VM オプション。
DTM	AgentConnectionTimeout	Secure Agent 通信で、タイムアウトするまでに待機を要求する秒数。デフォルトは 5 です。

タイプ	名前	説明
DTM	JVMOption1 - JVMOption5	<p>最大および最小の JVM ヒープサイズ、インテリジェント構造検出の最大レコードサイズ、特定のコネクタのプロキシ設定などの、データ統合サーバーの詳細プロパティを設定する JVM オプション。例えば、最大 JVM ヒープサイズをデフォルト値の 512 MB から 2048 MB に変更するには、JVMOption1 を '-Xmx2048m' に設定します。</p> <p>デフォルトでは、JVMOption1 - JVMOption5 を使用して、最大 5 つの詳細プロパティを設定できます。追加のプロパティを設定するには、JVMOption6 や JVMOption7 などの名前を付けた、データ統合サーバー用のカスタム DTM プロパティを追加します。オプション番号が連続していて、番号を飛ばしていないことを確認してください。</p> <p>設定できる JVM オプションの詳細については、データ統合のヘルプ、適切なコネクタのヘルプ、または Informatica Network の <a href="#">Knowledge Base</a> を参照してください。</p>
OS_PROFILE	EnableOSProfileForTaskExecution	<p>OS プロファイル機能を有効化または無効化します。値は true または false です。デフォルトは false です。</p> <p><b>注:</b> OS プロファイル機能は、Linux でのみ使用できます。</p> <p>デフォルトでは、タスクは Secure Agent と同じオペレーティングユーザーで実行されます。OS プロファイル機能を有効にすると、別のオペレーティングシステムユーザーでタスクを実行することができます。</p> <p>例えば、同じ Informatica Intelligent Cloud Services インストールを使用している複数の部門があり、分離が必要な場合などが挙げられます。</p> <p><b>ヒント:</b> 完了したタスクのセッションログで OS プロファイルユーザーの名前を表示することができます。</p>

タイプ	名前	説明
OS_PROFILE	FailTasksForMissingOsProfileMapping	<p>OS プロファイル機能が有効な場合は、特定のタスクのオペレーティングシステムユーザーへのマッピングが欠落していると、このプロパティによりタスクが失敗します。値は true または false です。デフォルトは true です。</p> <p>このプロパティを false に設定すると、このシナリオでタスクが失敗することはなく、代わりに Secure Agent のオペレーティングシステムユーザーでタスクが実行されます。</p>
OS_PROFILE	ShareSystemDirectories	<p>Secure Agent ユーザー用に作成されたシステムディレクトリをオペレーティングシステムユーザーと共有します。値は true または false です。デフォルトは false です。</p> <p>ユーザーパラメータなどの情報を共有する場合は、共有を有効にします。ただし、分離性を高め、エラーの可能性を減らすためには、これらのディレクトリを共有しないようにしてください。</p> <p>デフォルトの動作では、オペレーティングシステムユーザー用の新しいシステムディレクトリが <code>Data_Integration_Server/data/osprofiles_filesystem/&lt;profile_name&gt;</code> の下に作成されます</p> <p>システムディレクトリには、マッピング情報、セッションログ、およびユーザーパラメータが保存されません。</p>
OS_PROFILE	OSProfileUserMappingFile	<p>Secure Agent のユーザーまたは場所とオペレーティングシステムユーザー間のマッピング情報を含む YAML ファイルの場所。</p> <p>マッピングファイルへの変更は、Secure Agent の再起動なしで自動的に適用されます。</p> <p>このプロパティの詳細については、<a href="#">「OSProfileUserMappingFile の作成」 (ページ 31)</a> を参照してください。</p>
OS_PROFILE	OSProfileScriptForTaskExecution	<p>タスクの実行に使用されるスクリプトファイルの場所。</p> <p>このプロパティの詳細については、<a href="#">「OSProfileScriptForTaskExecution の設定」 (ページ 32)</a> を参照してください。</p>
<p><b>注:</b> Informatica グローバルカスタマサポートから指示された場合を除き、データ統合サーバーの他のプロパティ値は変更しないでください。</p>		

## OSProfileUserMappingFile の作成

OS\_PROFILE プロパティを定義する場合は、Secure Agent のユーザーまたは場所とオペレーティングシステムユーザー間のマッピング情報を含む YAML ファイルを作成する必要があります。このスクリプトは、データ統合サーバーサービスの OSProfileUserMappingFile プロパティで設定します。

次の例をテンプレートとして使用して YAML ファイルを作成します。

```
- profileName: userprofile1
  profileType: USER
  systemNames:
  - osp_idmc_1
  osMapping:
    osUser: osprofileuser1
    pmVariables:
      PMRootDir: /home/osprofileuser1/pmdata
      PMSessionLogDir: /mnt/shared/Vadi/osprofile
      PMBadFileDir: /home/osprofileuser1/pmbadfile
      PMCacheDir: /home/osprofileuser1/pmcache
      PMTargetFileDir: /home/osprofileuser1/pmtrgtfile
      PMSourceFileDir: /home/osprofileuser1/pmsrcfile
      PmExtProcDir: /home/osprofileuser1/pmextproc
      PMTempDir: /home/osprofileuser1/pmtemp
      PMLookupFileDir: /home/osprofileuser1/pmlookupfile
      PMStorageDir: /data/agent/userparam
- profileName: userprofile2
  profileType: USER
  systemNames:
  - osp_idmc_2
  osMapping:
    osUser: osprofileuser2
- profileName: locationprofile1
  profileType: LOCATION
  systemNames:
  - osprofilefoldertest1
  - osprofiletestfolder2
  osMapping:
    osUser: osprofileuser1
- profileName: locationprofile2
  profileType: LOCATION
  systemNames:
  - osprofilefoldertest1\test
  - osprofiletestfolder\test
  osMapping:
    osUser: osprofileuser2
```

次の表に、YAML ファイルの各プロパティとその説明を示します。

プロパティ	説明
profileName	プロファイルの名前。一意である必要があります。 このプロパティは必須です。
profileType	このマッピングが Secure Agent のユーザーまたは場所のどちらかに基づいているかことを示します。有効な値は USER または LOCATION です。 注: 同じタスクに「ユーザー」および「ロケーション」のプロファイルタイプが指定されている場合は、「ユーザー」が優先されます。
systemNames	このセクションには、Informatica Intelligent Cloud Services (IICS) のシステム名のリストが表示されます。 - profileType = USER の場合、システム名は IICS ユーザーのリストです。 - profileType = LOCATION の場合、システム名は IICS ロケーション名のリストです。すべての場所に対して絶対パスを使用します。
osMapping	このセクションには、オペレーティングシステムへのマッピングが表示されます。





3. 次のコマンドを実行して、pmimpprocess の権限を変更します:  

```
chmod 755 pmimpprocess
```
4. root ユーザーとしてログインするか、root として sudo を実行します。
5. 次のコマンドを実行して、pmimpprocess の所有権とアクセス権限を変更します:  

```
chown root:root pmimpprocess
chmod u+s pmimpprocess
```
6. 「[データ統合サーバーのプロパティ](#)」 (ページ 27) の OSPProfileScriptForTaskExecution プロパティの値として pmimpprocess の場所を入力します。  
 例:  

```
<Secure Agent installation directory>/apps/Data_Integration_Server/ext/pmimpprocess
```

## デフォルトのスクリプトの使用

スクリプトメソッドを使用する場合は、次の場所にスクリプトを配置します:

Data\_Integration\_Server/ext/infa-osprofile-dtm.sh

**注:** スクリプトメソッドは、技術的な知識のあるユーザーのみを対象としています。

次の画像は、デフォルトのスクリプトを示しています。

```
#!/bin/sh
input_args="$@"
env_var_file=$(mktemp)
chmod +r "${env_var_file}"
printenv | sed 's/(^[^=]*)=\\(.*)/export \\1="\\2"/!> "${env_var_file}"'
echo "sudo su - ${ENV_INFA_DTM_OSPROFILE_USER} -c . ${env_var_file}; cd ${PWD}; ${input_args}" >/tmp/xx
sudo su - ${ENV_INFA_DTM_OSPROFILE_USER} -c ". ${env_var_file}; cd ${PWD}; ${input_args}"
exit_code=$?
rm "${env_var_file}"
exit "${exit_code}"
```

必要に応じてスクリプトを更新します。

# データ統合サーバーアップグレード

新しいバージョンが利用可能になると、データ統合サーバーは自動的にアップグレードされます。

データ統合サーバーのアップグレードは自動的に行われるため、ユーザーの介入は必要ありません。アップグレードが利用可能になると、データ統合サーバーの新しいバージョンが起動し、古いバージョンはすべてのジョブを処理した後に停止します。CDC 連続抽出ジョブなどの実行時間の長いジョブがある場合、アップグレードプロセスはデータ統合サーバーの古いバージョンでジョブを終了し、アップグレードされたバージョンで新しいジョブとしてそれらのジョブを再開します。

ジョブの再開が発生すると、元のジョブは失敗ステータスとして表示され、[マイジョブ] ページに次のメッセージが表示されます。

Job stopped for Data Integration Server upgrade and will resume as a new job on the upgraded service.

元のジョブは、別のインスタンス名を持つ新しいジョブに置き換えられます。

CDC 連続抽出のマッピングタスクの設定の詳細については、オンラインヘルプの該当するコネクタのヘルプを参照してください。

## 第 7 章

# エラスティックサーバー

エラスティックサーバーは、詳細クラスタとクラスタで実行されるジョブを管理する Secure Agent サービスです。

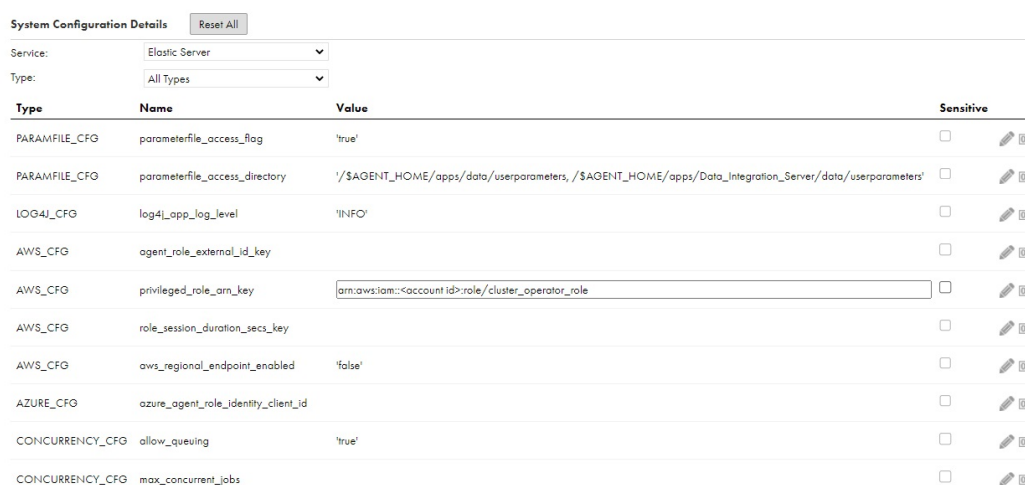
サービスのプロパティを設定して、AWS 環境でのクラスタオペレータロールや Secure Agent ロールなど、詳細クラスタをホストするクラウドプラットフォームに関する詳細を指定できます。また、プロパティを設定して、パラメータファイルへのアクセス、エラスティックサーバーがログファイルに書き込みを行う詳細レベル、および並行処理設定を指定することもできます。

詳細クラスタの詳細については、「[詳細クラスタ](#)」を参照してください。

## エラスティックサーバーのプロパティ

エラスティックサーバーの動作を変更するには、Secure Agent の編集時に **【システム構成の詳細】** 領域でエラスティックサーバーのプロパティを設定します。

次の図は、エラスティックサーバーのプロパティを示しています。



Type	Name	Value	Sensitive
PARAMFILE_CFG	parameterfile_access_flag	'true'	<input type="checkbox"/>
PARAMFILE_CFG	parameterfile_access_directory	'/\$AGENT_HOME/apps/data/userparameters,/\$AGENT_HOME/apps/Data_Integration_Server/data/userparameters'	<input type="checkbox"/>
LOG4J_CFG	log4j_app_log_level	'INFO'	<input type="checkbox"/>
AWS_CFG	agent_role_external_id_key		<input type="checkbox"/>
AWS_CFG	privileged_role_arn_key	arn:aws:iam::<account id>:role/cluster_operator_role	<input type="checkbox"/>
AWS_CFG	role_session_duration_secs_key		<input type="checkbox"/>
AWS_CFG	aws_regional_endpoint_enabled	'false'	<input type="checkbox"/>
AZURE_CFG	azure_agent_role_identity_client_id		<input type="checkbox"/>
CONCURRENCY_CFG	allow_queuing	'true'	<input type="checkbox"/>
CONCURRENCY_CFG	max_concurrent_jobs		<input type="checkbox"/>

設定できるエラスティックサーバーのプロパティを次に示します。

タイプ	名前	説明
PARAMFILE_CFG	parameterfile_access_flag	開発者が、Secure Agent マシンに保存されているパラメータファイルをダウンロードできるかどうかを示します。 デフォルトは'true'です。
PARAMFILE_CFG	parameterfile_access_directory	パラメータファイルのダウンロードを許可する Secure Agent マシン上のディレクトリのリスト。開発者は、指定したディレクトリまたはサブディレクトリのいずれかからパラメータファイルをダウンロードできます。 デフォルトは、' <code>/\$AGENT_HOME/apps/data/userparameters,/\$AGENT_HOME/apps/Data_Integration_Server/data/userparameters</code> 'です。
LOG4J_CFG	log4j_app_log_level	エラスティックサーバーがログファイルに書き込む詳細のレベル。「INFO」などの文字列としてログレベルを入力します。 ログレベルを大きくすると、エラスティックサーバーがログファイルに書き込むメッセージに、より優先度の高いログレベルのメッセージが含まれます。例えば、ログレベルが INFO の場合、ログには FATAL、ERROR、WARNING、および INFO コードのメッセージが記録されます。 有効な値は次のとおりです。 <ul style="list-style-type: none"> <li>- FATAL。サービスがシャットダウンする、または利用不可能になる修復不能なシステム障害が含まれます。</li> <li>- ERROR。接続の失敗、メタデータの保存または取得の失敗、サービスのエラーが含まれます。</li> <li>- WARNING。修復可能なシステム障害または警告が含まれます。</li> <li>- INFO。システムおよびサービスの変更にに関するメッセージが含まれます。</li> <li>- TRACE。ユーザー要求の失敗がログとして記録されます。</li> <li>- DEBUG。ユーザー要求がログとして記録されます。</li> </ul>
AWS_CFG	agent_role_external_id_key	Secure Agent がクラスタオペレータロールを使用する場合に Secure Agent で指定する外部 ID。クラスタオペレータロールの信頼関係で外部 ID を設定する場合に必要です。 このプロパティは、AWS 環境でのみ有効です。
AWS_CFG	privileged_role_arn_key	クラスタオペレータロールの ARN。 AWS 環境で個別のクラスタオペレータロールと Secure Agent ロールを設定する場合に必要です。 このプロパティは、AWS 環境でのみ有効です。

タイプ	名前	説明
AWS_CFG	role_session_duration_secs_key	AWS AssumeRole API のセッション時間 (秒単位)。デフォルトのセッション時間は 1,800 秒 (30 分) です。 クラスタオペレータロールに設定されている最大 CLI/API セッション期間をオーバーライドします。エラスティックサーバーに設定されているセッション期間がクラスタオペレータロールのセッション期間よりも長い場合、Secure Agent がクラスタオペレータロールを使用できない場合があります。このプロパティは、AWS 環境でのみ有効です。
AZURE_CFG	azure_agent_role_identity_client_id	マネージド ID agent_identity のクライアント ID。agent_identity がユーザー割り当てのマネージド ID であり、Secure Agent マシンに少なくとも 1 つの他のマネージド ID がある場合に必要です。 このプロパティは、Azure 環境でのみ有効です。
CONCURRENCY_CFG	allow_queuing	エラスティックサーバーが Spark タスクをキューに格納するかどうかを示します。デフォルトは true です。
CONCURRENCY_CFG	max_concurrent_jobs	エラスティックサーバーが処理できる同時 Spark タスクの最大数。

## エラスティックサーバーの並行処理

エラスティックサーバーは、Spark タスクを同時に処理し、追加の Spark タスクをキューに格納することができます。並行処理プロパティを設定してキューを有効または無効にし、同時 Spark タスクの最大数を設定できます。

次のエラスティックサーバーのプロパティにより、キューイングと並行処理を定義します。

### allow\_queuing

エラスティックサーバーが Spark タスクをキューに格納することができるようにします。デフォルト値は true に設定されています。

このプロパティが false に設定されている場合、エラスティックサーバーは、送信されたすべての Spark タスクを同時に処理します。エラスティックサーバー上の同時タスクの最大数をデフォルトの同時実行数よりも高い値に設定する場合は、それに応じて Java ヒープサイズを増やしてください。Java ヒープサイズを増やさずに処理を行うと、Secure Agent プロセスがクラッシュする可能性があります。

エラスティックサーバーは、データプレビューおよび SQL ELT の最適化ジョブをキューに格納せずにすぐに処理します。

### max\_concurrent\_jobs

エラスティックサーバーが処理できる同時 Spark タスクの最大数。エラスティックサーバーが同時 Spark タスクの最大数に達すると、追加の Spark タスクをキューに格納することができますようになります。

デフォルトでは、同時 Spark タスクの最大数はクラウドプラットフォームによって異なります。エラスティックサーバーは、フルマネージドクラスタとセルフサービスクラスタに対して次のデフォルトを使用します。

- AWS の Java ヒープスペース 2 GB あたり 500 件の同時 Spark タスク
- Google Cloud の Java ヒープスペース 2 GB あたり 375 件の同時 Spark タスク
- Microsoft Azure の Java ヒープスペース 2 GB あたり 250 件の同時 Spark タスク

ローカルクラスタは 15 件の Spark タスクを同時に実行できますが、並行処理数は変更されません。

Java ヒープサイズはデフォルトで 2 GB に設定されていることに注意してください。同時タスクの最大数をデフォルトよりも大きい値に設定する場合は、デフォルトの比率を使用して Java ヒープサイズを増やします。

