



Informatica® Multidomain MDM  
10.5 HotFix 4

**、、、バージョン 10.1、  
10.2、10.3、10.4、または  
10.5 からのアップグレー  
ド**

Informatica Multidomain MDM、、、バージョン 10.1、10.2、10.3、10.4、または 10.5 からのアップグレード

10.5 HotFix 4

2025 年 5 月

© 著作権 Informatica LLC 1998, 2025

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

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

Informatica、Informatica ロゴ、および ActiveVOS 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: 2025-06-12

# 目次

<b>序文</b> .....	10
Informatica のリソース.....	10
Informatica Network.....	10
Informatica ナレッジベース.....	10
Informatica マニュアル.....	10
Informatica 製品可用性マトリックス.....	11
Informatica Velocity.....	11
Informatica Marketplace.....	11
Informatica グローバルカスタマサポート.....	11
 <b>第 1 章 : アップグレードの概要</b> .....	12
概要.....	12
アップグレードプロセスの選択.....	13
アップグレードプロセスの選択についてのガイドライン.....	13
クリーンアップグレードのアップグレードプロセス.....	13
インプレースアップグレードのアップグレードプロセス.....	14
アップグレードの記録.....	15
 <b>第 2 章 : アップグレード前のタスク</b> .....	16
アップグレード前のタスク.....	16
Java Development Kit のインストール.....	16
JBoss EAP のバージョンのアップグレード.....	16
アップグレードの準備.....	17
環境の準備.....	17
既存のジョブの処理.....	22
MDM Hub 構成の更新.....	22
元のスキーマ所有者によるオペレーショナルリファレンスストアの登録.....	23
MDM アプリケーションサーバーのプロパティファイルの更新.....	26
BPM アップグレードの準備.....	26
ワークフローアダプタの選択.....	27
スタンドアロン ActiveVOS から埋め込み ActiveVOS への移行（インプレースアップグレード）.....	28
ビルドプロパティファイルの編集（インプレースアップグレード）.....	29
タイムラインバージョン抽出テーブルの設定.....	29
 <b>第 3 章 : データベースタスク</b> .....	34
データベースタスクの概要.....	34
Oracle データベースのセットアップ.....	34
手順 1. Oracle のインストールと設定.....	35
手順 2. データベースとテーブルスペースの作成.....	35

手順 3. データベース特権と接続の設定. . . . .	38
手順 4. ActiveVOS スキーマの作成. . . . .	38
Microsoft SQL Server のセットアップ. . . . .	40
手順 1. Microsoft SQL Server のインストールと設定. . . . .	40
手順 2. Microsoft SQL Server データファイルストアの作成. . . . .	41
手順 3. ODBC ドライバのインストール. . . . .	41
手順 4. ActiveVOS データベースの作成. . . . .	42
IBM Db2 データベースのセットアップ. . . . .	43
手順 1. IBM Db2 のインストールと設定. . . . .	43
手順 2. データベースとテーブルスペースの作成. . . . .	43
データベースとテーブルスペースの手動作成. . . . .	44
スクリプトを使用してデータベースとテーブルスペースを作成. . . . .	46
手順 3. データベースサーバーへのパッケージのバインド. . . . .	47
手順 4. ActiveVOS スキーマの作成. . . . .	48
<b>第 4 章 : アプリケーションサーバーのタスク. . . . .</b>	<b>49</b>
アプリケーションサーバーのタスクの概要. . . . .	49
JBoss のセットアップ. . . . .	49
Java 仮想マシンの設定. . . . .	50
完全なプロファイルのサーバープロパティの設定. . . . .	54
ActiveVOS コンソール管理ユーザーの作成. . . . .	56
タイムアウトプロパティの設定. . . . .	57
ActiveVOS を使用した最新の JBoss バージョンのサポートの設定. . . . .	57
JBoss の起動. . . . .	59
Oracle WebLogic のセットアップ. . . . .	60
Java 仮想マシンの設定. . . . .	61
WebLogic Server 認証の無効化. . . . .	65
ActiveVOS コンソール管理ユーザーの作成. . . . .	65
追加の Oracle WebLogic の設定. . . . .	66
スタンドアロンプロセスサーバーインスタンス用の WebLogic の設定. . . . .	66
複数の MDM Hub マスタデータベース用の WebLogic の設定. . . . .	69
HTTPS プロトコルの設定. . . . .	69
IBM WebSphere のセットアップ. . . . .	70
Java 仮想マシンの設定. . . . .	71
MDM Hub 環境でのパスワードの暗号化. . . . .	74
WebSphere 環境での安全なプロファイルの作成. . . . .	74
ActiveVOS コンソール管理ユーザーの作成. . . . .	75
MDM Hub デプロイメント用の SOAP 要求タイムアウトの設定. . . . .	76
追加の IBM WebSphere の設定. . . . .	77
スタンドアロンプロセスサーバーインスタンス用の WebSphere の設定. . . . .	77
複数の MDM Hub マスタデータベース用の WebSphere の設定. . . . .	80
HTTPS プロトコルの設定. . . . .	81
Informatica Data Director 用の WebSphere の設定. . . . .	81

<b>第 5 章 : Hub ストアのアップグレード</b>	<b>82</b>
Hub ストアのアップグレードの概要	82
Hub ストアのクローン作成 (クリーンアップグレード)	82
データベースを英語以外のロケールに設定	83
Verbose モードでの MDM Hub マスターデータベースのアップグレード	83
サイレントモードでの MDM Hub マスターデータベースのアップグレード	84
Verbose モードでのオペレーショナルリファレンスストアのデータベースのアップグレード	85
サイレントモードでのオペレーショナル参照ストアデータベースのアップグレード	88
ActiveVOS データベースの更新	89
アップグレードスクリプトが正常に実行されたことの確認	90
 <b>第 6 章 : Hub サーバーのアップグレード (インプレースアップグレード)</b>	 <b>91</b>
Hub サーバーのアップグレードの概要	91
グラフィカルモードでの Hub サーバーのアップグレード	92
コンソールモードでの Hub サーバーのアップグレード	95
サイレントモードでの Hub サーバーのアップグレード	98
プロパティファイルの設定	98
サイレントアップグレードの実行	99
patchInstallSetup スクリプトの実行	100
アップグレードドキュメントフォルダへの Hub サーバーのログファイルのコピー	101
Hub Server のアップグレードの再適用 (オプション)	101
 <b>第 7 章 : プロセスサーバーのアップグレード (インプレースアップグレード)</b>	 <b>103</b>
Process サーバーアップグレードの概要	103
グラフィカルモードでのプロセスサーバーのアップグレード	104
コンソールモードでのプロセスサーバーのアップグレード	106
サイレントモードでのプロセスサーバーのアップグレード	107
プロパティファイルの設定	107
プロセスサーバーのサイレントアップグレードの実行	108
Informatica Address Verification 5 統合へのアップグレード手順	109
一致ポピュレーションの設定	111
一致ポピュレーションの有効化	111
アップグレードドキュメントディレクトリへのプロセスサーバーのログファイルのコピー	113
プロセスサーバーのアップグレードの再適用 (オプション)	113
 <b>第 8 章 : アップグレード後のタスク</b>	 <b>115</b>
アップグレード後のタスク	115
Microsoft SQL Server 2017 用 JDBC ドライバの設定	116
更新プロパティ	116
JBoss のアップグレード後の作業	117

インブレースアップグレードのアップグレード後のタスクを実行する. . . . .	117
オブジェクト、カラム、および廃止されたオブジェクトへの参照の削除. . . . .	118
Hub サーバーをデプロイするための PostInstall スクリプトの実行（条件による）. . . . .	118
Hub コンソールクライアントの設定. . . . .	119
WebSphere 管理セキュリティの設定. . . . .	120
オペレーショナルリファレンスストアの登録解除. . . . .	120
EAR ファイルのアンインストールおよびデータソースの削除. . . . .	120
WebSphere 管理コンソールでの WebSphere 管理セキュリティの有効化. . . . .	121
Hub サーバーとプロセスサーバーのプロパティの設定. . . . .	121
Hub サーバーの PostInstallSetup スクリプトの手動実行. . . . .	122
プロセスサーバーの PostInstallSetup スクリプトの実行. . . . .	122
オペレーショナルリファレンスストアの登録. . . . .	122
クラスローダーの設定 WebSphere で. . . . .	127
オペレーショナルリファレンスストアの登録. . . . .	128
アップグレードされたメタデータの検証. . . . .	132
メタデータの検証. . . . .	132
検証結果の保存. . . . .	133
メタデータ検証メッセージの解決. . . . .	133
ローカライズされたスキーマの更新. . . . .	134
コンテンツセキュリティポリシーのカスタマイズ. . . . .	134
クリーンアップグレードのアップグレード後のタスクを実行する. . . . .	135
スキーマのパスワードの暗号化. . . . .	135
スキーマのパスワードの更新. . . . .	135
オペレーショナル参照ストア接続のテストおよび更新. . . . .	136
ActiveVOS 接続のテストおよび更新. . . . .	136
プロセスサーバーのテストおよび追加. . . . .	136
プラットフォームトランスフォーメーションに対するクレンジング関数の設定. . . . .	137
MDM Hub 環境レポートの確認. . . . .	138
MDM Hub 環境レポートの保存. . . . .	138
外部呼び出しとアプリケーションのアップグレード. . . . .	138
EJB プロトコル用の SiperianClient ライブラリクラスのアップグレード. . . . .	140
MDM Hub メタデータを準備する. . . . .	140
アップグレードテスト. . . . .	141
MDM Hub アップグレードテスト. . . . .	141
カスタムコードのアップグレードのテスト. . . . .	141
プロビジョニングツールアップグレードテスト. . . . .	141
ビジネスエンティティを使用する Data Director のアップグレードテスト. . . . .	142
サブジェクト領域を使用する Data Director のアップグレードテスト. . . . .	142
Hub サーバーの全般的なプロパティの設定. . . . .	143
Data Director および Hub サーバーのプロパティ. . . . .	143
Data Director のグローバルプロパティ. . . . .	144
ビジネスエンティティスキーマの生成. . . . .	144

<b>第 9 章 : 検索設定のアップグレード</b>	<b>145</b>
検索設定のアップグレードの概要	145
手順 1. Elasticsearch のインストールとセットアップ	145
インストール前のタスクの完了	146
Elasticsearch のインストール	147
Elasticsearch Java 仮想マシン (JVM) の設定	147
Elasticsearch プロパティファイルの設定	148
Elasticsearch の保護 (オプション)	149
分析プラグインのインストール	149
ストップワード、シノニム、文字マッピングの設定	149
Elasticsearch の起動	150
手順 2. 検索のための MDM Hub プロパティの設定	150
Hub サーバーでの検索の設定	150
プロセスサーバーでの検索の設定	151
手順 3. プロビジョニングツールを使用した検索の設定	152
Elasticsearch クラスタの設定	152
検索またはクエリ結果の表示の設定	153
手順 4. オペレーショナル参照ストアの検証	154
手順 5. 検索データのインデックス処理	155
Elasticsearch バージョン 7.17.0 へのアップグレード (オプション)	155
Elasticsearch バージョン 7.17.0 へのアップグレードの前提条件	156
Elasticsearch プロパティファイルの設定	156
Elasticsearch インデックスのアップグレード	157
 <b>第 10 章 : 階層のアップグレード</b>	 <b>160</b>
階層のアップグレードの概要	160
階層リレーションおよびネットワークリレーションを理解する	161
階層関係	162
ネットワークリレーション	162
階層マネージャリレーションのコピーおよび階層リレーションの作成	163
階層マネージャのリレーションのコピーおよび階層の作成	164
階層用の Hub サーバーの設定	165
リレーションベースオブジェクトをベースオブジェクトに戻す	165
階層へのアクセスの設定	165
階層マネージャリレーションのコピーおよびネットワークリレーションの作成	166
階層マネージャでリレーションをコピーすることによるネットワークリレーションの追加	166
リレーションベースオブジェクトをベースオブジェクトに戻す	167
 <b>第 11 章 : アプリケーションサーバーに対する ActiveVOS のインストール後のタスク</b>	 <b>168</b>
アプリケーションサーバーに対する ActiveVOS のインストール後のタスク	168

<b>第 12 章: ビジネスエンティティアダプタ用の ActiveVOS アップグレード後タスク</b> .....	169
ビジネスエンティティアダプタに対する ActiveVOS のアップグレード後のタスク.....	169
ビジネスエンティティワークフローアダプタの ActiveVOS URN の設定.....	170
ActiveVOS プロトコルの HTTPS への設定.....	170
ビジネスエンティティ用にカスタマイズしたワークフローの更新.....	171
ビジネスエンティティ用のワークフロー内のプレゼンテーションパラメータの更新.....	171
ビジネスエンティティのためのワークフロー内でのファイル添付の有効化.....	173
ActiveVOS 用の MDM Identity Service の設定.....	174
カスタム BeMDMWorkflow プロジェクト（インプレースアップグレード）.....	175
マージ解除ワークフローおよびマージワークフローのトリガの設定（インプレースアップグレード）.....	175
エンティティ 360 フレームワークタスクマネージャの追加.....	176
<b>第 13 章: サブジェクト領域アダプタ用の ActiveVOS アップグレード後タスク</b> .....	177
サブジェクト領域アダプタに対する ActiveVOS のアップグレード後のタスク.....	177
ActiveVOS URN の更新.....	178
ActiveVOS の信頼されたユーザーの確認.....	178
サブジェクト領域に基づく ActiveVOS ワークフローの Informatica Data Director タスク設定の更新.....	179
サブジェクト領域ベースの ActiveVOS アダプタの IDD 設定の更新.....	179
サブジェクト領域ワークフローアダプタのタスクトリガの設定.....	180
サブジェクト領域用にカスタマイズしたワークフローの更新.....	181
サブジェクト領域用のワークフロー内のプレゼンテーションパラメータの更新.....	181
サブジェクト領域のためのワークフロー内での添付の有効化.....	183
サブジェクト領域に基づく ActiveVOS ワークフローの再デプロイ.....	183
ビジネスエンティティおよびビジネスエンティティサービス構成ファイルの生成.....	184
<b>付録 A: アップグレードプロセスのトラブルシューティング</b> .....	185
<b>付録 B: FAQ（よくある質問）</b> .....	192
<b>付録 C: 既存の ActiveVOS タスクの処理</b> .....	194
既存の ActiveVOS タスクの処理の概要.....	194
移行プロパティ.....	194
移行スクリプトの実行プロパティファイルを指定.....	195
移行スクリプトの実行プロパティをコマンドラインに指定.....	196
<b>付録 D: メタデータキャッシュの設定</b> .....	197
メタデータキャッシュの構成（オプション）.....	197
Infinispan 属性.....	198
Infinispan 属性の編集.....	199

索引.....	200
---------	-----

# 序文

Multidomain MDM 実装を最新バージョンにアップグレードするには、Informatica<sup>(R)</sup> *Multidomain MDM* のアップグレードガイドの指示に従います。アップグレードの際は、必ず現在インストール済みのバージョンに適用される *Multidomain MDM* のアップグレードガイドを使用してください。

## 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 Network

Informatica Network は、Informatica ナレッジベースや Informatica グローバルカスタマサポートなど、多くのリソースへの入口です。Informatica Network を利用するには、<https://network.informatica.com> にアクセスしてください。

Informatica Network メンバーは、次のオプションを利用できます。

- ナレッジベースで製品リソースを検索できます。
- 製品の提供情報を表示できます。
- サポートケースを作成して確認できます。
- 最寄りの Informatica ユーザーグループネットワークを検索して、他のユーザーと共同作業を行えます。

### 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 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 製品可用性マトリックス

製品可用性マトリックス (PAM) には、製品リリースでサポートされるオペレーティングシステム、データベースなどのデータソースおよびターゲットが示されています。Informatica PAM は、<https://network.informatica.com/community/informatica-network/product-availability-matrices> で参照できます。

## Informatica Velocity

Informatica Velocity は、Informatica プロフェッショナルサービスが開発したヒントとベストプラクティスのコレクションで、多数のデータ管理プロジェクトから得た実体験に基づいています。Informatica Velocity には、世界中の組織と連携してデータ管理ソリューションを計画、開発、デプロイ、管理する Informatica コンサルタントによる集合知を表しています。

Informatica Velocity リソースには、<http://velocity.informatica.com> からアクセスしてください。Informatica Velocity についての質問、コメント、またはアイデアがある場合は、[ips@informatica.com](mailto:ips@informatica.com) から Informatica プロフェッショナルサービスにお問い合わせください。

## Informatica Marketplace

Informatica Marketplace は、お使いの Informatica 製品を拡張したり強化したりするソリューションを検索できるフォーラムです。Marketplace で、Informatica デベロッパーやパートナーからの多数のソリューションを活用すれば、生産性を向上したり、プロジェクトでの実装時間を短縮したりできます。Informatica Marketplace は、<https://marketplace.informatica.com> からアクセスしてください。

## Informatica グローバルカスタマサポート

電話または Informatica Network を介してグローバルカスタマサポートに連絡できます。

各地域の Informatica グローバルカスタマサポートの電話番号は、Informatica Web サイト (<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>) を参照してください。

Informatica Network のオンラインサポートリソースを見つけるには、<https://network.informatica.com> にアクセスして eSupport オプションを選択します。

# 第 1 章

## アップグレードの概要

この章では、以下の項目について説明します。

- [概要, 12 ページ](#)
- [アップグレードプロセスの選択, 13 ページ](#)
- [アップグレードの記録, 15 ページ](#)

### 概要

Informatica Multidomain MDM をアップグレードいただきありがとうございます。インストールされているバージョンからこのバージョンの Multidomain MDM に直接アップグレードできます。

**重要:** すべての MDM コンポーネントは、同じバージョンの Multidomain MDM にアップグレードする必要があります。

Multidomain MDM 環境には、開発環境、テスト環境、運用環境を含められます。これらの各環境をアップグレードする必要があります。ベストプラクティスとして、開発環境を最初にアップグレードします。アップグレードに関する問題を特定し、解決します。開発環境を正常にアップグレードしたら、テスト環境と運用環境を高度な信頼性でアップグレードできます。

#### はじめる前に

開始する前に、必要なガイドがあることを確認し、製品の可用性マトリックスを確認してください。

- インストールされている Multidomain MDM のバージョンの正しいアップグレードガイドがあることを確認します。タイトルページで、タイトルの「バージョンからのアップグレード」にあるメジャーバージョン番号を確認します。それがインストールされている製品のメジャーバージョン番号と一致していることを確認します。インストールされている製品のバージョン番号を確認するには、MDM Hub コンソールで、**[ヘルプ]** > **[バージョン情報]** をクリックします。**[インストールの詳細]** をクリックします。バージョン番号がリリース名のカラムに表示されます。アップグレード目的では、インストールされている製品に Hotfix が適用されているかどうかは問題になりません。
- ご使用のデータベースおよびアプリケーションサーバーに対応する *Multidomain MDM* のインストールガイドをダウンロードします。スムーズなアップグレードを実現するには、インストールガイドに記載されているいくつかのタスクを実行する必要があります。
- このバージョンの Multidomain MDM のシステム要件への変更点については、「製品可用性マトリックス」を確認してください。このマトリックスは、Informatica Network で見つけることができます。

<https://network.informatica.com/community/informatica-network/product-availability-matrices>。

# アップグレードプロセスの選択

Multidomain MDM のアップグレードには、次のオプションがあります。

## クリーンアップグレード

クリーンアップグレードでは、現在の MDM システム要件を満たすマシンで新しい環境を準備します。既存の環境から MDM Hub マスターデータベースとオペレーショナルリファレンスストアのクローンを作成し、クローンのデータベースを新しい環境にコピーし、Hub ストアアップグレードを実行します。『インストールガイド』の指示に従って、Hub サーバーとプロセスサーバーを新しい環境にインストールします。インフラストラクチャの計画の詳細については、*Multidomain MDM のインフラストラクチャプランニングガイド*を参照してください。

## インプレースアップグレード

インプレースアップグレードでは、現在の MDM システム要件に準拠するように、Multidomain MDM を実行するマシンを更新します。その後 MDM コンポーネントをアップグレードします。

## アップグレードプロセスの選択についてのガイドライン

アップグレードのタイプによって、選択できるアップグレードプロセスのタイプが決定します。

次の表に、アップグレードのタイプと、使用できるアップグレードプロセスを記載します。

アップグレードのタイプ Multidomain MDM	クリーンアップグレード	インプレースアップグレード
MDM の任意のバージョンへのアップグレードで、MDM システム要件を満たすようにアプリケーションサーバーをメジャーバージョンに更新する必要がある場合	○	×
MDM の任意のバージョンへのアップグレードで、MDM システム要件を満たすようにアプリケーションサーバーをメジャーバージョンに更新する必要がある場合	○	はい

## クリーンアップグレードのアップグレードプロセス

クリーンアップグレードのプロセスは、次のフェーズで構成されています。

フェーズ	タスク	手順の場所
1	データベースタスク	アップグレードガイド
2	アプリケーションサーバーのタスク	アップグレードガイド
3	アップグレード前のタスク	アップグレードガイド
4	Hub ストアのバックアップとクローンを作成し、Hub ストアを新しいマシンにコピーする	このタスクの実行は DBA に依頼
5	Hub ストアのアップグレード	アップグレードガイド
6	Hub サーバーのインストールとインストール後のタスク	インストールガイド

フェーズ	タスク	手順の場所
7	プロセスサーバーのインストールとインストール後のタスク	インストールガイド
8	アップグレード後のタスク	アップグレードガイド
9	アプリケーションサーバーに対する ActiveVOS のインストール後のタスク	インストールガイド
10	ビジネスエンティティアダプタに対する ActiveVOS のアップグレード後のタスク	アップグレードガイド
11	サブジェクト領域アダプタに対する ActiveVOS のアップグレード後のタスク	アップグレードガイド
12	リソースキットのインストールタスクと、必要に応じて、リソースキットのインストール後のタスク	インストールガイド

## インプレースアップグレードのアップグレードプロセス

インプレースアップグレードのプロセスは、次のフェーズで構成されています。

フェーズ	タスク	手順の場所
1	データベースタスク	アップグレードガイド
2	アプリケーションサーバーのタスク	アップグレードガイド
3	アップグレード前のタスク	アップグレードガイド
4	Hub ストアのアップグレード	アップグレードガイド
5	Hub サーバーのアップグレード	アップグレードガイド
6	プロセスサーバーのアップグレード	アップグレードガイド
7	リソースキットのアップグレード	アップグレードガイド
8	アップグレード後のタスク	アップグレードガイド
9	アプリケーションサーバーに対する ActiveVOS のインストール後のタスク	インストールガイド
10	ビジネスエンティティアダプタに対する ActiveVOS のインストール後のタスク	アップグレードガイド
11	サブジェクト領域アダプタに対する ActiveVOS のインストール後のタスク	アップグレードガイド

\* インストールされたバージョンと現在のバージョンとの間に複数のリリースがある場合、ベストプラクティスは、クリーンアップグレードを実行する、またはすべてのインストール前タスクを見直して、使用するデータベースまたはアプリケーションサーバーに対して発生した変更を見逃していないと確認することです。

# アップグレードの記録

アップグレードする前、アップロード中およびアップロード後に、Informatica MDM Hub 環境の詳細をキャプチャして、アップグレードを確認（および必要に応じてトラブルシューティング）する必要があります。アップグレードのトラブルシューティングに対するサポートが必要な場合、この情報のコピーを Informatica グローバルカスタマサポートに提供できます。

以下の表に、アップグレードに関する情報の参照先を示します。

アップグレード情報	確認できる場所
オペレーショナルリファレンスストア（ORS）メタデータ	Hub コンソールのリポジトリマネージャツールによる検証結果 アップグレード前とアップグレード後に ORS メタデータの検証結果を生成します。
構成設定	Hub コンソールのエンタープライズマネージャツールによる環境レポート 環境レポートには、Hub サーバー、Process サーバー、マスターデータベースおよび ORS データベースの情報が記録されています。アップグレード前とアップグレード後に環境レポートを保存します。
Hub サーバーおよび Process サーバーのアップグレードのアップグレードイベント	Informatica MDM Hub インストーラのログファイル
Hub サーバーと Process サーバーの情報	Hub サーバーと Process サーバーのログファイル
Hub コンソール情報	Hub コンソールのログファイル

## 関連項目：

- [「MDM Hub 環境レポートの保存」](#)（ページ 138）

## 第 2 章

# アップグレード前のタスク

この章では、以下の項目について説明します。

- [アップグレード前のタスク, 16 ページ](#)
- [アップグレードの準備, 17 ページ](#)
- [環境の準備, 17 ページ](#)
- [既存のジョブの処理, 22 ページ](#)
- [MDM Hub 構成の更新, 22 ページ](#)
- [MDM アプリケーションサーバーのプロパティファイルの更新, 26 ページ](#)
- [BPM アップグレードの準備, 26 ページ](#)
- [タイムラインバージョン抽出テーブルの設定, 29 ページ](#)

## アップグレード前のタスク

クリーンアップグレードまたはインプレースアップグレードを実行するかどうかにかかわらず、環境が適切に構成されていることを確認するため、アップグレード前のタスクを実行します。

### Java Development Kit のインストール

Oracle Java Runtime Environment (JRE) を使用する場合は、最新バージョンの MDM をインストールするマシンに Zulu JDK 17.0.10 または 17.0.11 をインストールします。

### JBoss EAP のバージョンのアップグレード

MDM を最新バージョンにアップグレードする前に、JBoss EAP をバージョン 8.0.0 に更新するかインストールしてください。

JBoss EAP 8.0.0 を Red Hat の公式ダウンロードサイトからダウンロードします。

# アップグレードの準備

アップグレードを準備するには、次のタスクを実行します。

アップグレードタスク	詳細
リリースノートの確認	リリースノートには、インストールおよびアップグレードプロセスの更新に関する情報が含まれています。 <b>重要:</b> アプリケーションサーバーとデータベースの一部のバージョンには、Multidomain MDM の実行時の既知の制限があります。推奨されたすべての回避策を必ず実行してください。
リリースガイドの確認	リリースガイドには、新しい機能と変更された動作に関する情報が含まれています。
最新のライセンスファイルの入手	Multidomain MDM のアップグレードソフトウェアを要求するときに、最新のライセンスファイルを要求してください。
アップグレード制限の確認	アップグレードを開始する前に、次の条件を確認します。 <ul style="list-style-type: none"><li>- Multidomain MDM 実装では、すべてのコンポーネントが同じバージョンである必要があります。複数のバージョンの Multidomain MDM がある場合は、各バージョンを別々の環境にインストールする必要があります。同じ環境に異なるバージョンの Multidomain MDM を共存させることはできません。</li><li>- Multidomain MDM のアップグレード中に、環境に大きな変更を加えることはできません。例えば、アップロードプロセス中に Oracle、IBM Db2、または Microsoft SQL Server データベースをアップグレードすることはできません。</li><li>- 付属しているアップグレードスクリプトを使用して、オペレーショナルリファレンスストア（ORS）をアップグレードしておく必要がある。アップグレードツールとしてリポジトリマネージャは使用することを想定されていない。これは、バージョン間で、アーティファクトが一部継承されない場合があったり、継承されてもアーティファクトの内容が不正確な場合があったりするためである。詳細については、『<i>Multidomain MDM のリリースノート</i>』を参照してください。</li></ul>
アップグレードドキュメントフォルダの作成	upgradedoc というフォルダを作成し、メタデータの検証結果、環境レポート、およびログファイルなど、アップグレードの全アーティファクトのコピーを保存します。アップグレード中に問題が発生した場合は、このディレクトリのコピーを Informatica グローバルカスタマサポートに送付して問題を分析してもらう必要があります。
MDM Hub 環境レポートの確認	Hub コンソールのエンタープライズマネージャツールを使用して、Hub サーバー、プロセスサーバー、MDM Hub マスターデータベース、およびオペレーショナルリファレンスストアのデータベースに対する MDM Hub の現在の設定を確認します。コンポーネントのバージョン履歴も確認します。 環境レポートのコピーをアップグレードドキュメントフォルダに保存します。

## 環境の準備

選択するアップグレードプロセスのためのタスクを実行します。

## クリーンアップグレード

クリーンアップグレードの場合、次のタスクを実行します。

タスク	説明
最小システム要件の確認	<p>マシンが、MDM Hub インストールのハードウェア要件とソフトウェア要件を満たしていることを確認します。ハードウェア要件は、データ、処理容量、およびビジネスルールによって異なります。</p> <p>MDM Hub をインストールするには、マシンが次の最小システム要件を満たしている必要があります。</p> <ul style="list-style-type: none"><li>- ディスク容量。4.9GB</li><li>- 開発環境の RAM。4GB</li></ul> <p>MDM Hub コンポーネントのランタイム物理メモリ要件を確認するには、次の計算式を使用してください。</p> <p>Total run-time memory requirement for MDM Hub components = JDK JVM max heap size of the application server + operating system native heap size</p>
Java Development Kit (JDK) のインストール	<p>MDM Hub をインストールするマシンに、サポートされているバージョンの JDK をインストールします。JDK は、MDM Hub インストーラにバンドルされていません。</p> <p>JBoss 環境では、Informatica が提供する Azul Zulu JDK のバージョンをインストールする必要があります。Informatica の発送部署に問い合わせて Azul Zulu JDK をダウンロードします。</p> <p><b>注:</b> アプリケーションサーバーマシンと Hub コンソールを起動するマシンで、同じ Java バージョンを使用します。</p>
Visual Studio 2019 の Visual C++再頒布可能パッケージのインストール (Windows のみ)	<p>Windows システムでは、Multidomain MDM は名前検索機能と照合機能をサポートするために Visual Studio 2019 の Visual C++再配付可能パッケージが必要です。</p>
環境変数の設定	<p>MDM Hub をインストールするための環境変数を設定します。</p> <p>適切な JDK を使用するには、次の環境変数を、JDK ディレクトリを参照するように設定します。</p> <ul style="list-style-type: none"><li>- JAVA_HOME。必須</li><li>- PATH。必須</li></ul> <p>Oracle 用の正しいロケール動作を設定するには、Oracle ローダー、MDM Hub コンポーネントなどのクライアントアプリケーション環境に、NLS_LANG 環境変数を設定します。</p> <p>NLS_LANG パラメータは以下の形式で指定します。</p> <p>NLS_LANG = &lt;language&gt;_&lt;territory&gt;.&lt;character set&gt;</p> <p><b>注:</b> 中国語、日本語、韓国語、またはアクセント記号付き文字を含んだレコードを保存および検索する場合は、文字セットを UTF-8 に設定します。</p> <p>NLS_LANG の設定の詳細については、Oracle のマニュアルを参照してください。</p> <p>スキーマを正常にエクスポートするには、次の環境変数を設定します。</p> <p>Windows の場合:</p> <pre>set JAVA_TOOL_OPTIONS= --add-opens=java.xml/com.sun.org.apache.xml.internal.serialize=ALL-UNNAMED</pre> <p>UNIX の場合:</p> <pre>export JAVA_TOOL_OPTIONS= "--add-opens=java.xml/com.sun.org.apache.xml.internal.serialize=ALL-UNNAMED"</pre>

タスク	説明
オペレーティングシステムのロケールの設定	Hub サーバー、MDM Hub マスタデータベース、オペレーショナル参照ストア、Hub コンソールに、同じオペレーティングシステムのロケールを設定します。
UNIX での X Window System の設定	インストーラを UNIX 上でグラフィカルモードで実行する場合は、X Window System を設定します。X Window System は、グラフィック表示サーバーです。X Window System の設定の詳細については、オペレーティングシステムのマニュアルを参照してください。
アプリケーションサーバーのルートようこそページへのアクセス権の無効化	セキュリティを向上するには、アプリケーションサーバーのルートようこそページへのアクセス権を無効にします。手順については、アプリケーションサーバーのマニュアルを参照してください。
セキュアでない TLS 暗号スイートの無効化	<p>セキュリティを向上するには、Multidomain MDM と一緒に使用する Java Runtime Environment で、セキュアでない TLS 暗号スイートを無効にします。</p> <ol style="list-style-type: none"> <li>1. <code>../jdk&lt;version&gt;/jre/lib/security/java.security</code> ファイルを開きます。</li> <li>2. <code>jdk.tls.disabledAlgorithms</code> プロパティを見つけ、セキュアでない暗号スイートの次のリストを含めるように値を更新します。</li> </ol> <pre>jdk.tls.disabledAlgorithms = SSLv3, RC4, MD5withRSA, DH keySize &lt; 1024, EC keySize &lt; 224, DES40_CBC, RC4_40, 3DES_EDE_CBC, EDH-RSA-DES-CBC3-SHA, ECDHE-RSA-DES-CBC3-SHA, DES-CBC3-SHA</pre> <p>プロパティの詳細については、JDK のマニュアルを参照してください。</p>

クリーンアップグレードの場合、次の追加のタスクを実行します。

タスク	詳細
メタデータの検証	<p>オペレーショナルリファレンスストア（ORS）に検証エラーがないことを確認します。ORS に無効なメタデータが含まれる場合に Hub ストアをアップグレードすると、予期しない結果が生成される場合があります。ハブコンソールでリポジトリマネージャを使用して、メタデータを検証します。検証上のすべての問題を解決し、メタデータを再検証して、検証上の問題が解決されていることを確認します。</p> <p>最終的な検証結果のコピーをアップグレードドキュメントフォルダ <code>upgradedoc</code> に保存します。MDM Hub コンソールのリポジトリマネージャツールを使用して、検証結果を検証して保存します。</p>
永続的な ID 実装の更新	永続的な ID を使用する場合は、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。Multidomain MDM のアップグレードされたバージョンと互換性があるように、永続的な ID 実装を更新する必要があります。

## インプレースアップグレード

インプレースアップグレードの場合、次のタスクを実行します。

タスク	詳細
システム要件を満たすための MDM 環境の更新	<p>オペレーティングシステム、アプリケーションサーバー、JDK、およびデータベースサーバーの更新が必要になる場合があります。システム要件については、本バージョンの MDM（Multidomain MDM）の「製品可用性マトリックス」を Informatica Network (<a href="https://network.informatica.com/community/informatica-network/product-availability-matrices/overview">https://network.informatica.com/community/informatica-network/product-availability-matrices/overview</a>) で参照してください。</p> <p>JBoss 環境では、Informatica が提供する Azul Zulu JDK のバージョンをインストールする必要があります。Informatica の発送部署に問い合わせて Azul Zulu JDK をダウンロードします。</p> <p>。現在の MDM システム要件に準拠するように、MDM を実行するマシンを更新します。</p>
Visual Studio 2019 の Visual C++再頒布可能パッケージのインストール (Windows のみ)	<p>Windows システムでは、Multidomain MDM は名前検索機能と照合機能をサポートするために Visual Studio 2019 の Visual C++再配付可能パッケージが必要です。</p>
メタデータの検証	<p>オペレーショナルリファレンスストア（ORS）に検証エラーがないことを確認します。ORS に無効なメタデータが含まれる場合に Hub ストアをアップグレードすると、予期しない結果が生成される場合があります。ハブコンソールでリポジトリマネージャを使用して、メタデータを検証します。検証上のすべての問題を解決し、メタデータを再検証して、検証上の問題が解決されていることを確認します。</p> <p>最終的な検証結果のコピーをアップグレードドキュメントフォルダ upgradedoc に保存します。</p> <p>MDM Hub コンソールのリポジトリマネージャツールを使用して、検証結果を検証して保存します。</p>
永続的な ID 実装の更新	<p>永続的な ID を使用する場合は、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。Multidomain MDM のアップグレードされたバージョンと互換性があるように、永続的な ID 実装を更新する必要があります。</p>

タスク	詳細
実装のバックアップ	<p>カスタマイズを保持し、環境を復元できるようにするには、現在の実装をバックアップします。アップグレード中に問題が発生した場合に、バックアップから環境を復元できます。</p> <p><b>スキーマのバックアップ</b></p> <p>スキーマのフルバックアップを実行します。アップグレードプロセスで行われたスキーマ変更はロールバックできません。アップグレード中に問題が発生した場合には、スキーマはバックアップからリストアできます。スキーマのバックアップ手順については、データベースのマニュアルを参照してください。</p> <p><b>Hub Server とプロセスサーバーがインストールされているディレクトリのバックアップ</b></p> <p>Hub サーバーとプロセスサーバーをインストールする前に、ご使用の環境の Hub サーバーとプロセスサーバーのインストールフォルダをバックアップします。</p> <p><b>Data Director アプリケーションのバックアップ</b></p> <p>サブジェクト領域と使用する Data Director アプリケーションのエクスポートに関する詳細については、『<i>Multidomain MDM Data Director の実装ガイド</i>』を参照してください。</p> <p><b>カスタマイズの登録またはバックアップ</b></p> <p>カスタムクエリ、カスタム関数、ユーザーイグジットなど、ハブコンソールに登録するカスタマイズは、アップグレードプロセス時に保持されます。</p> <p>Hub コンソールに登録していないカスタマイズの最新ソースコードをバックアップします。登録していないカスタマイズは、アップグレード後には使用できなくなります。</p> <p><b>カスタマイズしたクレンジングエンジン設定ファイルのバックアップ</b></p> <p>カスタマイズしたクレンジングエンジン設定ファイルをすべてバックアップします。</p>
JAVA_HOME 環境変数の設定	<p>JBoss または WebLogic。アップグレードを既存のインストールディレクトリに適用する場合、次の手順を実行します。</p> <ol style="list-style-type: none"> <li>JBoss 環境で、次のファイルから JAVA_HOME 環境変数設定を手動で削除します。 <ul style="list-style-type: none"> <li>Windows: &lt;MDM Hub installation directory: MDM Hub のインストールディレクトリ&gt;\setSiperianEnv.bat</li> <li>UNIX: &lt;MDM Hub installation directory: MDM Hub のインストールディレクトリ&gt;/setSiperianEnv.sh</li> </ul> </li> <li>JBoss または WebLogic の環境では、JAVA_HOME 環境変数を、サポートされている JDK のバージョンのパスに設定します。</li> </ol>
アプリケーションサーバーのルートようこそページへのアクセス権の無効化	<p>セキュリティを向上するには、アプリケーションサーバーのルートようこそページへのアクセス権を無効にします。手順については、アプリケーションサーバーのマニュアルを参照してください。</p>

## 既存のジョブの処理

既存のジョブを処理するには、次のタスクを実行します。

アップグレードタスク	詳細
レコードを含むステージングテーブルでのロードジョブの実行	Microsoft SQL Server 環境において、レコードを含むステージングテーブル上でロードバッチジョブを実行します。 ステージングテーブルにレコードが含まれる場合にオペレーショナル参照ストアをアップグレードすると、ログファイルのサイズが利用可能なハードドライブの容量を超えるためにアップグレードが失敗する可能性があります。
ステージジョブの完了とステージテーブルの内容の削除	オペレーショナル参照ストアをアップグレードする前に、実行中のステージジョブを完了し、ステージテーブルの内容を削除します。 ステージテーブルの内容を削除しないと、Microsoft SQL Server 環境では、オペレーショナル参照ストアのアップグレード完了時間が、予期される時間より長くなります。

## MDM Hub 構成の更新

MDM Hub 構成を更新するには、次のタスクを実行します。

アップグレードタスク	詳細
ORS ユーザーに SYS.V_\$PARAMETER の選択権限を付与する	選択権限を付与するには、以下の SQL 文を実行します。 <code>grant select on SYS.V_\$PARAMETER to &lt;Operational Reference Store user&gt;;</code>
サブジェクト領域の設定で名前から特殊文字を削除する	プロビジョニングツールでは、名前に特殊文字を使用することはできません。サブジェクト領域の設定からビジネスエンティティスキーマを生成する場合は、最初に、サブジェクト領域の設定で名前に特殊文字があれば削除しておく必要があります。
カラム名に予約語が含まれないようにする	Multidomain MDM の最近のバージョンでは、追加のキーワードが予約済みとしてマークされています。実装で予約済みのキーワードを使用しないようにします。 予約語の完全な一覧については、 <i>Multidomain MDM の設定ガイド</i> の「Requirements for Defining Schema Objects」セクションを参照してください。 データが含まれるカラムの名前を変更するスクリプトを要求するには、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。

アップグレードタスク	詳細
マップされたカラムのデータ型の互換性を確認する	<p>オペレーションリファレンスストアに Oracle データベースを使用する場合、ランディングテーブルからステージングテーブルへのマッピングをチェックして、マップされたカラムのデータ型に互換性があるかどうかを確認します。</p> <p>以前のバージョンでは、マップされたカラムのデータ型に互換性がない、ランディングテーブルからステージングテーブルへのマッピングを作成できましたが、このバージョンから、マップされたカラムのデータ型は互換性があることが必須になっています。データをロードする際にデータ型が一致していないと、ロードは失敗し、ログファイルにエラーが記録されます。</p> <p>カラムのマッピングの詳細については、『<i>Multidomain MDM の設定ガイド</i>』を参照してください。</p>
ユーザー設定を C_REPOS_DS_PREF テーブルと C_REPOS_DS_PREF_DETAIL テーブルからバックアップする	<p>Data Director 環境に、カラムの表示または非表示に関するユーザー設定が含まれている場合、このバージョンで暗号化ハッシュアルゴリズムが変更されているため、アップグレード時はその設定は失われます。アップグレード後、ユーザー設定を再作成します。</p> <p>Data Director グローバルプロパティの詳細については、『<i>Multidomain MDM Data Director の実装ガイド</i>』を参照してください。</p>
ユーザーに有効な電子メールアドレスがあることを確認する	<p>既存のユーザーが、Hub コンソールのユーザー名に関連付けられた有効な電子メールアドレスを持っていることを確認します。MDM Hub にアクセスするためのパスワードをリセットするには、ユーザーは有効な電子メールアドレスを持っている必要があります。</p> <p>Hub コンソールでのユーザー情報の編集に関する詳細については、『<i>Multidomain MDM のセキュリティガイド</i>』を参照してください。</p> <p><b>注:</b> Hub コンソールの管理者ユーザーの電子メールアドレスは変更できません。管理者ユーザーの電子メールアドレスを変更するには、CMX_SYSTEM スキーマの C_REPOS_USER テーブルを</p>
インデックスを登録する	<p>カスタムインデックスを含むスキーマをアップグレードする場合は、カスタムインデックスを登録しておく必要があります。カスタムインデックスを登録するには、SIF の API の RegisterCustomIndex を使用します。</p> <p>RegisterCustomIndex SIF API の詳細については、『<i>Multidomain MDM サービスの統合フレームワークガイド</i>』を参照してください。</p>

## 元のスキーマ所有者によるオペレーションリファレンスストアの登録

オペレーションリファレンスストア（ORS）データベースのいずれかがプロキシユーザーで登録されている場合は、ORS データベースを元のスキーマ所有者で登録してください。アップグレード操作の実行に必要な特権を確実に得るには、元のスキーマ所有者で ORS を登録する必要があります。アップグレード後は、ORS をプロキシユーザーで登録してもかまいません。元のデフォルトのスキーマ所有者は CMX\_ORO です。

- Hub コンソールを起動します。  
[データベースの変更] ダイアログボックスが表示されます。
- MDM Hub マスタデータベースを選択して、[接続] をクリックします。
- 設定ワークベンチにあるデータベースツールを選択します。
- [書き込みロック] > [ロックの取得] の順に選択します。

5. **【データベースの登録】** ボタンをクリックします。

**Informatica MDM Hub 接続ウィザード**が表示され、データベースタイプの選択が求められます。

6. データベースのタイプを選択して **【次へ】** をクリックします。

7. データベースの接続プロパティを設定します。

- a. Oracle の接続方式を選択して、**【次へ】** をクリックします。

以下の Oracle 接続方式を選択できます。

#### **サービス**

サービス名を使用して Oracle に接続します。

#### **SID**

Oracle のシステム ID を使用して Oracle に接続します。

サービス名と SID 名の詳細については、Oracle のマニュアルを参照してください。

**【接続プロパティ】** ページが表示されます。

- b. 選択する接続タイプの接続プロパティを指定し、**【次へ】** をクリックします。

以下の接続プロパティを設定することができます。

#### **データベース表示名**

Hub コンソールに表示する必要があるオペレーショナルリファレンスストアの名前。

#### **マシン識別子**

Hub ストアインスタンスからのレコードを一意に識別するためにキーに割り当てられるプレフィックス。

#### **データベースホスト名**

Oracle データベースをホストするサーバーの IP アドレスまたは名前。

#### **SID**

サーバー上で実行される Oracle データベースのインスタンスを参照する Oracle システム識別子。**【SID】** フィールドは、**【SID】** 接続タイプを選択した場合に表示される。

#### **サービス**

Oracle データベースへの接続に使用する Oracle サービスの名前。**【サービス】** フィールドは、**【サービス】** 接続タイプを選択した場合に表示される。

#### **ポート**

Oracle データベースサーバー上で実行される Oracle リスナの TCP ポート。デフォルトは 1521。

#### **Oracle TNS 名**

ネットワーク上で認識されているデータベースの名前（アプリケーションサーバーの TNSNAMES.ORA ファイルで定義）。

例: mydatabase.mycompany.com

Oracle TNS 名は、Oracle データベースのインストール時に設定します。Oracle TNS 名の詳細については、Oracle のマニュアルを参照してください。

#### **スキーマ名**

オペレーショナルリファレンスストアの名前。

### ユーザー名

ORS の元のユーザー名を指定する。デフォルトは CMX\_ORS。

### パスワード

ORS の元のユーザーに関連付けられているパスワード。

**注:** cmxserver.properties ファイルで

cmx.server.database.authentication.method=windowsauthentication である場合、デフォルトのパスワード ChangeMe は、\*\*\*\*\*と表示されます。アプリケーションは、Windows 認証を使用してデータソースに接続します。

サマリページが表示されます。

- c. サマリを確認し、追加の接続プロパティを指定します。

以下の表に、設定可能な追加の接続プロパティを示します。

プロパティ	説明
接続 URL	接続 URL。接続ウィザードでは、デフォルトで接続 URL が生成されます。次のリストに、Oracle 接続タイプの接続 URL のフォーマットを示します。 <b>サービス接続タイプ</b> <code>jdbc:oracle:thin:@//database_host:port/service_name</code> <b>SID 接続タイプ</b> <code>jdbc:oracle:thin:@//database_host:port:sid</code> [サービス] 接続タイプの場合、カスタマイズし、後で別の接続 URL をテストするオプションがあります。
登録後にデータソースを作成する	登録後にアプリケーションサーバーのデータソースを作成する場合は選択します。 <b>注:</b> このオプションを選択しない場合、手動でデータソースを設定する必要があります。

- d. サービス接続タイプには、デフォルトの URL を変更する場合、**【編集】** ボタンをクリックし、URL を指定して **【OK】** をクリックします。

8. **【完了】** をクリックします。

**【データベースの登録】** ダイアログボックスが表示されます。

9. **【OK】** をクリックします。

MDM Hub で ORS が登録されます。

# MDM アプリケーションサーバーのプロパティファイルの更新

Hub サーバーをアップグレードすると、MDM Hub サーバーでの JBoss EAP の最新バージョンへの自動更新が失敗します。

最新の MDM Hub サーバーにアップグレードする前に、次の場所にある `cmxserver.properties` ファイルで `cmx.appserver.version` プロパティをバージョン 8 に更新してください。

- `infadm\hub\cleanse\resources\cmxcleanse.properties`
- `infadm\hub\server\resources\cmxserver.properties`

例: `cmx.appserver.version=8`

## BPM アップグレードの準備

ビジネスプロセス管理システムをアップグレードするには、適切なワークフローアダプタを選択することから開始します。スタンドアロン ActiveVOS を使用している場合、埋め込み ActiveVOS に移行し、`build.properties` ファイルを編集する必要があります。使用する環境に ActiveVOS がインストールされている場合、アップグレードタスクに従って `build.properties` ファイルを編集します。Informatica は ActiveVOS 9.2.4.6 のみをサポートします。

## ワークフローアダプタの選択

ワークフローアダプタの次のアップグレードオプションを確認し、ニーズに合ったアップグレードオプションを選択します。

次の表で、現在のワークフローアダプタとそのアップグレードオプションについて説明します。

現在のワークフローアダプタ	アップグレードオプション
SIPERIAN	<p>推奨されていません。</p> <p><b>オプション 1</b></p> <p>SIPERIAN をプライマリワークフローアダプタとして保持します。</p> <p><b>オプション 2</b></p> <p>ビジネスエンティティに基づいて、BE ActiveVOS ワークフローアダプタにアップグレードします。BE ActiveVOS ワークフローアダプタを使用するには、ビジネスエンティティ構成が必要です。</p> <p>詳細については、「<i>Migrating IDD Applications to the Business Entity Data Model</i>」を参照してください。</p>
Informatica ActiveVOS	<p>このワークフローアダプタはサブジェクト領域に基づいており、SIF API を使用して動作します。</p> <p><b>オプション 1</b></p> <p>Informatica ActiveVOS をプライマリワークフローアダプタとして保持します。</p> <p><b>オプション 2</b></p> <p>ビジネスエンティティに基づいて、BE ActiveVOS ワークフローアダプタにアップグレードします。BE ActiveVOS ワークフローアダプタを使用するには、ビジネスエンティティ構成が必要です。</p> <p>詳細については、『<i>Multidomain MDM Data Director の移行ガイド</i>』を参照してください。</p>
BE ActiveVOS	<p>このワークフローアダプタは、ビジネスエンティティに基づいており、ビジネスエンティティサービスを使用して動作します。</p> <p>製品可用性マトリックスで定義されている ActiveVOS のバージョンを使用する必要があります。環境内で古いバージョンが検出された場合は、アップグレードプロセスによって必要なバージョンの ActiveVOS がインストールされます。</p>

## スタンドアロン ActiveVOS から埋め込み ActiveVOS への移行（インプレースアップグレード）

スタンドアロン ActiveVOS を使用している場合、埋め込み ActiveVOS に移行する必要があります。スタンドアロン ActiveVOS はサポートされません。