例えば、Microsoft Azure で 500 件の Spark タスクを同時に実行する場合は、Java ヒープサイズを 4 GB に増やします。Java ヒープサイズが 2 GB のままである場合、エラスティックサーバーは最大 250 件の Spark タスクを同時に処理し、残りの 250 件の Spark タスクはキューに格納されます。

エラスティックサーバーには、2 GB の Java ヒープ領域あたり 1,000 件の Spark タスクという上限があります。Java ヒープサイズを 2 GB よりも大きい容量にせずに、同時タスクの最大数を 1,000 件を超える値に設定すると、エラスティックサーバーは起動に失敗します。

Secure Agent マシンのサイズ変更の詳細については、「[データ統合のパフォーマンスのチューニング](#)」を参照してください。

## 第 8 章

# ファイル統合サービス

ファイル統合サービスを使用して、組織とリモートファイルサーバーの間でファイルを転送します。

ファイル統合サービスは、エージェントが AS2 などの高度なファイル転送プロトコルの実行に使用する Secure Agent のサービスです。

組織でリモートパートナーからファイルを受信できるようにする前に、ファイルサーバーを設定しておく必要があります。管理者の [ファイルサーバー] ページで、ファイル統合サービスに関連付けられる組織のファイルサーバーを設定します。設定には、ファイルサーバーの詳細、暗号化の方法、および許可されるファイルタイプなどのプロパティが含まれます。

ファイル統合サービスを停止または開始するには、サービスを使用するファイルサーバーを停止または開始します。

ファイルサーバーの設定については、「ファイルサーバー」を参照してください。

ファイル統合サービスを使用するには、組織が適切なライセンスを持っている必要があります。ファイル統合サービスを設定するには、管理者ロールが割り当てられている必要があります。

## 第 9 章

# GitRepoConnectApp

組織がオンプレミスのソース管理リポジトリを使用している場合、GitRepoConnectApp サービスは Informatica Intelligent Cloud Services およびソース管理リポジトリ間の通信を管理します。

Secure Agent は、Secure Agent マシンへのリモートソース管理リポジトリのローカルコピーの作成時に GitRepoConnectApp サービスを使用します。また、このサービスを使用して、リモートリポジトリからソース管理操作に関する情報を取得します。

## ローカルリポジトリのベースディレクトリ

ソース管理リポジトリがオンプレミスの場合、Secure Agent は Informatica Intelligent Cloud Services アセットを格納するリポジトリブランチのローカルコピーを作成します。Secure Agent マシンでローカルリポジトリの場所を設定できます。

デフォルトでは、Secure Agent は次のディレクトリにローカルリポジトリを作成します。

```
<Secure Agent インストールディレクトリ>/apps/GitRepoConnectApp/<ベースディレクトリ>/<クライアント URL>/<組織 ID>/<ブランチ>/<リモートリポジトリ名>
```

このファイルパスでは、ベースディレクトリは GitRepoConnectApp サービスの **[git\_local\_repository\_path]** プロパティによって管理されます。

デフォルトでは、**git\_local\_repository\_path** は `../data/git_repository/` に設定されています。そのため、Secure Agent は次のディレクトリにローカル Git リポジトリを作成します。

```
<Secure Agent インストールディレクトリ>/apps/GitRepoConnectApp/data/git_repository/<クライアント URL>/<組織 ID>/<ブランチ>/<リモートリポジトリ名>
```

**[git\_local\_repository\_path]** プロパティを編集することで、ベースディレクトリを変更できます。例えば、このプロパティを `../MYREPO/PROD` に設定した場合、Secure Agent は、次のディレクトリにローカル Git リポジトリを作成します。

```
<Secure Agent インストールディレクトリ>/apps/GitRepoConnectApp/MYREPO/PROD/<クライアント URL>/<組織 ID>/<ブランチ>/<リモートリポジトリ名>
```

バックスラッシュ文字 (\) を含むベースディレクトリを指定するには、別のバックスラッシュ文字でエスケープします。

[`git_local_repository_path`] プロパティを設定する場合は、次のガイドラインを使用してください。

- 親ディレクトリ (..) を省略した場合このプロパティを設定すると、Secure Agent は GitRepoConnectApp サービスのバージョンにサブディレクトリを作成します。リポジトリのローカルコピーは、次のディレクトリに保存されます。

<Secure Agent インストールディレクトリ>/apps/GitRepoConnectApp/<GitRepoConnectApp バージョン>/<ベースディレクトリ>/<クライアント URL>/<組織 ID>/<ブランチ>/<リモートリポジトリ名>

この場合、Secure Agent は GitRepoConnectApp サービスが更新されるたびに新しいローカルリポジトリディレクトリを作成するため、Secure Agent マシンのディスク領域が大量に消費される可能性があります。

- このプロパティを設定する場合は、ローカルリポジトリディレクトリが複数のエージェントによって共有されないようにしてください。それぞれの Secure Agent マシンには、リポジトリの独自のローカルコピーが必要です。

## GitRepoConnectApp のプロパティ

GitRepoConnectApp サービスの動作を変更または最適化するには、サービスのプロパティを設定します。Secure Agent の編集時に、[システム構成の詳細] 領域でサービスプロパティを設定します。

以下の図に、GitRepoConnectApp のプロパティを示します。

▼ System Configuration Details

Service:	GitRepoConnectApp	
Type:	All Types	
Type	Name	Value
LOG4J	rootLogger	'INFO'
GIT_REPO_CONNECT_APP_CONF	host	'localhost'
GIT_REPO_CONNECT_APP_CONF	address	'127.0.0.1'
GIT_REPO_CONNECT_APP_CONF	git_local_repository_path	'../data/git_repository/'
GIT_REPO_CONNECT_APP_CONF	JVM_MIN_MEMORY	'32m'
GIT_REPO_CONNECT_APP_CONF	JVM_MAX_MEMORY	'256m'



以下のサービスのプロパティを設定できます。

タイプ	名前	説明
GIT_REPO_CONNECT_APP_CONF	git_local_repository_path	Secure Agent マシンのローカル Git リポジトリのベースディレクトリ。 ベースディレクトリは、次のディレクトリに作成されます。 <Secure Agent インストールディレクトリ>/apps/GitRepoConnectApp/ デフォルトは../data/git_repository/です。そのため、Secure Agent は次のディレクトリにローカル Git リポジトリを作成します。 <Secure Agent インストールディレクトリ>/apps/GitRepoConnectApp/git_repository/<クライアント URL>/<組織 ID>/<ブランチ>/<リモートリポジトリ名>
GIT_REPO_CONNECT_APP_CONF	JVM_MIN_MEMORY	サービスの開始時に GitRepoConnectApp サービスに割り当てられるメモリの量。 デフォルトは 32 MB です。
GIT_REPO_CONNECT_APP_CONF	JVM_MAX_MEMORY	GitRepoConnectApp サービスに割り当てられる最大メモリ。 デフォルトは 256 MB です。
<p><b>注:</b> Informatica グローバルカスタマサポートから指示された場合を除き、このプロパティ値は変更しないでください。</p>		

# 第 10 章

## IDMC Data Gateway Service

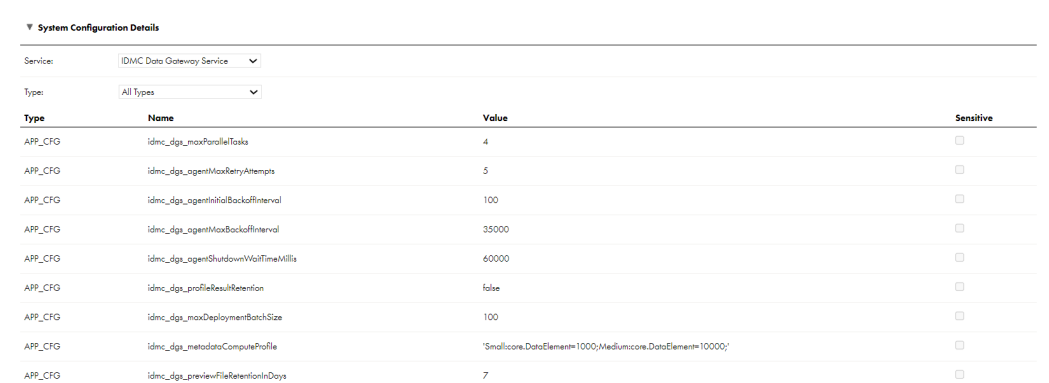
IDMC Data Gateway Service を使用すると、CLAIRE GPT でデータ探索タスクを実行できます。ソースデータを探索し、データセットのサンプルをプレビューして、サンプルデータの取得に使用された SQL コードを確認し、後で参照できるようにサンプルデータを CSV ファイルに保存することができます。CLAIRE GPT でデータ探索タスクを実行するには、Secure Agent で IDMC Data Gateway Service が稼働していることを確認します。

IDMC Data Gateway Service のパフォーマンスを最適化するには、サービスのプロパティを設定します。

### IDMC Data Gateway Service のプロパティ

IDMC Data Gateway Service の動作を変更または最適化するには、Secure Agent の編集時に **【システム構成の詳細】** セクションでプロパティを設定します。

次の画像は、IDMC Data Gateway Service のプロパティを示しています。



The screenshot shows the 'System Configuration Details' for the 'IDMC Data Gateway Service'. It lists various configuration properties with their types, names, values, and whether they are sensitive.

Type	Name	Value	Sensitive
APP_CFG	idmc_dgs_maxParallelTasks	4	<input type="checkbox"/>
APP_CFG	idmc_dgs_agentMaxRetryAttempts	5	<input type="checkbox"/>
APP_CFG	idmc_dgs_agentInitialBackoffInterval	100	<input type="checkbox"/>
APP_CFG	idmc_dgs_agentMaxBackoffInterval	35000	<input type="checkbox"/>
APP_CFG	idmc_dgs_agentShutdownWaitTimeMillis	60000	<input type="checkbox"/>
APP_CFG	idmc_dgs_profileResultRetention	false	<input type="checkbox"/>
APP_CFG	idmc_dgs_maxDeploymentBatchSize	100	<input type="checkbox"/>
APP_CFG	idmc_dgs_methodDataComputeProfile	'Smallcore.DataElement=1000;Mediumcore.DataElement=10000'	<input type="checkbox"/>
APP_CFG	idmc_dgs_previewFileRetentionInDays	7	<input type="checkbox"/>

**【システム構成の詳細】** セクションで、IDMC Data Gateway Service の次のようなシステムプロパティを設定することができます。

タイプ	名前	説明	サンプル値	デフォルト値
APP_CFG	idmc_dgs_agentMaxRetryAttempts	-	3	5
APP_CFG	idmc_dgs_agentShutdownWaitTimeMillis	-	30000	60000

タイプ	名前	説明	サンプル値	デフォルト値
APP_CFG	idmc_dgs_connectionPoolMaxSize	Secure Agent 用にキャッシュされるデータソース接続の最大数。	最小値は 10 です。 最大値は 1000 です。	100
APP_CFG	idmc_dgs_connectionPoolCacheExpiryDuration	データソースとの接続が未使用であるときに開いたままになる最大期間。 注: パラメータの値は分単位で指定する必要があります。	最小値は 10 です。 最大値は 120 です。	30 分
APP_CFG	idmc_dgs_minHeapSize	JVM の最小ヒープサイズを変更するためのオプション。 注: パラメータの値は一重引用符で囲む必要があります。	'128m'	'128m'
APP_CFG	idmc_dgs_maxHeapSize	JVM の最大ヒープサイズを変更するためのオプション。 注: パラメータの値は一重引用符で囲む必要があります。	'512m'	'512m'
APP_CFG	idmc_dgs_JVM_ARGS	デバッグパラメータを追加する、または JVM メモリ構成パラメータを変更するオプション。	-Xms32m -Xmx512m	-Xms512m
APP_CFG	idmc_dgs_queryExecutionTimeout	データベースクエリがデータソースに対して実行される最大期間。 注: パラメータの値は秒単位で指定する必要があります。	最小値は 1 です 値に上限はありません。	60 秒

タイプ	名前	説明	サンプル値	デフォルト値
APP_LOG 4J	name	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'AppLogPropertiesConfig'	'AppLogPropertiesConfig'
APP_LOG 4J	rootLogger_level	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'warn'	'info'
APP_LOG 4J	rootLogger_appenderRefs	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'applog'	'applog'
APP_LOG 4J	rootLogger_appenderRefs_applog_ref	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'LOGFILE'	'LOGFILE'
APP_LOG 4J	appenders	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'applog'	'applog'

タイプ	名前	説明	サンプル値	デフォルト値
APP_LOG 4J	appender_applog_type	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'RollingFile'	'RollingFile'
APP_LOG 4J	appender_applog_name	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'LOGFILE'	'LOGFILE'
APP_LOG 4J	appender_applog_filePattern	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'%d{MM-dd-yy-HH-mm-ss}-%i'	'%d{MM-dd-yy-HH-mm-ss}-%i'
APP_LOG 4J	appender_applog_layout_type	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'PatternLayout'	'PatternLayout'
APP_LOG 4J	appender_applog_layout_pattern	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'%d %d{z} %p [%c] - %m%n'	'%d %d{z} %p [%c] - %m%n'

タイプ	名前	説明	サンプル値	デフォルト値
APP_LOG 4J	appender_applog_policies_type	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'Policies'	'Policies'
APP_LOG 4J	appender_applog_policies_size_type	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'SizeBasedTriggeringPolicy'	'SizeBasedTriggeringPolicy'
APP_LOG 4J	appender_applog_policies_size_size	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'5MB'	'10MB'
APP_LOG 4J	appender_applog_strategy_type	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'DefaultRolloverStrategy'	'DefaultRolloverStrategy'
APP_LOG 4J	appender_applog_strategy_max	すべての SecureAgent アプリケーションの共通プロパティ。 注: パラメータの値は一重引用符で囲む必要があります。	'5'	'5'

タイプ	名前	説明	サンプル値	デフォルト値
APP_LOG BACK	logback_log_file_pattern	すべての SecureAgent ア プリケーション の共通プロパテ イ。 注: パラメータの 値は一重引用符 で囲む必要があ ります。	'{yyyy-ww}'	'{yyyy-ww}'
APP_LOG BACK	logback_log_max_file_size	すべての SecureAgent ア プリケーション の共通プロパテ イ。 注: パラメータの 値は一重引用符 で囲む必要があ ります。	'5MB'	'10MB'
APP_LOG BACK	logback_log_max_history	すべての SecureAgent ア プリケーション の共通プロパテ イ。 注: パラメータの 値は一重引用符 で囲む必要があ ります。	'20'	'10'
APP_LOG BACK	logback_root_level	すべての SecureAgent ア プリケーション の共通プロパテ イ。 注: パラメータの 値は一重引用符 で囲む必要があ ります。	'DEBUG'	'INFO'

注: APP\_CFG はアプリケーション設定プロパティ、APP\_LOG4J はログ記録関連のアプリケーションプロパティ、APP\_LOGBACK はログファイルローテーション関連のアプリケーションプロパティを指します。

# 第 11 章

## 一括取り込み（ファイル）

Secure Agent グループが使用する一括取り込みファイルの動作を変更または最適化するには、Administrator のランタイム環境に一括取り込みプロパティを設定します。

以下のプロパティを設定する事ができます。

タイプ	名前	説明
AGENT_RUNTIME_SETTINGS	file-listener-snapshot-dir	新しいファイルリスナコンポーネントのスナップショットが追加されるディレクトリ。次のディレクトリパスを追加できます。 <ul style="list-style-type: none"><li>- MassIngestionRuntime ディレクトリに対する相対パス。例: ../data/monitor。</li><li>- 絶対パス。以下に例を示します。 &lt;Secure agent installation directory&gt;/apps/MassIngestionRuntime/data/monitor <i>Secure agent installation directory</i> には Secure Agent がインストールされているディレクトリの名前が入ります。</li></ul> <b>注:</b> グループに複数の Secure Agent が存在する場合は、すべてのエージェントで共有されるスナップショットディレクトリを使用します。
AGENT_RUNTIME_SETTINGS	mi-task-workspace-dir	ファイル取り込みタスクがファイルをターゲットに転送するとき中間ステージング領域として使用するエージェント内のディレクトリです。エージェント内のカスタムの場所のディレクトリ。このパスは、共有場所、マウントされた場所、およびエージェント内のデフォルトの場所とは異なる場所にすることができます。
AGENT_RUNTIME_SETTINGS	mi-task-project-dir	ファイル取り込みタスクがプロジェクトファイルを保存するディレクトリ。エージェント内のカスタムの場所のディレクトリ。このパスは、共有場所、マウントされた場所、およびエージェント内のデフォルトの場所とは異なる場所にすることができます。
AGENT_RUNTIME_SETTINGS	mi-task-logs-dir	ファイル取り込みタスクがタスクログファイルを保存するディレクトリ。エージェント内のカスタムの場所のディレクトリ。このパスは、共有場所、マウントされた場所、およびエージェント内のデフォルトの場所とは異なる場所にすることができます。



タイプ	名前	説明
AGENT_RUNTIME_SETTINGS	mi-task-quarantine-dir	<p>ウイルススキャンの実行時に検出された感染ファイルをファイル取り込みタスクが保存するディレクトリ。エージェント内のカスタムの場所のディレクトリ。このパスは、共有場所、マウントされた場所、およびエージェント内のデフォルトの場所とは異なる場所にすることができます。</p> <p>例: userdata\quarantine  <b>注:</b> 検疫ディレクトリを自動的にクリーンアップするには、検疫場所のエージェントプロパティを/tmp/informatica/fmi/quarantineのようなシステム一時ファイルの場所に設定します。</p>
AGENT_RUNTIME_SETTINGS	agent-dedup-repository	<p>スキップされた重複ファイルに関する情報は、Informatica Intelligent Cloud Services (IICS) に保存されます。スキップされた重複ファイルの情報を Secure Agent に保存するには、プロパティを true に設定します。</p> <p>デフォルトは false です。</p> <p>スキップされた重複に関する情報の保存については、一括取り込みファイルのガイドを参照してください。</p>
AGENT_RUNTIME_SETTINGS	mi-dedup-snapshot-dir	<p>スキップされた重複ファイルに関する情報を Secure Agent に保存するためのパスを入力します。</p> <p>agent-dedup-repository プロパティが true に設定されている場合のみ適用されます。</p>
AGENT_RUNTIME_SETTINGS	file-listener-max-pool-size	<p>ファイルリスナを実行するスレッドの最大数。デフォルトは 20 です。</p>
AGENT_RUNTIME_SETTINGS	file-listener-core-pool-size	<p>スレッドの合計数。デフォルトは 20 です。</p>
AGENT_RUNTIME_SETTINGS	fmi-task-max-pool-size	<p>ファイル取り込みタスクを実行するスレッドの最大数。デフォルトは 50 です。</p>
AGENT_RUNTIME_SETTINGS	fmi-task-core-pool-size	<p>スレッドの初期数または最小数。デフォルトは 20 です。</p>
AGENT_RUNTIME_SETTINGS	ftp-receive-socket-buffer-size	<p>FTP インバウンドパケットのバッファサイズ。デフォルトは 16 バイトです。</p>
AGENT_RUNTIME_SETTINGS	ftp-send-socket-buffer-size	<p>FTP アウトバウンドパケットのバッファサイズ。デフォルトは 16 バイトです。</p>
AGENT_RUNTIME_SETTINGS	http-client-timeout	<p>Informatica Intelligent Cloud Services へのエージェントの要求のタイムアウト時間 (秒単位)。デフォルトは 30 秒です。</p>

タイプ	名前	説明
PGP_SETTINGS	public-keyring-path	パブリックキーリングを保存するディレクトリ。次のディレクトリパスを追加できます。 <ul style="list-style-type: none"> <li>- 一括取り込みがインストールされるディレクトリに対する相対パス。以下に例を示します。  <code>../data/pubring.pkr</code>  <i>pubring.pkr</i>にはパブリックキーリングを保存するファイルの名前が入ります。</li> <li>- 絶対パス。以下に例を示します。  <code>&lt;Secure agent installation directory&gt;/apps/MassIngestionRuntime/data/pubring.pkr</code>  <i>pubring.pkr</i>にはパブリックキーリングを保存するファイルの名前が、<i>Secure agent installation directory</i>にはエージェントがインストールされているディレクトリの名前が入ります。</li> </ul>
PGP_SETTINGS	secret-keyring-path	シークレットキーリングを保存するディレクトリ。次のディレクトリパスを追加できます。 <ul style="list-style-type: none"> <li>- 一括取り込みがインストールされるディレクトリに対する相対パス。以下に例を示します。  <code>../data/secring.pkr</code>  <i>secring.pkr</i>にはシークレットキーリングを保存するファイルの名前が入ります。</li> <li>- 絶対パス。以下に例を示します。  <code>&lt;Secure agent installation directory&gt;/apps/MassIngestionRuntime/data/secring.pkr</code>  <i>secring.pkr</i>にはシークレットキーリングを保存するファイルの名前が、<i>Secure Agent installation directory</i>にはエージェントがインストールされているディレクトリの名前が入ります。</li> </ul>
JVM_SETTINGS	app-heap-size	一括取り込みファイルアプリケーションの最小および最大ヒープサイズ。 デフォルトは-Xms256m -Xmx2048m です。
JVM_SETTINGS	lcm-heap-size	ライフサイクル管理スクリプトの最小および最大ヒープサイズ。 デフォルトは-Xms32m -Xmx128m です。

Secure Agent を編集する場合は、**[カスタム構成の詳細]** 領域で次のプロパティを設定できます。

タイプ	名前	説明
AGENT_RUNTIME_SETTINGS	ComplexFileDisableWriteChecksum	crc ファイルを無視するには、値を True に設定します。ジョブは、Hadoop Files V2 をソースとし、Snowflake Cloud Data Warehouse V2 をターゲットとして正常に実行されます。

#### フォルダパスを指定するためのガイドライン

フォルダパスは、共有場所、マウントされた場所、および Secure Agent 内のデフォルトの場所とは異なる場所にすることができます。

次の表に、ソースフォルダパスの前後のスラッシュの使い方を示します。

ソース	フォルダパス
Windows	<フォルダパス> 例: C:\temp
Linux	/<フォルダパス>/ 例: /root/path
Windows の共有の場所	<フォルダパス>にスラッシュ (\) を追加 例えば、パス\\INV12B2B01\Shared\path は、\\\\INV12B2B01\\Shared\\path と指定されます

## 第 12 章

# メタデータ基盤アプリケーション

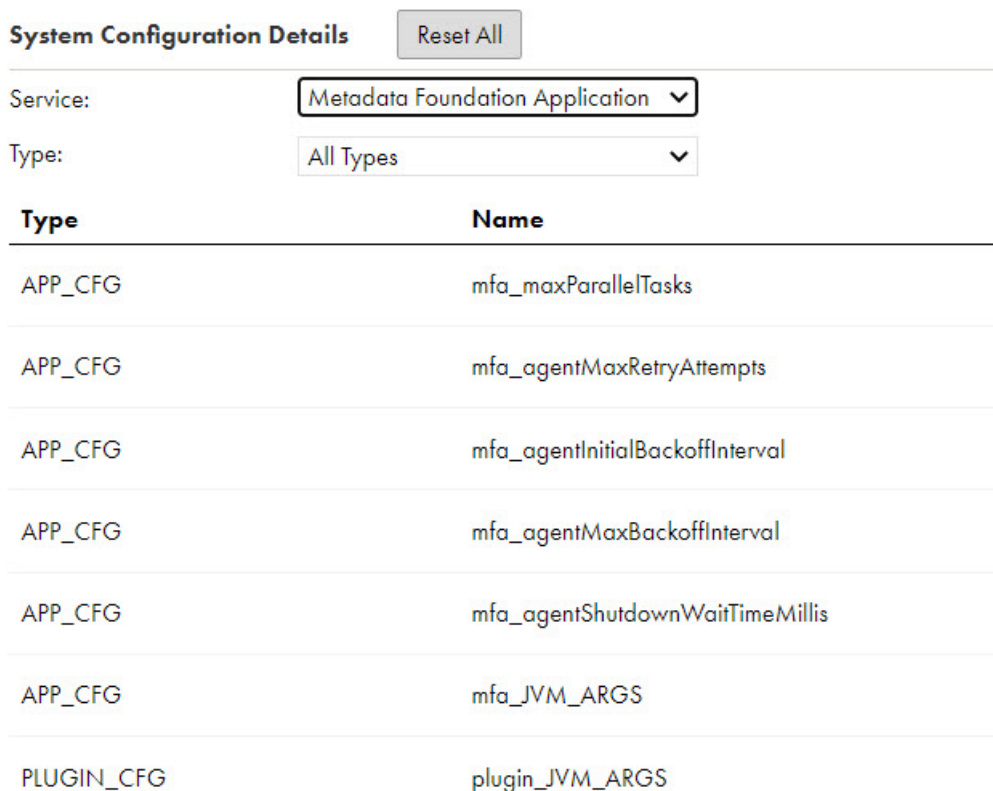
メタデータ基盤アプリケーションサービスを使用すると、組織内で設定されているソースシステムからメタデータを抽出し、抽出したメタデータを Secure Agent を介してメタデータコマンドセンターにアップロードできます。

いくつかのサービスプロパティを設定して、メタデータ基盤アプリケーションサービスのパフォーマンスを最適化できます。

## メタデータ基盤アプリケーションのプロパティ

メタデータ基盤アプリケーションサービスの動作を変更する、または最適化するには、Secure Agent の編集時にプロパティを **システム構成の詳細** セクションで設定します。

次の画像は、メタデータ基盤アプリケーションサービスのプロパティを示しています。



Type	Name
APP_CFG	mfa_maxParallelTasks
APP_CFG	mfa_agentMaxRetryAttempts
APP_CFG	mfa_agentInitialBackoffInterval
APP_CFG	mfa_agentMaxBackoffInterval
APP_CFG	mfa_agentShutdownWaitTimeMillis
APP_CFG	mfa_JVM_ARGS
PLUGIN_CFG	plugin_JVM_ARGS

以下のメタデータ基盤アプリケーションサービスのプロパティを設定できます。

タイプ	名前	説明	サンプル値	デフォルト値
APP_CFG	mfa_maxParallelTasks	Secure Agentで同時実行中のタスクの数。	5	4
APP_CFG	mfa_agentMaxRetryAttempts	-	3	5
APP_CFG	mfa_agentInitialBackoffInterval	-	500	100
APP_CFG	mfa_agentMaxBackoffInterval	-	20000	35000
APP_CFG	mfa_agentShutdownWaitTimeMillis	-	30000	60000
APP_CFG	mfa_JVM_ARGS	デバッグパラメータを追加する、またはJVMメモリ構成パラメータを変更するオプション。	agentlib:jdwp=transport=dt_socket,address=8084,server=y,suspend=y	-
PLUGIN_CFG	plugin_JVM_ARGS	デバッグパラメータを追加する、またはJVMメモリ構成パラメータを変更するオプション。	agentlib:jdwp=transport=dt_socket,address=8083,server=y,suspend=y	-
TRANSFER_SVC_CFG	transfer_svc_JVM_ARGS	デバッグパラメータを追加する、またはJVMメモリ構成パラメータを変更するオプション。	agentlib:jdwp=transport=dt_socket,address=8085,server=y,suspend=y	-
TRANSFER_SVC_CFG	transfer_svc_batchSize	バッチで処理できるコンテンツファイルの数。 このオプションを使用して、アップロードサービスを使用してアップロードされるCSVファイルの生成を最適化します。	4	1

タイプ	名前	説明	サンプル値	デフォルト値
TRANSFER_SVC_CFG	transfer_svc_stagingMaxRetry	コンテンツのステージング中に失敗した場合の再試行回数。	5	3
TRANSFER_SVC_CFG	transfer_svc_parallelTaskExecutorsSize	同時実行中のタスクの数。	5	4
TRANSFER_SVC_CFG	transfer_svc_contentBatchExecutorsSize	アップロードサービスを使用した同時コンテンツアップロードの数。	5	4
TRANSFER_SVC_CFG	transfer_svc_stagingBatchExecutorsSize	同時コンテンツステージングタスクの数。	5	4
TRANSFER_SVC_CFG	transfer_svc_contentWorkersBean	ステージング、取り込み、およびログ転送を行うサービス Bean を転送します。 ステージングを無効にするには、取り込みとログ転送のみを指定します。	ingestion,log	ingestion,staging,log
TRANSFER_SVC_CFG	transfer_svc_ingestionMaxRetry	失敗した場合のアップロードと一括取り込みの再試行回数。	5	3
TRANSFER_SVC_CFG	transfer_svc_contentMaxRetentionTimeInMin	アップロード後にメタデータの抽出結果を保持または削除するオプション。 デフォルトの動作では、結果が削除されます。	600	0

タイプ	名前	説明	サンプル値	デフォルト値
TRANSFER_SVC_CFG	transfer_svc_postDriverCompletionMaxWaitTimeInSec	長時間実行中または応答しない取り込みタスクが終了した後の秒数。 デフォルトの動作では、長時間実行中または応答しない取り込みタスクは終了しません。	10	-1
APP_LOG4J	name	すべてのSecureAgentアプリケーションの共通プロパティ。	AppLogPropertiesConfig	AppLogPropertiesConfig Xmx2048m
APP_LOG4J	rootLogger_level	すべてのSecureAgentアプリケーションの共通プロパティ。	warn	info
APP_LOG4J	rootLogger_appenderRefs	すべてのSecureAgentアプリケーションの共通プロパティ。	applog	applog
APP_LOG4J	rootLogger_appenderRefs_applog_ref	すべてのSecureAgentアプリケーションの共通プロパティ。	LOGFILE	LOGFILE
APP_LOG4J	appenders	すべてのSecureAgentアプリケーションの共通プロパティ。	applog	applog
APP_LOG4J	appender_applog_type	すべてのSecureAgentアプリケーションの共通プロパティ。	RollingFile	RollingFile
APP_LOG4J	appender_applog_name	すべてのSecureAgentアプリケーションの共通プロパティ。	LOGFILE	LOGFILE

タイプ	名前	説明	サンプル値	デフォルト値
APP_LOG4J	appender_app_log_filePattern	すべてのSecureAgentアプリケーションの共通プロパティ。	%d{MM-dd-yy-HH-mm-ss}-%i	%d{MM-dd-yy-HH-mm-ss}-%i
APP_LOG4J	appender_app_log_layout_type	すべてのSecureAgentアプリケーションの共通プロパティ。	PatternLayout	PatternLayout
APP_LOG4J	appender_app_log_layout_pattern	すべてのSecureAgentアプリケーションの共通プロパティ。	%d %d{z} %p [%c] - %m%n	%d %d{z} %p [%c] - %m%n
APP_LOG4J	appender_app_log_policies_type	すべてのSecureAgentアプリケーションの共通プロパティ。	ポリシー	ポリシー
APP_LOG4J	appender_app_log_policies_size_type	すべてのSecureAgentアプリケーションの共通プロパティ。	SizeBasedTriggeringPolicy	SizeBasedTriggeringPolicy
APP_LOG4J	appender_app_log_policies_size_size	すべてのSecureAgentアプリケーションの共通プロパティ。	5 MB	10 MB
APP_LOG4J	appender_app_log_strategy_type	すべてのSecureAgentアプリケーションの共通プロパティ。	DefaultRolloverStrategy	DefaultRolloverStrategy
APP_LOG4J	appender_app_log_strategy_max	すべてのSecureAgentアプリケーションの共通プロパティ。	5	5
APP_LOGBACK	logback_log_file_pattern	すべてのSecureAgentアプリケーションの共通プロパティ。	{yyyy-ww}	{yyyy-ww}



タイプ	名前	説明	サンプル値	デフォルト値
APP_LOGBACK	logback_log_max_file_size	すべてのSecureAgentアプリケーションの共通プロパティ。	5 MB	10 MB
APP_LOGBACK	logback_log_max_history	すべてのSecureAgentアプリケーションの共通プロパティ。	20	10
APP_LOGBACK	logback_root_level	すべてのSecureAgentアプリケーションの共通プロパティ。	デバッグ	情報

## 第 13 章

# メタデータプラットフォームサービス

メタデータプラットフォームサービスを使用すると、メタデータコマンドセンターで実行するジョブのプロファイリングアクティビティを実行できます。メタデータコマンドセンターのランタイム環境にアクティブなメタデータプラットフォームサービスが存在しない場合、プロファイリングは失敗します。

いくつかのサービスプロパティを設定して、メタデータプラットフォームサービスのパフォーマンスを最適化できます。

## メタデータプラットフォームサービスのプロパティ

メタデータプラットフォームサービスの動作を変更または最適化する場合は、Secure Agent の編集時にプロパティを **【システム構成の詳細】** セクションで設定します。

次の図は、メタデータプラットフォームサービスのプロパティの一部を示しています。

### ▼ System Configuration Details

Service:	Metadata Platform Service	▼
Type:	All Types	▼
Type	Name	Value
APP_CFG	mps_maxParallelTasks	4
APP_CFG	mps_agentMaxRetryAttempts	5
APP_CFG	mps_agentInitialBackoffInterval	100
APP_CFG	mps_agentMaxBackoffInterval	35000
APP_CFG	mps_agentShutdownWaitTimeMillis	60000
APP_CFG	mps_profileResultRetention	false
APP_CFG	mps_maxDeploymentBatchSize	100

次のようなメタデータプラットフォームサービスのプロパティを設定できます。

タイプ	名前	説明	サンプル値	デフォルト値
APP_CFG	mps_maxParallelTasks	同時に実行できるバッチプロファイリングサブタスクの最大数。	3	4
APP_CFG	mps_agentMaxRetryAttempts	-	3	5
APP_CFG	mps_agentInitialBackoffInterval	-	500	100
APP_CFG	mps_agentMaxBackoffInterval	-	20000	35000
APP_CFG	mps_agentShutdownWaitTimeMillis	-	30000	60000
APP_CFG	mps_profileResultRetention	-	false	false

タイプ	名前	説明	サンプル値	デフォルト値
APP_CFG	mps_maxDeploymentBatchSize	バッチプロファイリングサブタスクに含めることができるオブジェクトの最大数。	50	100 注: デプロイメントバッチサイズのパラメータの値は 1 以上である必要があります。
APP_CFG	mps_metadataComputeProfile	ロードタイプを定義します。	-	'Small:core.DataElement=1000;Medium:core.DataElement-10000;'
APP_CFG	mps_previewFileRetentionInDays	プロファイリング機能のカタログソースジョブに関連付けられたファイルが保存される最大日数。例えば、データプレビューの結果などがあります。	5	7
APP_CFG	mps_JVM_ARGS	デバッグパラメータを追加する、または JVM メモリ構成パラメータを変更するオプション。	-Xms32m -Xmx512m	-
TRANSFER_SVC_CFG	transfer_svc_batchSize	バッチで処理できるコンテンツファイルの数。 このオプションを使用して、アップロードサービスを使用してアップロードされる CSV ファイルの生成を最適化します。	4	1
TRANSFER_SVC_CFG	transfer_svc_stagingMaxRetry	コンテンツのステージング中に失敗した場合の再試行回数。	5	3
TRANSFER_SVC_CFG	transfer_svc_parallelTaskExecutorsSize	同時実行中のタスクの数。	5	4

タイプ	名前	説明	サンプル値	デフォルト値
TRANSFER_SVC_CFG	transfer_svc_contentBatchExecutorsSize	アップロードサービスを使用した同時コンテンツアップロードの数。	5	4
TRANSFER_SVC_CFG	transfer_svc_stagingBatchExecutorsSize	同時コンテンツステージングタスクの数。	5	4
TRANSFER_SVC_CFG	transfer_svc_contentWorkersBean	ステージング、取り込み、およびログ転送を行うサービス Bean を転送します。 ステージングを無効にするには、取り込みとログ転送のみを指定します。	ingestion,log	ingestion,staging,log
TRANSFER_SVC_CFG	transfer_svc_ingestionMaxRetry	失敗した場合のアップロードと一括取り込みの再試行回数。	5	3
TRANSFER_SVC_CFG	transfer_svc_contentMaxRetentionTimeInMin	アップロード後にメタデータの抽出結果を保持または削除するオプション。 デフォルトの動作では、結果が削除されます。	600	0
TRANSFER_SVC_CFG	transfer_svc_postDriverCompletionMaxWaitTimeInSec	長時間実行中または応答しない取り込みタスクが終了した後の秒数。 デフォルトの動作では、長時間実行中または応答しない取り込みタスクは終了しません。	10	-1