次の表で、スタンドアロン ActiveVOS から埋め込み ActiveVOS に移行するのに必要なアップグレード前のタスクについて説明します。

アップグレードタスク	説明
ActiveVOS データソースの削除	スタンドアロン ActiveVOS サーバーが Multidomain MDM と同じ JBoss インスタンス、WebSphere プロファイル、または WebLogic ドメインにある場合、ActiveVOS のデータソースをアプリケーションサーバーから削除します。
WebSphere 環境での URN マッピングの編集	<ol style="list-style-type: none"><li>ActiveVOS コンソールを起動します。ブラウザで、正しいホスト名とポート番号に置き換えて、以下の URL を入力します。<ul style="list-style-type: none"><li>保護されている接続。https://&lt;host&gt;:&lt;port&gt;/activevos</li><li>保護されていない接続。http://&lt;host&gt;:&lt;port&gt;/activevos</li></ul></li><li>ActiveVOS コンソールの【ホーム】ページで、【管理】 &gt; 【サーバーの設定】 &gt; 【URN マッピング】をクリックします。</li><li>URN java:comp/env/jdbc/ActiveVOS が URL java:comp/env/jdbc/ActiveVOS にマッピングされていることを確認します。</li></ol>

**重要:** アップグレードプロセスの一部として Hub サーバーインストーラを実行する場合、必ず埋め込み ActiveVOS をインストールしてください。

## ビルドプロパティファイルの編集（インプレースアップグレード）

ActiveVOS が MDM 環境にインストールされている場合、build.properties ファイルを編集するにはアップグレードタスクを実行する必要があります。

次の表で、build.properties ファイルを編集するのに必要なアップグレードタスクについて説明します。

アップグレードタスク	詳細
IBM Db2 へのデータベース接続プロパティの設定	<p>IBM Db2 環境では、アップグレード前に、ActiveVOS データベース接続プロパティを &lt;MDM Hub installation directory: MDM Hub のインストールディレクトリ&gt;/hub/server/bin の build.properties ファイルに追加します。</p> <p>次のサンプルは、サンプルエントリが含まれる ActiveVOS データベース接続プロパティを示しています。</p> <pre>activevos.db.type=db2 activevos.db.server=localhost activevos.db.port=50000 activevos.db.user=AVOS activevos.db.dbname=INFA102 activevos.db.schemaname=AVOS activevos.db.jdbc.url=jdbc:db2://localhost:50000/INFA102 activevos.b4p.url=http://localhost:9080/active-bpel/services/AeB4PTaskClient-taskOperations</pre>
ActiveVOS インストールディレクトリの構成	<p>アップグレードプロセスでインストールされている ActiveVOS のバージョンがシステム要件を満たしていないことが検出されると、アップグレードプロセスによって、必要なバージョンの ActiveVOS が新しいディレクトリにインストールされます。</p> <p>以前のインストールと同じディレクトリに ActiveVOS をインストールするには、build.properties ファイル内の次のエントリを削除するか、コメントアウトします。</p> <pre>activevosinstall.dir=&lt;ActiveVOS installation directory&gt;</pre>
ActiveVOS の設定	<p>JBoss 環境の場合にのみ適用されます。ActiveVOS 設定プロパティ sip.appserver.web.url がデフォルトのポート番号 8080 に設定されていない場合は、プロパティとポート番号を次の値に手動で更新する必要があります。</p> <pre>-Dsip.appserver.web.url= http://localhost:8079</pre> <p>ポート番号は 8079 か他の未使用のポート番号に設定できます。</p> <p>&lt;MDM Hub インストールディレクトリ&gt;/hub/server/bin の build.properties ファイル内のプロパティを更新します</p>

## タイムラインバージョン抽出テーブルの設定

ベースオブジェクトでタイムラインが有効になっている場合は、タイムラインバージョン抽出テーブルを作成して入力します。createtve スクリプトにより、タイムラインバージョン抽出テーブルが作成され、populatetve スクリプトによりレコードバージョンデータがテーブルに入力されます。

タイムラインバージョン抽出テーブルの詳細については、『*Multidomain MDM 設定ガイド*』を参照してください。

1. Business Entity Timeline パッケージから pre\_upgrade\_util.zip ファイルをダウンロードします。
2. pre\_upgrade\_util.zip ファイルを抽出します。
3. 次のディレクトリをコピーします。  
<MDM Hub インストールディレクトリ>/hub/server/resources directory/certificates。

4. 証明書ディレクトリを次のディレクトリに貼り付けます。  
pre\_upgrade\_util/resources
5. タイムラインバージョン抽出テーブルを作成するには、コマンドプロンプトを開き、次のコマンドを実行します。  
UNIX の場合:  

```

chmod 777 setInfaEnv.sh
cd bin
chmod 777 sip_ant.sh
chmod 777 exec_cmd.sh
sip_ant.sh createtve

```

Windows の場合:  

```

cd pre_upgrade_util/bin
sip_ant.bat createtve

```
6. プロンプトに回答します。  
Oracle 環境の場合、次の情報を提供してください。

プロンプト	説明
データベースタイプ (ORACLE、DB2、MSSQL) を入力	[データベースタイプ]。ORACLE を指定する。
Oracle Net の接続 ID (TNS_Name) を入力	Oracle TNS 名。
オペレーショナル参照ストア のユーザー名[mdm_sample] を入力	オペレーショナル参照ストアスキーマにアクセスするためのユーザー名。
ORS のユーザーパスワード を入力	オペレーショナル参照ストアユーザーのパスワード。
アップグレードスクリプトディレクトリ[C:\BaseSoftware] を入力	オペレーショナル参照ストアスキーマに対して実行するスクリプトの主な場所。 次のディレクトリでスクリプトを見つけます: pre_upgrade_util\bin \oracle\scripts。

IBM Db2 環境の場合、次の情報を提供してください。

**注:** DB2 データベース名は大文字で入力する必要があります。

プロンプト	説明
データベースタイプ (ORACLE、DB2、MSSQL) を入力	データベースタイプ。DB2 を指定する。
DB2 データベース名 [M104HF3]を入力	DB2 データベース名。 <b>注:</b> DB2 データベース名は大文字で入力する必要があります。
オペレーショナル参照ストア のデータベース名 [mdm_sample]を入力	スクリプトを適用するオペレーショナル参照ストアデータベースの名前。

プロンプト	説明
オペレーショナル参照ストアのユーザー名[mdm_sample]を入力	オペレーショナル参照ストアデータベースにアクセスするためのユーザー名。
オペレーショナル参照ストアのユーザーパスワードを入力	オペレーショナル参照ストアユーザーのパスワード。
アップグレードスクリプトディレクトリ[C:\BaseSoftware]を入力	オペレーショナル参照ストアスキーマに対して実行するスクリプトの主な場所。 次のディレクトリでスクリプトを見つけます: pre_upgrade_util\bin\oracle\scripts。

Microsoft SQL 環境の場合は、次の情報を入力します。

プロンプト	説明
データベースタイプ (ORACLE、DB2、MSSQL) を入力	[データベースタイプ]。MSSQL を指定する。
Microsoft SQL データベースのホスト名[localhost]を入力	Microsoft SQL Server の場所。
Microsoft SQL データベースのポート番号[1433]を入力	Microsoft SQL Server リスナが使用するポート番号。
オペレーショナル参照ストアのユーザー名[mdm_sample]を入力	オペレーショナル参照ストアデータベースにアクセスするためのユーザー名。
オペレーショナル参照ストアのユーザーパスワードを入力	オペレーショナル参照ストアユーザーのパスワード。
アップグレードスクリプトディレクトリ[C:\BaseSoftware]を入力	オペレーショナル参照ストアスキーマに対して実行するスクリプトの主な場所。 次のディレクトリでスクリプトを見つけます: pre_upgrade_util\bin\oracle\scripts。

- タイムラインバージョン抽出テーブルにレコードバージョンデータを入力するには、次のコマンドを実行します。

UNIX の場合:

```
chmod 777 setInfaEnv.sh
cd bin
chmod 777 sip_ant.sh
chmod 777 exec_cmd.sh
sip_ant.sh populatetve
```

Windows の場合:

```
cd pre_upgrade_util/bin
sip_ant.bat populatetve
```

- プロンプトに回答します。

Oracle サービス環境または SID 環境の場合は、次の情報を入力します。

プロンプト	説明
データベースタイプ (ORACLE、DB2、MSSQL) を入力	[データベースタイプ]。ORACLE を指定する。
Oracle 接続タイプ (SERVICE、SID) を入力	接続タイプ。 以下の値を使用する。 - SERVICE。Oracle に接続するサービス名を使用する。 - SID。Oracle に接続する Oracle システム ID を使用する。
マスターデータベースの Oracle JDBC 名 [jdbc:oracle:thin:@// localhost:1521]を入力	JDBC 接続文字列。
マスターデータベース名 [orcl]を入力	Oracle TNS 名。
マスタースキーマのユーザー 名[cmx_system]を入力	cmx_system スキーマにアクセスするためのユーザー名。
マスタースキーマのユーザー パスワードを入力	データベースユーザーのパスワード。
ORS ID [orcl-MDM_SAMPLE] を入力	DATABASE_ID カラムの下での C_REPOS_DATABASE テーブルにあるデータベース名。
再開ブール値 (true、false) を入力	失敗した場合にスクリプトを再実行するかどうかを示します。
最終実行日 (NULL、yyyy- mm-dd) を入力	ベースオブジェクトの最終更新日。
入力される TABLE_NAME (例: C_ACCOUNT または ALL) を入力	特定のテーブルまたはすべてのテーブルに入力します。特定のテーブルに入力するには、テーブル名を使用します。例: C_ACCOUNT。すべてのテーブルに入力するには、ALL を使用します。

IBM Db2 環境の場合、次の情報を提供してください。

プロンプト	説明
データベースタイプ (ORACLE、DB2、MSSQL) を入力	データベースタイプ。DB2 を指定する。
DB2 JDBC 名を入力	JDBC 接続文字列
DB2 マスターデータベース名 [MAIN]を入力	DB2 データベース名。 <b>注:</b> DB2 データベース名は大文字で入力する必要があります。
マスターデータベースのユーザー名[cmx_system]を入力	cmx_system データベースにアクセスするためのユーザー名。

プロンプト	説明
マスターデータベースのユーザーパスワードを入力	データベースユーザーのパスワード。
ORS ID [orcl-MDM_SAMPLE] を入力	DATABASE_ID カラムの下 C_REPOS_DATABASE テーブルにあるデータベース名。
再開ブール値 (true、false) を入力	失敗した場合にスクリプトを再実行するかどうかを示します。
最終実行日 (NULL、yyyy-mm-dd) を入力	ベースオブジェクトの最終更新日。
入力される TABLE_NAME (例: C_ACCOUNT または ALL) を入力	特定のテーブルまたはすべてのテーブルに入力します。特定のテーブルに入力するには、テーブル名を使用します。例: C_ACCOUNT。すべてのテーブルに入力するには、ALL を使用します。

Microsoft SQL 環境の場合は、次の情報を入力します。

プロンプト	説明
データベースタイプ (ORACLE、DB2、MSSQL) を入力	[データベースタイプ]。MSSQL を指定する。
MSSQL JDBC 名 [jdbc:sqlserver://localhost:1433] を入力	JDBC 接続文字列
マスタースキーマのユーザー名 [cmx_system] を入力	cmx_system データベースにアクセスするためのユーザー名。
マスタースキーマのユーザーパスワードを入力	データベースユーザーのパスワード。
ORS ID [orcl-MDM_SAMPLE] を入力	DATABASE_ID カラムの下 C_REPOS_DATABASE テーブルにあるデータベース名。
再開ブール値 (true、false) を入力	失敗した場合にスクリプトを再実行するかどうかを示します。
最終実行日 (NULL、yyyy-mm-dd) を入力	ベースオブジェクトの最終更新日。
入力される TABLE_NAME (例: C_ACCOUNT または ALL) を入力	特定のテーブルまたはすべてのテーブルに入力します。特定のテーブルに入力するには、テーブル名を使用します。例: C_ACCOUNT。すべてのテーブルに入力するには、ALL を使用します。

## 第 3 章

# データベースタスク

この章では、以下の項目について説明します。

- [データベースタスクの概要, 34 ページ](#)
- [Oracle データベースのセットアップ, 34 ページ](#)
- [Microsoft SQL Server のセットアップ, 40 ページ](#)
- [IBM Db2 データベースのセットアップ, 43 ページ](#)

## データベースタスクの概要

Multidomain MDM と連動するように、データベースを設定します。データベースの手順に従います。

クリーンアップグレードを実行する場合、Multidomain MDM の新規インストールの実行時と同様に、データベースのすべての手順を実行します。

データベースのバージョンが引き続きサポートされているインプレースアップグレードを実行する場合、このバージョンの Multidomain MDM と連動するように、データベースが設定されていることを確認します。

## Oracle データベースのセットアップ

MDM Hub マスターデータベースおよびオペレーショナル参照ストアを作成する前に、データベース環境を設定します。

データベース環境を設定するには、次のタスクを実行します。

1. Oracle のインストールおよび設定
2. データベースインスタンスを設定します。
3. データベースの特権と接続を設定します。
4. ActiveVOS スキーマを作成します。

## 手順 1. Oracle のインストールと設定

Oracle のマニュアルに記載されている手順に従って、Oracle データベースをインストールして設定できます。

次の表に、Oracle インスタンスを配置する各マシンで実行する必要がある Oracle のインストールおよび設定タスクを示します。:

タスク	説明
Oracle のインストール	サポートされているバージョンの Oracle データベースのインストール <b>注:</b> Oracle マルチテナント機能を使用する場合は、MDM Hub インストール用にプラグラブルデータベース (PDB) を設定します。
クライアントとユーティリティのインストール	MDM Hub との通信や MDM Hub プロセスの実行をする Oracle クライアントおよびユーティリティソフトウェアをインストールします。 Hub サーバーまたはプロセスサーバーを実行する各マシンに、次のソフトウェアをインストールします。 - Oracle クライアント - SQL*Loader
リサイクルビンの無効化	リサイクルビン (USER_RECYCLEBIN と DBA_RECYCLEBIN) を無効にします。リサイクルビンによって MDM Hub のプロセスが妨害されることがあります。 リサイクルビンをシステムレベルまたはセッションレベルで無効にします。また、リサイクルビン内の既存のオブジェクトを消去します。
初期化パラメータの設定	Oracle 初期化パラメータを init.ora ファイルに設定します。 Oracle 初期化パラメータの詳細については、『MDM Multidomain Edition パフォーマンスのチューニングの概要』を参照してください。

Oracle のインストールと設定の詳細については、Oracle のマニュアルを参照してください。

## 手順 2. データベースとテーブルスペースの作成

Oracle データベースのインストールと設定の後、データベースとテーブルスペースを作成して設定します。

次の表に、データベースを設定するために実行する必要があるタスクを示します。

タスク	説明
データベースの作成	データベースインスタンスごとにデータベースを作成します。
テーブルスペースの作成	MDM Hub データ用のテーブルスペースを作成します。MDM Hub にロードするデータ量に基づいて、デフォルトのテーブルスペースサイズおよびデータファイル数を調整します。 次のテーブルスペースを作成します。 - CMX_DATA。MDM Hub のメタデータおよびユーザーデータが含まれます。 - CMX_INDX。MDM Hub で作成および使用されるインデックスが含まれます。 - CMX_TEMP。MDM Hub 用の一時テーブルが含まれます。 - BPM_DATA。ビジネスプロセス管理用の ActiveVOS データが含まれます。 <b>注:</b> 複数の MDM Hub マスタデータベースを作成する場合は、MDM Hub マスタデータベースごとに一意のテーブルスペースを作成します。

## オンプレミスのテーブルスペースの作成

オンプレミスの Oracle を使用する場合、ローカル管理または手動設定を行う永続テーブルスペースとしてテーブルスペースを作成します。

**注:** 大量のデータロードのデータベース管理を簡素化するには、ビッグファイルのテーブルスペースを作成します。ただし、必要に応じてスモールファイルのテーブルスペースを作成できます。

1. SYSTEM などのデータベース管理者ユーザーとして Oracle にログインします。
2. テーブルスペースを作成します。

次の表に、テーブルスペースを作成するサンプルの SQL 文を示します。

テーブルスペース名	サンプルの SQL 文
CMX_DATA	<pre>CREATE BIGFILE TABLESPACE CMX_DATA NOLOGGING DATAFILE '&lt;Oracle install directory&gt;/CMX_DATA1.dbf' SIZE 2048M REUSE EXTENT MANAGEMENT LOCAL;</pre> <p><b>注:</b> カスタムテーブルスペース名の詳細については、<a href="#">「カスタムテーブルスペース名用のデータベース環境の設定」</a> (ページ 37) を参照してください。</p>
CMX_INDX	<pre>CREATE BIGFILE TABLESPACE CMX_INDX NOLOGGING DATAFILE '&lt;Oracle install directory&gt;/CMX_INDX1.dbf' SIZE 2048M REUSE EXTENT MANAGEMENT LOCAL;</pre> <p><b>注:</b> カスタムテーブルスペース名の詳細については、<a href="#">「カスタムテーブルスペース名用のデータベース環境の設定」</a> (ページ 37) を参照してください。</p>
CMX_TEMP	<pre>CREATE BIGFILE TABLESPACE CMX_TEMP NOLOGGING DATAFILE '&lt;Oracle install directory&gt;/CMX_TEMP1.dbf' SIZE 2048M REUSE EXTENT MANAGEMENT LOCAL;</pre>
BPM_DATA	<pre>CREATE BIGFILE TABLESPACE BPM_DATA NOLOGGING DATAFILE '&lt;Oracle install directory&gt;/BPM_DATA1.dbf' SIZE 2048M REUSE EXTENT MANAGEMENT LOCAL;</pre>

## Amazon Relational Database Service でのテーブルスペースの作成

Amazon Relational Database Service (RDS) for Oracle を使用する場合、Amazon RDS でテーブルスペースを永続テーブルスペースとして作成します。

**注:** 大量のデータロードのデータベース管理を簡素化するには、ビッグファイルのテーブルスペースを作成します。ただし、必要に応じてスモールファイルのテーブルスペースを作成できます。

1. データベース管理者ユーザーとして Amazon RDS for Oracle にログインします。
2. テーブルスペースを作成します。

次の表に、テーブルスペースを作成するサンプルの SQL 文を示します。

デフォルトのテーブルスペース名	サンプルの SQL 文
CMX_DATA	<pre>CREATE BIGFILE TABLESPACE CMX_DATA DATAFILE SIZE 2048M AUTOEXTEND ON NEXT 2048M;</pre> <p>注: リポジットリマネージャ検証エラーを防止するために、デフォルトのテーブルスペース名を変更しないでください。</p>
CMX_INDX	<pre>CREATE BIGFILE TABLESPACE CMX_INDX DATAFILE SIZE 2048M AUTOEXTEND ON NEXT 2048M;</pre> <p>注: リポジットリマネージャ検証エラーを防止するために、デフォルトのテーブルスペース名を変更しないでください。</p>
CMX_TEMP	<pre>CREATE BIGFILE TABLESPACE CMX_TEMP DATAFILE SIZE 2048M AUTOEXTEND ON NEXT 2048M;</pre>

## カスタムテーブルスペース名用のデータベース環境の設定

使用するテーブルスペース名が、CMX\_INDX または CMX\_TEMP のデフォルトテーブルスペース名と異なる場合は、初期化パラメータの DEFERRED\_SEGMENT\_CREATION を無効にします。このパラメータを無効にするのは、リポジットリマネージャの検証エラーを防止するためです。

- ▶ DEFERRED\_SEGMENT\_CREATION を無効にするには、次の SQL 文を実行して、データベースを再起動します。  

```
ALTER SYSTEM SET DEFERRED_SEGMENT_CREATION=FALSE SCOPE=BOTH;
```

## 手順 3. データベース特権と接続の設定

データベースの特権と接続を設定します。

次の表で、データベースの特権と接続を設定するために実行する必要があるタスクについて説明します。

タスク	説明
データベース管理者ユーザーへの特権の付与	<p>データベース管理者ユーザーを使用して MDM Hub マスタデータベースとオペレーショナル参照ストアを作成する場合は、データベース管理者ユーザーに特権を付与します。ユーザーは、分散トランザクションと DBMS_LOCK のオプションを付与する特権を持っている必要があります。</p> <p>データベース管理者ユーザーに特権を付与するには、オプションを付与する特権を持つユーザーとしてデータベースに接続し、次の SQL 文を実行します。</p> <pre>GRANT SELECT ON sys.pending_trans\$ TO &lt;DBA user&gt; with grant option; GRANT SELECT ON sys.dba_pending_transactions TO &lt;DBA user&gt; with grant option; GRANT SELECT ON sys.dba_2pc_pending TO &lt;DBA user&gt; with grant option; GRANT EXECUTE ON sys.dbms_xa TO &lt;DBA user&gt; with grant option; GRANT EXECUTE ON sys.dbms_lock TO &lt;DBA user&gt; with grant option;</pre>
Oracle TNS 名の追加	<p>Oracle データベースに接続するには、TNS 名前エントリを Hub サーバーおよび Process サーバーマシン上の tnsnames.ora ファイルに追加します。</p> <p>TNS 名前エントリを追加するには、次の構文を使用します。</p> <pre>&lt;TNS NAME&gt; = (DESCRIPTION =   (ADDRESS_LIST =     (ADDRESS = (PROTOCOL = TCP)(Host = &lt;Oracle server host name&gt;)(Port = &lt;Oracle server port&gt;))   )   (CONNECT_DATA =     (SERVICE_NAME = &lt;Oracle SID&gt;)   ) )</pre> <p><b>注:</b> Hub サーバーおよび Process サーバーマシン上の TNS 名は同じでなければなりません。</p>
データベース接続のテスト	<p>Hub サーバーまたは Process サーバーを実行する各マシンから、データベースへの接続をテストします。</p> <p>SQL*Plus で、次の SQL 文の構文を使用します。</p> <pre>sqlplus &lt;user name&gt; /&lt;password&gt; @&lt;TNS Name&gt;</pre>

## 手順 4. ActiveVOS スキーマの作成

ActiveVOS をインストールするには、ActiveVOS スキーマを作成する必要があります。スキーマを作成するには、create\_bpm スクリプトを実行します。

複数の MDM Hub マスタデータベースを作成する場合は、MDM Hub マスタデータベースごとに ActiveVOS スキーマを作成します。

**注:** Oracle マルチテナント機能を使用する場合は、プラグブルデータベース (PDB) に ActiveVOS スキーマを作成します。

1. コマンドプロンプトを開き、次のディレクトリに移動します。  
<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin
2. 次のコマンドを実行します。  
UNIX の場合: ./sip\_ant.sh create\_bpm  
Windows の場合: sip\_ant.bat create\_bpm

3. 表示されるプロンプトに回答します。

プロンプトでは、デフォルトのテキストが括弧内に表示されます。デフォルト値を使用して次のプロンプトに進むには、**Enter** キーを押します。

プロパティ	説明
Database Type	データベースのタイプ。Oracle データベースの場合、Oracle を指定します。データベースタイプは、MDM Hub マスターデータベースとオペレーショナル参照ストア用に選択したデータベースタイプと同じである必要があります。
Oracle Connection Type	接続タイプ。以下の値を使用する。 - SERVICE。Oracle に接続するサービス名を使用する。 - SID。Oracle に接続する Oracle システム ID を使用する。
ActiveVOS Database Host Name	データベースをホストするマシンの名前。
ActiveVOS Database Port	データベースリスナが使用するポート番号。
データベースサービス名	Oracle サービスの名前。このプロパティは、選択した Oracle 接続タイプが SERVICE の場合に必要になります。
Oracle Net Connect Identifier (TNS Name)	Oracle TNS 名。
Database SID	Oracle システム ID の名前。このプロパティは、選択した Oracle 接続タイプが SID の場合に必要になります。
DBA User Name	データベース管理者ユーザーのユーザー名。
DBA Password	管理者ユーザーのパスワード。
ActiveVOS User Name	ActiveVOS サーバーの管理者ユーザーの名前。
ActiveVOS User Password	管理者ユーザーのパスワード。
ActiveVOS User Tablespace	MDM ワークフローに関連するレコードを格納するテーブルスペースの名前。
ActiveVOS User Temp Tablespace	一時テーブルスペースの名前。

4. スキーマを作成したら、次のディレクトリの sip\_ant.log ファイルを確認します。

<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin

sip\_ant.log ファイルには、ActiveVOS のスキーマを作成するために sip\_ant スクリプトを実行したときに発生したすべてのエラーが記録されます。

# Microsoft SQL Server のセットアップ

MDM Hub マスターデータベースおよびオペレーショナル参照ストアを作成する前に、データベース環境を設定します。

データベース環境を設定するには、次のタスクを実行します。

1. Microsoft SQL Server をインストールして設定します。
2. データファイルストアを作成します。
3. ODBC ドライバをインストールします。
4. ActiveVOS スキーマを作成します。

## 手順 1。Microsoft SQL Server のインストールと設定

Microsoft SQL Server のマニュアルに書かれた手順に従って、Microsoft SQL Server をインストールして設定できます。

次の表に、Microsoft SQL Server のインストールおよび設定のタスクを示します。

タスク	説明
Microsoft SQL Server のインストール	サポートされているバージョンの Microsoft SQL Server のインストール <b>注:</b> データベースエンジン認証セキュリティモードとして混合モードを指定したことを確認します。
JDBC ドライバの設定	Multidomain MDM を、Microsoft SQL Server 2017 を使用する環境にインストールする場合、次のタスクを実行します。 1. Microsoft の Web サイトから、サポートされている最新バージョンの Microsoft JDBC ドライバをダウンロードします。 2. ドライバファイルを、Microsoft SQL Server をインストールしたマシンの Binn ディレクトリにコピーします。
分散トランザクションの設定	MDM Hub には分散トランザクション環境が必要であるため、分散トランザクション用に Microsoft SQL Server を設定します。 分散トランザクション用に Microsoft SQL Server を設定するには、XA トランザクション用に MS DTC を有効にし、JDBC 分散トランザクションコンポーネントを設定します。
TCP/IP ネットワークプロトコルの有効化	MDM Hub 環境が必要とする TCP/IP ネットワークプロトコルを使用するように Microsoft SQL Server を設定します。

## 分散トランザクション用の Microsoft SQL Server の設定

Microsoft SQL Server を起動する前に、必ず分散トランザクション用に設定してください。MDM Hub には分散型トランザクション環境が必要です。

1. Microsoft SQL Server を起動したときに Microsoft 分散トランザクションコーディネーター (MS DTC) サービスが起動したことを確認するには、サービスマネージャで MS DTC を[自動] に設定します。
2. XA トランザクション用に MS DTC を有効にします。
3. SNA LU 6.2 トランザクション用に MS DTC を有効にします。

4. JDBC 分散トランザクションコンポーネントを設定します。
  - a. Microsoft Web サイトからマシン上のディレクトリに、サポートされているバージョンの Microsoft SQL Server JDBC ドライバをダウンロードして、抽出します。
  - b. sqljdbc\_xa.dll ファイルを、Microsoft SQL Server をインストールしたマシンの Binn ディレクトリにコピーします。
  - c. Microsoft SQL Server インスタンス上で、xa\_install.sql スクリプトを管理者として実行します。  
xa\_install.sql スクリプトは、JDBC ドライバの xa ディレクトリにあります。
  - d. スクリプトにより、Microsoft SQL Server マスターデータベースに SqlJDBCXAUser のロールが作成されたことを確認します。

分散トランザクション用に Microsoft SQL Server をインストールおよび設定する方法の詳細については、Microsoft SQL Server のドキュメントを参照してください。

## TCP/IP ネットワークプロトコルの有効化

SQL Server 構成マネージャを使用して、MDM Hub 環境で必要な TCP/IP ネットワークプロトコルを有効にします。

1. SQL Server 構成マネージャを起動します。
2. コンソールペインで、**[SQL Server ネットワークの構成]** を展開し、**[MSSQLSERVER のプロトコル]** をクリックします。
3. 詳細ペインで、**[TCP/IP]** を右クリックし、**[有効化]** をクリックします。
4. SQL Server サービスを再起動します。  
TCP/IP ネットワークプロトコルが有効になります。

## 手順 2. Microsoft SQL Server データファイルストアの作成

Microsoft SQL Server データファイルストアがない場合は、必ず作成します。MDM Hub マスタデータベースとオペレーショナル参照ストアを作成する際に、Microsoft SQL Server データファイルストアへのパスを指定する必要があります。

## 手順 3. ODBC ドライバのインストール

UNIX または Linux 上にアプリケーションサーバーをインストールする場合、データソースが作成できるようにするために、Microsoft SQL Server 用の ODBC ドライバをインストールします。

- ▶ オペレーティングシステム用の ODBC ドライバをダウンロードしてインストールします。
  - Linux の場合: Microsoft Web サイトから Microsoft SQL Server ODBC ドライバをダウンロードします。
  - UNIX の場合: unixODBC プロジェクトのホームページから UNIX ODBC ドライバをダウンロードします。

## 手順 4.ActiveVOS データベースの作成

ActiveVOS をインストールするには、ActiveVOS データベースを作成する必要があります。データベースを作成するには、create\_bpm スクリプトを実行します。

複数の MDM Hub マスタデータベースを作成する場合は、MDM Hub マスタデータベースごとに ActiveVOS データベースを作成します。

1. コマンドプロンプトを開き、次のディレクトリに移動します。

<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin

2. 表示されるプロンプトに回答します。

プロンプトでは、デフォルトのテキストが括弧内に表示されます。デフォルト値を使用して次のプロンプトに進むには、**Enter** キーを押します。

プロパティ	説明
Database Type	使用するデータベースのタイプ。Microsoft SQL Server の場合、MSSQL を指定します。データベースタイプは、MDM Hub マスタデータベースとオペレーショナル参照ストア用に選択したデータベースタイプと同じである必要があります。
ActiveVOS User Name	ActiveVOS サーバーの管理者ユーザーの名前。ユーザー名は、Microsoft SQL Server データベースの名前と同じです。
ActiveVOS User Password	管理者ユーザーのパスワード。
ActiveVOS Collation Name	ActiveVOS データベース照合の名前。 例えば、Latin1_General_CI_AS です。
ActiveVOS Database Host Name	データベースをホストするマシンの名前。
ActiveVOS Database Path	データベースの場所へのパス。
DBA User Name	SA ユーザーのユーザー名。
DBA Password	SA ユーザーアカウントのパスワード。

3. データベースを作成した後に、次のディレクトリの sip\_ant.log ファイルを確認します。

<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin

sip\_ant.log ファイルには、ActiveVOS データベースを作成するために sip\_ant スクリプトを実行したときに発生したすべてのエラーが記録されます。

# IBM Db2 データベースのセットアップ

MDM Hub マスターデータベースおよびオペレーショナル参照ストアを作成する前に、データベース環境を設定します。

データベース環境を設定するには、次のタスクを実行します。

1. IBM Db2 をインストールして設定します。
2. データベースとテーブルスペースを作成します。
3. データベースサーバー上にパッケージをバインドします。
4. ActiveVOS スキーマを作成します。

## 手順 1。IBM Db2 のインストールと設定

IBM Db2 のマニュアルに書かれた手順に従って、IBM Db2 をインストールして設定できます。

次の表に、IBM Db2 インスタンスを配置する各マシンで実行する必要がある IBM Db2 のインストールおよび設定タスクを示します。

タスク	説明
IBM Db2 のインストール	サポートされているバージョンの IBM Db2 をインストールします。
クライアントとユーティリティのインストール	MDM Hub との通信や MDM Hub プロセスの実行をする IBM Db2 クライアントおよびユーティリティソフトウェアをインストールします。 Hub サーバーまたはプロセスサーバーを実行する各マシンに、次のソフトウェアをインストールします。 <ul style="list-style-type: none"><li>- Db2 クライアント</li><li>- Db2 クライアント用の Db2 Java ユーティリティ</li></ul> IBM Db2 データベースを各 Db2 クライアントからカタログ化するようにします。
IBM Db2 ドライバの設定	IBM Db2 ドライバを設定するには、db2jcc.jar および db2jcc_license_cu.jar ドライバファイルをソースディレクトリからターゲットディレクトリにコピーします。 ソース: <IBM DB2 のインストールディレクトリ>/java ターゲット: <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/lib
MDM Hub スキーマユーザーの作成	次の MDM Hub スキーマにアクセスするユーザーを作成します。 <ul style="list-style-type: none"><li>- MDM Hub マスターデータベース</li><li>- オペレーショナル参照ストア</li></ul>

## 手順 2。データベースとテーブルスペースの作成

IBM Db2 のインストールと設定の後、データベースとテーブルスペースを作成して設定します。データベースインスタンスごとにデータベースを作成する必要があります。

**注:** 複数の MDM Hub マスタデータベースを作成する場合は、MDM Hub マスタデータベースごとに一意のテーブルスペースを作成します。

次の表に、MDM Hub スキーマに必要なテーブルスペースを示します。

テーブルスペース名	説明
CMX_DATA	オペレーショナル参照ストアスキーマ用のデフォルトのテーブルスペース。MDM Hub のメタデータおよびユーザーデータが含まれます。
CMX_INDX	MDM Hub で作成して使用するインデックスが含まれているテーブルスペース。
CMX_TEMP	MDM Hub で作成して使用する一時テーブルが含まれているテーブルスペース。
CMX_REPOS	オペレーショナル参照ストアのオブジェクトが含まれているテーブルスペース。
CMX_USER_TEMP	オペレーショナル一時テーブルが含まれる一時テーブルスペース。
CMX_SYS_TEMP	SQL 操作の一時テーブルスペース。

データベースとテーブルスペースを作成するには、以下のいずれかの手順を使用します。

- データベースとテーブルスペースの手動作成
- スクリプトを使用してデータベースとテーブルスペースを作成

## データベースとテーブルスペースの手動作成

データベースとテーブルスペースを手動で作成できます。互換性ベクトルをオンにして、UTF-8 TERRITORY US ロケールでデータベースを作成してください。

### Db2 環境変数および Db2 レジストリ変数の設定

データベースを手動で作成する場合は、MDM Hub に必要な Db2 環境変数と Db2 レジストリ変数を設定します。

次のコマンドを使用して、Db2 環境変数および Db2 レジストリ変数を設定します。

```
db2set DB2CODEPAGE=1208
db2set DB2_COMPATIBILITY_VECTOR=
db2set DB2_DEFERRED_PREPARE_SEMANTICS=YES
db2set DB2_RESTORE_GRANT_ADMIN_AUTHORITIES=ON
db2set DB2_HASH_JOIN=YES
db2set DB2_ANTIJOIN=YES
db2set DB2_INLIST_TO_NLJN=NO
db2set DB2_SELECTIVITY=ALL
db2set DB2_SKIPINSERTED=YES
db2set DB2_SKIPDELETED=YES
db2set DB2_EXTENDED_OPTIMIZATION=ON,ENHANCED_MULTIPLE_DISTINCT,IXOR,SNHD
db2set DB2NTNOCACHE=ON
db2set DB2_REDUCED_OPTIMIZATION=REDUCE_LOCKING
```

### データベースインスタンスのデータベース管理者設定の設定

データベースインスタンスのデータベース管理者設定を最適化する必要があります。

次のコマンドを使用して、データベース管理者設定を最適化します。

```
db2 update dbm cfg using MON_HEAP_SZ AUTOMATIC
db2 update dbm cfg using JAVA_HEAP_SZ 2048
db2 update dbm cfg using AGENT_STACK_SZ 256
db2 update dbm cfg using SHEAPTHRES 0
db2 update dbm cfg using INTRA_PARALLEL YES
```

**注:** このコマンドで指定される値は、MDM Hub の最小要件です。

## データベース設定パラメータの設定

データベースの設定パラメータを設定します。

次のコマンドを使用して、データベースの設定パラメータを設定します。

```
db2 update db cfg using LOCKLIST AUTOMATIC
db2 update db cfg using MAXLOCKS AUTOMATIC
db2 update db cfg using PKCACHESZ 128000
db2 update db cfg using DBHEAP AUTOMATIC
db2 update db cfg using CATALOGCACHE_SZ 25000
db2 update db cfg using LOGBUFSZ 4096
db2 update db cfg using UTIL_HEAP_SZ 50000
db2 update db cfg using BUFFPAGE 250
db2 update db cfg using STMTHEAP AUTOMATIC
db2 update db cfg using APPLHEAPSZ AUTOMATIC
db2 update db cfg using APPL_MEMORY AUTOMATIC
db2 update db cfg using STAT_HEAP_SZ AUTOMATIC
db2 update db cfg using LOGFILSIZ 128000
db2 update db cfg using LOGPRIMARY 10
db2 update db cfg using LOGSECOND 200
db2 update db cfg using auto_reval deferred_force
db2 update db cfg using decflt_rounding round_half_up
db2 update db cfg using SHEAPTHRES_SHR AUTOMATIC
db2 update db cfg using DFT_DEGREE 1
```

**注:** このコマンドで指定される値は、MDM Hub の最小要件です。

## SYSIBMADM モジュールへの特権の付与

モジュール UTL\_DIR、UTL\_FILE、および DBMS\_SQL SYSIBMADM に特権を付与する必要があります。

次のコマンドを使用して、モジュールに特権を付与します。

```
GRANT EXECUTE ON MODULE SYSIBMADM.UTL_DIR TO PUBLIC WITH GRANT OPTION
GRANT EXECUTE ON MODULE SYSIBMADM.UTL_FILE TO PUBLIC WITH GRANT OPTION
GRANT EXECUTE ON MODULE SYSIBMADM.DBMS_SQL TO PUBLIC WITH GRANT OPTION
```

## データベースマスター用のバッファプールの定義

バッファプール REPOS\_POOL および CMX\_POOL を定義します。

次のコマンドを使用して、バッファプールを定義します。

```
CREATE BUFFERPOOL REPOS_POOL IMMEDIATE SIZE 1500 PAGESIZE 32 K
CREATE BUFFERPOOL CMX_POOL IMMEDIATE SIZE 3000 PAGESIZE 32 K
```

## テーブルスペースの作成

MDM Hub スキーマに必要なテーブルスペースを作成する必要があります。

次のリストの順序で、テーブルスペースを作成します。

1. CMX\_DATA
2. CMX\_INDX
3. CMX\_REPOS
4. CMX\_TEMP
5. CMX\_USER\_TEMP
6. CMX\_SYS\_TEMP

次の文を使用して、MDM Hub スキーマ用のテーブルスペースを作成します。

```
CREATE TABLESPACE CMX_DATA PAGESIZE 32 K
  MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\CMX_DATA\cmx_data01.dat' 500 M )
  EXTENTSIZE 16
  AUTORESIZE YES
  OVERHEAD 10.5
  PREFETCHSIZE 16
  BUFFERPOOL CMX_POOL
```

```
CREATE TABLESPACE CMX_INDX PAGESIZE 32 K
  MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\CMX_INDX\cmx_indx01.dat' 500 M )
  EXTENTSIZE 16
  AUTORESIZE YES
  OVERHEAD 10.5
  PREFETCHSIZE 16
  BUFFERPOOL CMX_POOL
```

```
CREATE TABLESPACE CMX_REPOS PAGESIZE 32 K
  MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\CMX_REPOS\cmx_repos01.dat' 500 M )
  EXTENTSIZE 16
  AUTORESIZE YES
  OVERHEAD 10.5
  PREFETCHSIZE 16
  BUFFERPOOL REPOS_POOL
```

```
CREATE TABLESPACE CMX_TEMP PAGESIZE 32 K
  MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\CMX_TEMP\cmx_temp01.dat' 500 M )
  EXTENTSIZE 16
  AUTORESIZE YES
  OVERHEAD 10.5
  PREFETCHSIZE 16
  BUFFERPOOL CMX_POOL
```

```
CREATE USER TEMPORARY TABLESPACE CMX_USER_TEMP PAGESIZE 32 K
  MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\USER_TEMP\cmx_user_temp01.dat' 500 M )
  EXTENTSIZE 16
  AUTORESIZE YES
  OVERHEAD 10.5
  PREFETCHSIZE 16
  BUFFERPOOL CMX_POOL
```

```
CREATE SYSTEM TEMPORARY TABLESPACE CMX_SYS_TEMP PAGESIZE 32 K
  MANAGED BY DATABASE USING ( FILE '<Db2 storage path>\SYSTEM_TEMP\cmx_sys_temp01.dat' 500 M )
  EXTENTSIZE 16
  AUTORESIZE YES
  OVERHEAD 10.5
  PREFETCHSIZE 16
  BUFFERPOOL CMX_POOL
```

必要に応じて、削除されたテーブルリカバリ機能を有効にしてテーブルスペースを作成するには、CREATE TABLESPACE 文に次の句を追加します。

```
DROPPED TABLE RECOVERY ON
```

## スクリプトを使用してデータベースとテーブルスペースを作成

MDM Hub ディストリビューションには、データベースを作成するスクリプトおよび関連付けられたテーブルスペースが含まれています。スクリプトを実行するには、Db2 データディレクトリに対する書き込みおよび実行権限のある管理特権が必要です。

UNIX では、データベースを作成する前に、database.properties ファイル内の db2.storage.path プロパティを正しいデータベースストレージへのパスで更新します。database.properties ファイルは以下のディレクトリにあります。

<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin/db2

1. コマンドプロンプトを開き、次のディレクトリに移動します。

<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin

2. データベースを作成するには、次のコマンドを実行します。

UNIX の場合: ./sip\_ant.sh create\_db

Windows の場合: sip\_ant.bat create\_db

3. 以下の表に説明する内容に従ってプロンプトに回答します。

プロンプト	説明
データベースタイプを入力 (ORACLE、MSSQL、DB2)	データベースタイプ。DB2 を指定する。
データベースインスタンス名を入力 [db2]	データベースインスタンスの名前。デフォルトは db2。
データベース名を入力 [SIP97]	データベースの名前。デフォルトは SIP97。
データベースストレージへのパスを 入力 [C:\DB2DATA]	データベースを格納する必要があるディレクトリへのパス。デ フォルトは C:\DB2DATA。 <b>注:</b> UNIX では、デフォルト値を受け入れます。 database.properties ファイルで指定するデータベースストレ ージへのパスが使用されます。
DBA ユーザー名を入力 [DB2ADMIN]	管理者ユーザーの名前。デフォルトは DB2ADMIN です。
DBA のパスワードを入力	管理者ユーザーのパスワード。

このスクリプトでは、データベースおよび次のテーブルスペースが作成されます。

- CMX\_DATA
- CMX\_INDX
- CMX\_TEMP
- CMX\_REPOS
- CMX\_USER\_TEMP
- CMX\_SYS\_TEMP

データベースが正常に作成されていることを確認するには、<MDM Hub distribution directory: MDM Hub ディストリ  
ビューションディレクトリ>/database/bin ディレクトリの sip\_ant.log ファイルを確認します。

## 手順 3. データベースサーバーへのパッケージのバインド

IBM Db2 クライアントがデータベースサーバーに接続して Db2 コマンドを実行できるように、データベースサ  
ーバーにパッケージをバインドします。

1. IBM Db2 コマンドウィンドウを開き、次のディレクトリに変更します。

<IBM Db2 のインストールディレクトリ>/SQLLIB/bnd

2. 次のコマンドを実行してデータベースに接続します。

db2 connect to <database name: データベース名> user <database user: データベースユーザー> using <database  
user password: データベースユーザーのパスワード>

**注:** データベースユーザーがバインド権限を持っている必要があります。

3. 次のバインドコマンドを実行します。

```
db2 bind @db2cli.lst blocking all grant public sqlerror continue CLIPKG 10
```

必要なパッケージがデータベースサーバーにバインドされます。

## 手順 4.ActiveVOS スキーマの作成

ActiveVOS をインストールするには、ActiveVOS スキーマを作成する必要があります。スキーマを作成するには、create\_bpm スクリプトを実行します。

複数の MDM Hub マスタデータベースを作成する場合は、MDM Hub マスタデータベースごとに ActiveVOS スキーマを作成します。

1. コマンドプロンプトを開き、次のディレクトリに移動します。

<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin

2. 次のコマンドを実行します。

UNIX の場合: ./sip\_ant.sh create\_bpm

Windows の場合: sip\_ant.bat create\_bpm

3. 表示されるプロンプトに回答します。

プロンプトでは、デフォルトのテキストが括弧内に表示されます。デフォルト値を使用して次のプロンプトに進むには、**Enter** キーを押します。

プロパティ	説明
Database Type	使用するデータベースのタイプ。IBM Db2 の場合、DB2 と指定します。データベースタイプは、MDM Hub マスタデータベースとオペレーショナル参照ストア用に選択したデータベースタイプと同じである必要があります。
ActiveVOS Database Host Name	データベースをホストするマシンの名前。
ActiveVOS Database TCP/IP Port	データベースリスナーが使用するポート番号。
ActiveVOS Database Name	データベースの名前。
ActiveVOS Database Schema/ User Name	ActiveVOS サーバーの管理者ユーザーの名前。
ActiveVOS User Password	管理者ユーザーのパスワード。
DBA User Name	データベース管理者ユーザーのユーザー名。
DBA Password	管理者ユーザーのパスワード。
ActiveVOS Tablespace Name	MDM ワークフローに関連するレコードを格納するテーブルスペースの名前。

4. スキーマを作成したら、次のディレクトリの sip\_ant.log ファイルを確認します。

<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin

sip\_ant.log ファイルには、ActiveVOS のスキーマを作成するために sip\_ant スクリプトを実行したときに発生したすべてのエラーが記録されます。

## 第 4 章

# アプリケーションサーバーのタスク

この章では、以下の項目について説明します。

- [アプリケーションサーバーのタスクの概要, 49 ページ](#)
- [JBoss のセットアップ, 49 ページ](#)
- [Oracle WebLogic のセットアップ, 60 ページ](#)
- [追加の Oracle WebLogic の設定, 66 ページ](#)
- [IBM WebSphere のセットアップ, 70 ページ](#)
- [追加の IBM WebSphere の設定, 77 ページ](#)

## アプリケーションサーバーのタスクの概要

Multidomain MDM と連動するように、アプリケーションサーバーを設定します。アプリケーションサーバーの手順に従います。

クリーンアップグレードを実行する場合、Multidomain MDM の新規インストールの実行時と同様に、アプリケーションサーバーのすべての手順を実行します。

アプリケーションサーバーが引き続きサポートされているインプレースアップグレードを実行する場合、このバージョンの Multidomain MDM と連動するように、アプリケーションサーバーが設定されていることを確認します。

## JBoss のセットアップ

MDM Hub は、JBoss クラスタ環境またはスタンドアロンの JBoss インスタンスにインストールできます。JBoss のマニュアルに従って、JBoss 環境をインストールし、設定します。MDM Hub を JBoss クラスタ環境またはスタンドアロンの JBoss インスタンスにインストールするかどうかにかかわらず、JBoss スタンドアロン設定をインストールし、設定の完全なプロファイルを使用します。

JBoss のクラスタは、1 つまたは複数のマシン上にある 1 つまたは複数のノードで構成されています。クラスタノードを設けるすべてのマシンに JBoss をインストールして設定します。クラスタ環境では、JBoss がインストールされたディレクトリの構造がすべてのクラスタノードで確実に同じになるようにしてください。

**注:** データベースサーバーと同じタイムゾーンのアプリケーションサーバーをインストールします。

## Java 仮想マシンの設定

Java 仮想マシン（JVM）を設定するには、JAVA\_OPTS 環境変数を使用して、Java のオプションを設定します。

Java のオプションは、次のファイルで設定できます。

UNIX の場合:<JBoss のインストールディレクトリ>/bin/standalone.conf

Windows の場合:<JBoss のインストールディレクトリ>\bin\standalone.conf.bat

次の表で、Java のオプションの設定を説明します。

Java のオプション	説明
-server	起動速度は遅くなりますが、後続の操作は高速になります。
-De360.connection.channel -De360.mdm.host -De360.mdm.port	<p>アプリケーションサーバーの通信プロトコル、ホストおよびポート。</p> <p>4447 以外の JBoss ポートに MDM Hub アプリケーションをデプロイするには、次の Java オプションを設定します。</p> <ul style="list-style-type: none"><li>-De360.connection.channel。使用する通信プロトコルに設定します。有効な値は HTTP および HTTPS です。デフォルトは HTTP です。</li><li>-De360.mdm.host JBoss ホストの IP アドレスを設定します。</li></ul> <p>環境で HTTPS 通信プロトコルを使用していて、セキュリティ証明書が完全修飾ドメイン名（FQDN）に発行されている場合、FQDN に設定します。</p> <ul style="list-style-type: none"><li>-De360.mdm.port ビジネスエンティティを持つ Data Director が MDM Hub と通信するリスナポートを指定します。デフォルトは 8080 です。</li></ul> <p>これらのパラメータを設定しない場合、エンティティ 360 フレームワークに基づく Data Director の画面が期待どおりに動作しない可能性があります。</p>
-Didd.mdm.host -Didd.mdm.port -Didd.protocol	<p>サブジェクト領域を使用する Data Director に必要です。</p> <p>サブジェクト領域を使用する Data Director をデプロイする場合、次の Java オプションを設定します。</p> <ul style="list-style-type: none"><li>-Didd.mdm.host。アプリケーションサーバーホストのホスト名または IP アドレスに設定します。</li><li>-Didd.mdm.port。サーバーの初期化中にサブジェクト領域アプリケーションとともに Data Director によって内部的に使用される必須プロパティ。アプリケーション用の JVM で使用する HTTP または HTTPS リスナポートを指定します。デフォルトは 8080 です。</li><li>-Didd.protocol。サーバーの初期化中にサブジェクト領域のアプリケーションをデプロイするために使用される必須プロパティ。使用する通信プロトコルが HTTP か HTTPS かを指定します。デフォルトは HTTP です。</li></ul>