タイプ	名前	説明	サンプル値	デフォルト値
APP_LOG4J	name	すべてのSecureAgentアプリケーションの共通プロパティ。	AppLogPropertiesConfig	AppLogPropertiesConfig
APP_LOG4J	rootLogger_level	すべてのSecureAgentアプリケーションの共通プロパティ。	warn	info
APP_LOG4J	rootLogger_appenderRefs	すべてのSecureAgentアプリケーションの共通プロパティ。	applog	applog
APP_LOG4J	rootLogger_appenderRefs_applog_ref	すべてのSecureAgentアプリケーションの共通プロパティ。	LOGFILE	LOGFILE
APP_LOG4J	appenders	すべてのSecureAgentアプリケーションの共通プロパティ。	applog	applog
APP_LOG4J	appender_applog_type	すべてのSecureAgentアプリケーションの共通プロパティ。	RollingFile	RollingFile
APP_LOG4J	appender_applog_name	すべてのSecureAgentアプリケーションの共通プロパティ。	LOGFILE	LOGFILE
APP_LOG4J	appender_applog_filePattern	すべてのSecureAgentアプリケーションの共通プロパティ。	%d{MM-dd-yy-HH-mm-ss}-%i	%d{MM-dd-yy-HH-mm-ss}-%i
APP_LOG4J	appender_applog_layout_type	すべてのSecureAgentアプリケーションの共通プロパティ。	PatternLayout	PatternLayout

タイプ	名前	説明	サンプル値	デフォルト値
APP_LOG4J	appender_app_log_layout_pattern	すべてのSecureAgentアプリケーションの共通プロパティ。	%d %d{z} %p [%c] - %m%n	%d %d{z} %p [%c] - %m%n
APP_LOG4J	appender_app_log_policies_type	すべてのSecureAgentアプリケーションの共通プロパティ。	ポリシー	ポリシー
APP_LOG4J	appender_app_log_policies_size_type	すべてのSecureAgentアプリケーションの共通プロパティ。	SizeBasedTriggeringPolicy	SizeBasedTriggeringPolicy
APP_LOG4J	appender_app_log_policies_size_size	すべてのSecureAgentアプリケーションの共通プロパティ。	5 MB	10 MB
APP_LOG4J	appender_app_log_strategy_type	すべてのSecureAgentアプリケーションの共通プロパティ。	DefaultRolloverStrategy	DefaultRolloverStrategy
APP_LOG4J	appender_app_log_strategy_max	すべてのSecureAgentアプリケーションの共通プロパティ。	5	5
APP_LOGBACK	logback_log_file_pattern	すべてのSecureAgentアプリケーションの共通プロパティ。	{yyyy-ww}	{yyyy-ww}
APP_LOGBACK	logback_log_max_file_size	すべてのSecureAgentアプリケーションの共通プロパティ。	5 MB	10 MB

タイプ	名前	説明	サンプル値	デフォルト値
APP_LOGBACK	logback_log_max_history	すべての SecureAgent アプリケーションの共通プロパティ。	20	10
APP_LOGBACK	logback_root_level	すべての SecureAgent アプリケーションの共通プロパティ。	デバッグ	情報



## 第 14 章

# プロセスサーバー

プロセスサーバーとは、アプリケーションの統合のプロセス、コネクタ、および接続を実行する Secure Agent サービスです。

Secure Agent にアプリケーション統合のアセットをデプロイしたら、プロセスサーバーにもデプロイします。アセットを実行すると、プロセスサーバーによって実行されます。

PostgreSQL データベースには Secure Agent のプロセスサーバーサービスが付属しており、プロセスサーバーが収集および生成したメタデータが格納されます。

システムの次の場所にある PostgreSQL ディレクトリを検索します。

```
<Secure Agent installation directory>\apps\process-engine\data\PostGreSql
```

プロセスサーバーパッケージには PostgreSQL データベースが含まれています。ただし、別のバージョンの PostgreSQL データベースをプロセスサーバーに接続することもできます。また、既存のクラスタのデータベースダンプを PostgreSQL データベースに移行することもできます。詳細については、ドキュメントポータル の [How-To library](#) セクションの「設定」を参照してください。

## プロセスサーバーのプロパティ

プロセスサーバーの動作を変更または最適化するには、プロセスサーバーのプロパティを設定します。サーバー、Secure Agent グループ、Java 仮想マシン、コネクタ、データベース、およびログプロパティを設定できます。

次の図に、一部のプロセスサーバーのプロパティを示しています。

server	host-name	'localhost'
server	shutdown-port	7005
server	key-alias	'localhost'
server	key-store	'../conf/ae.keystore'
server	key-store-password	'password'
server	trust-store	'../conf/ae.cacerts'
server	trust-store-password	'changeit'
server	ldap-enabled-realm	false
server	ldap-properties	- key: connectionURL _value: ldap://\$[host_name]:10389 - key: connectionName _value: uid=admin,ou=system - key: connectionPassword _value: \$[pe.ldap.password] - key: authentication _value: simple - key: userBase _value: ou=people,DC=\$[host_name],DC=informatica,DC=com - key: userSearch _value: {uid={01}} - key: roleBase _value: ou=groups,DC=\$[host_name],DC=informatica,DC=com - key: roleName _value: cn - key: roleSearch _value: {uniqueMember={01}}
server	ssl-enabled-protocols	'TLSv1.2'
server	ephemeral-DH-key-size	2048
server	use-secure-ciphers-only	true

以下のサーバープロパティを設定できます。

名前	通信方式	説明
host-name	Secure Agent チャンネル	プロセスエンジンサーバーのホスト名。
shutdown-port	Secure Agent チャンネル	プロセスサーバー Tomcat のシャットダウンポート。
key-alias	HTTPS	HTTPS 通信のセキュリティキーが含まれるキーストアレコードの識別子。
key-store	HTTPS	アプリケーションの統合が HTTPS 通信に使用するキーストアファイルのパスとファイル名。 Secure Agent をインストールすると、次のデフォルトの場所にキーストアがインストールされます。 <Secure Agent インストールディレクトリ>/apps/process-engine/conf/ae.keystore また、相対パスを入力することもできます。例えば、現在の作業ディレクトリが Secure Agent インストールディレクトリである場合、ae.keystore ファイルを指すように次の値を入力します。 ../conf/ae.keystore 注: ファイルパスにはスラッシュのみを含めることができます (/)。
key-store-password	HTTPS	キーストアのパスワード。デフォルトは password です。

名前	通信方式	説明
trust-store	HTTPS	<p>アプリケーションの統合が HTTPS 通信に使用するトラストストアファイルのパスとファイル名。</p> <p>Secure Agent をインストールすると、デフォルトの場所にトラストストアがインストールされます。</p> <p>&lt;Secure Agent インストールディレクトリ&gt;/apps/process-engine/conf/ae.cacerts</p> <p>また、相対パスを入力することもできます。例えば、現在の作業ディレクトリが Secure Agent インストールディレクトリである場合、ae.cacerts ファイルを指すように次の値を入力します。</p> <p>../conf/ae.cacerts</p> <p>注: ファイルパスにはスラッシュのみを含めることができます (/)。</p> <p>サービスエンドポイント認証用の公開証明書をインポートする場合は、次の場所に配置します。</p> <p>&lt;Secure Agent インストールディレクトリ&gt;/apps/process-engine/conf/certs</p>
trust-store-password	HTTPS	<p>トラストストアのパスワード。デフォルトは changeit です。パスワードは変更できます。</p>
ldap-enabled-realm	HTTP/ HTTPS	<p>認証に LDAP プロバイダを使用する場合は、このプロパティを true に設定します。クラスタ化された Secure Agent を使用する場合は、LDAP プロバイダを認証の一括管理用として使用します。</p>
ldap-properties	HTTP/ HTTPS	<p>設定する必要がある LDAP プロパティ。LDAP プロバイダに合うように既存のプロパティを編集します。</p> <p>注: LDAP パスワードは画面に表示されません。\$(pe.ldap.password)の値は環境変数 PE_LDAP_PASSWORD から取得されます。</p>
ssl-enabled-protocols	HTTPS	<p>使用する TLS プロトコル。デフォルトのプロトコルは最も安全なプロトコルである TLSv1.2 です。互換性の問題が発生した場合のみ、この値を TLSv1.0 または TLSv1.1 などの古いバージョンに変更します。を参照してください。</p>
ephemeral-DH-key-size	HTTPS	<p>安全なアルゴリズムのキーの長さ。デフォルトは 2048 です。互換性の問題が発生した場合のみ、この値を変更します。</p>
use-secure-ciphers-only	HTTPS	<p>エンドポイントの呼び出し時に使用する暗号セットを安全な暗号のみに制限します。デフォルトは true です。互換性の問題が発生した場合のみ、この値を [False] に変更します。</p>
fips-enabled	HTTPS	<p>Secure Agent で連邦情報処理標準 (FIPS) モードを有効にするには、このプロパティを true に設定します。FIPS モードを有効にすると、Windows は FIPS 検証済みの暗号化アルゴリズムを使用します。</p> <p>デフォルトは false です。</p>

次の Secure Agent グループ（UI では「クラスタ」）のプロパティを設定できます。

名前	通信方式	説明
name	HTTP/ HTTPS	Secure Agent グループの名前。
primary-node	HTTP/ HTTPS	Secure Agent をマスターエージェントにする場合は、このプロパティを [True] に設定します。マスターエージェントを選択する場合は、Secure Agent クラスタを作成します。クラスタでは、すべての Secure Agent がマスター Secure Agent の PostgreSQL データベースを共有します。
load-balance-url	HTTP/ HTTPS	Secure Agent にデプロイされたプロセスの呼び出しに使用できるロードバランサ URL。 ロードバランサを使用する場合に適用されます。

次の Java 仮想マシンのプロパティを設定できます。

名前	通信方式	説明
min-heap	Secure Agent チャネル	プロセスサーバーが Tomcat JVM に割り当てる最小ヒープメモリ。
max-heap	Secure Agent チャネル	プロセスサーバーが Tomcat JVM に割り当てる最大ヒープメモリ。
additional-properties	Secure Agent チャネル	Tomcat JVM セットに追加できるカスタムシステムプロパティ。例えば、カスタムプロパティ-Dsun.net.inetaddr.ttl=60 を設定できます。

以下のコネクタプロパティを設定できます。

名前	通信方式	説明
http-port	HTTP	Secure Agent がデータを送信する HTTP ポート。デフォルトのポートは 7080 です。 REST および SOAP エンドポイントの URL の構造について詳しくは、アプリケーションの統合ヘルプを参照してください。
http-maxThreads	HTTP	プロセスサーバーが HTTP を介してアプリケーションの統合で作成する接続の最大数。
http-connectionTimeout	HTTP	プロセスサーバーが HTTP 接続の応答を待機する最大時間（ミリ秒単位）。
https-port	HTTPS	Secure Agent がデータを送信する HTTPS ポート。デフォルトのポートは 7443 です。 REST および SOAP エンドポイントの URL の構造について詳しくは、アプリケーションの統合ヘルプを参照してください。
https-maxThreads	HTTPS	プロセスサーバーが HTTPS を介してアプリケーションの統合で作成する接続の最大数。
https-connectionTimeout	HTTPS	プロセスサーバーが HTTPS 接続の応答を待機する最大時間（ミリ秒単位）。

名前	通信方式	説明
secure-channel maxThreads	Secure Agent チャンネル	プロセスサーバーがアプリケーションの統合で作成する接続の最大数。
secure-channel-connectionTimeout	Secure Agent チャンネル	プロセスサーバーが接続の応答を待機する最大時間（ミリ秒単位）。

以下のデータベースプロパティを設定できます。

名前	通信方式	説明
type	Secure Agent チャンネル	プロセスサーバーが実行されるデータベースタイプ。 <b>重要:</b> この設定は変更しないでください。アプリケーションの統合 Secure Agent は他のデータベースをサポートしていません。
driver	Secure Agent チャンネル	プロセスサーバーが実行されるデータベースドライバ。 <b>重要:</b> この設定は変更しないでください。Informatica Cloud Secure Agent は他のデータベースをサポートしていません。
URL	Secure Agent チャンネル	プロセスサーバーがデータベースにアクセスするときの URL。 <b>重要:</b> この設定は変更しないでください。Informatica Cloud Secure Agent は他のデータベースをサポートしていません。
maxActive	Secure Agent チャンネル	プロセスサーバーのデータベースに同時に割り当てられるアクティブな接続の最大数。
maxIdle	Secure Agent チャンネル	プロセスサーバーのデータベースで一度にアイドル状態のままにすることができる接続の最大数。アイドル状態の接続数がこの数を越えると、プロセスサーバーは接続を解放します。
maxWait	Secure Agent チャンネル	接続が存在しない場合にプロセスサーバーのデータベースが待機する最大時間。
connection-properties	Secure Agent チャンネル	データベース接続プロパティのキーと値のペア。デフォルトでは、一部のキーが使用できます。 デフォルトのキーは削除しないでください。ただし、これらのキーの値は変更することができます。 他のキーと値のペアを追加できます。例えば、次のキーと値のペアを追加できます。 キー: autoReconnect 値: true

別のログデータソースを作成し、プロセスログを既存のデータベースからデータソースにリダイレクトする場合は、次のプロパティを設定します。

名前	通信方式	説明
logUrl	Secure Agent チャネル	プロセスサーバーがログデータソースにアクセスしてプロセスログデータをリダイレクトする URL。 デフォルトは jdbc:postgresql://localhost:5432/activevos です。
logMaxActive	Secure Agent チャネル	プロセスサーバーのデータベースに同時に割り当てられるアクティブな接続の最大数のログ。 デフォルトは 50 です。
logMaxIdle	Secure Agent チャネル	プロセスサーバーのデータベース内で一度にアイドル状態のままになった接続の最大数のログ。 デフォルトは 5 です。
logMaxWait	Secure Agent チャネル	利用可能な接続がない場合に、プロセスサーバーのデータベースが接続を待機した最大時間のログ。 デフォルトは 30000 です。
logConnection-properties	Secure Agent チャネル	ログデータベース接続プロパティのキーと値のペア。デフォルトでは次のキーを使用できます。 <ul style="list-style-type: none"> <li>- key: timeBetweenEvictionRunsMillis value: 300000</li> <li>- key: testOnBorrow value: true</li> <li>- key: testWhileIdle value: true</li> </ul> デフォルトのキーは削除しないでください。ただし、これらのキーの値は変更することができます。 他のキーと値のペアを追加できます。例えば、次のキーと値のペアを追加できます。 <ul style="list-style-type: none"> <li>- key: autoReconnect value: true</li> </ul>

Secure Agent での個別のログデータソースの設定の詳細については、「[別のログデータソースの設定](#)」(ページ 72)を参照してください。

以下のログプロパティを設定できます。

名前	通信方式	説明
org.apache.catalina.core.ContainerBase.Catalina_localhost_level	Secure Agent チャネル	仮想マシンで Tomcat をホストするときの localhost.log ファイルでのログのレベル。 デフォルトは INFO です。
org.apache.catalina.core.ContainerBase.Catalina_localhost_manager_level	Secure Agent チャネル	仮想マシンで Tomcat をホストするときの manager.log ファイルでのログのレベル。 デフォルトは INFO です。
org.apache.catalina.core.ContainerBase.Catalina_localhost_host-manager_level	Secure Agent チャネル	仮想マシンで Tomcat をホストするときの host-manager.log ファイルでのログのレベル。 デフォルトは INFO です。

名前	通信方式	説明
log4j2_root_level	Secure Agent チャンネル	ROOT ロガーのログレベル。 デフォルトは INFO です。
additional-logging	Secure Agent チャンネル	特定のクラスのログペアの名前レベル。 デフォルトは次のとおりです: - name: org.apache.camel.component.file.remote.SftpOperations level: ERROR

[カスタム構成の詳細] セクションでは、次のカスタムプロパティを設定できます。

名前	タイプ	説明
https-clientAuth	コネクタ	プロセスサーバーのアップグレード後に相互認証を有効にするには、このプロパティを true に設定します。 このプロパティの設定の詳細については、 <a href="#">「プロセスサーバーの相互認証を有効にする」 (ページ 89)</a> を参照してください。
replication_upgrade	db	Secure Agent の PostgreSQL データベースのレプリケーションアップグレードを有効にするには、このプロパティを true に設定します。 このプロパティの設定の詳細については、 <a href="#">「レプリケーション技術を使用した PostgreSQL データベースのアップグレード」 (ページ 87)</a> を参照してください。
ssl-implementation	サーバー	Tomcat で使用される ssl 実装のクラス名をオーバーライドするには、このプロパティを設定します。 デフォルトは org.apache.tomcat.util.net.jsse.JSSEImplementation です。

変更を有効にするためには、カスタムプロパティを追加または編集した後にプロセスサーバーを再起動する必要があります。

カスタムプロパティの追加の詳細については、[第 16 章、「Secure Agent サービスプロパティの設定」 \(ページ 96\)](#)を参照してください。

## デフォルト接続データベースのプロパティ

次の表では、connection-properties データベースプロパティで利用可能なデフォルトキーについて説明します。

キー	説明
timeBetweenEvictionRuns	アイドル状態のオブジェクト evictor スレッドの実行と実行の間にプロセスサーバーが待機する時間 (ミリ秒)。
testOnBorrow value	プロセスサーバーはプールからオブジェクトを借用する前にオブジェクトを検証します。プロセスサーバーがオブジェクトを検証できない場合、プールからこのオブジェクトは削除されます。次に、プロセスサーバーは別のオブジェクトを借用しようとします。