Java のオプション	説明
- Dio.undertow.legacy.cookie.ALLOW_HTTP_SEPARATORS_IN_V0	JBoss バージョン 7.3 を使用している場合は必須プロパティ。プロパティをシステムプロパティとして設定し、Cookie 値の切り捨てによって必要な資格情報が欠落したビジネスエンティティサービス要求の REST API の認証エラーを防ぐには、true に設定します。デフォルトは false です。JBoss バージョン 7.1 または 7.2 ではこのプロパティを設定する必要はありません。
-Ddb2.jcc.charsetDecoderEncoder	MDM Hub サンプルオペレーショナル参照ストアを使用する必要があります。JDBC ドライバで、UTF-8 文字列でないバイトシーケンスの代わりに、Unicode 置換文字 (U+FFFD) を返すことができます。3 に設定します。
-Djava.net.preferIPv4Stack	Java で Internet Protocol バージョン 4 (IPv4) を使用するかどうかを指定します。オペレーティングシステムで Internet Protocol バージョン 6 (IPv6) を使用する場合は、プロパティを true に設定します。true に設定すると、アプリケーションは IPv4 ホストとのみ通信します。デフォルトは false です。
-Djava.net.preferIPv6Addresses	Java で Internet Protocol バージョン 6 (IPv6) を使用するかどうかを指定します。ホストに IPv4 アドレスと IPv6 アドレスの両方があり、オペレーティングシステムが IPv6 を使用している場合は、後方互換性のために IPv4 を使用します。アプリケーションが IPv6 ホストとのみ通信する場合は、true に設定します。デフォルトは false です。
-Djavax.net.ssl.trustStore -Djavax.net.ssl.trustStorePassword	プロセスサーバーで HTTPS ポートを使用する場合に必要です。 -Djavax.net.ssl.trustStore Java オプションは、クライアント証明書の検証に使用するトラストストアファイルへのパスを指定します。 -Djavax.net.ssl.trustStorePassword Java オプションは、トラストストアファイルにアクセスするためのパスワードを指定します。
-Djava.security.egd	Linux 環境での Data Director の起動時間を短縮します。値を file:/dev/./urandom に設定します。
-Djboss.as.management.blocking.timeout	JBoss がデプロイされるまで待機する時間 (秒単位)。JBoss を確実に起動するには、この値を 5000 に設定します。この時間は、ご使用の環境に応じて調整します。デフォルトは 300 です。 このパラメータを設定しない場合は、JBoss デプロイメントのタイムアウトが発生することがあります。
-Djgroups.bind_addr	JGroup がメッセージの送受信を行う必要があるインターフェース。 マルチノード環境またはクラスタ環境が必要です。各ノードが専用のネットワークインターフェースにバインドされていることを確認します。

Java のオプション	説明
-DFrameworksLogConfigurationPath	log4j.xml ファイルのパス。 次のディレクトリが作成された後、ディレクトリ内にある log4j.xml ファイルを見つけます。 - /hub/cleanse/conf - /hub/server/conf
-Dmdm.node.groupid	MDM Hub 実装で Java 仮想マシンのグループ ID を指定します。Hub サーバーとプロセスサーバーを論理的にグループ化する場合にのみ必要です。
-DUseESLegacyFqSearch	フィールド検索で、ビジネスエンティティタイプの子ノード内から完全一致を返すかどうかを指定します。複数のフィールドでフィールド検索を実行する場合にのみ適用できます。複数のクエリフィールドが子レベルにある場合、検索時に同じ子ノードの検索値を含むレコードが返されるようにするかどうかを指定します。true に設定すると、異なる子ノードの子レベルのクエリフィールドに一致する可能性のあるレコードが返されます。デフォルトは false です。
-Dfile.encoding -Dorg.apache.catalina.connector.URI_ENCODING	Informatica Data Director を使用する場合、および REST API を使用してレコードを検索する場合に必要です。 UTF-8 文字を含むレコードを確実に検索および保存できるようにするには、両方の Java のオプションを UTF-8 に設定します。
-Dorg.apache.coyote.http11.Http11Protocol.MAX_HEADER_SIZE	HTTP ヘッダーの最大サイズ（バイト単位）。ヘッダーのサイズが小さいと、検索要求が失敗する可能性があります。16384 に設定します。
-Dtask.pageSize=<タスクの最大数>	各要求に対して取得される ActiveVOS タスクの最大数を指定します。デフォルトは 5000 です。環境に多数のタスクがある場合は、この数値を増やします。
-Dstricttransportsecurity.flag	HTTP 要求を使用して Data Director にアクセスするすべての試行を、代わりに HTTPS 要求に変換する必要があるかどうかを指定します。true に設定します。
-Xms	初期ヒープサイズ 2048m に設定します。
-Xmx	最大 JVM ヒープサイズ。6 GB 以上に設定します。 例えば、-Xmx を 6144m に設定するには、次の JAVA_OPTIONS 環境変数設定を使用します。 <pre>set "JAVA_OPTIONS=-server ... -Xmx6144m"</pre>
-XX:+UseCodeCacheFlushing	コードキャッシュがいっぱいになったときに、JVM がコンパイルされたコードを破棄するかどうかを指定します。

Java のオプション	説明
-XX:ReservedCodeCacheSize	JIT コードのキャッシュサイズ。MDM Hub 環境のパフォーマンスを高めるには、512m に設定します。
-XX:MaxMetaspaceSize	最大メタスペースサイズ: JVM がメモリを使い切るのを防ぐには、1G に設定します。

## Java 仮想マシンの追加の設定

MDM インストーラを実行する前、または JBoss を設定する前に、Java 仮想マシンの追加の設定を行います。

次のファイルで、ActiveVOS を使用した RedHat JBoss EAP に必要な追加の Java オプションを設定します。

UNIX の場合: <JBoss インストールディレクトリ>/bin/standalone.conf.sh

- ActiveVOS ログインページを表示するには、次のプロパティを追加します。  
`JAVA_OPTS="$JAVA_OPTS -Dae.mail.default.charset=US-ASCII -Dae.web.filter.https.force=false --add-opens=java.xml/com.sun.org.apache.xerces.internal.dom=ALL-UNNAMED"`
- プロビジョニングサイトのログイン認証を有効にするには、次のプロパティを追加します。  
`JBoss_JAVA_SIZING="-Xms1G -Xmx5G -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=1G --add-exports=java.xml/com.sun.org.apache.xml.internal.serialize=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED --add-opens=java.base/jdk.internal.loader=ALL-UNNAMED"`
- サーバーが長時間実行されているときに JBoss サーバーのログファイルが大きくなることを防ぎ、SEED Operational Reference Store (ORS) の階層が正常に有効になるようにするには、次のプロパティを追加します。  
`JAVA_OPTS="$JAVA_OPTS -Djdk.serialFilter=*`
- Generate Operational Reference Store (ORS) API と JMS スキーマイベントマネージャを実装するには、次のプロパティを追加します。  
`JAVA_OPTS="$JAVA_OPTS --add-exports=jdk.compiler/com.sun.tools.javac.main=ALL-UNNAMED"`
- 階層ワークフローでの承認後にワークフローが進行するようにするには、次のプロパティを追加します。  
`JAVA_OPTS="$JAVA_OPTS -Dio.undertow.legacy.cookie.ALLOW_HTTP_SEPARATORS_IN_V0=true"`

Windows の場合: <JBoss インストールディレクトリ>\bin\standalone.conf.bat

- ActiveVOS ログインページを表示するには、次のプロパティを追加します。  
`set "JAVA_OPTS=%JAVA_OPTS% -Dae.mail.default.charset=US-ASCII -Dae.web.filter.https.force=false --add-opens=java.xml/com.sun.org.apache.xerces.internal.dom=ALL-UNNAMED"`
- プロビジョニングサイトのログイン認証を有効にするには、次のプロパティを追加します。  
`set "JBoss_JAVA_SIZING=-Xms1G -Xmx5G -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=1G --add-exports=java.xml/com.sun.org.apache.xml.internal.serialize=ALL-UNNAMED --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.net=ALL-UNNAMED --add-opens=java.base/jdk.internal.loader=ALL-UNNAMED"`
- サーバーが長時間実行されているときに JBoss サーバーのログファイルが大きくなることを防ぎ、SEED Operational Reference Store (ORS) の階層が正常に有効になるようにするには、次のプロパティを追加します。  
`set "JAVA_OPTS=%JAVA_OPTS% -Djdk.serialFilter=*`
- Generate Operational Reference Store (ORS) API と JMS スキーマイベントマネージャを実装するには、次のプロパティを追加します。  
`set "JAVA_OPTS=%JAVA_OPTS% --add-exports=jdk.compiler/com.sun.tools.javac.main=ALL-UNNAMED"`

- 階層ワークフローでの承認後にワークフローが進行するようにするには、次のプロパティを追加します。  
set "JAVA\_OPTS=%JAVA\_OPTS% -Dio.undertow.legacy.cookie.ALLOW\_HTTP\_SEPARATORS\_IN\_V0=true"

## Java 仮想マシンの論理的なグループ化の例

Java 仮想マシン（JVM）をグループ化することにより、Hub サーバーとプロセスサーバーの論理グループを取得できます。Hub サーバーアプリケーションとプロセスサーバーアプリケーションを論理 JVM グループにデプロイすると、Hub サーバーアプリケーションとプロセスサーバーアプリケーション間のすべての通信がグループ内にとどまります。JVM をグループ化するには、MDM Hub 環境の各 JVM にグループ ID を割り当てます。

**注:** プロセスサーバーのグループ化は、クレンジングと一致プロセスのみに適用されます。論理グループは MDM Hub の内部サーバーキャッシュには適用されません。

次の表は、論理 JVM グループの例を示しています。

JVM グループ	JVM	Hub サーバー	プロセスサーバー
Group1	JVM1	○	はい
Group1	JVM4	-	○
Group2	JVM2	○	はい
Group3	JVM3	-	○

JVM1 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group1
```

JVM2 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group2
```

JVM3 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group3
```

JVM4 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group1
```

JVM を設定し、Hub サーバーとプロセスサーバーをデプロイすると、グループには次の特性があります。

- Group1 には 2 つのプロセスサーバーがあり、Group2 には 1 つのプロセスサーバーがあり、Group3 には 1 つのプロセスサーバーがあります。
- すべてのクレンジングとバッチ呼び出しは、検索を除き、自分のグループにとどまります。例えば、Group1 の Hub サーバーでのリアルタイム呼び出しは、Group1 プロセスサーバー（JVM1 および JVM4）にのみ影響します。

## 完全なプロファイルのサーバープロパティの設定

スタンドアロンモードの完全なプロファイルのサーバープロパティは、standalone-full.xml ファイルで設定します。ファイルは以下のディレクトリにあります。<JBoss のインストールディレクトリ>/standalone/configuration

サーバーのプロパティを設定するには、JBoss コマンドラインインタフェース (CLI) から、設定用のコマンドを実行します。

JBoss CLI を使用するには、次の手順を実行します。

1. 次のディレクトリに移動します。<JBoss のインストールディレクトリ>/bin
2. JBoss CLI を起動するには、次の手順を実行します。  
UNIX の場合：jboss-cli.sh  
Windows の場合：jboss-cli.bat
3. サーバーに接続するには、次のコマンドを実行します。  
connect

サーバーのプロパティ設定の詳細については、JBoss のマニュアルを参照してください。

## トランザクションタイムアウト

トランザクションタイムアウトプロパティは、MDM Hub トランザクションが完了するまでの待機時間を秒単位で指定します。この値は、使用している環境に基づいて設定します。

### コマンド:

```
/subsystem=transactions:write-attribute(name=default-timeout,value=<timeout in seconds>)
```

### 設定例:

```
<subsystem xmlns="urn:jboss:domain:transactions:4.0">
  ...
  <coordinator-environment default-timeout="3600"/>
</subsystem>
```

## 最大ポストサイズ

最大ポストサイズプロパティは、アップロードするファイルの最大サイズをバイト単位で設定します。Data Director アプリケーションにアタッチするファイルのサイズ制限の値を設定します。

値を 20000000 以上に設定します。デフォルトは 10000000 です。

**注:** MDM Hub をインストールした後、cmxserver.properties ファイルの Hub Server プロパティの cmx.file.max\_file\_size\_mb に同じ値を設定します。

### コマンド:

```
/subsystem=undertow/server=default-server/<listener type>=:write-attribute(name=max-post-size,value=<maximum file size in bytes>)
```

### 設定例:

```
<subsystem xmlns="urn:jboss:domain:undertow:4.0">
  ...
  <server name="default-server">
    <http-listener name="default" socket-binding="http" redirect-socket="https" enable-http2="true" max-post-size="20000000"/>
    <https-listener name="https" socket-binding="https" security-realm="ApplicationRealm" enable-http2="true" max-post-size="20000000"/>
    ...
  </host>
</server>
  ...
</subsystem>
```

## リモート接続コネクタ

リモート接続コネクタプロパティは、リモート接続コネクタポートとソケットバインディングを設定します。

リモートマシンから Hub コンソールにログインするには、ポートを 4447 に設定し、ソケットバインディングをリモート接続に設定します。

**注:** デフォルトでは、リモーティングコネクタセキュリティは無効です。MDM Hub のリモーティングコネクタセキュリティを設定する場合は、サポートされている Elytron セキュリティフレームワークのみを設定していることを確認してください。

#### コマンド:

```
/socket-binding-group=standard-sockets/socket-binding=remoting:add(port=4447)
```

```
/subsystem=remoting/connector=remoting-connector:add(socket-binding=remoting)
```

#### リモーティングコネクタポートの設定例:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-offset="${jboss.socket.binding.port-offset:0}">
    ...
    <socket-binding name="remoting" port="4447"/>
    ...
</socket-binding-group>
```

#### リモーティングコネクタソケットバインディングの設定例:

```
<subsystem xmlns="urn:jboss:domain:remoting:4.0">
    ...
    <connector name="remoting-connector" socket-binding="remoting"/>
    ...
</subsystem>
```

## ActiveVOS コンソール管理ユーザーの作成

ActiveVOS を使用する場合、アプリケーションサーバーコンテナで abAdmin ロールを使用して ActiveVOS コンソールの管理者ユーザーを作成します。ActiveVOS を使用する場合、abAdmin ロールを使用して ActiveVOS コンソールの管理者ユーザーを作成します。管理者ユーザーを作成しない場合は、Hub サーバーのデプロイメントに失敗します。Hub サーバーインストーラから ActiveVOS コンソールの管理者ユーザーの資格情報を入力するように要求されたら、ActiveVOS コンソールの管理者ユーザー名およびパスワードを使用します。

1. 次のディレクトリに変更します。  
    <JBoss のインストールディレクトリ>/bin
2. add-user ユーティリティを実行するには、次のスクリプトを使用します。  
    UNIX の場合: add-user.sh  
    Windows の場合: add-user.bat
3. 表示されるプロンプトに回答します。  
    次の表に、各プロンプトで指定する値の説明を示します。

プロンプト	指定する値
どのタイプのユーザーを追加しますか? a) 管理ユーザーまたは b) アプリケーションユーザー	アプリケーションユーザーを選択するには、b を入力します。
レルム (ApplicationRealm)	レルム名。standalone-full.xml ファイルに追加した login-module で指定したレルム名を入力します。
ユーザー名	ActiveVOS コンソールの管理者名。

プロンプト	指定する値
パスワード	JBoss のパスワード標準に準拠しているパスワード。
このユーザーをどのロールにしますか?	abAdmin。
レルム<realm name>のユーザー<user name>を追加しようとしています。よろしいですか?	ユーザーを追加するには、yes を入力します。
この新しいユーザーは、ある AS プロセスを別の AS プロセスに接続するために使用されますか?	はい。

4. WebSphere コンソールにログインして、ActiveVOS コンソール管理者ユーザーを作成します。  
**注:** ActiveVOS コンソールのユーザーは、インストール後またはアップグレード後のプロセスで postInstallSetup または patchInstallSetup スクリプトを実行するときに abAdmin ロールにマップされます。
5. WebLogic コンソールにログインします。
6. abAdmin ロールを作成します。
7. ActiveVOS コンソール管理ユーザーを作成します。
8. 管理者ユーザーを abAdmin ロールに割り当てます。

## タイムアウトプロパティの設定

Operational Reference Store (ORS) 固有の API 生成および Java Message Service (JMS) スキーマ生成中に発生する可能性のある API タイムアウトエラーに対処するには、standalone-full.xml ファイルでタイムアウトプロパティを設定します。このファイルには次のディレクトリからアクセスできます: <JBoss のインストールディレクトリ>/standalone/configuration。サーバーのプロパティを設定するには、JBoss コマンドラインインタフェース (CLI) から設定の関連コマンドを実行します。

1. テキストエディタを使用して standalone-full.xml を開きます。
  2. <security-domain>タグを見つけます。
  3. 次のセクションに示すように、read-timeout 属性値と write-timeout 属性の値を更新します。
- ```
<server name="default-server">
  <http-listener name="default" socket-binding="http" max-post-size="50000000" redirect-socket="https"
enable-http2="true" read-timeout="180000" write-timeout="180000"/>
  <https-listener name="https" socket-binding="https" ssl-context="applicationSSC" enable-http2="true"/>
  <host name="default-host" alias="localhost">
    <location name="/" handler="welcome-content"/>
    <http-invoker http-authentication-factory="application-http-authentication"/>
  </host>
</server>
```

## ActiveVOS を使用した最新の JBoss バージョンのサポートの設定

ActiveVOS を使用した RedHat JBoss EAP 8 での Java Standard Edition 17 のサポートのセキュリティ設定の変更を手動で設定する必要があります。

### JBoss 環境での信頼されたユーザーの作成

ActiveVOS ワークフローエンジンを使用するには、アプリケーションサーバーで abTrust、abServiceConsumer、および abTaskClient のロールを持つ信頼されたユーザーを作成します。信頼されたユ

ーザーは、MDM Hub と ActiveVOS で信頼されます。このユーザーが MDM Hub と ActiveVOS 間の通信の安全性を高めます。

この信頼されたユーザーは、Hub コンソールの ActiveVOS ワークフローアダプタユーザーと同じユーザーです。信頼されたユーザーの名前は、アプリケーションサーバーの管理者ユーザーと同じ名前にすることはできません。

1. コマンドプロンプトから add-user.bat を実行します。

- UNIX の場合: <JBoss install director>/bin/add-user.sh
- Windows の場合: <JBoss install director>\bin\add-user.bat

2. 表示されるプロンプトに回答します。

**注:** プロンプトでは、デフォルト値が括弧内に表示されます。デフォルト値を使用して次のプロンプトに進むには、**Enter** キーを押します。

**どのタイプのユーザーを追加しますか? a) 管理ユーザーまたは b) アプリケーションユーザー**

b を入力してアプリケーションユーザーを選択します。

**レルム (ApplicationRealm)**

standalone-full.xml ファイルに追加した login-module で指定したのと同じレルム名を入力します。

**ユーザー名**

信頼されたユーザーのユーザー名を入力します。

**パスワード**

JBoss のパスワード標準に準拠しているパスワードを入力します。

**このユーザーをどのロールにしますか?**

abTrust, abServiceConsumer, abTaskClient を入力します。

**レルム<realmname>のユーザー<username>を追加しようとしています。よろしいですか?**

ユーザーを追加するには、yes を入力します。

**この新しいユーザーは、ある AS プロセスを別の AS プロセスに接続するために使用されますか?**

yes を入力します。

3. アプリケーションサーバーを再起動します。

## RedHat JBoss EAP 8.x での Java Standard Edition 17 のサポートのためのアプリケーションサーバーの設定

RedHat JBoss EAP 8.x での Java Standard Edition 17 のサポートについて、次のセキュリティ設定の変更を設定します。

1. テキストエディタを使用して standalone-full.xml を開きます。

2. <security-domain>タグを見つけます。

3. 次の行を<security-domain>タグの直下に追加します。

```
<security-domain name="ActiveVOS" default-realm="ActiveVOSSecurityRealm"
    permission-mapper="default-permission-mapper">
    <realm name="ActiveVOSSecurityRealm" role-decoder="groups-to-roles"/>
    <realm name="local"/>
</security-domain>
```

4. 次の ActiveVOSSecurityRealm プロパティを<security-realms>タグに追加します。

```
<properties-realm name="ActiveVOSSecurityRealm">
    <users-properties path="applicationUsers.properties" relative-
```

```

to="jboss.server.config.dir"/>
<groups-properties path="applicationRoles.properties" relative-
to="jboss.server.config.dir"/>
</properties-realm>

```

5. 次の HTTP 認証設定を http-authentication-factory タグに追加します。

```

<http-authentication-factory name="activevos-http-authentication" security-domain="ActiveVOS"
http-server-mechanism-factory="global">
  <mechanism-configuration>
    <mechanism mechanism-name="BASIC">
      <mechanism-realm realm-name="ActiveVOSSecurityRealm"/>
    </mechanism>
    <mechanism mechanism-name="FORM">
      <mechanism-realm realm-name="ActiveVOSSecurityRealm"/>
    </mechanism>
    <mechanism mechanism-name="CLIENT_CERT">
      <mechanism-realm realm-name="ActiveVOSSecurityRealm"/>
    </mechanism>
  </mechanism-configuration>
</http-authentication-factory>

```

6. 次の例に示すように、urn:jboss:domain:undertow:12.0 サブシステムタグ内の default-security-domain の値を ActiveVOS に変更します。

```

<subsystem xmlns="urn:jboss:domain:undertow:12.0" default-server="default-server"
default-virtual-host="default-host"
default-servlet-container="default"
default-security-domain="ActiveVOS"
statistics-enabled="${wildfly.undertow.statistics-enabled:${wildfly.statistics-
enabled:false}}">

```

7. applicationUsers.properties ファイルと applicationRoles.properties ファイルを<jboss-server-path>\standalone\configuration ディレクトリに追加します。

## JBoss の起動

Hub サーバーと Process サーバーをインストールする前に、JBoss アプリケーションサーバーを起動します。お使いの環境に基づいて、スタンドアロンの JBoss インスタンスか JBoss クラスターノードのどちらかを開始します。

### スタンドアロンの JBoss インスタンスの起動

スタンドアロンの JBoss インスタンスを使用する場合は、MDM Hub コンポーネントをインストールする各インスタンスを開始します。

1. 次のディレクトリに移動します。

```
<JBoss のインストールディレクトリ>/bin
```

2. JBoss インスタンスを開始するには、次のコマンドを実行します。

UNIX の場合: standalone.sh -c standalone-full.xml -b 0.0.0.0 -Djboss.as.management.blocking.timeout=5000

Windows の場合: standalone.bat -c standalone-full.xml -b 0.0.0.0 -Djboss.as.management.blocking.timeout=5000

JBoss は利用可能なネットワークインタフェースで起動し、/etc/hosts ディレクトリ内のホストファイルに定義された現在のホストをリスンします。お使いの環境に基づいて、タイムアウト間隔を調整します。

3. 同じマシンに複数の JBoss インスタンスがある場合、2 番目以降の JBoss インスタンスを開始するには、次の引数をスタートアップコマンドに追加します。