キー	説明
testWhileIdle	プロセスサーバーは、アイドル状態のオブジェクト evictor（存在する場合）によってオブジェクトを検証します。プロセスサーバーがオブジェクトを検証できない場合、プールからこのオブジェクトは削除されます。
validationQuery value	呼び出し元に返す前にこのプールの接続を検証する SQL クエリ。このプロパティを指定する場合は、クエリが 1 つ以上の行を返す SQL SELECT ステートメントである必要があります。

## ログレベル

次の表に、プロセスサーバーの【ログ】プロパティで設定できるレベルを示しています。

レベル	説明
SEVERE	エラーを記録します。
WARNING	潜在的に有害な状況を記録します。
INFO	アプリケーションの進行状況の概要を表す情報イベントを記録します。
CONFIG	INFO レベルよりも詳細な情報イベントを記録します。
FINE	アプリケーションのデバッグに使用できる詳細な情報イベントを記録します。
FINER	FINE レベルよりも詳細な情報イベントを記録します。
FINEST	すべてのイベントを記録します。

## 別のログデータソースの設定

別のログデータソースを設定して、プロセスログを既存のデータベースからデータソースにリダイレクトすることで、既存のデータベースの負荷を軽減できます。

Secure Agent 上に別のログデータソースを作成するには、次の手順を実行します。

1. 次のように、ログデータベース環境のユーザー名とパスワードを使用して環境変数を作成します。

```
PE_DB_LOG_USERNAME
```

```
PE_DB_LOG_PASSWORD
```

この手順はオプションです。ログデータベースに別の環境変数を指定しない場合は、デフォルトのデータベースユーザー名とパスワード、つまり PE\_DB\_USERNAME と PE\_DB\_PASSWORD が使用されます。

2. 別のデータベースまたはスキーマを使用する場合は、次の DDL を実行して、そのスキーマ内に AeProcessLogData テーブルの構造を作成します。

```
CREATE TABLE AeProcessLogData
(
  ProcessId BIGINT NOT NULL,
  SequenceId BIGINT NOT NULL,
  PlanId BIGINT NOT NULL,
  TenantContextId VARCHAR(32),
  LocationPath TEXT NOT NULL,
  InstanceLocationId INT NOT NULL,
  DefLocationId INT NOT NULL,
```



```

CorrelationId INT NOT NULL,
EventId INT NOT NULL,
SessionId INT NOT NULL,
SourceId INT NOT NULL,
FaultName VARCHAR(255),
AncillaryStr TEXT,
AncillaryInt INT,
EventTime BIGINT NOT NULL,
DataDocument TEXT,
PRIMARY KEY (ProcessId, SequenceId)
);
CREATE INDEX AeLogDataPidInsId ON AeProcessLogData(PlanId, ProcessId, InstanceLocationId);

```

3. プロセスサーバーのプロパティで次のログデータソースのプロパティを設定します。

- logUrl
- logMaxActive
- logMaxIdle
- logMaxWait
- logConnection-properties

ログデータソースのプロパティの設定の詳細については、[「プロセスサーバーのプロパティ」](#) (ページ 65) を参照してください。

4. 変更を有効にするには、Secure Agent を再起動します。

## プロセスサーバーのサイズ決定に関する推奨事項

作業負荷に応じて Secure Agent のプロセスサーバーサービスを設定します。

リソースを最適化するには、次のサイズ決定に関する推奨事項を参照してください。

推奨事項	小	中	大
プロセス数	75	175	350
リソースキャッシュ (MB)	75	175	350
作業マネージャの最小スレッドプール	50	100	150
作業マネージャの最大スレッドプール	250	500	750
JVM 最小ヒープ (MB)	デフォルト	768	1024
JVM 最大ヒープ (MB)	デフォルト	デフォルト	4096

デフォルトの [JVM 最小ヒープ] は 512 MB で、デフォルトの [JVM 最大ヒープ] は 1536 MB です。

例えば、[プロセス数]、[リソースキャッシュ]、[作業マネージャの最小スレッドプール]、および [作業マネージャの最大スレッドプール] に関連するプロパティを設定するには、次の手順を実行します。

1. アプリケーション統合コンソールで、[サーバーの環境設定] をクリックします。
2. [サーバー設定] ページで、Secure Agent を選択します。
3. [サーバーの設定] タブをクリックします。

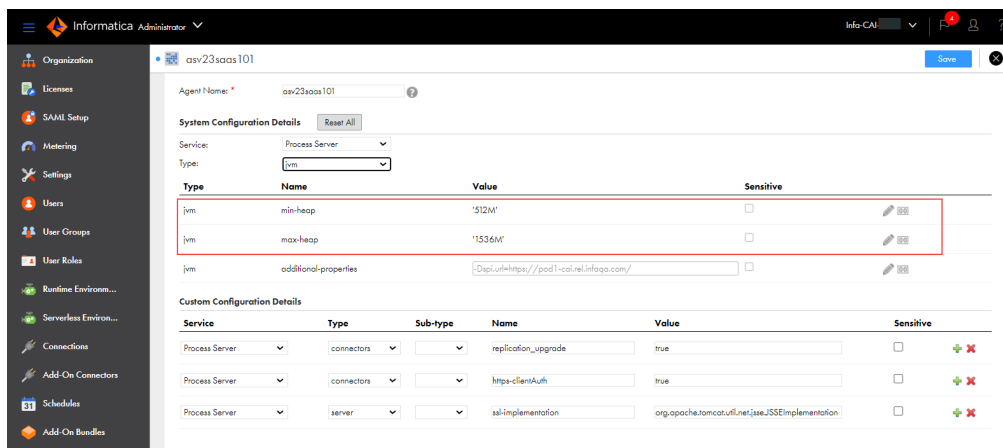
- 次の図に示すように、[プロセス数]、[リソースキャッシュ]、[作業マネージャの最小スレッドプール]、および [作業マネージャの最大スレッドプール] に関連するプロパティ値を更新します。

Property Name	Value
Process Activity Execution Limit:	10000
Process Count:	250
Process Idle Timeout (seconds):	10
Terminate suspended processes after (days):	0
Suspended processes termination timer interval (minutes):	1440
Suspended processes termination limit per run:	-1
Suspend long running processes after (days):	0
Running processes suspension timer interval (minutes):	1440
Running processes suspension limit per run:	-1
Resource Cache (MB):	100
Screen Cache Size (MB):	50
Unmatched Correlated Receive Timeout (seconds):	30
Web Service Invoke Timeout (seconds):	600
Web Service Reply Timeout (seconds):	600
Maximum HTTP Connections:	500
Maximum HTTP Connections Per Host:	100
Max Connection Timeout (seconds):	0
Message size limit:	5242880
Message TTL (seconds):	90
Message with attachments size limit:	5242880
Default Work Manager Thread Pool Min:	5
Default Work Manager Thread Pool Max:	100
System Work Manager Thread Pool Min:	1
System Work Manager Thread Pool Max:	50
Invoke Work Manager Thread Pool Min:	25
Invoke Work Manager Thread Pool Max:	300
Work Manager Threads For Alarms Max:	15
Work Manager Threads For Process Migration Max:	50
Work Manager Threads Per Process Max:	10

- [保存] をクリックします。
- 変更を有効にするためにはプロセスサーバーを再起動する必要があります。  
アプリケーション統合コンソールのプロセスサーバー設定プロパティの詳細については、[「Process Server properties」](#) を参照してください。  
[JVM 最小ヒープ] および [JVM 最大ヒープ] を設定するには、次の手順を実行します。

- Administrator で、[ランタイム環境] をクリックします。
- [ランタイム環境] ページで、Secure Agent の名前をクリックします。  
Secure Agent グループ内の Secure Agent を一覧表示するには、Secure Agent グループの展開が必要になる場合があります。
- [詳細] タブをクリックします。
- 右上隅の [編集] をクリックします。
- [システム構成の詳細] セクションまで下にスクロールします。

6. サービスとして **【プロセスサーバー】** を選択します。
7. 設定プロパティのタイプに **【jvm】** を選択します。
8. **【jvm min-heap】** および **【jvm max-heap】** プロパティを含む行で、**【エージェント設定の編集】** アイコンをクリックし、プロパティ値を更新します。  
次の図に、**【jvm min-heap】** プロパティと **【jvm max-heap】** プロパティを示します。



9. **【保存】** をクリックします。
10. 変更を有効にするためにはプロセスサーバーを再起動する必要があります。

Administrator のプロセスサーバープロパティの詳細については、[「プロセスサーバーのプロパティ」](#) (ページ 65) を参照してください。

UNIX オペレーティングシステムでプロセスサーバーを起動すると、次のエラーが表示されることがあります。

Cannot write to temp location [/tmp]

このエラーが発生するのは、UNIX が 1 つのプロセスで作成できるファイルの数を制限しているためです。1 つのプロセスで作成できるファイルの最大数は 1024 です。

このエラーを回避するには、開くファイルの制限を少なくともデフォルト値である 1024 の 10 倍に増やすことをお勧めします。システム管理者に、最大ユーザープロセスなどのその他の関連パラメータの値を増やすように依頼します。

Secure Agent のためのプロセスサーバーのサイズ決定の詳細については、以下のドキュメントを参照してください。

<https://knowledge.informatica.com/s/article/DOC-17439>

## Secure Agent との通信

Informatica Intelligent Cloud Services は、Secure Agent チャンネルまたは HTTP や HTTPS 直接リンクを介して Secure Agent からプロセスサーバーにデータを送信します。

Secure Agent は、次の 2 つの方法でプロセスサーバーと通信します。

### Secure Agent チャンネル

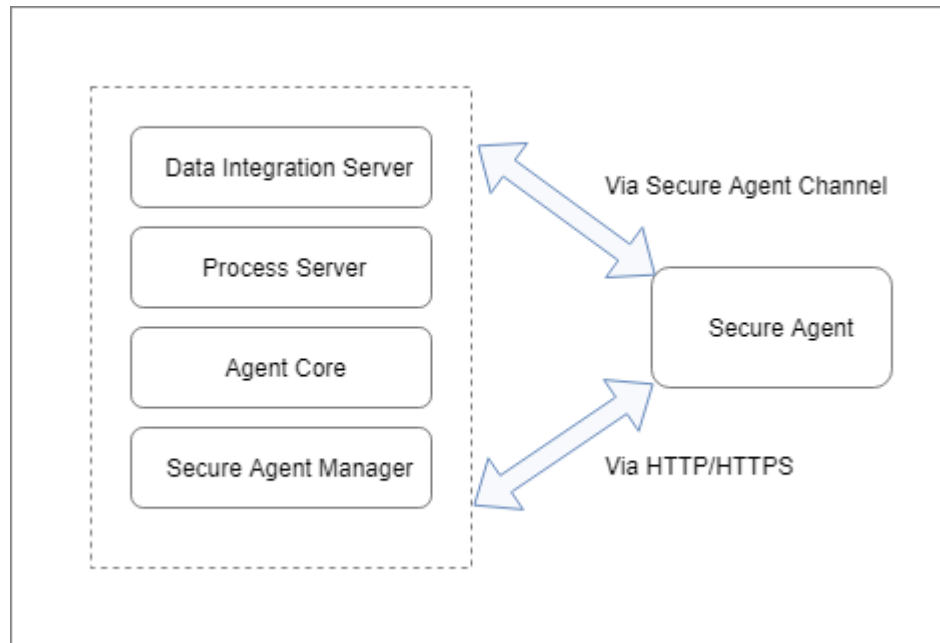
接続された各 Secure Agent とプロセスサーバー間にトンネルを作成するセキュアチャンネル。

## HTTP または HTTPS

Secure Agent がプロセスサーバーにデータを直接送信する場合のプロトコル。この通信方法を使用すると、Informatica Intelligent Cloud Services は認証プロバイダに対して資格情報を検証します。

各プロセスサーバープロパティが使用する通信方法の詳細については、[Process Server Properties \(ページ 65\)](#) を参照してください。

次の図に、Secure Agent とプロセスサーバー間の 2 つの通信方法を示します。



## プロセスサーバーのための Secure Agent の設定

ビジネス要件に基づいて、単一の Secure Agent または Secure Agent グループにアセットをデプロイします。プロセスサーバーが Secure Agent グループを使用している場合は、要件に基づいて、Secure Agent 負荷分散構成または Secure Agent クラスタ設定を使用できます。

アプリケーションの統合プロセス、接続、またはサービスコネクタを Secure Agent にデプロイする場合、これらのアセットを Secure Agent のプロセスサーバーサービスにデプロイします。そのプロセスサーバーサービスを使用するすべての Secure Agent は、同じ PostgreSQL データベースを使用します。

アセットを次の Secure Agent 構成に割り当てることができます。

### 単一の Secure Agent

単一の Secure Agent はグループ内の唯一のエージェントであったり、グループの複数のエージェントの 1 つであったりする可能性があります。

詳細については、「[単一の Secure Agent へのデプロイ](#)」(ページ 77)を参照してください。

### Secure Agent グループ

Secure Agent グループには複数のエージェントが含まれます。

次のような Secure Agent 設定を使用することで、Secure Agent グループを使用してプロセスサーバーサービスをデプロイできます。

### Secure Agent 負荷分散

アセットを Secure Agent グループにデプロイすると、Informatica Intelligent Cloud Services によって負荷分散が実行されます。ステートレス要求を処理する場合や OData 要求専用の Secure Agent を使用する場合に、要求を分散させるために Secure Agent 負荷分散構成を使用します。

### Secure Agent クラスタ

Secure Agent クラスタは、1つのマスタ Secure Agent を持つエージェントグループです。すべてのプロセスサーバーでプロセス実行アクティビティに関する情報を受信するようにする場合は、Secure Agent クラスタ構成を使用します。

詳細については、[Deploy to a Secure Agent group \(ページ 78\)](#)を参照してください。

以下の表は、さまざまなシナリオにおける Secure Agent のプロセス実行の概要を示しています。

	単一の Secure Agent	Secure Agent グループ	
		Secure Agent 負荷分散	Secure Agent クラスタ
使用可能なエージェント	プロセスが実行される。	プロセスが実行される。	プロセスが実行される。
使用不可能なエージェント	プロセスは実行されない。	プロセスは使用可能な Secure Agent のいずれかで実行される。	プロセスは使用可能な Secure Agent のいずれかで実行される。
エージェントが実行中に停止	プロセスは実行されない。	Secure Agent が停止したときにプロセスが停止される。	別の Secure Agent でプロセスの実行が継続される。

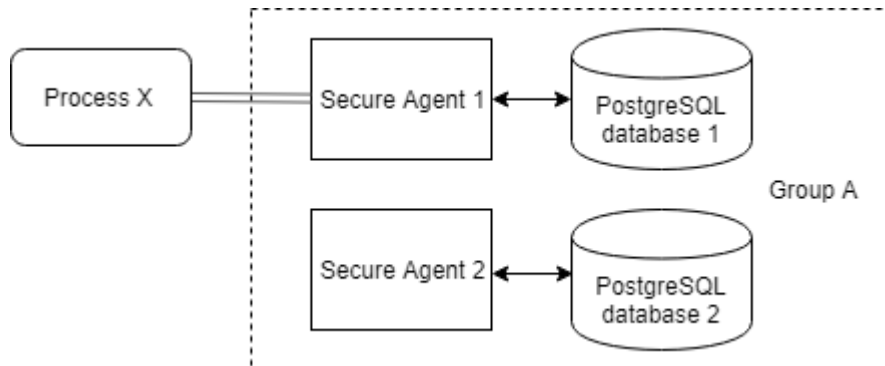
Secure Agent でのプロセスサーバーの負荷分散とクラスタリングの詳細については、ナレッジベースの記事 ([DOC-17437](#))を参照してください。

## 単一の Secure Agent へのデプロイ

アセットをグループ内の単一のエージェントに直接デプロイできます。

アセットを単一の Secure Agent にデプロイする場合、Secure Agent グループ内の他のプロセスサーバーはいずれも、アセット定義を受信しません。

次の図は、プロセス X を Secure Agent 1 に直接デプロイした場合の構成例を示しています。



プロセス X を実行できるのは Secure Agent 1 だけです。Secure Agent 1 が使用不可能になると、プロセスは実行されません。

## Secure Agent グループへのデプロイ

Secure Agent グループには複数のエージェントが含まれます。アセットを Secure Agent グループにデプロイすることができます。

ビジネス要件に応じて、Secure Agent グループを使用することで、次のような Secure Agent 設定を使用してプロセスサーバーサービスをデプロイできます。

### Secure Agent 負荷分散構成

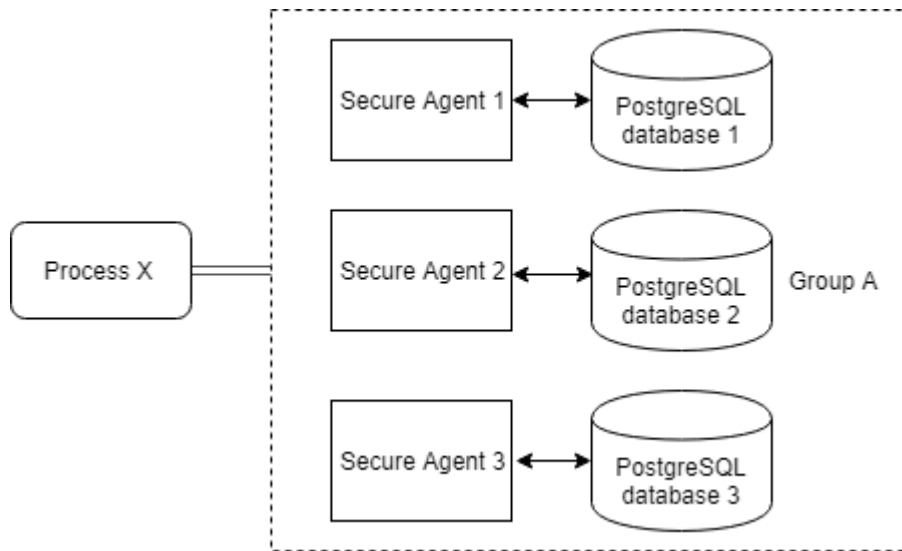
ステータス要求を処理する場合や OData 要求専用の Secure Agent を使用する場合に、要求を分散させるために Secure Agent 負荷分散構成を使用します。

アセットを Secure Agent グループにデプロイした場合は、負荷分散された構成を使用します。Informatica Intelligent Cloud Services によって負荷分散が実行されます。グループに複数の Secure Agent を追加して、プロセスサーバー間でタスクの分散を調整することができます。実行時に、受信された要求が Informatica Intelligent Cloud Services によって、使用可能な Secure Agent にラウンドロビン方式でディスパッチされません。

また、load-balance-url プロセスサーバープロパティを設定して、カスタムロードバランサを使用することもできます。詳細については、[Process Server Properties \(ページ 65\)](#)を参照してください。

グループ内のすべての Secure Agent は個別の PostgreSQL データベースを使用します。アセットを Secure Agent グループにデプロイすると、グループ内のすべてのプロセスサーバーが、新規のアセット定義または更新されたアセット定義に関する詳細を受信します。ただし、グループ内の他のプロセスサーバーはアセットの実行アクティビティに関する詳細を受信しません。例えば、プロセス実行中に Secure Agent で障害が発生しても、プロセスはグループ内の別の Secure Agent で継続して実行されません。

次の図は、プロセス X を Secure Agent グループ A にデプロイした場合の構成例を示しています。



プロセス X を変更して再パブリッシュすると、3 つすべての Secure Agent が更新された定義を受信します。すべての Secure Agent がプロセスを実行できます。

例えば、プロセスが開始され、Secure Agent 1 と Secure Agent 2 が使用不可能な場合、負荷分散された構成によって、Secure Agent 3 がプロセス X を実行することが保証されます。ただし、Secure Agent 1 と Secure Agent 2 は、プロセスが失敗したか正常に終了したかどうかについての情報を受信しません。プロセス X の実行中に Secure Agent 3 が停止した場合、プロセスはそれ以降実行されません。

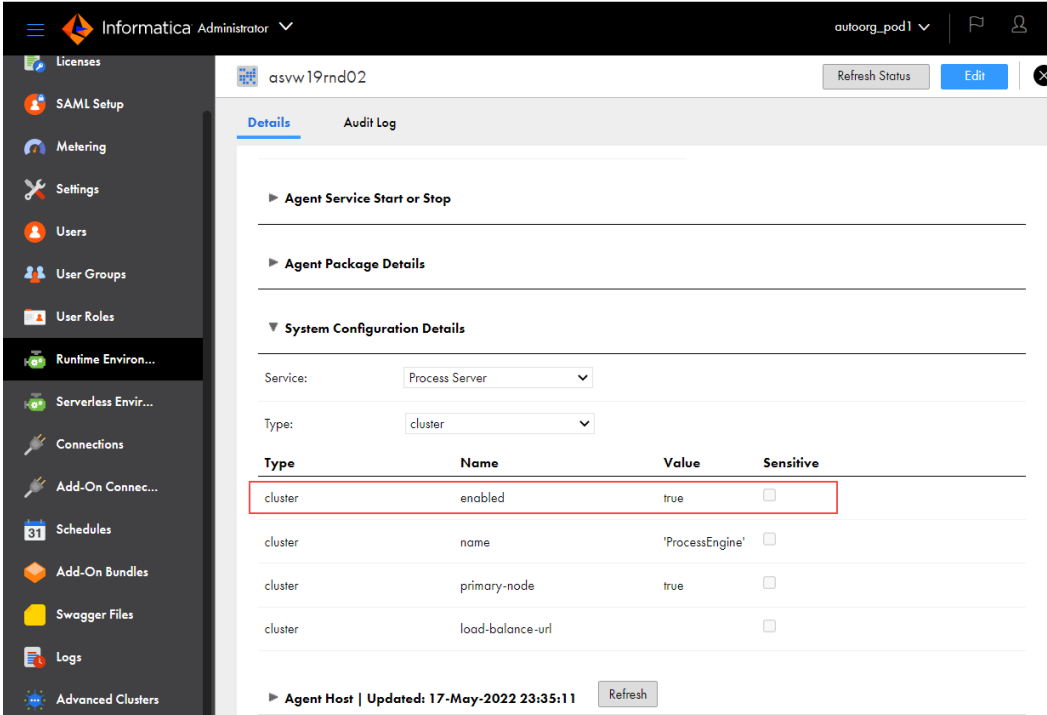
## Secure Agent クラスタ設定

実行時間の長いプロセスを処理しており、あるノードから別のノードへのペイロード処理のリカバリを実行する必要がある場合は、Secure Agent クラスタ設定を使用します。クラスタ設定は、単一のデータベースサーバーをすべてのクラスタノードと共有します。

Secure Agent クラスタは、1つのマスタ Secure Agent を持つエージェントグループです。アセットを Secure Agent クラスタにデプロイすることができます。

アセットを Secure Agent クラスタにデプロイすると、すべてのプロセスサーバーはプロセス実行アクティビティに関する情報を受信します。マスタ Secure Agent は情報を受信し、他のすべての Secure Agent に送信します。プロセス実行中に Secure Agent で障害が発生すると、プロセスはクラスタ内の別の Secure Agent で継続して実行されます。

プロセスサーバーのクラスタ設定を有効にするには、Administrator で **【ランタイム環境】** タブをクリックします。次の図に示すように、**【システム構成の詳細】** セクションで、クラスタ対応のプロセスサーバープロパティ値を true に設定します。



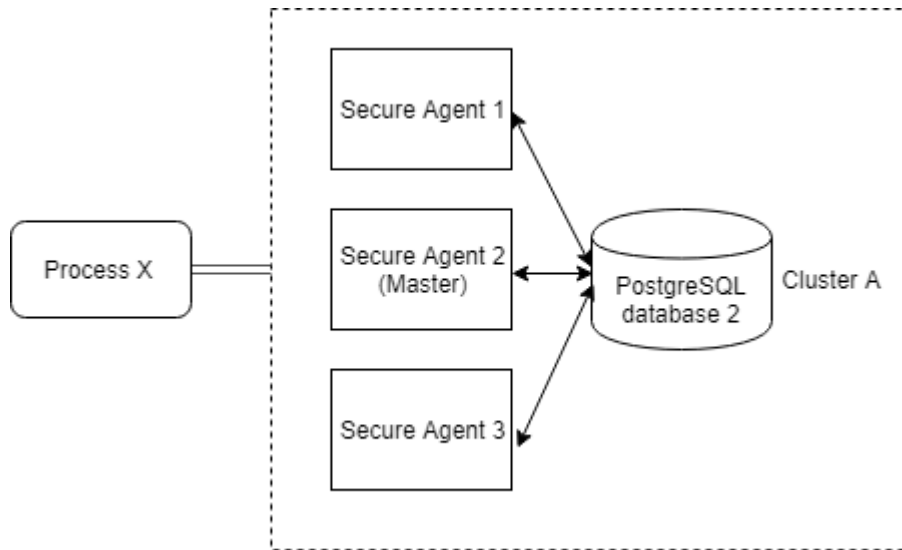
The screenshot shows the Informatica Administrator interface for a Process Server. The left sidebar contains navigation options like Licenses, SAML Setup, Metering, Settings, Users, User Groups, User Roles, Runtime Environ..., Serverless Envir..., Connections, Add-On Connec..., Schedules, Add-On Bundles, Swagger Files, Logs, and Advanced Clusters. The main content area shows the configuration for 'asvw19rnd02'. Under 'System Configuration Details', there is a table of properties:

Type	Name	Value	Sensitive
cluster	enabled	true	<input type="checkbox"/>
cluster	name	'ProcessEngine'	<input type="checkbox"/>
cluster	primary-node	true	<input type="checkbox"/>
cluster	load-balance-url		<input type="checkbox"/>

クラスタ内のすべてのプロセスサーバーはマスタエージェントの PostgreSQL データベースを共有します。

マスタ Secure Agent を定義するには、primary-node プロセスサーバープロパティを使用してください。詳細については、[Process Server Properties \(ページ 65\)](#)を参照してください。

次の図は、プロセス X を Secure Agent クラスタ A にデプロイした場合の構成例を示しています。



Secure Agent 3 がプロセス X の実行を開始し、途中でこのエージェントが停止した場合、Secure Agent 1 または Secure Agent 2 がそのプロセスの実行を継続します。

Secure Agent グループの詳細については、「ランタイム環境」を参照してください。

## PostgreSQL データベースのインストールとアップグレードの前提条件

PostgreSQL データベースをインストールまたはアップグレードする前に、次の前提条件が満たされていることを確認してください。

- 十分な空きディスク容量がある。Informatica では、最小空きディスク容量を次のディレクトリの Data フォルダの 2 倍のサイズにすることを勧めます。  
<Secure Agent installation directory>\apps\process-engine\data\PostGreSql\Data
- スクリプトを実行して PostgreSQL データベースをインストールまたはアップグレードするユーザーは、次のディレクトリ内のフォルダの読み取り、書き込み、および変更の権限を持っている必要がある。  
<Secure Agent installation directory>\apps\process-engine\data
- PostgreSQL データベースとプロセスサーバーを停止する。Secure Agent が詳細クラスターと関連付けられている場合は、すべてのノードでプロセスサーバーを停止する必要があります。  
**注:** このステップは、upgrade コマンドラインオプションを使用して db\_upgrade スクリプトを実行する場合に必要です。ただし、check コマンドラインオプションを使用して db\_upgrade スクリプトを実行する場合はオプションです。したがって、check コマンドラインオプションを使用すると、db\_upgrade スクリプトをダウンタイムなしで実行できます。
- Linux オペレーティングシステムでは、オペレーティングシステムユーザーのロケールは、PostgreSQL データベースのエンコーディングと一致するように UTF-8 エンコーディングを使用する必要がある。使用しない場合、プロセスサーバーの起動に失敗します。
- Linux オペレーティングシステムでは、GNU C (GLIBC) ライブラリファイルがバージョン 2.14 以降であることを確認する。バージョン番号を見つけるには、次のコマンドを実行します: `ldd --version`



# Windows での PostgreSQL データベースの管理

PostgreSQL データベースを管理するには、バイナリとユーティリティスクリプトを使用します。

**重要:** PostgreSQL データベースを管理するには、システム管理者権限を持たないユーザーとしてログインする必要があります。システム管理者は、PostgreSQL バイナリとユーティリティスクリプトを実行できません。

PostgreSQL バイナリに基づいていくつかのユーティリティスクリプトが作成されています。これらのユーティリティスクリプトを使用すると、PostgreSQL データベースを簡単に管理できます。

次のディレクトリには、PostgreSQL データベースのファイルが含まれます。

- PostgreSQL バイナリ: <Secure Agent installation directory>\apps\process-engine\data\db\postgresql-windows-x64-binaries\pgsql\bin
- PostgreSQL ユーティリティスクリプト: <Secure Agent installation directory>\apps\process-engine\data\db\util
- PostgreSQL ログ: <Secure Agent installation directory>\apps\process engine\logs\PostGreSql\postgresql.log
- PostgreSQL データ: <Secure Agent installation directory>\apps\process engine\data\PostGreSql\Data

PostgreSQL スクリプトの詳細については、<https://www.postgresql.org/docs/current/static/index.html> で PostgreSQL のヘルプを参照してください。

以降の一部のセクションには、次のデフォルト値を使用するサンプルコマンドが含まれています。

- デフォルトのデータベース名: activevos
- デフォルトのデータベースユーザー名: bpeluser
- デフォルトのデータベースパスワード: bpel

## Windows での PostgreSQL データベースのバックアップ

PostgreSQL データベースをバックアップするには、スクリプト db\_backup.bat を使用します。

PostgreSQL データベースをバックアップするには、次の手順を実行します。

1. <Secure Agent installation directory>\apps\process-engine\data\db\util に移動します。
2. 次のコマンドを実行します。  
db\_backup.bat <dbusername> <dbpassword> <バックアップファイルへのパスと、「.dump」という拡張子を持つバックアップファイルの名前> <dbport>

例えば、次のコマンドを実行すると、Secure Agent はバックアップファイル「BackupFile1.dump」を C:\postgre\backup に作成します。

```
db_backup.bat bpeluser bpel "C:\postgre\backup\BackupFile1.dump" 6432
```

**注:** dbport 引数はオプションです。デフォルトのポート 5432 とは異なるポートを使用する場合は、dbport 引数を指定します。

## Windows での PostgreSQL データベースのリストア

PostgreSQL データベースをバックアップファイルからリストアするには、コマンド db\_restore.bat を使用します。

PostgreSQL データベースファイルをバックアップファイルからリストアするには、次の手順を実行します。

1. <Secure Agent installation directory>\apps\process-engine\data\db\util に移動します。

2. 次のコマンドを実行します。

```
db_restore.bat <dbusername> <dbpassword> <ダンプファイルへのパス> <dbport>
```

例えば、次のコマンドを実行すると、ファイル BackupFile1.dump を使用して PostgreSQL データベースがリストアされます。

```
db_restore.bat bpeluser bpel "C:\postgre\backup\BackupFile1.dump" 6432
```

**注:** dbport 引数はオプションです。デフォルトのポート 5432 とは異なるポートを使用する場合は、dbport 引数を指定します。

## Windows での PostgreSQL データベースのリセット

PostgreSQL データベースをシャットダウンしてから、db\_reset.bat コマンドを使用してリセットします。

PostgreSQL データベースを元の状態にリセットするには、次の手順を実行します。

1. <Secure Agent installation directory>\apps\process-engine\data\db\util に移動します。
2. サーバーをシャットダウンするには、次のコマンドを実行します。  
server\_stop.bat
3. PostgreSQL データベースをリセットするには、次のコマンドを実行します。  
db\_reset.bat

## Windows での PostgreSQL サーバーの起動

Windows で PostgreSQL サーバーを起動するには、次の手順を実行します。

1. <Secure Agent installation directory>\apps\process-engine\data\db\util に移動します。
2. 次のスクリプトを実行します。  
server\_start.bat

**注:** デフォルトの 5432 ポートを使用しない場合は、次のように引数としてポート番号を渡す必要があります。 server\_start.bat <port\_number>

例: server\_start.bat 6789

## Windows での PostgreSQL サーバーの停止

Windows で PostgreSQL サーバーを停止するには、次の手順を実行します。

1. <Secure Agent installation directory>\apps\process-engine\data\db\util に移動します。
2. 次のスクリプトを実行します。  
server\_stop.bat

## Windows での PostgreSQL サーバーステータスの取得

Windows で PostgreSQL サーバーのステータスを取得するには、次の手順を実行します。

1. <Secure Agent installation directory>\apps\process-engine\data\db\util に移動します。
2. 次のスクリプトを実行します。  
server\_status.bat

**注:** デフォルトの 5432 ポートを使用しない場合は、次のように引数としてポート番号を渡す必要があります。 server\_status.bat <port\_number>

例: server\_status.bat 6789

## Windows での PostgreSQL データベースのクリーンアップ

PostgreSQL データベースをクリーンアップし、古いタブルを削除して領域を確保します。スクリプト `db_maintenance.bat` を使用して、PostgreSQL データベースをクリーンアップします。

デフォルトで、PostgreSQL データベースの自動クリーンアップが行われます。データベースを手動でクリーンアップする場合は、次の手順を実行します。

1. `<Secure Agent installation directory>\apps\process-engine\data\db\util` に移動します。
2. データベース全体をクリーンアップするには、次のコマンドを実行します。  
`db_maintenance.bat <dbusername> <dbpassword> <dbport> vacuum`
3. 単一テーブルをクリーンアップするには、次のコマンドを実行します。  
`db_maintenance.bat <dbusername> <dbpassword> <dbport> vacuum <tablename>`

例えば、次のコマンドを実行すると、「`aeprocesslogdata`」テーブルがクリーンアップされます。

```
db_maintenance.bat bpeluser bpel 5432 vacuum aeprocesslogdata
```

**注:** デフォルトのポート 5432 を使用する場合でも、`dbport` 引数は必須です。

## Windows での PostgreSQL データベースの再インデックス化

PostgreSQL でデータをクリーンアップした後、インデックスをクリーンアップして使用領域を解放するには、再インデックス化オプションを使用します。スクリプト `db_maintenance.bat` を使用して、PostgreSQL データベースを再インデックス化します。

PostgreSQL データベースを再インデックス化するには、次の手順を実行します。

1. `<Secure Agent インストールディレクトリ>\apps\process-engine\data\db\util` に移動します。
2. データベース全体を再インデックス化するには、次のコマンドを実行します。  
`db_maintenance.bat <dbusername> <dbpassword> <dbport> reindex`
3. 単一テーブルを再インデックス化するには、次のコマンドを実行します。  
`db_maintenance.bat <dbusername> <dbpassword> <dbport> reindex <tablename>`

例えば、次のコマンドを実行すると、「`aeprocesslogdata`」テーブルが再インデックス化されます。

```
db_maintenance.bat bpeluser bpel 5432 reindex aeprocesslogdata
```

**注:** デフォルトのポート 5432 を使用する場合でも、`dbport` 引数は必須です。

## Windows でのトランザクションログのリセット

制御情報の破損が原因で PostgreSQL サーバーが起動しない場合は、コマンド `pg_resetxlog.exe` を使用して制御情報をリセットします。

PostgreSQL データベースの制御情報をリセットするには、次の手順を実行します。

1. `<Secure Agent installation directory>\apps\process-engine\data\db\postgresql-windows-x64-binaries\pgsql\bin` に移動します。
2. 次のコマンドを実行します。  
`pg_resetxlog.exe -D <path to postgresql data directory>`

例えば、次のコマンドを実行すると、「`Data`」ディレクトリのトランザクションログがリセットされます。

```
pg_resetxlog.exe -D "C:\postgre\apps\process-engine\data\PostGreSql\Data"
```

# Linux での PostgreSQL データベースの管理

PostgreSQL データベースを管理するには、バイナリとユーティリティスクリプトを使用します。

**重要:** PostgreSQL データベースを管理するには、ルートアクセス権を持たないユーザーとしてログインする必要があります。ルートユーザーは、PostgreSQL バイナリとユーティリティスクリプトを実行できません。

PostgreSQL バイナリに基づいていくつかのユーティリティスクリプトが作成されています。これらのユーティリティスクリプトを使用すると、PostgreSQL データベースを簡単に管理できます。