```
-Djboss.socket.binding.port-offset=<port offset range such as, 0,100,200,...n>
```

## JBoss クラスタノードの起動

JBoss クラスタ環境を使用する場合は、MDM Hub コンポーネントをインストールするクラスタノードを開始します。

1. 次のディレクトリに移動します。

<JBoss のインストールディレクトリ>/bin

2. JBoss クラスタノードを開始するには、クラスタノードのあるマシンで次のコマンドを実行します。

UNIX の場合: `standalone.sh -c standalone-full.xml -b 0.0.0.0 -Djboss.node.name=<Name of the cluster node> -Djboss.server.base.dir=./<node path> -Djboss.as.management.blocking.timeout=5000 -u <multicast address> -Djgroups.bind_addr=<bind address> -Djboss.socket.binding.port-offset=<port offset value> -Djboss.partition.name=<Partition name>`

Windows の場合: `standalone.bat -c standalone-full.xml -b 0.0.0.0 -Djboss.node.name=<Name of the cluster node> -Djboss.server.base.dir=./<node path> -Djboss.as.management.blocking.timeout=5000 -u <multicast address> -Djgroups.bind_addr=<bind address> -Djboss.socket.binding.port-offset=<port offset value> -Djboss.partition.name=<Partition name>`

複数のクラスタノードを同じマシン上で実行している場合、ポートオフセット値を設定します。デフォルトのパーティション名を使用するか、クラスタに属すすべてのノードでパーティション名が同じであることを確認してください。お使いの環境に基づいて、タイムアウト間隔を調整します。

## Oracle WebLogic のセットアップ

**注:** バージョン 10.5 HotFix 4 では、MDM での Oracle WebLogic のサポートは延期され、使用することはできません。

### Oracle Weblogic

Oracle WebLogic は、現時点で Java 17 と Jakarta EE 10 の両方をサポートするバージョンを利用することができないため、延期されています。対応する互換性のあるバージョンを Oracle がリリースするまでサポートは延期されます。Informatica は、このようなバージョンが利用可能になった場合に、今後の MDM リリースでサポートを提供する予定です。

### ユーザーに対するアドバイス

- Oracle WebLogic を使用している場合、リリース後に MDM 10.4 HotFix 4 にアップグレードすることはできません。
- 現在のバージョンの MDM を引き続き使用するか、Java 8 をサポートし、Oracle WebLogic との互換性が維持される MDM 10.5 HotFix 3 へのアップグレードを検討するかを選択することができます。

### 予定されている開発

- Informatica は、Java 17 および Jakarta EE 10 互換バージョンのリリースに伴い、将来の MDM リリースで Oracle WebLogic のサポートを提供する予定です。
- その後にサポートされる MDM バージョンが利用可能になった場合は、すぐにアップグレードすることができます。

MDM Hub は、Oracle WebLogic クラスタ環境またはスタンドアロンの WebLogic インスタンスにインストールできます。WebLogic のマニュアルに記載されている説明に従って、WebLogic をインストールし、設定します。

**重要:** Multidomain MDM と、バージョン 12.2.1.3 を含む Oracle WebLogic の一部のバージョンの連携には、既知の制限事項がいくつかあります。開始する前に、『*Multidomain MDM のリリースノート*』のインストールとアップグレードに関する章を参照してください。

Hub サーバーと Process サーバーを WebLogic アプリケーションサーバーにインストールする前に、WebLogic のドメインを作成します。WebLogic の管理コンソールを使用して、Hub サーバーおよび Process サーバーのアプリケーションのためのドメインを作成します。詳細については、WebLogic のマニュアルを参照してください。

**注:** データベースサーバーと同じタイムゾーンのアプリケーションサーバーをインストールします。

## Java 仮想マシンの設定

Java 仮想マシン (JVM) を設定するには、JAVA\_OPTIONS 環境変数を使用して、Java のオプションを設定します。

Java のオプションは、次のファイルで設定できます。

UNIX の場合: <WebLogic domain: WebLogic ドメイン>/bin/setDomainEnv.sh

Windows の場合: <WebLogic domain: WebLogic ドメイン>\bin\setDomainEnv.cmd

次の表で、Java のオプションを説明します。

Java のオプション	説明
-server	起動速度は遅くなりますが、後続の操作は高速になります。
-Djgroups.bind_addr	JGroup がメッセージの送受信を行う必要があるインターフェース。 マルチノード環境またはクラスタ環境が必要です。各ノードが専用のネットワークインターフェースにバインドされていることを確認します。
-Djava.net.preferIPv4Stack	Java で Internet Protocol バージョン 4 (IPv4) を使用するかどうかを指定します。オペレーティングシステムで Internet Protocol バージョン 6 (IPv6) を使用する場合は、true に設定します。
-Djava.security.egd	Linux 環境での Informatica Data Director の起動時間を短縮します。 値を次のように設定します。 Djava.security.egd=file:/dev/./urandom
-Doracle.jdbc.J2EE13Compliant	oracle.jdbc.J2EE13Compliant システム変数を設定します。true に設定します。 このパラメータを true に設定しない場合、Java データベース接続 (JDBC) の問題が発生する可能性があります。
-Djavax.wsdl.factory.WSDLFactory	WebLogic 12.2.1 以降の環境でデフォルトの名前空間プレフィックスを持つサービス WSDL を呼び出すために必要です。com.ibm.wsdl.factory.WSDLFactoryImpl に設定します。
-DANTLR_USE_DIRECT_CLASS_LOADING	MDM Hub がデプロイされている WebLogic 12.2.1 以降の環境を起動するために必要です。true に設定します。
-Dmdm.node.groupid	MDM Hub 実装で Java 仮想マシンのグループ ID を指定します。Hub サーバーとプロセスサーバーを論理的にグループ化する場合にのみ必要です。

Java のオプション	説明
-De360.connection.channel	<p>アプリケーションサーバーがエンティティ 360 フレームワークに基づいて Hub サーバーと Data Director 画面間の通信に使用する必要のある通信プロトコル。有効な値は HTTP および HTTPS です。デフォルトは HTTP です。</p> <p>次のサポートしている Java オプションを設定していることを確認します。</p> <ul style="list-style-type: none"> <li>-De360.mdm.host</li> <li>-De360.mdm.port</li> </ul>
-DUseESLegacyFqSearch	<p>フィールド検索で、ビジネスエンティティタイプの子ノード内から完全一致を返すかどうかを指定します。複数のフィールドでフィールド検索を実行する場合にのみ適用できます。複数のクエリフィールドが子レベルにある場合、検索時に同じ子ノードの検索値を含むレコードが返されるようにするかどうかを指定します。true に設定すると、異なる子ノードの子レベルのクエリフィールドに一致する可能性のあるレコードが返されます。デフォルトは false です。</p>
-Dweblogic.security.SSL.trustedCAKeyStore -Dweblogic.security.SSL.enable.renegotiation -Dweblogic.security.SSL.verbose	<p>HTTPS が WebLogic に対して有効な場合に必要です。</p> <p>-Dweblogic.security.SSL.trustedCAKeyStore Java オプションは、信頼されている証明書を含むキーストアへのパスを指定します。キーストアへの絶対パスを設定します。</p> <p>-Dweblogic.security.SSL.enable.renegotiation Java オプションは SSL の再ネゴシエーションを有効にします。true に設定します。</p> <p>-Dweblogic.security.SSL.verbose Java オプションは追加の SSL デバッグを有効にします。true に設定します。</p>
-De360.mdm.host	<p>アプリケーションサーバーホスト。</p> <p>環境で HTTPS 通信プロトコルを使用していて、セキュリティ証明書が完全修飾ドメイン名 (FQDN) に発行されている場合、FQDN に設定します。</p> <p>Java のオプションは、次のいずれかのシナリオで設定します。</p> <ul style="list-style-type: none"> <li>- MDM Hub アプリケーションを管理対象サーバーにのみデプロイするには、各管理対象サーバーのスクリプトに Java のオプションを追加します。管理対象サーバーの IP アドレスまたはシンボリック名に設定します。</li> <li>- MDM Hub アプリケーションを 7001 以外の WebLogic ポートにデプロイするには、アプリケーションサーバーの起動スクリプトに Java のオプションを追加します。WebLogic ホストの IP アドレスまたはシンボリック名に設定します。</li> </ul> <p>このパラメータを設定しない場合、エンティティ 360 フレームワークに基づく Data Director の画面が期待どおりに動作しない可能性があります。</p>

Java のオプション	説明
-De360.mdm.port	<p>アプリケーションサーバーのポート。</p> <p>Java のオプションは、次のいずれかのシナリオで設定します。</p> <ul style="list-style-type: none"> <li>- MDM Hub アプリケーションを管理対象サーバーにのみデプロイするには、各管理対象サーバーのスクリプトに Java のオプションを追加します。管理対象サーバーのポート番号に設定します。</li> <li>- MDM Hub アプリケーションを 7001 以外の WebLogic ポートにデプロイするには、アプリケーションサーバーの起動スクリプトに Java のオプションを追加して、そのポート番号に設定します。</li> </ul> <p>このパラメータを設定しない場合、エンティティ 360 フレームワークに基づく Data Director の画面が期待どおりに動作しない可能性があります。</p>
-Didd.mdm.host -Didd.mdm.port -Didd.protocol	<p>サブジェクト領域を使用する Data Director に必要です。</p> <p>サブジェクト領域を使用する Data Director をデプロイするには、次の Java のオプションを設定します。</p> <ul style="list-style-type: none"> <li>-Didd.mdm.host。アプリケーションサーバーホストのホスト名または IP アドレスに設定します。</li> <li>-Didd.mdm.port。サーバーの初期化中にサブジェクト領域アプリケーションとともに Data Director によって内部的に使用される必須プロパティ。アプリケーション用の JVM で使用する HTTP または HTTPS リスナポートを指定します。デフォルトは 8080 です。</li> <li>-Didd.protocol。サーバーの初期化中にサブジェクト領域のアプリケーションをデプロイするために使用される必須プロパティ。使用する通信プロトコルが HTTP か HTTPS かを指定します。デフォルトは HTTP です。</li> </ul>
-Dfile.encoding -Dweblogic.http.URIDecodeEncoding	<p>Data Director を使用する場合、および REST API を使用してレコードを検索する場合に必要です。</p> <p>UTF-8 文字を含むレコードを確実に検索および保存できるようにするには、両方の Java のオプションを UTF-8 に設定します。</p>
-DFrameworksLogConfigurationPath	log4j.xml ファイルのコンフィギュレーションパスを設定します。
-Dtask.pageSize=<タスクの最大数>	各要求に対して取得される ActiveVOS タスクの最大数を指定します。デフォルトは 5000 です。環境に多数のタスクがある場合は、この数値を増やします。
-Dstricttransportsecurity.flag	HTTP 要求を使用して Data Director にアクセスするすべての試行を、代わりに HTTPS 要求に変換する必要があるかどうかを指定します。true に設定します。
-Xms	初期ヒープサイズ 2048m に設定します。
-Xmx	<p>最大 JVM ヒープサイズ。6 GB 以上に設定します。</p> <p>例えば、次の USER_MEM_ARGS 環境変数設定を使用します。</p> <pre>USER_MEM_ARGS="-Xms2048m -Xmx6000m" export USER_MEM_ARGS</pre>

Java のオプション	説明
XX:+UseCodeCacheFlushing	コードキャッシュがいっぱいになったときに、JVM がコンパイルされたコードを破棄するかどうかを指定します。
-XX:ReservedCodeCacheSize	JIT コードのキャッシュサイズ。MDM Hub 環境のパフォーマンスを高めるには、512m に設定します。

## Java 仮想マシンの論理的なグループ化の例

Java 仮想マシン（JVM）をグループ化することにより、Hub サーバーとプロセスサーバーの論理グループを取得できます。Hub サーバーアプリケーションとプロセスサーバーアプリケーションを論理 JVM グループにデプロイすると、Hub サーバーアプリケーションとプロセスサーバーアプリケーション間のすべての通信がグループ内にとどまります。JVM をグループ化するには、MDM Hub 環境の各 JVM にグループ ID を割り当てます。

**注:** プロセスサーバーのグループ化は、クレンジングと一致プロセスのみに適用されます。論理グループは MDM Hub の内部サーバーキャッシュには適用されません。

次の表は、論理 JVM グループの例を示しています。

JVM グループ	JVM	Hub サーバー	プロセスサーバー
Group1	JVM1	○	はい
Group1	JVM4	-	○
Group2	JVM2	○	はい
Group3	JVM3	-	○

JVM1 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group1
```

JVM2 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group2
```

JVM3 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group3
```

JVM4 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group1
```

JVM を設定し、Hub サーバーとプロセスサーバーをデプロイすると、グループには次の特性があります。

- Group1 には 2 つのプロセスサーバーがあり、Group2 には 1 つのプロセスサーバーがあり、Group3 には 1 つのプロセスサーバーがあります。
- すべてのクレンジングとバッチ呼び出しは、検索を除き、自分のグループにとどまります。例えば、Group1 の Hub サーバーでのリアルタイム呼び出しは、Group1 プロセスサーバー（JVM1 および JVM4）にのみ影響します。

## WebLogic Server 認証の無効化

MDM Hub は、WebLogic Server 認証を無効にする必要がある HTTP 基本認証を使用します。WebLogic Server 認証を無効にするには、config.xml ファイルを編集します。

1. 次の WebLogic ディレクトリに移動します。  
<WebLogic installation directory: Weblogic のインストールディレクトリ>/user\_projects/domains/<user domain>/config
2. テキストエディタで config.xml ファイルを開きます。
3. <security-configuration>要素内に次の要素を追加します。  
<enforce-valid-basic-auth-credentials>  
false  
</enforce-valid-basic-auth-credentials>

## ActiveVOS コンソール管理ユーザーの作成

ActiveVOS を使用する場合、アプリケーションサーバーコンテナで abAdmin ロールを使用して ActiveVOS コンソールの管理者ユーザーを作成します。ActiveVOS を使用する場合、abAdmin ロールを使用して ActiveVOS コンソールの管理者ユーザーを作成します。管理者ユーザーを作成しない場合は、Hub サーバーのデプロイメントに失敗します。Hub サーバーインストーラから ActiveVOS コンソールの管理者ユーザーの資格情報を入力するように要求されたら、ActiveVOS コンソールの管理者ユーザー名およびパスワードを使用します。

1. 次のディレクトリに変更します。  
<JBoss のインストールディレクトリ>/bin
2. add-user ユーティリティを実行するには、次のスクリプトを使用します。  
UNIX の場合: add-user.sh  
Windows の場合: add-user.bat
3. 表示されるプロンプトに回答します。  
次の表に、各プロンプトで指定する値の説明を示します。

プロンプト	指定する値
どのタイプのユーザーを追加しますか? a) 管理ユーザーまたは b) アプリケーションユーザー	アプリケーションユーザーを選択するには、b を入力します。
レルム (ApplicationRealm)	レルム名。standalone-full.xml ファイルに追加した login-module で指定したレルム名を入力します。
ユーザー名	ActiveVOS コンソールの管理者名。
パスワード	JBoss のパスワード標準に準拠しているパスワード。
このユーザーをどのロールにしますか?	abAdmin。
レルム<realm name>のユーザー<user name>を追加しようとしています。よろしいですか?	ユーザーを追加するには、yes を入力します。
この新しいユーザーは、ある AS プロセスを別の AS プロセスに接続するために使用されますか?	はい。

4. WebSphere コンソールにログインして、ActiveVOS コンソール管理者ユーザーを作成します。  
**注:** ActiveVOS コンソールのユーザーは、インストール後またはアップグレード後のプロセスで postInstallSetup または patchInstallSetup スクリプトを実行するときに abAdmin ロールにマップされます。
5. WebLogic コンソールにログインします。
6. abAdmin ロールを作成します。
7. ActiveVOS コンソール管理ユーザーを作成します。
8. 管理者ユーザーを abAdmin ロールに割り当てます。

## 追加の Oracle WebLogic の設定

MDM Hub 環境の要件に基づいて、WebLogic の追加設定を行います。

次の表に、実行可能な設定を示します。

設定	説明
スタンドアロンプロセスサーバーインスタンス用の WebLogic の設定	次のシナリオでは、スタンドアロンプロセスサーバーインスタンス用に WebLogic を設定する必要があります。 - Hub サーバーがインストールされていない WebLogic インスタンスに Process サーバーインスタンスをインストールする必要がある。 - 複数のスタンドアロン Process サーバーインスタンスをインストールする。
複数の MDM Hub マスタデータベース用の WebLogic の設定	複数の MDM Hub マスタデータベースインスタンスを設定する場合に必要です。
HTTPS プロトコルの設定	MDM Hub 通信用に HTTPS プロトコルを設定する場合に必要です。

## スタンドアロンプロセスサーバーインスタンス用の WebLogic の設定

複数のスタンドアロン Process サーバーインスタンスをインストールする場合、適切なデータソースを使用するように WebLogic を設定します。また、Hub サーバーをインストールしていない WebLogic インスタンスに Process サーバーインスタンスをインストールする場合、データソースを設定します。

以下のタスクを実行して、適切なデータソースを使用するように WebLogic を設定します。

1. JDBC ドライバをインストールします。
2. MDM Hub マスタデータベースのデータソースを作成します。
3. オペレーショナル参照ストアのデータソースを作成します。

### 手順 1.JDBC ドライバのインストール

MDM Hub マスタデータベースとオペレーショナル参照ストア（ORS）のデータソースを作成する前に、JDBC ドライバをインストールします。

サポートされているバージョンの JDBC ドライバの取得方法については、Oracle にお問い合わせください。

サポートされているバージョンの JDBC ドライバの取得方法については、Microsoft にお問い合わせください。

サポートされているバージョンの JDBC ドライバの取得方法については、IBM お問い合わせください。

1. JDBC ドライバを次のディレクトリにコピーします。

<WebLogic installation directory: WebLogic のインストールディレクトリ>/wlserver/server/lib

2. 次のファイルの `_CLASSPATH` 変数に、JDBC ドライバへのパスを追加します。

UNIX の場合:<WebLogic domain: WebLogic ドメイン>/bin/commEnv.sh

Windows の場合:<WebLogic domain: WebLogic ドメイン>\bin\commEnv.cmd

**注:** 他の WebLogic Server ライブラリへのパスの前に、JDBC ドライバへのパスを配置します。

## 手順 2.MDM Hub マスタデータベースのデータソースの作成

JDBC ドライバのインストール後、Process サーバマシンで、MDM Hub マスタデータベースのデータソースを作成します。

1. WebLogic 管理コンソールで、**[ロックして編集]** ボタンをクリックしてロックを取得します。
2. **[サービス] > [JDBC] > [データソース]** をクリックし、**[新規作成]** をクリックします。  
**[JDBC データソースのプロパティ]** ページが表示されます。
3. 以下のデータソースのプロパティを指定してください。

プロパティ	説明
名前	JDBC データソースの名前。「MDM Master Data Source」という名前を付けます。
JNDI 名	JDBC データソースが関連付けられる場所への JNDI パス。jdbc/siperian-cmx_system-ds と指定します。
データベースタイプ	接続先にするデータベースのタイプ。 <b>[Oracle]</b> を選択します。 <b>[MS SQL Server]</b> を選択します。
データベースドライバ	データベースへの接続に使用する JDBC ドライバ。 <b>[Oracle ドライバ (Thin XA)]</b> を選択します。 <b>[MS SQL Server ドライバ (Type 4 XA) バージョン: 2005 以降]</b> を選択します。

4. **[次へ]** をクリックし、**[次へ]** をもう一度クリックします。  
**[接続プロパティ]** ページが表示されます。
5. 次の接続プロパティの値を入力します。

プロパティ	説明
データベース名	接続するデータベースの名前。
ホスト名	データベースをホストするサーバーの DNS 名または IP アドレス。7001 以外の WebLogic ポートに MDM Hub アプリケーションをデプロイするには、WebLogic ホストのホスト名と IP アドレスを設定します。localhost は使用しないでください。
ポート	データベースサーバーが接続リクエストをリスンするポート。
データベースユーザー名	データソース内の各接続に使用するデータベースユーザー名。

プロパティ	説明
パスワード	データベースユーザーアカウントのパスワード。
パスワードの確認	データベースユーザーアカウントのパスワード。

6. **【次へ】** をクリックします。  
**【データベース接続のテスト】** ページが表示されます。
7. **【設定のテスト】** をクリックして、ドライバ接続をテストします。  
テストに失敗した場合は、**【接続プロパティ】** ページ内の値を更新し、成功するまで接続を繰り返し試行する必要があります。
8. **【次へ】** をクリックし、データソースをデプロイするサーバーを選択します。
9. **【完了】** をクリックし、**【変更のアクティブ化】** をクリックします。

### 手順 3. オペレーショナル参照ストアのデータソースの作成

Process サーバマシンで、各オペレーショナル参照ストアのデータストアを作成します。

1. WebLogic 管理コンソールで、**【ロックして編集】** ボタンをクリックしてロックを取得します。
2. **【サービス】** > **【JDBC】** > **【データソース】** をクリックし、**【新規作成】** をクリックします。  
**【JDBC データソースのプロパティ】** ページが表示されます。
3. 以下のデータソースのプロパティを指定してください。

プロパティ	説明
名前	JDBC データソースの名前。「MDM ORS Data Source」という名前を付けます。
JNDI 名	JDBC データソースが関連付けられる場所への JNDI パス。「jdbc/siperian-<oracle host name>-<oracle sid>-<Operational reference Store name>-ds」と指定します。 jdbc/siperian-<Microsoft SQL Server host name>-<Operational reference Store name>-ds を指定します。
データベースタイプ	接続先にするデータベースのタイプ。 <b>【Oracle】</b> を選択します。 <b>【MS SQL Server】</b> を選択します。
データベースドライバ	データベースへの接続に使用する JDBC ドライバ。 <b>【Oracle ドライバ (Thin XA)】</b> を選択します。 <b>【MS SQL Server ドライバ (Type 4 XA) バージョン: 2005 以降】</b> を指定します。

4. **【次へ】** をクリックし、**【次へ】** をもう一度クリックします。  
**【接続プロパティ】** ページが表示されます。

5. 次の接続プロパティの値を入力します。

プロパティ	説明
データベース名	接続するデータベースの名前。
ホスト名	データベースをホストするサーバーの DNS 名または IP アドレス。7001 以外の WebLogic ポートに MDM Hub アプリケーションをデプロイするには、WebLogic ホストのホスト名と IP アドレスを設定します。localhost は使用しないでください。
ポート	データベースサーバーが接続リクエストをリスンするポート。
データベースユーザー名	データソース内の各接続に使用するデータベースユーザー名。
パスワード	データベースユーザーアカウントのパスワード。
パスワードの確認	データベースユーザーアカウントのパスワード。

6. **【次へ】** をクリックします。  
**【データベース接続のテスト】** ページが表示されます。
7. **【設定のテスト】** をクリックして、ドライバ接続をテストします。  
テストに失敗した場合は、**【接続プロパティ】** ページ内の値を更新し、成功するまで接続を繰り返し試行する必要があります。
8. **【次へ】** をクリックし、データソースをデプロイするサーバーを選択します。
9. **【完了】** をクリックし、**【変更のアクティブ化】** をクリックします。

## 複数の MDM Hub マスタデータベース用の WebLogic の設定

複数の MDM Hub マスタデータベースインスタンスを設定する場合、MDM Hub マスタデータベースインスタンスと同数の WebLogic ドメインを設定します。各 MDM Hub マスタデータベースインスタンスには、独自の MDM Hub インスタンスが必要です。そのため、同数の WebLogic ドメインを作成して各 MDM Hub インスタンスを個別の WebLogic ドメインでデプロイします。

## HTTPS プロトコルの設定

MDM Hub の通信には HTTPS プロトコルを設定できます。WebLogic Server 管理コンソールを使用して HTTPS プロトコルを設定します。または、デフォルトの JDK セキュリティ証明書を使用して HTTPS を有効化できます。

**注:** キーストアを構成する前に、Java Keytool または OpenSSL のいずれかを使用して、WebLogic サーバー用の自己署名証明書を生成します。

- WebLogic Server 管理コンソールで、**【環境】** セクションの下にある **【サーバー】** をクリックします。
- 【サーバーのサマリ】** ページで、サーバーのリストから **【AdminServer(admin)】** リンクをクリックします。
- 【SSL リスニングポート有効】** オプションを有効にし、SSL リスニングポート番号を入力します。
- 【SSL】** タブをクリックし、**【ホスト名の検証】** から **【なし】** を選択します。
- ID およびトラストキーストアを設定するサーバーの名前を選択します。
- 【設定】** > **【キーストア】** をクリックします。

7. 次の設定オプションに対するパスワードを変更します。

設定オプション	説明
デモ ID キーストアのパスフレーズ	デモ ID キーストアの暗号化されたパスワード。
デモトラストキーストアのパスフレーズ	デモトラストキーストアの暗号化されたパスワード。
Java 標準トラストキーストアのパスフレーズ	Java 標準トラストキーストアのパスワード。

8. 変更を保存します。

## IBM WebSphere のセットアップ

MDM Hub は、IBM WebSphere クラスタ環境またはスタンドアロンの WebSphere インスタンスにインストールできます。WebSphere のマニュアルで説明されている手順に従って、WebSphere をインストールして設定します。

**注:** バージョン 10.5 HotFix 4 では、MDM での IBM WebSphere Traditional のサポートは延期され、使用することはできません。

IBM WebSphere

IBM WebSphere Traditional は、Java 17 および Jakarta EE 10 と互換性がないため、現在のリリースでは提供されていません。WebSphere のサポートは、IBM が Java 17 および Jakarta EE 10 互換バージョンをリリースするまで、次回のリリースに延期されます。このようなバージョンが利用可能になった場合、Informatica は今後の MDM リリースで IBM WebSphere のサポートを拡張する予定です。

### ユーザーに対するアドバイス

- IBM WebSphere を使用している場合、リリース後に MDM 10.4 HotFix 4 にアップグレードすることはできません。
- 現在のバージョンの MDM を引き続き使用するか、Java 8 をサポートし、IBM WebSphere との互換性が維持される MDM 10.5 HotFix 3 へのアップグレードを検討するかを選択することができます。

### 予定されている開発

- Informatica は、Java 17 および Jakarta EE 10 互換バージョンのリリースに伴い、将来の MDM リリースで IBM WebSphere のサポートを提供する予定です。
- その後にサポートされる MDM バージョンが利用可能になった場合は、すぐにアップグレードすることができます。

WebSphere インストールディレクトリのパスにスペースを含めないようにします。

**注:** データベースサーバーと同じタイムゾーンのアプリケーションサーバーをインストールします。

次の表で、インストール前に設定するプロパティ、値、およびプロパティを設定する場所の簡単な説明を示します。

カスタムプロパティ	値	説明
com.ibm.ws.scripting.echoparams	false	このプロパティは、次のディレクトリにある <code>wsadmin.properties</code> ファイルで設定します: <WebSphere のインストールディレクトリ>\WebSphere\AppServer\profiles\<プロファイル名>\properties ActiveVOS データベースのパスワードが <code>patchinstallSetup.log</code> ファイルのスクリプトテキストに表示されないようにするには、この値を <code>false</code> に設定します。デフォルトでは <code>true</code> です。

## Java 仮想マシンの設定

Java 仮想マシン (JVM) を設定するには、`JAVA_OPTIONS` 環境変数を使用して、Java のオプションを設定します。Java のオプションを編集または追加した場合は、JVM を再起動します。

WebSphere クラスタ環境を使用する場合は、次のクラスタコンポーネントで Java のオプションを設定します。

- サーバー。クラスタ内の各サーバーで、必要なすべての Java のオプションを設定します。
- デプロイメントマネージャ。必要なすべての Java のオプションを設定します。
- ノードエージェント。Java の `-Xmx` オプションと `-Xms` オプションを使用して、ヒープサイズのみを設定します。

次の表で、Java のオプションの設定を説明します。

Java のオプション	説明
<code>-server</code>	起動速度は遅くなりますが、後続の操作は高速になります。
<code>-Djava.net.preferIPv4Stack</code>	Java で Internet Protocol バージョン 4 (IPv4) を使用するかどうかを指定します。オペレーティングシステムで Internet Protocol バージョン 6 (IPv6) を使用する場合は、 <code>true</code> に設定します。
<code>-Djava.security.egd</code>	Linux 環境での Data Director の起動時間を短縮します。値を <code>file:/dev/.urandom</code> に設定します。
<code>-DUseESLegacyFqSearch</code>	フィールド検索で、ビジネスエンティティタイプの子ノード内から完全一致を返すかどうかを指定します。複数のフィールドでフィールド検索を実行する場合にのみ適用できます。複数のクエリフィールドが子レベルにある場合、検索時に同じ子ノードの検索値を含むレコードが返されるようにするかどうかを指定します。 <code>true</code> に設定すると、異なる子ノードの子レベルのクエリフィールドに一致する可能性のあるレコードが返されます。デフォルトは <code>false</code> です。
<code>-Ddb2.jcc.charsetDecoderEncoder</code>	MDM Hub サンプルオペレーショナル参照ストアを使用する必要があります。JDBC ドライバで、UTF-8 文字列でないバイトシーケンスの代わりに、Unicode 置換文字 (U+FFFF) を返すことができます。3 に設定します。

Java のオプション	説明
-Dcom.ibm.crypto.provider.DoRSATypeChecking	<p>Java が秘密キーおよび復号化と公開キーによるデータの RSA 型暗号化を許可するかどうかを指定します。MDM Hub インストーラがライセンス証明書を読み取り、パスワードハッシュが MDM Hub で機能するために必要です。false に設定します。</p> <p>-Dcom.ibm.crypto.provider.DoRSATypeChecking を false に設定しないと、Hub サーバーが起動せず、ライセンスエラーが発生する可能性があります。</p>
-Djgroups.bind_addr	<p>JGroup がメッセージの送受信を行う必要があるインターフェース。</p> <p>マルチノード環境またはクラスタ環境で必要です。各ノードが専用のネットワークインターフェースにバインドされていることを確認します。</p>
-DDe360.mdm.host -DDe360.mdm.port -DDe360.connection.channel	<p>アプリケーションサーバーの通信プロトコル、ホストおよびポート。</p> <p>2809 以外の Bootstrap ポートに MDM Hub アプリケーションをデプロイするには、次の Java オプションを設定します。</p> <ul style="list-style-type: none"> <li>-DDe360.connection.channel。使用する通信プロトコルに設定します。有効な値は HTTP および HTTPS です。デフォルトは HTTP です。</li> <li>-DDe360.mdm.hostWebSphere ホストの IP アドレスを設定します。</li> </ul> <p>環境で HTTPS 通信プロトコルを使用していて、セキュリティ証明書が完全修飾ドメイン名 (FQDN) に発行されている場合、FQDN に設定します。</p> <ul style="list-style-type: none"> <li>-DDe360.mdm.port2809 ではなく、設定済みの WebSphere Bootstrap ポートを設定します。</li> </ul> <p>このパラメータを設定しない場合、エンティティ 360 フレームワークに基づく Data Director の画面が期待どおりに動作しない可能性があります。</p>
-Ddidd.mdm.host -Ddidd.mdm.port -Ddidd.protocol	<p>サブジェクト領域を使用する Data Director に必要です。サブジェクト領域を使用する Data Director をデプロイする場合、次の Java オプションを設定します。</p> <ul style="list-style-type: none"> <li>-Ddidd.mdm.host。アプリケーションサーバーホストのホスト名または IP アドレスに設定します。</li> <li>-Ddidd.mdm.port。サーバーの初期化中にサブジェクト領域アプリケーションとともに Data Director によって内部的に使用される必須プロパティ。アプリケーション用の JVM で使用する HTTP または HTTPS リスナーポートを指定します。デフォルトは 8080 です。</li> <li>-Ddidd.protocol。サーバーの初期化中にサブジェクト領域のアプリケーションをデプロイするために使用される必須プロパティ。使用する通信プロトコルが HTTP か HTTPS かを指定します。デフォルトは HTTP です。</li> </ul>
-DFrameworksLogConfigurationPath	log4j.xml ファイルのパス。

Java のオプション	説明
-Doracle.jdbc.J2EE13Compliant	Oracle ドライバのシステム変数が J2EE に完全に準拠するかどうかを指定します。true に設定します。 このパラメータを true に設定しない場合、Java データベース接続（JDBC）の問題が発生する可能性があります。
-Dmdm.node.groupid	MDM Hub 実装で Java 仮想マシンのグループ ID を指定します。Hub サーバーとプロセスサーバーを論理的にグループ化する場合にのみ必要です。
-Dfile.encoding -Dclient.encoding.override	Informatica Data Director を使用する場合、および REST API を使用してレコードを検索する場合に必要です。 UTF-8 文字を含むレコードを確実に検索および保存できるようにするには、両方の Java のオプションを UTF-8 に設定します。
-Dstricttransportsecurity.flag	HTTP 要求を使用して Data Director にアクセスするすべての試行を、代わりに HTTPS 要求に変換する必要があるかどうかを指定します。true に設定します。
-XX:codecachetotal	JIT コードのキャッシュサイズ。MDM Hub 環境のパフォーマンスを高めるには、512m に設定します。
-Xmx	最大 JVM ヒープサイズ。6 GB 以上に設定します。 例えば、-Xmx を 6144m に設定するには、次の JAVA_OPTIONS 環境変数設定を使用します。 <pre>set "JAVA_OPTIONS=-server ... -Xmx6144m"</pre>
-Xms	初期ヒープサイズ 2048m に設定します。
-Xmso	プロセスサーバーの JVM に必要です。オペレーティングシステムスレッドの初期スタックサイズ。システムスレッドのスタックサイズが低いために、アプリケーションサーバーが予期せずシャットダウンしてしまうことがないようにします。4096k に設定します。
-Xss	初期スタックサイズ。2000k に設定します。
XX:+UseCodeCacheFlushing	コードキャッシュがいっぱいになったときに、JVM がコンパイルされたコードを破棄するかどうかを指定します。
-Dtask.pageSize=<タスクの最大数>	各要求に対して取得される ActiveVOS タスクの最大数を指定します。デフォルトは 5000 です。環境に多数のタスクがある場合は、この数値を増やします。

## Java 仮想マシンの論理的なグループ化の例

Java 仮想マシン（JVM）をグループ化することにより、Hub サーバーとプロセスサーバーの論理グループを取得できます。Hub サーバーアプリケーションとプロセスサーバーアプリケーションを論理 JVM グループにデプロイすると、Hub サーバーアプリケーションとプロセスサーバーアプリケーション間のすべての通信がグ

グループ内にとどまります。JVM をグループ化するには、MDM Hub 環境の各 JVM にグループ ID を割り当てます。

**注:** プロセスサーバーのグループ化は、クレンジングと一致プロセスのみに適用されます。論理グループは MDM Hub の内部サーバーキャッシュには適用されません。

次の表は、論理 JVM グループの例を示しています。

JVM グループ	JVM	Hub サーバー	プロセスサーバー
Group1	JVM1	○	はい
Group1	JVM4	-	○
Group2	JVM2	○	はい
Group3	JVM3	-	○

JVM1 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group1
```

JVM2 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group2
```

JVM3 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group3
```

JVM4 では、起動スクリプトに次の Java オプションを追加します。

```
-Dmdm.node.groupid=Group1
```

JVM を設定し、Hub サーバーとプロセスサーバーをデプロイすると、グループには次の特性があります。

- Group1 には 2 つのプロセスサーバーがあり、Group2 には 1 つのプロセスサーバーがあり、Group3 には 1 つのプロセスサーバーがあります。
- すべてのクレンジングとバッチ呼び出しは、検索を除き、自分のグループにとどまります。例えば、Group1 の Hub サーバーでのリアルタイム呼び出しは、Group1 プロセスサーバー（JVM1 および JVM4）にのみ影響します。

## MDM Hub 環境でのパスワードの暗号化

MDM Hub 環境でログファイルに表示されるパスワードなどの機密データを暗号化するには、WebSphere でスク립ティング管理を設定します。

1. 次のディレクトリにある wsadmin.properties ファイルを開きます。

```
<WebSphere installation directory: WebSphere インストールディレクトリ>/profiles/<Application server profile name: アプリケーションサーバーのプロファイル名>/properties
```

2. Java プロパティ com.ibm.ws.scripting.echoparams を false に設定します。

## WebSphere 環境での安全なプロファイルの作成

WebSphere で、Multidomain MDM および Informatica ActiveVOS で使用するためのセキュアプロファイルを設定します。

1. コマンドラインから、次のサンプルコードに示すように安全なプロファイルを作成します。

Windows の場合:

```
<app_server_root>\bin\manageprofiles.bat -create -profileName AppSrv01  
-profilePath <app_server_root>\profiles\AppSrv01  
-templatePath <app_server_root>\profileTemplates\default  
-adminUserName administrator -adminPassword password1 -enableAdminSecurity true
```

UNIX の場合:

```
<app_server_root>/bin/manageprofiles.sh -create -profileName AppSrv01  
-profilePath <app_server_root>/profiles/AppSrv01  
-templatePath <app_server_root>/profileTemplates/default  
-adminUserName administrator -adminPassword password1 -enableAdminSecurity true
```

2. WebSphere コンソールで、セキュリティトランスポートタイプを SSL-Supported に変更します。
  - a. **【セキュリティ】**を展開し、**【グローバル・セキュリティ】**をクリックします。
  - b. **【認証】**の下で**【RMI/IIOP セキュリティ】**を展開し、**【CSiv2 インバウンド通信】**をクリックします。
  - c. CSiv2 Transport Layer の下の**【Transport】**リストから、**【SSL サポート】**を選択します。
  - d. **【適用】**をクリックしてから**【保存】**をクリックします。
  - e. **【CSiv2 アウトバウンド通信】**をクリックします
  - f. CSiv2 Transport Layer の下の**【Transport】**リストから、**【SSL サポート】**を選択します。
  - g. **【適用】**をクリックしてから**【保存】**をクリックします。
3. WebSphere コンソールで、アプリケーションセキュリティが設定されていることを確認します。
  - a. **【セキュリティ】**を展開し、**【グローバル・セキュリティ】**をクリックします。
  - b. **【アプリケーション・セキュリティ】**の下で**【アプリケーション・セキュリティを使用可能にする】**を選択します。
  - c. **【適用】**をクリックしてから**【保存】**をクリックします。
4. 統合リポジトリをセットアップします。
  - a. **【セキュリティ】**を展開し、**【グローバル・セキュリティ】**をクリックします。
  - b. **【ユーザー・アカウント・リポジトリ】**の下で**【使用可能なレルム定義】**リストから、**【統合リポジトリ】**を選択します。
  - c. **【構成】**をクリックします。
  - d. レルムの**【リポジトリ】**の下で、**【組み込みリポジトリを使用する】**をクリックします。
  - e. 管理者ユーザーのパスワードを指定します。
  - f. **【適用】**をクリックしてから**【保存】**をクリックします。
5. WebSphere プロファイルを再起動します。

## ActiveVOS コンソール管理ユーザーの作成

ActiveVOS を使用する場合、アプリケーションサーバーコンテナで abAdmin ロールを使用して ActiveVOS コンソールの管理者ユーザーを作成します。ActiveVOS を使用する場合、abAdmin ロールを使用して ActiveVOS コンソールの管理者ユーザーを作成します。管理者ユーザーを作成しない場合は、Hub サーバーのデプロイメントに失敗します。Hub サーバーインストーラから ActiveVOS コンソールの管理者ユーザーの資格情報を入力するように要求されたら、ActiveVOS コンソールの管理者ユーザー名およびパスワードを使用します。

1. 次のディレクトリに変更します。

```
<JBoss のインストールディレクトリ>/bin
```
2. add-user ユーティリティを実行するには、次のスクリプトを使用します。

UNIX の場合: add-user.sh

Windows の場合: add-user.bat

3. 表示されるプロンプトに回答します。

次の表に、各プロンプトで指定する値の説明を示します。

プロンプト	指定する値
どのタイプのユーザーを追加しますか? a) 管理ユーザーまたは b) アプリケーションユーザー	アプリケーションユーザーを選択するには、b を入力します。
レルム (ApplicationRealm)	レルム名。standalone-full.xml ファイルに追加した login-module で指定したレルム名を入力します。
ユーザー名	ActiveVOS コンソールの管理者名。
パスワード	JBoss のパスワード標準に準拠しているパスワード。
このユーザーをどのロールにしますか?	abAdmin。
レルム<realm name>のユーザー<user name>を追加しようとしています。よろしいですか?	ユーザーを追加するには、yes を入力します。
この新しいユーザーは、ある AS プロセスを別の AS プロセスに接続するために使用されますか?	はい。

4. WebSphere コンソールにログインして、ActiveVOS コンソール管理者ユーザーを作成します。

**注:** ActiveVOS コンソールのユーザーは、インストール後またはアップグレード後のプロセスで postInstallSetup または patchInstallSetup スクリプトを実行するときに abAdmin ロールにマップされます。

5. WebLogic コンソールにログインします。
6. abAdmin ロールを作成します。
7. ActiveVOS コンソール管理ユーザーを作成します。
8. 管理者ユーザーを abAdmin ロールに割り当てます。

## MDM Hub デプロイメント用の SOAP 要求タイムアウトの設定

MDM Hub コンポーネントのデプロイメントがタイムアウトにならないように、SOAP 要求タイムアウトのプロパティを設定します。インストールが成功したら、このプロパティをデフォルト値にリセットします。

1. 次のディレクトリにある soap.client.props ファイルを開きます。

<WebSphere installation directory: WebSphere インストールディレクトリ>/profiles/<Application server profile name>/application-server.properties

2. com.ibm.SOAP.requestTimeout プロパティを 1800 以上に設定します。

# 追加の IBM WebSphere の設定

MDM Hub 環境の要件に基づいて、WebSphere の追加設定を行います。

次の表に、実行可能な設定を示します。

設定	説明
スタンドアロンプロセスサーバーインスタンス用の WebSphere の設定	次のシナリオでは、スタンドアロンプロセスサーバーインスタンス用に WebSphere を設定する必要があります。 <ul style="list-style-type: none"><li>- Hub サーバーがインストールされていない WebSphere インスタンスに Process サーバーインスタンスをインストールする必要がある。</li><li>- 複数のスタンドアロン Process サーバーインスタンスをインストールする。</li></ul>
複数の MDM Hub マスタデータベース用の WebSphere の設定	複数の MDM Hub マスタデータベースインスタンスを設定する場合に必要です。
HTTPS プロトコルの設定	MDM Hub 通信用に HTTPS プロトコルを設定する場合に必要です。
Informatica Data Director 用の WebSphere の設定	Data Director を使用する場合に必要です。

## スタンドアロンプロセスサーバーインスタンス用の WebSphere の設定

複数のスタンドアロン Process サーバーインスタンスをインストールする場合、適切なデータソースを使用するように WebSphere を設定します。また、Hub サーバーをインストールしていない WebSphere インスタンスに Process サーバーインスタンスをインストールする場合、データソースを設定します。

以下のタスクを実行して、適切なデータソースを使用するように WebSphere を設定します。

1. JDBC ドライバをインストールします。
2. MDM Hub マスタデータベースのデータソースを作成します。
3. オペレーショナル参照ストアのデータソースを作成します。

### 手順 1.JDBC ドライバのインストール

MDM Hub マスタデータベースとオペレーショナル参照ストア（ORS）のデータソースを作成する前に、JDBC ドライバをインストールします。

サポートされているバージョンの JDBC ドライバの取得方法については、Oracle にお問い合わせください。

サポートされているバージョンの JDBC ドライバの取得方法については、Microsoft にお問い合わせください。

サポートされているバージョンの JDBC ドライバの取得方法については、IBM にお問い合わせください。

- ▶ JDBC ドライバを次のディレクトリにコピーします。

<WebSphere installation directory>/lib

### 手順 2.MDM Hub マスタデータベースのデータソースの作成

プロセスサーバーマシンに JDBC ドライバをインストールした後で、MDM Hub マスタデータベース用のデータソースを作成します。

1. WebSphere Application Server の管理コンソールを起動します。

2. ドライバライブラリの場所を指定します。
  - a. コンソールナビゲーションツリーで、**【環境】**を展開します。
  - b. **【WebSphere 変数】** リンクをクリックします。
  - c. JDBC 変数が次の JDBC ドライバディレクトリを参照するように更新します。  
 <WebSphere installation directory: WebSphere インストールディレクトリ>/lib
3. MDM Hub マスタデータベースのデータソースで使用するセキュリティアカウントを作成します。
  - a. コンソールのナビゲーションツリーで、**【セキュリティ】**を展開します。
  - b. **【セキュアな管理、アプリケーション、インフラストラクチャ】** リンクをクリックします。
  - c. **【認証】** で **【Java 認証および承認サービス】** を展開し、**【J2C 認証データ】** をクリックします。
  - d. **【新規】** をクリックし、以下のプロパティを指定します。

プロパティ	説明
エイリアス	MDM Hub マスタデータベースの名前。
ユーザー ID	MDM Hub マスタデータベースに接続するためのユーザー名。
パスワード	MDM Hub マスタデータベースにアクセスするためのパスワード。

- e. **【OK】** をクリックします。
4. JDBC プロバイダを作成します。
  - a. **【リソース】** > **【JDBC】** を展開し、**【JDBC プロバイダ】** をクリックします。  
**【JDBC プロバイダ】** ページが表示されます。
  - b. データソースを使用するアプリケーションの範囲を選択します。
  - c. **【新規】** をクリックし、以下のプロパティを指定します。

プロパティ	説明
データベースタイプ	データベースの種類。 <b>【Oracle】</b> を選択します。
プロバイダタイプ	JDBC プロバイダのタイプ。 <b>【Oracle JDBC ドライバ】</b> を選択します。
実装タイプ	データソース実装のタイプ。 <b>【XA データソース】</b> を選択します。
名前	JDBC プロバイダの名前。名前を「Informatica MDM Oracle JDBC Provider (XA)」に変更します。

- d. **【次へ】** をクリックし、**【完了】** をクリックします。
5. MDM Hub マスタデータベースのデータソースを作成します。
  - a. 作成した JDBC プロバイダをクリックします。  
**【設定】** ページが表示されます。
  - b. **【追加プロパティ】** の下で、**【データソース】** をクリックします。  
**【データソース】** ページが表示されます。

- c. **【新規】** をクリックします。
- d. 以下のデータソースのプロパティを指定してください。

プロパティ	説明
名前	データソース名。「MDM Master Data Source」と指定します。
JNDI 名	JDBC データソースが関連付けられる場所への JNDI パス。jdbc/siperian-cmx_system-ds と指定します。 <b>注:</b> JNDI 名は小文字で指定する必要があります。
管理されるコンポーネントの認証エイリアス	マスターデータベースデータソースの認証エイリアス。[<host name>/cmx_system] を選択します。

- e. **【次へ】** をクリックし、**【完了】** をクリックします。

### 手順 3. オペレーショナル参照ストアのデータソースの作成

プロセスサーバマシンに JDBC ドライバをインストールした後で、各オペレーショナル参照ストア用のデータソースを作成します。

1. WebSphere Application Server の管理コンソールを起動します。
2. ドライバライブラリの場所を指定します。
  - a. コンソールナビゲーションツリーで、**【環境】** を展開します。
  - b. **【WebSphere 変数】** リンクをクリックします。
  - c. JDBC 変数が次の JDBC ドライバディレクトリを参照するように更新します。  
<WebSphere installation directory>: WebSphere インストールディレクトリ>/lib
3. オペレーショナル参照ストアで使用するセキュリティアカウントを作成します。
  - a. コンソールのナビゲーションツリーで、**【セキュリティ】** を展開します。
  - b. **【セキュアな管理、アプリケーション、インフラストラクチャ】** リンクをクリックします。
  - c. **【認証】** で **【Java 認証および承認サービス】** を展開し、**【J2C 認証データ】** をクリックします。
  - d. **【新規】** をクリックし、以下のプロパティを設定します。

プロパティ	説明
エイリアス	オペレーショナル参照ストアの名前。
ユーザー ID	オペレーショナル参照ストアに接続するためのユーザー名。
パスワード	オペレーショナル参照ストアにアクセスするためのパスワード。

- e. **【OK】** をクリックします。
4. JDBC プロバイダを作成します。
  - a. **【リソース】** > **【JDBC】** を展開し、**【JDBC プロバイダ】** をクリックします。  
**【JDBC プロバイダ】** ページが表示されます。

- b. データソースを使用するアプリケーションのスコープを選択します。
- c. **【新規】** をクリックし、以下のプロパティを指定します。

プロパティ	説明
データベースタイプ	データベースの種類。【Oracle】を選択します。
プロバイダタイプ	JDBC プロバイダのタイプ。【Oracle JDBC ドライバ】を選択します。
実装タイプ	データソース実装のタイプ。【XA データソース】を選択します。
名前	JDBC プロバイダの名前。名前を「Informatica MDM Oracle JDBC Provider (XA)」に変更します。

- d. **【次へ】** をクリックし、**【完了】** をクリックします。
5. オペレーショナル参照ストアのデータソースを作成します。
    - a. 作成した JDBC プロバイダをクリックします。  
**【設定】** ページが表示されます。
    - b. **【追加プロパティ】** の下で、**【データソース】** をクリックします。  
**【データソース】** ページが表示されます。
    - c. **【新規】** をクリックします。
    - d. 以下のデータソースのプロパティを指定してください。

プロパティ	説明
名前	データソース名。「MDM ORS Data Source」と指定します。
JNDI 名	JDBC データソースが関連付けられる場所への JNDI パス。jdbc/siperian-<Oracle host name>-<Oracle SID>-<Operational Reference Store name>-dsjdbc/siperian-<IBM Db2 host name>-<IBM Db2 database name>-<Operational Reference Store name>-dsjdbc/siperian-<Microsoft SQL Server host name>-<Operational Reference Store name>-ds を指定します。 <b>注:</b> JNDI 名は小文字で指定する必要があります。
管理されるコンポーネントの認証エイリアス	マスターデータベースデータソースの認証エイリアス。<host name>/<Operational Reference Store name>を選択します。
ドライバクラス名	JDBC ドライバクラス。com.microsoft.sqlserver.jdbc.SQLServerXADataSource に設定します。

- e. **【次へ】** をクリックし、**【完了】** をクリックします。

## 複数の MDM Hub マスターデータベース用の WebSphere の設定

複数の MDM Hub マスターデータベースインスタンスを設定する場合、MDM Hub マスターデータベースインスタンスと同数の WebSphere プロファイルを設定します。各 MDM Hub マスターデータベースインスタンスには、独自の MDM Hub インスタンスが必要です。そのため、同数の WebSphere プロファイルを作成して各 MDM Hub インスタンスを個別の WebSphere プロファイルでデプロイします。

## HTTPS プロトコルの設定

Hub サーバー、プロセスサーバー、ActiveVOS などの、MDM Hub コンポーネント間の通信に HTTPS プロトコルを使用するには、WebSphere アプリケーションサーバーで HTTPS プロトコルを設定します。

1. SSL 対応の WebSphere ポートを作成します。
2. 自己署名証明書を許可するように WebSphere を設定します。
3. 次のカスタム JVM プロパティを設定します。

カスタム JVM プロパティ	説明
javax.net.ssl.keyStore	キーストアの場所。
javax.net.ssl.keyStorePassword	キーストアのパスワード。
javax.net.ssl.keyStoreType	キーストアのタイプ。
javax.net.ssl.trustStore	トラストストアの場所。
javax.net.ssl.trustStorePassword	トラストストアのパスワード。
javax.net.ssl.trustStoreType	トラストストアのタイプ。

HTTPS プロトコルの設定の詳細については、WebSphere のマニュアルを参照してください。

## Informatica Data Director 用の WebSphere の設定

Data Director を使用する場合、WebSphere を設定し、変更が反映されるように WebSphere を再起動します。

以下の設定を確実に実行します。

- Web コンテナのカスタムプロパティを設定します。  
WebSphere コンソールを使用して `com.ibm.ws.webcontainer.invokerequestlistenerforfilter` を `true` に設定します。Web コンテナのカスタムプロパティの設定手順については、WebSphere のマニュアルを参照してください。
- タスクの管理をサポートするには、タイムアウトプロパティの値を 2 倍に増やします。  
WebSphere コンソールを使用して以下のタスクを実行します。
  1. **WebSphere コンソールの [サーバー] > [サーバータイプ] > [WebSphere アプリケーションサーバー] > <target server name>** に移動します。
  2. **[コンテナサービス]** カテゴリで、**[トランザクションサービス]** をクリックし、タイムアウトプロパティの値を増やします。

## 第 5 章

# Hub ストアのアップグレード

この章では、以下の項目について説明します。

- [Hub ストアのアップグレードの概要, 82 ページ](#)
- [Hub ストアのクローン作成（クリーンアップグレード）, 82 ページ](#)
- [データベースを英語以外のロケールに設定, 83 ページ](#)
- [Verbose モードでの MDM Hub マスターデータベースのアップグレード, 83 ページ](#)
- [サイレントモードでの MDM Hub マスターデータベースのアップグレード, 84 ページ](#)
- [Verbose モードでのオペレーショナルリファレンスストアのデータベースのアップグレード, 85 ページ](#)
- [サイレントモードでのオペレーショナル参照ストアデータベースのアップグレード, 88 ページ](#)
- [ActiveVOS データベースの更新, 89 ページ](#)
- [アップグレードスクリプトが正常に実行されたことの確認, 90 ページ](#)

## Hub ストアのアップグレードの概要

Hub ストアとは、MDM Hub マスターデータベースおよび 1 つまたは複数のオペレーショナルリファレンスストア（ORS）のデータベースが含まれたデータベースです。データベースをアップグレードするには、配布内容に含まれているスクリプトを使用します。

データベース環境を英語以外のロケールに設定する場合は、Hub ストアをアップグレードする前に、必ず文字セットを Unicode 文字にします。アップグレードが正常に完了したら、設定するロケールを選択できます。ロケールは、データベースレベルにおいてでなく、ユーザーアカウント設定として保存されます。

**注:** データベースディレクトリまたはフォルダ名へのパスにスペースを含めないようにしてください。ディレクトリまたはフォルダ名にスペースがあるパスを指定すると、アップグレードは失敗します。

## Hub ストアのクローン作成（クリーンアップグレード）

クリーンアップグレードを実行する場合は、DBA に依頼して、マスターデータベースとオペレーショナルリファレンスストアのバックアップとクローンを作成します。クローンのデータベースを新しい環境にコピーします。マスターデータベースとオペレーショナルリファレンスストアが新しい環境でアップグレードされます。

## データベースを英語以外のロケールに設定

Hub ストアデータベースの環境を英語以外のロケールに設定する場合は、アップグレードスクリプトを実行する前に、データベースの環境で Unicode 文字セットが使用されていることを確認します。文字セットを設定するには、データベースの環境変数を使用します。

**注:** このタスクは Microsoft SQL Server 環境には必要ありません。

アップグレードスクリプトにより、メタデータが英語に翻訳され、翻訳キーがメタデータに関連付けられます。アップグレードが成功したら、各 MDM Hub コンソールユーザーはユーザーインターフェースとデータベース用にサポートされているロケールを選択できるようになります。各ユーザーのロケール選択はあらゆるユーザーデータとともにマスターデータベースに保存されます。

例えば、Oracle データベース環境に韓国語ロケールで作成されている MDM Hub ストアを考えます。アップグレードする前に、データベースの環境変数 `NLS_LANG` を必ず `KOREAN_KOREA.AL32UTF8` (韓国語の Unicode) に設定しておきます。アップグレード後は、ロケールを韓国語のままにすることも、サポートされている別のロケールを選択することもできます。

次の表に、文字セットを設定するのに使用できるデータベース環境変数を示します。

データベース	環境変数名
Oracle	<code>NLS_LANG</code>
IBM Db2	<code>DB2CODEPAGE</code>

データベース環境変数の詳細については、使用するデータベースのマニュアルを参照してください。

## Verbose モードでの MDM Hub マスタデータベースのアップグレード

MDM Hub マスタデータベースをアップグレードするには、アップグレードスクリプトを実行します。

**注:** Hub コンソールを使用してメタデータの変更を行わなかった場合、データベースのアップグレードスクリプトが失敗する可能性があります。スクリプトが失敗した場合、Informatica グローバルカスタマサポートに連絡してください。

1. コマンドプロンプトを開きます。
2. 次のディレクトリに移動します。
  - UNIX の場合: <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin
  - Windows の場合: <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>\database\bin
3. 次のコマンドを使って MDM Hub マスタデータベースのアップグレードスクリプトを実行します。
  - UNIX の場合: `sip_ant.sh updatemasterdatabase`
  - Windows の場合: `sip_ant.bat updatemasterdatabase`
4. プロンプトに回答します。

5. マスターデータベースのアップグレード時に次のプロンプトが表示されたら、下記のように回答します。

This upgrade should be performed by a DBA to grant 'create sequence' privileges for the master database. The master database does not have 'create sequence' privileges, you can either grant it now (manually) and then move forward or re-start the upgrade, or direct this process to do so for you now, and continue the current upgrade.