次のディレクトリには、PostgreSQL データベースのファイルが含まれます。

- PostgreSQL バイナリ: <Secure Agent installation directory>/apps/process-engine/data/db/postgresql-linux-x64-binaries/pgsql/bin
- PostgreSQL ユーティリティスクリプト: <Secure Agent installation directory>/apps/process-engine/data/db/util
- PostgreSQL ログ: <Secure Agent installation directory>/apps/process-engine/logs/PostgreSQL/postgresql.log
- PostgreSQL データ: <Secure Agent installation directory>/apps/process-engine/data/PostgreSQL/Data

PostgreSQL スクリプトの詳細については、<https://www.postgresql.org/docs/current/static/index.html> で PostgreSQL のヘルプを参照してください。

一部のセクションには、次のデフォルト値を使用するサンプルコマンドが含まれています。

- デフォルトのデータベース名: activevos
- デフォルトのデータベースユーザー名: bpeluser
- デフォルトのデータベースパスワード: bpel

## Linux での PostgreSQL データベースのバックアップ

PostgreSQL データベースをバックアップするには、スクリプト db\_backup を使用します。

PostgreSQL データベースをバックアップするには、次の手順を実行します。

1. <Secure Agent installation directory>/apps/process-engine/data/db/util に移動します。
2. 次のコマンドを実行します。  
db\_backup.sh <dbusername> <dbpassword> <バックアップファイルの名前とバックアップファイルへのパス> .dump <dbport>

例えば、次のコマンドを実行すると、Secure Agent はバックアップファイル「backupfile1.dump」を/home/data/myfolder/に作成します。

```
db_backup.sh bpeluser bpel "/home/data/myfolder/backupfile1.dump" 6432
```

**注:** dbport 引数はオプションです。デフォルトのポート 5432 とは異なるポートを使用する場合は、dbport 引数を指定します。

## Linux での PostgreSQL データベースのリストア

PostgreSQL データベースをバックアップファイルからリストアするには、スクリプト db\_restore.sh を使用します。

PostgreSQL データベースファイルをバックアップファイルからリストアするには、次の手順を実行します。

1. <Secure Agent installation directory>/apps/process-engine/data/db/util に移動します。

2. 次のコマンドを実行します。  
`db_restore.sh <dbusername> <dbpassword> <ダンプファイルへのパス> <dbport>`

例えば、次のコマンドを実行すると、ファイル `backupfile1.dump` を使用して PostgreSQL データベースがリストアされます。

```
db_restore.sh bpeluser bpel "/home/data/myfolder/backupfile1.dump" 6432
```

**注:** `dbport` 引数はオプションです。デフォルトのポート 5432 とは異なるポートを使用する場合は、`dbport` 引数を指定します。

## Linux での PostgreSQL データベースのリセット

最初に PostgreSQL データベースをシャットダウンしてから、スクリプト `db_reset.sh` を使用してリセットします。

PostgreSQL データベースを元の状態にリセットするには、次の手順を実行します。

1. `<Secure Agent installation directory>/apps/process-engine/data/db/util` に移動します。
2. サーバーをシャットダウンするには、次のスクリプトを実行します。  
`server_stop.sh`
3. PostgreSQL データベースをリセットするには、次のスクリプトを実行します。  
`db_reset.sh`

## Linux での PostgreSQL サーバーの起動

Linux で PostgreSQL サーバーを起動するには、次の手順を実行します。

1. `<Secure Agent installation directory>/apps/process-engine/data/db/util` に移動します。
2. 次のスクリプトを実行します。  
`server_start.sh`

**注:** デフォルトの 5432 ポートを使用しない場合は、次のように引数としてポート番号を渡す必要があります。  
`server_start.sh <port_number>`

例: `server_start.sh 6789`

## Linux での PostgreSQL サーバーの停止

PostgreSQL サーバーを停止するには、次の手順を実行します。

1. `<Secure Agent installation directory>/apps/process-engine/data/db/util` に移動します。
2. 次のスクリプトを実行します。  
`server_stop.sh`

## Linux での PostgreSQL サーバーステータスの取得

Linux で PostgreSQL サーバーのステータスを取得するには、次の手順を実行します。

1. `<Secure Agent installation directory>/apps/process-engine/data/db/util` に移動します。
2. 次のスクリプトを実行します。  
`server_status.sh`

**注:** デフォルトの 5432 ポートを使用しない場合は、次のように引数としてポート番号を渡す必要があります。  
`server_status.sh <port_number>`

例: `server_status.sh 6789`

## Linux での PostgreSQL データベースのクリーンアップ

PostgreSQL データベースをクリーンアップし、古いタブルを削除して領域を確保します。スクリプト `db_maintenance.sh` を使用して、PostgreSQL データベースをクリーンアップします。

デフォルトで、PostgreSQL データベースの自動クリーンアップが行われます。データベースを手動でクリーンアップする場合は、次の手順を実行します。

1. `<Secure Agent installation directory>/apps/process-engine/data/db/util` に移動します。
2. データベース全体をクリーンアップするには、次のコマンドを実行します。  
`db_maintenance <dbusername> <dbpassword> <dbport> vacuum`
3. 単一テーブルをクリーンアップするには、次のコマンドを実行します。  
`db_maintenance.sh <dbusername> <dbpassword> <dbport> vacuum <tablename>`

例えば、次のコマンドを実行すると、「`aeprocesslogdata`」テーブルがクリーンアップされます。

```
db_maintenance.sh bpeluser bpel 5432 vacuum aeprocesslogdata
```

**注:** デフォルトのポート 5432 を使用する場合でも、`dbport` 引数は必須です。

## Linux での PostgreSQL データベースの再インデックス化

PostgreSQL でデータをクリーンアップした後、インデックスをクリーンアップして使用領域を解放するには、再インデックス化オプションを使用します。スクリプト `db_maintenance.sh` を使用して、PostgreSQL データベースを再インデックス化します。

PostgreSQL データベースを再インデックス化するには、次の手順を実行します。

1. `<Secure Agent installation directory>/apps/process-engine/data/db/util` に移動します。
2. データベース全体を再インデックス化するには、次のコマンドを実行します。  
`db_maintenance <dbusername> <dbpassword> <dbport> reindex`
3. 単一テーブルを再インデックス化するには、次のコマンドを実行します。  
`db_maintenance.sh <dbusername> <dbpassword> <dbport> reindex <tablename>`

例えば、次のコマンドを実行すると、「`aeprocesslogdata`」テーブルが再インデックス化されます。

```
db_maintenance.sh bpeluser bpel 5432 reindex aeprocesslogdata
```

**注:** デフォルトのポート 5432 を使用する場合でも、`dbport` 引数は必須です。

## Linux でのトランザクションログのリセット

制御情報の破損が原因で PostgreSQL サーバーが起動しない場合は、コマンド `pg_resetxlog` を使用して制御情報をリセットします。

PostgreSQL データベースの制御情報をリセットするには、次の手順を実行します。

1. `<Secure Agent installation directory>/apps/process-engine/data/db/postgresql-linux-x64-binaries/pgsql/bin` に移動します。
2. 次のコマンドを実行します。  
`pg_resetxlog -D <path to postgresql data directory>`

例えば、次のコマンドを実行すると、「`Data`」ディレクトリのトランザクションログがリセットされます。

```
pg_resetxlog -D "home/apps/process engine/data/PostGreSql/Data"
```

# PostgreSQL データベースのアップグレード

PostgreSQL データベースをバージョン 9.5.2、12.4、または 12.6 からバージョン 13.5 にアップグレードできます。バージョン 13.5 ではセキュリティ、パフォーマンス、および拡張性が強化されています。

Informatica が提供するスクリプトを手動で実行することで、都合に合わせてアップグレードを行うことができます。このスクリプトを実行すると既存のデータベースバージョンのバックアップが作成され、アップグレードで問題が発生した場合でも古いデータベースバージョンに復元することが可能です。

PostgreSQL データベースをアップグレードする方法の詳細については、次のコミュニティ記事を参照してください。

<https://knowledge.informatica.com/s/article/DOC-18945>

## レプリケーション技術を使用した PostgreSQL データベースのアップグレード

プロセスサーバーの PostgreSQL データベースのバージョンがバージョン 13.5 より前の場合、プロセスサーバーの再起動時にデータベースをアップグレードできます。

プロセスサーバーの PostgreSQL データベースのアップグレードは、デフォルトで無効になっています。Secure Agent の PostgreSQL データベースのレプリケーションアップグレードを有効にするには、Secure Agent のカスタムプロパティを追加する必要があります。

Secure Agent のカスタムプロパティを追加するには、次の手順を実行します。

1. [Administrator] で、[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、Secure Agent の名前をクリックします。  
Secure Agent グループ内の Secure Agent を一覧表示するには、Secure Agent グループの展開が必要になる場合があります。
3. [詳細] タブをクリックします。
4. 右上隅の [編集] をクリックします。
5. [カスタム構成の詳細] 領域までスクロールダウンします。
6. 次の画像は、[カスタム構成の詳細] 領域を示しています。

Custom Configuration Details

Service	Type	Sub-type	Name	Value	Sensitive
					<input type="checkbox"/>

7. すでに設定されているカスタムプロパティがある場合、[追加] アイコンをクリックして、新しいプロパティ行を追加します。
8. サービスとして [プロセスサーバー] を選択します。
9. 設定プロパティの種類として [db] を選択します。
10. プロパティ名として「replication\_upgrade」、値として「true」を入力します。
11. [保存] をクリックします。  
プロセスサーバーサービスのステータスが [再起動が必要です] と表示されます。

プロセスサーバーを初めて再起動すると、データベースは既存のデータベースバージョンからデータのレプリケーションを開始します。プロセスサーバーを再起動するたびに、データベースはレプリケーションステータスを検証します。レプリケーションの完了後、プロセスサーバーを次に再起動するときには、古いバージョンの PostgreSQL データベースはシャットダウンされ、最新バージョンのデータベースが自動的に起動されます。

これにより、Informatica が提供するスクリプトを実行して、プロセスサーバーの PostgreSQL データベースを手動で最新バージョンにアップグレードする必要がなくなります。

## PostgreSQL 構成ファイル

PostgreSQL データベースをインストールすると、postgresql.conf ファイルが次のディレクトリに自動的に作成されます。

```
<SecureAgent インストールディレクトリ>\apps\process-engine\data\PostGreSql\Data
```

postgresql.conf ファイルの設定パラメータにより、監査、認証、暗号化、およびその他の動作に関連するサーバープロパティのデフォルト値を定義します。

postgresql.conf ファイルは、新しいバージョンの PostgreSQL データベースをインストールまたはアップグレードするたびに上書きされます。postgresql.conf ファイルが上書きされると変更が失われるため、カスタマイズされた動作に対してこのファイルを更新しないでください。

user.conf ファイルを使用して、postgresql.conf ファイルで定義されているデフォルト値を上書きします。

user.conf ファイルが存在しない場合は、postgresql.conf ファイルと同じディレクトリにファイルを作成し、値を上書きします。PostgreSQL データベースを再起動すると、変更が有効になります。

## PostgreSQL ログローテーションの設定

PostgreSQL ログには、Secure Agent でパッケージ化されている PostgreSQL データベースのログ情報が含まれています。

PostgreSQL ログは次のディレクトリに含まれています。

```
<Secure Agent インストールディレクトリ>\apps\process-engine\logs\PostGreSql\postgresql.log
```

時間の経過とともに PostgreSQL ログのサイズは非常に大きくなるため、管理が困難になる可能性があります。ログローテーションを設定してファイルサイズを縮小し、ファイルの管理を簡単にすることができます。時間またはファイルサイズに基づいてログローテーションを設定できます。

1. user.conf という名前のファイルが存在しない場合は、次の場所に作成します。

```
<SecureAgent インストールディレクトリ>\apps\process-engine\data\PostGreSql\Data
```

user.conf ファイルは、postgresql.conf ファイルで定義されている値を上書きします。

2. 次のいずれかの手順を実行します。

- 時間に基づいてログをローテーションするには、user.conf ファイルに次のプロパティを追加します。

```
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'  
log_rotation_age=<value in minutes>
```

例えば、log\_rotation\_age プロパティの値を 1440 に設定すると、ログファイルは毎日ローテーションされます。

- ファイルサイズに基づいてログをローテーションするには、user.conf ファイルに次のプロパティを追加します。

```
log_filename = 'postgresql-%Y-%m-%d_%H%M%S.log'  
log_rotation_size=<value in kilobytes>  
log_truncate_on_rotation=on
```



例えば、log\_rotation\_size プロパティの値を 10240 に設定すると、ファイルサイズが 10 MB を超えた場合にログファイルがローテーションされます。

3. user.conf ファイルを保存します。
4. PostgreSQL データベースを再起動して、変更を有効にします。

## プロセスサーバーに対するパブリック証明書とプライベートキーの設定

アプリケーション統合プロセスと接続を使用して SSL 対応エンドポイントに接続する場合は、パブリック証明書またはプライベートキー、あるいはその両方が必要です。プロセスと接続用のパブリック証明書をプライベートキーを Secure Agent にインポートする必要があります。

プロセスサーバーのパブリック証明書とプライベートキーを設定した後に、プロセスサーバーの相互認証を有効にします。

### プロセスと接続用のパブリック証明書とプライベートキーのインポート

Web サービス、キュー、JDBC 接続などの SSL 対応エンドポイントに接続するには、パブリック証明書やプライベートキーが必要です。

プロセスまたは接続がこれらのエンドポイントへの SSL 対応接続を確立するためにパブリッシュされる Secure Agent マシンに証明書をインポートする必要があります。

パブリック証明書やプライベートキーをインポートするには、次の手順を実行します。

- パブリック証明書の場合は、証明書ファイルを以下の場所に配置して、SecureAgent を再起動します。  
<Secure Agent installation directory>/apps/process-engine/conf/certs
- プライベートキーの場合は、キーを以下の場所の ae.keystore ファイルにインポートして、Secure Agent を再起動します。

<Secure Agent installation directory>/apps/process-engine/conf

上記の certs フォルダに x509 形式のパブリック証明書ファイルをインポートし、配置する必要があります。使いやすさとアップグレードとの互換性を確保するために、証明書とキーを同じ場所にインポートする必要があります。

さらに、Informatica Keystore 内に秘密プライベートキーをインポートするには、秘密鍵が同じキーストア形式、つまり PKCS12 ".p12" である必要があります。例えば、秘密鍵が ".pfx" 形式で提供された場合、それを ".p12" に変換する必要があります。これは、証明書プロバイダーで確認できます。

localhost ではなく、ドメイン名で Secure Agent に接続するには、証明書に接続し、certs フォルダにコピーするドメイン名に基づいて証明書を生成できます。

### プロセスサーバーの相互認証を有効にする

プロセスサーバーを最新のパッケージでアップグレードした後に、次のいずれかの方法を使用してプロセスサーバーの相互認証を有効にします。

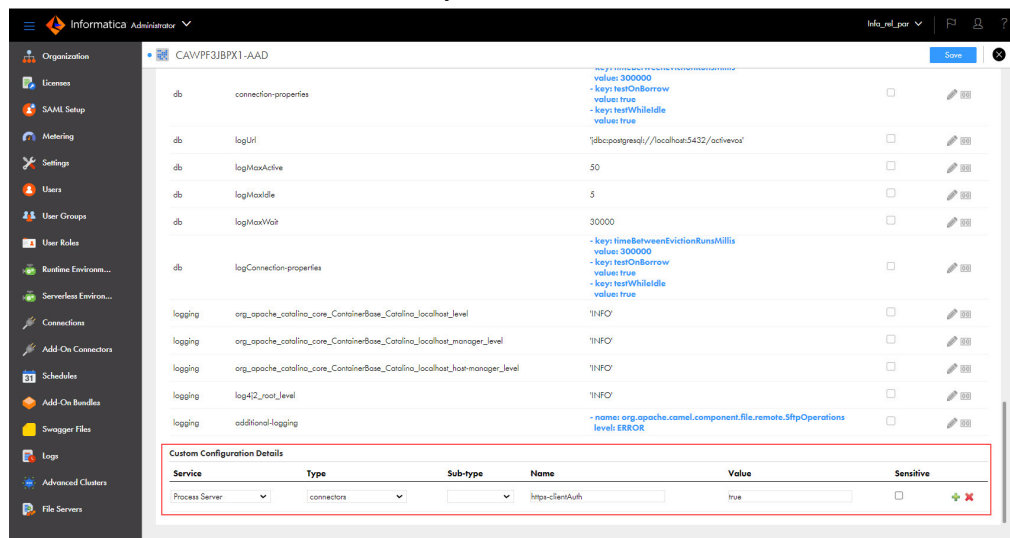
- Administrator で相互認証を有効にします。
- server.xml.mustache ファイルを手動で更新します。

Administrator で相互認証を有効にするには、次の手順を実行します。

1. [Administrator] で、[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、Secure Agent の名前をクリックします。

Secure Agent グループ内の Secure Agent を一覧表示するには、Secure Agent グループの展開が必要になる場合があります。

3. **【詳細】** タブをクリックします。
4. 右上隅の **【編集】** をクリックします。
5. **【カスタム構成の詳細】** 領域までスクロールダウンします。
6. すでに設定されているカスタムプロパティがある場合は、**【追加】** アイコンをクリックして、新しいプロパティ行を追加します。
7. サービスとして **【プロセスサーバー】** を選択します。
8. 設定プロパティのタイプに **【コネクタ】** を選択します。
9. 次の図に示すように、プロパティ名に **https-clientAuth** と入力し、値に **true** と入力します。



10. **【保存】** をクリックします。  
プロセスサーバーサービスのステータスに **【再起動が必要です】** と表示されます。
11. 変更を有効にするためにはプロセスサーバーを再起動する必要があります。  
server.xml.mustache ファイルを手動で更新するには、次の手順を実行します。

1. Secure Agent マシンにログインします。
2. 次のディレクトリに移動します。  
`<Secure Agent installation directory>/downloads/package-process-engine.<latest_version>/package/app/conf/`
3. server.xml.mustache ファイルを編集し、clientAuth プロパティの値を `want` から `true` に変更します。
4. server.xml.mustache ファイルを保存します。
5. 変更を有効にするには、Secure Agent を再起動します。

**注:** デフォルトのキーストアは `ae.keystore` で、localhost 証明書を使用してインストールされます。

プロセスサーバーのキーストアとトラストストアの設定については、ナレッジベースの記事 [611562](#) の添付書類を参照してください。

# スループットを向上させるためのスレッドプールプロファイルの設定

AMQP や Kafka などのイベントベースのコネクタを使用する場合、デフォルトのスレッドプールプロファイルのスレッドプールサイズを増やすことで、スレッド数を増やすことができます。スレッドプールサイズを増やすには、Secure Agent マシンの `aeEngineConfig.xml.mustache` ファイルを更新します。

デフォルトでは、スレッドプールサイズは 10 スレッドに制限されており、最大 10 個のメッセージを接続によって同時に処理できます。Informatica では、スループットを向上させたい場合にのみスレッドプールサイズを増やすことをお勧めします。これは、追加のスレッドが一部のリソースを占有し、プールが大きくなりすぎることが推奨されないためです。スレッドプールサイズは、すべてのイベントベースの接続に適用されます。

また、アプリケーション統合コンソールの **[サーバー設定]** ページで、スレッドプールのプロファイル設定を構成することもできます。**[サーバーの環境設定]** ページと Secure Agent マシンの `aeEngineConfig.xml.mustache` ファイルの両方でスレッドプールプロファイル設定を構成した場合、**[サーバーの環境設定]** ページで設定されたプロパティが優先されます。**[サーバー設定]** ページのスレッドプールプロファイル設定の詳細については、[Server Properties](#) を参照してください。

スレッドプールプロファイルを設定するには、次の手順を実行します。

1. Secure Agent マシンにログインします。
2. 次のディレクトリに移動します。

```
<Secure Agent インストールディレクトリ>/downloads/package-process-engine.<latest_version>/package/app/webapps/process-engine/WEB-INF/classes
```

3. テキストエディタで `aeEngineConfig.xml.mustache` ファイルを開き、次のエントリを検索します。

```
<entry name="IAeESBManager">...</entry>
```

4. IAeESBManager エントリ内で、`camelContext` という名前のサブエントリを検索します。

`cluster.enabled` オプションのステータスに基づき、次のエントリを見つけます。

- `cluster.enabled` オプションが無効になっている場合、次の `camelContext` エントリを編集します。

```
<entry name="Class" value="com.activeee.rt.camel.AeCamelIntegrationManager"/>
<entry name="camelContext">
  <entry name="Class" value="com.activeee.rt.camel.core.AeDefaultCamelContext"/>
</entry>
```

- `cluster.enabled` オプションが有効になっている場合、次の `camelContext` エントリを編集します。

```
<entry name="Class" value="com.activeee.rt.cluster.AeClusterDistributedCamelManager"/>
<entry name="camelContext">
  <entry name="Class" value="com.activevos.socrates.connectors.camel.AeSocratesCamelContext"/>
</entry>
```

5. 新しい `threadpoolProfile` サブエントリを `camelContext` エントリ内に追加します。

`cluster.enabled` オプションが無効になっている場合、次の行を追加します。

```
<entry name="camelContext">
  <entry name="Class" value="com.activeee.rt.camel.core.AeDefaultCamelContext"/>
  <!-- New subentry -->
  <entry name="threadpoolProfile">
    <entry name="PoolSize" value="<PoolSizeValue>"/>
    <entry name="MaxPoolSize" value="<MaxPoolSizeValue>"/>
    <entry name="MaxQueueSize" value="<MaxQueueSizeValue>"/>
    <entry name="KeepAliveTime" value="<KeepAliveTimeValue>"/>
    <entry name="TimeUnit" value="SECONDS"/>
    <entry name="AllowCoreThreadTimeout" value="true"/>
    <entry name="RejectedPolicy" value="CallerRuns"/>
  </entry>
</entry>
```

cluster.enabled オプションが有効になっている場合、次の行を追加します。

```
<entry name="camelContext">
  <entry name="Class" value="com.activevos.socrates.connectors.camel.AeSocratesCamelContext"/>
  <!-- New subentry -->
  <entry name="threadpoolProfile">
    <entry name="PoolSize" value="<PoolSizeValue>"/>
    <entry name="MaxPoolSize" value="<MaxPoolSizeValue>"/>
    <entry name="MaxQueueSize" value="<MaxQueueSizeValue>"/>
    <entry name="KeepAliveTime" value="<KeepAliveTimeValue>"/>
    <entry name="TimeUnit" value="SECONDS"/>
    <entry name="AllowCoreThreadTimeout" value="true"/>
    <entry name="RejectedPolicy" value="CallerRuns"/>
  </entry>
</entry>
```

6. 必要に応じてスレッドプールサイズを増やします。

**注:** プールは、SecureAgent で実行中のすべてのイベントベースの接続によって使用されます。

7. aeEngineConfig.xml.mustache ファイルを保存します。
8. 変更を有効にするには、Secure Agent を再起動します。

## platform.yaml ファイル内のプロパティのオーバーライド

アプリケーションの統合に関連する設定の詳細をプロセスサーバーが取得し、Secure Agent サービスと通信するために使用するプロパティのデフォルト値をオーバーライドできます。

この操作を行うには、platform.yaml ファイルのコピーを作成し、プロパティ値を更新してファイルに user-platform.yaml という名前を付け、次のディレクトリに保存します:

```
<Secure Agent のインストールディレクトリ>/apps/process-engine/conf/
```

platform.yaml ログファイルは、次のディレクトリから取得することができます:

```
<Secure Agent のインストールディレクトリ>\apps\process-engine\<latest_process_engine_version>\conf
```

user-platform.yaml ファイルでプロパティが指定されている場合、プロセスサーバーは稼働中にその特定の値を使用します。プロパティが指定されていない場合、プロセスサーバーは platform.yaml ファイルで定義された値を使用します。

Secure Agent が新しいバージョンにアップグレードされた場合でも、user-platform.yaml ファイルのプロパティは変更されません。

## カスタム user-platform.yaml ファイルの作成

必要なプロパティとそのカスタム値を含むカスタム user-platform.yaml ファイルを作成して、Secure Agent の platform.yaml ファイルのプロパティをオーバーライドできます。

user-platform.yaml ファイルを作成および編集するには、platform.yaml ファイルと同じレベルのアクセス権限が割り当てられている必要があります。

カスタムの user-platform.yaml ファイルを作成するには、次の手順を実行します。

1. Secure Agent マシンにログインします。
2. 次のディレクトリに user-platform.yaml という名前のファイルを作成します。  
<Secure Agent のインストールディレクトリ>/apps/process-engine/conf/

3. カスタマイズするプロパティを、次のディレクトリにある platform.yaml ファイルから user-platform.yaml ファイルにコピーし、必要に応じて値をオーバーライドします。

<Secure Agent のインストールディレクトリ>\apps\process-engine\<latest\_process\_engine\_version>\conf

例えば、cacheExpiryMillis プロパティと lockTimeoutMillis プロパティの値をオーバーライドする必要があります。

これらのプロパティは、次の例に示すように、デフォルトの platform.yaml ファイルの ID セクションで取得することができます。

```
serviceHosts:
  ids:
    context: /identity-service/api/v1/
    agentContext : /identity-service/agent/api/v1/
    cache:
      enabled: true
      maxCacheItems: 2500
      cacheExpiryMillis: 60000
      lockTimeoutMillis: 30000
      refreshAheadEnabled: true
      queueCapacity: 500
      corePoolSize: 1
      maxPoolSize: 5
      keepAliveSec: 120
      timeToRefreshSec: 15
      evictOnLoadMiss: false
    clientPool:
      socketTimeout : 60000
      connectionTimeout : 60000
      poolMaxTotal : 40
      poolMaxPerRoute : 30
    hystrix:
      executionTimeoutInMilliseconds : 60000
      fallbackEnabled: false
```

次のサンプルに示すように、これらのプロパティを platform.yaml ファイルからコピーし、user-platform.yaml ファイルの値を更新します。

```
serviceHosts:
  ids:
    context: /identity-service/api/v1/
    agentContext : /identity-service/agent/api/v1/
    cache:
      cacheExpiryMillis: 200000
      lockTimeoutMillis: 150000
```

**注:** デフォルトの platform.yaml ファイルと同じ階層および構文に従っていることを確認する必要があります。

4. user-platform.yaml ファイルを保存します。
5. 変更を有効にするためにはプロセスサーバーを再起動する必要があります。

プロセスサーバーの稼働中に、user-platform.yaml ファイルで指定されたプロパティによって platform.yaml ファイルのプロパティがオーバーライドされます。

## トラブルシューティング

プロパティをオーバーライドするときに、platform.yaml ファイルと比較して user-platform.yaml ファイルに正しくないプロパティまたは値が含まれていると、予期しない動作が発生する可能性があります。

例えば、手動による操作が原因で予期した結果が得られなくなる次のようなシナリオについて考えてみます。

- IDS-SESSION-ID を取得するために agentContext プロパティが使用されている場合。platform.yaml ファイルの agentContext プロパティの値が /identity-service/agent/api/v1/ で、ユーザーが user-platform.yaml ファイルに /identity-service/agent/api/v2/ という値を入力した場合、エンドポイントが無効であるため、プロセスサーバーでは IDS-SESSION-ID の取得が失敗します。そのため、Secure Agent は基本認証メカニズムを使用します。
- cacheExpiryMillis プロパティ値が数値である必要がある場合。ただし、英数字の値を入力すると、エラーが発生することなく値は文字列として扱われます。

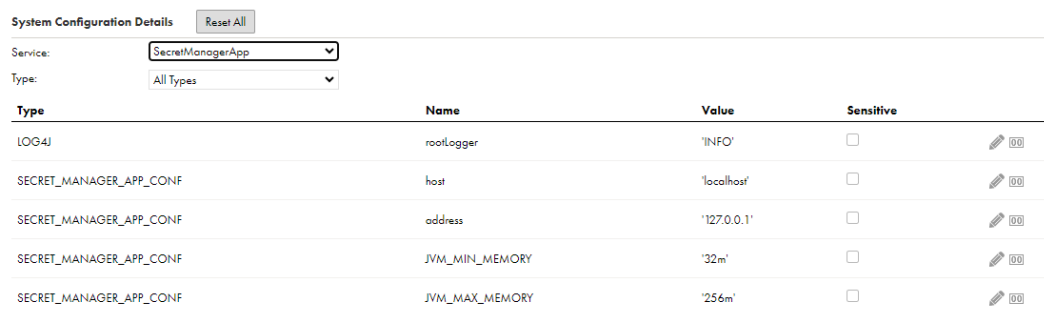
## 第 15 章











# SecretManagerApp

SecretManagerApp サービスは、組織が AWS Secrets Manager や Azure Key Vault などの外部シークレットマネージャを使用している場合に、Informatica Intelligent Cloud Services とシークレットマネージャ間の通信を管理する Secure Agent サービスです。

SecretManagerApp サービスの動作を変更または最適化する場合は、Secure Agent の編集時に [システム構成の詳細] セクションでプロパティを設定します。

次の図に、SecretManagerApp サービスプロパティの一部を示します。



Type	Name	Value	Sensitive	
LOG4J	rootLogger	'INFO'	<input type="checkbox"/>	 
SECRET_MANAGER_APP_CONF	host	'localhost'	<input type="checkbox"/>	 
SECRET_MANAGER_APP_CONF	address	'127.0.0.1'	<input type="checkbox"/>	 
SECRET_MANAGER_APP_CONF	JVM_MIN_MEMORY	'32m'	<input type="checkbox"/>	 
SECRET_MANAGER_APP_CONF	JVM_MAX_MEMORY	'256m'	<input type="checkbox"/>	 

次の SecretManagerApp サービスのプロパティを設定できます。

タイプ	名前	説明
SECRET_MANAGER_APP_CONF	JVM_MIN_MEMORY	サービスの開始時に SecretManagerApp サービスに割り当てられるメモリの量。 デフォルトは 32 MB です。
SECRET_MANAGER_APP_CONF	JVM_MAX_MEMORY	SecretManagerApp サービスに割り当てられる最大メモリ。 デフォルトは 256 MB です。

**注:** Informatica グローバルカスタマサポートから指示された場合を除き、他の SecretManagerApp サービスの値は変更しないでください。

## 第 16 章

# Secure Agent サービスプロパティの設定

Secure Agent サービスプロパティを設定するには、**【ランタイム環境】** ページを開いて Secure Agent を編集します。Secure Agent サービスのプロパティ値を変更、マスク、およびリセットできます。サービスのカスタムプロパティを追加および削除できます。また、Secure Agent 名を変更することもできます。

カスタムプロパティはコネクタ固有です。カスタムプロパティの詳細については、該当するコネクタのヘルプを参照してください。

**警告:** グループレベルのプロパティ設定を使用する Secure Agent グループのエージェントに対して、エージェントレベルの Secure Agent サービスプロパティ設定を構成しないでください。エージェントレベルのプロパティ設定を構成する場合は、エージェントプロパティを構成する前に、グループレベルのプロパティ設定を削除してください。グループレベルのプロパティ設定の詳細については、*REST API リファレンス*のランタイム環境に関する説明を参照してください。

1. **【ランタイム環境】** ページで、Secure Agent の名前をクリックします。  
Secure Agent グループ内の Secure Agent を一覧表示するには、Secure Agent グループの展開が必要になる場合があります。
2. **【詳細】** タブをクリックします。
3. 右上隅の **【編集】** をクリックします。
4. Secure Agent の名前を変更するには、**【エージェント名】** フィールドに新しい名前を入力します。
5. サービスプロパティを編集するには、次の手順を実行します。
  - a. **【システム構成の詳細】** 領域で、サービスを選択します。
  - b. 設定プロパティの種類を選択します。
  - c. 編集するプロパティを含む行で、**【エージェント設定の編集】** アイコンをクリックします。
  - d. プロパティ値を変更するには、新しいプロパティ値を入力します。  
プロパティが機密プロパティである場合、プロパティを編集すると既存の値がクリアされます。
  - e. プロパティに機密データが含まれており、Secure Agent の詳細ページで値をマスクする場合は、**【機密】** オプションを有効にします。  
機密オプションを有効にすると、入力した値がマスクされます。フィールドが複数行のテキストフィールドである場合、変更を保存した後に値がマスクされます。
  - f. プロパティをシステムデフォルト値にリセットするには、**【エージェント設定をシステムデフォルトにリセット】** アイコンをクリックします。



6. サービスのカスタムプロパティを追加するには、次の手順を実行します。

a. **【カスタム構成の詳細】** 領域までスクロールダウンします。

次の画像は、**【カスタム構成の詳細】** 領域を示しています。

Custom Configuration Details					
Service	Type	Sub-type	Name	Value	Sensitive
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>  

b. すでに設定されているカスタムプロパティがある場合、**【追加】** アイコンをクリックして、新しいプロパティ行を追加します。

c. 設定するサービスを選択します。

d. 設定プロパティの種類を選択します。

e. 設定プロパティタイプにサブタイプがある場合は、適切なサブタイプを選択します。

例えば、ログレベルを決定するには、サブタイプとして **【情報】** または **【デバッグ】** を選択します。

f. プロパティの名前と値を入力します。

g. プロパティに機密データが含まれており、Secure Agent の詳細ページで値をマスクする場合は、**【機密】** オプションを有効にします。

7. カスタムプロパティを削除するには、カスタムプロパティの隣にある **【削除】** アイコンをクリックします。

8. すべての設定プロパティをデフォルト設定にリセットするには、**【すべてリセット】** をクリックします。

9. **【保存】** をクリックします。

# 索引

## C

Cloud アプリケーション統合コミュニティ  
URL [6](#)  
Cloud 開発者コミュニティ  
URL [6](#)

## G

GitRepoConnectApp  
ローカルリポジトリディレクトリ [39](#)  
概要 [39](#)

## I

Informatica Intelligent Cloud Services  
Web サイト [6](#)  
Informatica グローバルカスタマサポート  
連絡先情報 [7](#)

## N

NetworkRetryInterval  
データ統合サーバーのプロパティ [26](#)  
NetworkTimeoutPeriod  
データ統合サーバーのプロパティ [26](#)

## S

SecretManagerApp  
概要 [95](#)  
Secure Agent  
サービスプロパティの設定 [96](#)  
GitRepoConnectApp サービスの概要 [39](#)  
GitRepoConnectApp の設定プロパティ [40](#)  
IDMC Data Gateway Service [42](#)  
IDMC Data Gateway Service のプロパティ [42](#)  
SecretManagerApp サービスの概要 [95](#)  
エージェント名の変更 [96](#)  
エラスティックサーバーサービスの概要 [34](#)  
エラスティックサーバーの構成プロパティ [34](#)  
カスタム構成のプロパティ [96](#)  
サービスの概要 [8](#)  
データ統合サーバーサービスの概要 [26](#)  
データ統合サーバーの設定プロパティ [27](#)  
ネットワーク中断設定 [26](#)  
ファイル取り込みの設定プロパティ [48](#)  
マスキング設定のプロパティ [96](#)  
メタデータプラットフォームサービス [58](#)  
メタデータプラットフォームサービスのプロパティ [58](#)  
メタデータ基盤アプリケーション [52](#)  
メタデータ基盤アプリケーションのプロパティ [52](#)

Secure Agent (続く)

共通統合コンポーネントプロパティ [18](#)  
Secure Agent サービス  
CMI ストリーミングエージェント [15](#)  
データベース取り込みエージェントの環境変数 [25](#)  
データベース取り込みサービスのプロパティ [21](#)

## W

Web サイト [6](#)

## あ

アップグレード通知 [7](#)

## え

エラスティックサーバー  
概要 [34](#)

## か

カスタム構成のプロパティ  
Secure Agent [96](#)

## し

システムステータス [7](#)

## す

ステータス  
Informatica Intelligent Cloud Services [7](#)  
ストリーミング取り込み  
Secure Agent [15](#)

## そ

ソース管理  
ローカルリポジトリディレクトリの設定 [39](#)

## て

データ統合サーバー  
概要 [26](#)

ふ

ファイル統合サービス [38](#)

め

メンテナンスの停止 [7](#)