Do you want the process to create this privilege? Yes/No

- **No** と回答すると、アップグレードプロセスは、特権が確かにユーザーに付与されたかどうかを再びチェックし、TNS 名のプロンプトに戻る。
- **Yes** と回答した場合には、アップグレードプロセスを続行するために、次のプロンプトに回答する必要がある。

DBA のユーザー名を入力してください。

DBA パスワードを入力してください。

6. JBoss 環境で、アプリケーションサーバーを再起動します。
7. CMX\_SYSTEM アップグレードログファイルのコピーをアップグレードドキュメントディレクトリに保存します。変更スクリプトごとにログファイルが 1 つずつあります。

これらのファイルはアップグレードプロセスによって次の場所に保存されます。

- UNIX の場合:

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/server/resources/database/db\_changes/<database name: データベース名>/Master

- Windows の場合:

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>\server\resources\database\db\_changes\<database name: データベース名>\Master

## サイレントモードでの MDM Hub マスターデータベースのアップグレード

サイレントモードで MDM Hub マスターデータベースをアップグレードするには、環境に応じた適切なコマンドでアップグレードスクリプトを実行します。

**注:** コマンドラインにサイレントアップグレードプロセスを表示するには、-Dnoprompt を true に設定します。コマンドラインに開始時刻と終了時刻のみを表示するには、-Dnoprompt を true -silent に設定します。

1. コマンドプロンプトを開きます。
2. 次のディレクトリに移動します。
  - UNIX。<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin
  - Windows。<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>\database\bin
3. Oracle 環境で MDM Hub マスターデータベースをアップグレードするには、次のコマンドを実行します。

UNIX。

```
sip_ant.sh updatemasterdatabase -Dmaster.tnsname=<TNS name> -Dmaster.connectiontype=<SID or SERVICE> -  
Dmaster.server=<host name> -Dmaster.port=<port> -Dmaster.sid=<SID name> -Dmaster.username=<MDM Hub Master  
Database username> -Dmaster.password=<MDM Hub Master Database password> -Dcmx.username=<administrator  
username> -Dcmx.password=<administrator password> -Dcmx.server.masterdatabase.type=ORACLE -Dnoprompt=true
```

Windows。

```
sip_ant.bat updatemasterdatabase -Dmaster.tnsname=<TNS name> -Dmaster.connectiontype=<SID or SERVICE> -  
Dmaster.server=<host name> -Dmaster.port=<port> -Dmaster.sid=<SID name> -Dmaster.username=<MDM Hub Master  
Database username> -Dmaster.password=<MDM Hub Master Database password> -Dcmx.username=<administrator  
username> -Dcmx.password=<administrator password> -Dcmx.server.masterdatabase.type=ORACLE -Dnoprompt=true
```

4. IBM Db2 環境で MDM Hub マスターデータベースをアップグレードするには、次のコマンドを実行します。

UNIX。

```
sip_ant.sh updatemasterdatabase -Dcmx.server.masterdatabase.type=DB2 -Dmaster.hostname=<host name>  
-Dmaster.port=<port> -Dmaster.username=<MDM Hub Master Database username> -Dmaster.password=<MDM Hub  
Master Database password> -Ddba.username=<DBA username>  
-Ddba.password=<DBA password> -Dnoprompt=true
```

Windows。

```
sip_ant.bat updatemasterdatabase -Dcmx.server.masterdatabase.type=DB2 -Dmaster.hostname=<host name>  
-Dmaster.port=<port> -Dmaster.username=<MDM Hub Master Database username> -Dmaster.password=<MDM Hub  
Master Database password> -Ddba.username=<DBA username>  
-Ddba.password=<DBA password> -Dnoprompt=true
```

5. Microsoft SQL Server 環境で MDM Hub マスターデータベースをアップグレードするには、次のコマンドを実行します。

UNIX。

```
sip_ant.sh updatemasterdatabase -Dmaster.hostname=<hostname> -Dmaster.port=<port> -Dmaster.username=<MDM  
Hub Master Database username>  
-Dmaster.password=<MDM Hub Master Database password>  
-Dcmx.username=<administrator username> -Dcmx.password=<administrator password> -  
Dcmx.server.masterdatabase.type=MSSQL  
-Dmaster.database=<MDM Hub Master Database name> -Dnoprompt=true
```

Windows。

```
sip_ant.bat updatemasterdatabase -Dmaster.hostname=<hostname> -Dmaster.port=<port> -Dmaster.username=<MDM  
Hub Master Database username>  
-Dmaster.password=<MDM Hub Master Database password>  
-Dcmx.username=<administrator username> -Dcmx.password=<administrator password> -  
Dcmx.server.masterdatabase.type=MSSQL  
-Dmaster.database=<MDM Hub Master Database name> -Dnoprompt=true
```

## Verbose モードでのオペレーショナルリファレンス ストアのデータベースのアップグレード

オペレーショナルリファレンスストア（ORS）のデータベースをアップグレードするには、アップグレードスクリプトを実行します。ORS のデータベースをアップグレードする前に、MDM Hub マスターデータベースをアップグレードします。

**注:** Hub コンソールを使用してメタデータの変更を行わなかった場合、データベースのアップグレードスクリプトが失敗する可能性があります。スクリプトが失敗した場合、Informatica グローバルカスタマサポートに連絡してください。

1. アプリケーションサーバーを停止します。
2. コマンドプロンプトを開きます。
3. 次のディレクトリに移動します。
  - UNIX の場合: <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin
  - Windows の場合: <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>\database\bin

4. 次のコマンドを使って、オペレーショナルリファレンスストアのアップグレードスクリプトを実行します。
    - UNIX。./sip\_ant.sh updateorsdatabase
    - Windows。sip\_ant.bat updateorsdatabase
  5. プロンプトに回答します。
- Oracle 環境の場合、次の情報を提供してください。

プロンプト	説明
データベースタイプを入力 (ORACLE、MSSQL、DB2)	データベースタイプ。ORACLE を指定する。
Oracle 接続タイプ (SERVICE、SID) を入力。 [SERVICE]	接続タイプ。以下の値を使用する。 SERVICE Oracle に接続するサービス名を使用する。 SID Oracle に接続する Oracle システム ID を使用する。 デフォルトは SERVICE。
オペレーショナル参照ストアのデータベースホスト名を入力 [localhost]	データベースを実行しているホストの名前。
オペレーショナル参照ストアデータベースのポート番号を入力します。[1521]	データベースリスナーが使用するポート番号。デフォルトは 1521。
オペレーショナル参照ストアのデータベースサービス名を入力 [orcl]	Oracle サービスの名前。このプロンプトは、選択した Oracle 接続タイプが SERVICE の場合に表示される。
Oracle Net の接続 ID (TNS 名) を入力。[orcl]	Oracle TNS 名。デフォルトは orcl。
ORS DB の接続 URL: "jdbc:oracle:thin:@//<host_name>:<port>/<service_name>"。 接続 URL を変更しますか (y/n) [n]	Oracle 接続タイプ SERVICE の接続 URL。デフォルトの接続 URL を変更する場合は、y を入力する。デフォルトの接続 URL を使用する場合は、n を入力する。
データベース SID を入力。[orcl]	Oracle システム ID の名前。このプロンプトは、選択した Oracle 接続タイプが SID の場合に表示される。
オペレーショナル参照ストアのデータベース名を入力 [cmx_ors]	オペレーショナル参照ストアのデータベースの名前。デフォルトは cmx_ors。
オペレーショナル参照ストアのデータベースのユーザーパスワードを入力	オペレーショナル参照ストアにアクセスするためのパスワード。
リストからロケール名を入力します (de、en_US、fr、ja、ko、zh_CN) [en_US]	オペレーティングシステムのロケール。
DBA ユーザー名を入力 [sys]	DBA レベルの権限を持つユーザーの名前。

プロンプト	説明
DBA のパスワードを入力	DBA レベルの権限を持つユーザーのパスワード。
ORS タイムラインの粒度の整数コードを次のように入力: 年 5、月 4、日 3、時 2、分 1、秒 0 [3]	使用するタイムラインの単位を指定する。

IBM Db2 環境の場合、次の情報を提供してください。

プロンプト	説明
データベースタイプの入力 (ORACLE、MSSQL、DB2)	データベースタイプ。DB2 を指定する。
オペレーショナル参照ストアのデータベースホスト名を入力 [localhost]	データベースを実行しているホストの名前。
オペレーショナルリファレンスストアデータベースのポート番号を入力 [50000]	データベースリスナが使用するポート番号。デフォルトは 50000。
オペレーショナルリファレンスストアのデータベース名を入力 [SIP97]	データベースの名前。デフォルトは SIP97。
オペレーショナル参照ストアのデータベース名を入力 [cmx_ors]	オペレーショナル参照ストアのデータベースの名前。デフォルトは cmx_ors。
オペレーショナル参照ストアのデータベースのユーザーパスワードを入力	オペレーショナル参照ストアにアクセスするためのパスワード。
リスト (de、en_US、fr、ja、ko、zh_CN) からロケール名を入力。[en_US]	オペレーティングシステムのロケール。デフォルトは en_US。
DBA ユーザー名を入力 [sys]	DBA レベルの権限を持つユーザーの名前。
DBA のパスワードを入力	DBA レベルの権限を持つユーザーのパスワード。
ORS タイムラインの粒度の整数コードを次のように入力: 年 5、月 4、日 3、時 2、分 1、秒 0 [3]	使用するタイムラインの単位を指定する。

Microsoft SQL Server 環境の場合、次の情報を提供してください。

プロンプト	説明
データベースタイプの入力 (ORACLE、MSSQL、DB2)	データベースタイプ。MSSQL を指定する。
オペレーショナル参照ストアのデータベースホスト名を入力 [localhost]	データベースを実行しているホストの名前。
オペレーショナルリファレンスストアデータベースのポート番号を入力 [1433]	データベースリスナが使用するポート番号。デフォルトは 1433。

プロンプト	説明
オペレーショナル参照ストアのデータベース名を入力 [cmx_ors]	オペレーショナル参照ストアのデータベースの名前。デフォルトは cmx_ors。
オペレーショナル参照ストアのデータベースのユーザーパスワードを入力	オペレーショナル参照ストアにアクセスするためのパスワード。
リスト (de、en_US、fr、ja、ko、zh_CN) からロケール名を入力。[en_US]	オペレーティングシステムのロケール。デフォルトは en_US。
DBA ユーザー名を入力 [sys]	DBA レベルの権限を持つユーザーの名前。
DBA のパスワードを入力	DBA レベルの権限を持つユーザーのパスワード。
ORS タイムラインの粒度の整数コードを次のように入力: 年 5、月 4、日 3、時 2、分 1、秒 0 [3]	使用するタイムラインの単位を指定する。

6. CMX\_ORS アップグレードログファイルのコピーをアップグレードドキュメントディレクトリに保存します。変更スクリプトごとにログファイルが 1 つずつあります。

これらのファイルはアップグレードプロセスによって次の場所に保存されます。

- UNIX の場合:

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/server/resources/database/db\_changes/<database name: データベース名>/ORS

- Windows の場合:

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>\server\resources\database\db\_changes\<database name: データベース名>\ORS

**重要:** sip\_ant ログファイルは、コマンドラインから sip\_ant を実行するたびに上書きされます。sip\_ant スクリプトを実行して別の ORS をアップグレードする前に、バックアップコピーを保存する必要があります。

## 関連項目：

- [「MDM Hub 環境レポートの保存」 \(ページ 138\)](#)

# サイレントモードでのオペレーショナル参照ストアデータベースのアップグレード

サイレントモードでオペレーショナル参照ストアデータベースをアップグレードするには、環境に応じた適切なコマンドでアップグレードスクリプトを実行します。

**注:** コマンドラインにサイレントアップグレードプロセスを表示するには、-Dnoprompt を true に設定します。コマンドラインにサイレントアップグレードプロセスの開始時刻と終了時刻のみを表示するには、-Dnoprompt を true -silent に設定します。

1. コマンドプロンプトを開きます。
2. 次のディレクトリに移動します。

- UNIX。 <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/database/bin
  - Windows。 <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>\database\bin
3. Oracle 環境でオペレーショナル参照ストアをアップグレードするには、次のコマンドを実行します。
- UNIX。
- ```
sip_ant.sh updateorsdatabase -Dors.tnsname=<tns name> -Dors.connectiontype=sid -Dors.hostname=<host name> -Dors.port=<port> -Dors.sid=<Oracle SID> -Ddba.username=<Database administrator username> -Ddba.password=<Database administrator password> -Dors.username=<ORS username> -Dors.password=<ORS password> -Dcmx.server.masterdatabase.type=oracle -Dnoprompt=true
```
- Windows。
- ```
sip_ant.bat updateorsdatabase -Dors.tnsname=<tns name> -Dors.connectiontype=sid -Dors.hostname=<host name> -Dors.port=<port> -Dors.sid=<Oracle SID> -Ddba.username=<Database administrator username> -Ddba.password=<Database administrator password> -Dors.username=<ORS username> -Dors.password=<ORS password> -Dcmx.server.masterdatabase.type=oracle -Dnoprompt=true
```
4. IBM Db2 環境でオペレーショナル参照ストアをアップグレードするには、次のコマンドを実行します。
- UNIX。
- ```
sip_ant.sh updateorsdatabase -Dors.hostname=<host name> -Dors.database=%db2database% -Dors.port=<port> -Dors.username=<ORS username> -Dors.password=<ORS password> -Dcmx.server.masterdatabase.type=db2 -Dnoprompt=true
```
- Windows。
- ```
sip_ant.bat updateorsdatabase -Dors.hostname=<host name> -Dors.database=%db2database% -Dors.port=<port> -Dors.username=<ORS username> -Dors.password=<ORS password> -Dcmx.server.masterdatabase.type=db2 -Dnoprompt=true
```
5. Microsoft SQL Server 環境でオペレーショナル参照ストアをアップグレードするには、次のコマンドを実行します。
- UNIX。
- ```
sip_ant.sh updateorsdatabase -Dors.hostname=<hostname> -Dors.port=<port> -Dors.username=<ORS username> -Dors.password=<ORS password> -Dors.database=<database name> -Dcmx.server.masterdatabase.type=MSSQL -Dnoprompt=true
```
- Windows。
- ```
sip_ant.bat updateorsdatabase -Dors.hostname=<hostname> -Dors.port=<port> -Dors.username=<ORS username> -Dors.password=<ORS password> -Dors.database=<database name> -Dcmx.server.masterdatabase.type=MSSQL -Dnoprompt=true
```

## ActiveVOS データベースの更新

ActiveVOS を以前のバージョンから 9.2.7 にアップグレードする場合は、インストールファイルを抽出した後で、次の手順を必ず実行してください。

1. 次のディレクトリに移動します。  
<AVOS\_INS\_PATH>\server\server-enterprise\jboss\_config\ddls\updates\enterprise
2. データベースタイプに応じて、SQL パッチファイルを開きます。
  - **ActiveBPEL\_Enterprise-Oracle-Patch-<バージョン>** (Oracle データベースの場合)
  - **ActiveBPEL\_Enterprise-SQLServer-Patch-<バージョン>** (SQL Server データベースの場合)
  - **ActiveBPEL\_Enterprise-DB2-Patch-<バージョン>** (DB2 データベースの場合)

- **ActiveBPEL\_Enterprise-MySQL-Patch-<バージョン>** (MySQL データベースの場合)
3. 選択したパッチファイルから、任意のデータベースクエリツールで SQL クエリを実行します。

## アップグレードスクリプトが正常に実行されたことの確認

C\_REPOS\_DB\_CHANGE テーブルをチェックして、Hub ストアのアップグレードスクリプトが正常に実行されたことを確認します。

スクリプトは、前回のアップグレード中に実行されなかった場合には、アップグレードプロセス中に実行されます。スクリプトが失敗したことが C\_REPOS\_DB\_CHANGE テーブルで示された場合は、Informatica グローバルカスタマサポートに連絡してください。

## 第 6 章

# Hub サーバーのアップグレード (インプレースアップグレード)

この章では、以下の項目について説明します。

- [Hub サーバーのアップグレードの概要, 91 ページ](#)
- [グラフィカルモードでの Hub サーバーのアップグレード, 92 ページ](#)
- [コンソールモードでの Hub サーバーのアップグレード, 95 ページ](#)
- [サイレントモードでの Hub サーバーのアップグレード, 98 ページ](#)
- [patchInstallSetup スクリプトの実行, 100 ページ](#)
- [アップグレードドキュメントフォルダへの Hub サーバーのログファイルのコピー, 101 ページ](#)
- [Hub Server のアップグレードの再適用 \(オプション\) , 101 ページ](#)

## Hub サーバーのアップグレードの概要

Hub サーバーでは MDM のコア/一般サービス（アクセス、セキュリティ、およびセッション管理など）が実行されます。Hub サーバーはアプリケーションサーバー環境にデプロイされます。

**注:** 本章はインプレースアップグレードについてのみです。クリーンアップグレードについては、*Multidomain MDM のインストールガイド*の手順に従って、ご使用のアプリケーションサーバーおよびデータベース環境に Hub サーバーをインストールしてください。

Hub サーバーは、グラフィカルモード、コンソールモード、またはサイレントモードでアップグレードできます。Hub サーバーをグラフィカルモードまたはコンソールモードでアップグレードするには、ディストリビューション内に用意されている Hub サーバーのインストーラを実行します。Hub サーバーをサイレントモードでアップグレードするには、サイレントインストーラのプロパティファイルを設定します。

サポートされていないバージョンの ActiveVOS サーバーを使用する場合、Hub サーバーのアップグレード中に、ActiveVOS サーバーをインストールするように求められます。

Hub サーバーのインストーラでは、パッチインストールとフルインストールが区別され、インストール時にユーザーは既存の MDM Hub インストールをターゲットとして指定します。既存の Hub サーバーインストールを上書きする前に、MDM Hub インストーラは重要なファイルのバックアップを作成します。

# グラフィカルモードでの Hub サーバーのアップグレード

Hub サーバーをグラフィカルモードでアップグレードするには、Hub サーバーのインストーラを実行します。

**ヒント:** 埋め込みの ActiveVOS を使用する場合、アップグレードプロセスで、更新されたバージョンの ActiveVOS を新しいディレクトリにインストールすることが求められます。既存の ActiveVOS インストールを上書きすることを選ぶ場合、アップグレードを開始する前に、../hub/server/bin/build.properties ファイルを開き、activevos.install.dir = <AVOS installed path>プロパティをコメントアウトします。その他のアップグレード前タスクについては、「[BPM アップグレードの準備](#)」(ページ 26)を参照してください。

1. Hub サーバーのインストールに使用したユーザー名を使用してログインします。
2. Hub サーバーがデプロイされているアプリケーションサーバーを起動します。
3. コマンドプロンプトを開き、配布ディレクトリ内の Hub サーバーのインストーラに移動します。デフォルトでは、インストーラは以下のディレクトリにあります。
  - UNIX の場合: <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/<operating system name: オペレーティングシステム名>/mrmsserver
  - Windows の場合: <MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>\windows\mrmsserver
4. 次のコマンドを実行します。
  - UNIX の場合: hub\_install.bin
  - Windows の場合: hub\_install.exe
5. **【導入】** ページで、**【次へ】** をクリックします。  
**【使用許諾契約を確認】** ページが表示されます。
6. 使用許諾契約の条件を確認して同意します。**【次へ】** をクリックします。
7. Hub サーバーのインストール場所を選択します。Hub サーバーのインストールフォルダには siperian-mrm.ear ファイルが含まれます。**【次へ】** をクリックします。
8. Customer 360 または Supplier 360 を使用する場合は、次のタスクを実行します。
  - a. すべてのドラフトレコードが Customer 360 または Supplier 360 で送信されていることを確認します。  
ドラフトレコードを送信しない場合は、それらを確認して削除するかアップグレードを続行することができます。
  - b. **【続行】** をクリックします。
9. バージョンの警告メッセージが表示された場合は、**【OK】** をクリックして既存の Hub サーバーのインストールをアップグレードします。
10. **【Hub コンソールのサーバーの詳細の構成】** ページで、次の詳細を入力します。
  - パブリックアクセス可能なホスト名。アプリケーションサーバーがバインドするサーバーの IP アドレスまたはパブリックアクセス可能なホスト名 (FQDN)。
  - HTTP ポート。Hub コンソールが使用する必要のあるサーバーの HTTP ポート。  
アプリケーションサーバー用に HTTPS が有効な場合は、アップグレード後に、build.properties ファイルでプロパティを編集して Hub コンソールクライアントを設定します。
11. 以前のインストールで WebLogic をアプリケーションサーバーとして使用した場合、WebLogic Admin のパスワードを入力するように求めるメッセージが Hub サーバーのインストーラに表示されます。WebLogic のパスワードを入力します。

12. ビジネスプロセス管理に ActiveVOS を使用する場合、このバージョンの Multidomain MDM にサポートされる埋め込みの ActiveVOS のバージョンをインストールします。

**重要:** スタンドアロン ActiveVOS はサポートされません。サポート対象バージョンの埋め込み ActiveVOS がすでにインストールされている場合、この手順のいくつかのサブ手順を省略できます。サブ手順 c. を実行してデータベース詳細とサブ手順 12f. に進み、ActiveVOS サーバー管理コンソールのユーザー名とパスワードを指定します。

- a. **【はい】** を選択します。
- b. **【ActiveVOS インストールディレクトリを選択】** ページで、デフォルトのインストールパスを受け入れるか、別の場所を選択します。**【次へ】** をクリックします。
- c. **【ActiveVOS 用の J2EE データソースを設定】** ページで、ActiveVOS データベーススキーマを作成したときに指定したデータベース情報を入力し、**【次へ】** をクリックします。  
**注:** スタンドアロン ActiveVOS から埋め込み ActiveVOS に移行する場合、スタンドアロン ActiveVOS スキーマの詳細を入力します。
- d. **【ActiveVOS Web サービスの URL を指定】** ページで、デフォルトの URL を受け入れるか、または ActiveVOS Web サービスを呼び出すために使用する URL を指定します。URL にアプリケーションサーバーに接続するための正しいポート番号が含まれていることを確認します。**【次へ】** をクリックします。

この URL は、インストール後のセットアップスクリプトによって、ActiveVOS Web サービスの呼び出し、定義済みの MDM ワークフローの ActiveVOS へのデプロイ、および URN マッピングの作成に使用されます。

- e. **【ActiveVOS インストーラを選択】** ページで、**【選択】** をクリックします。配布パッケージの ActiveVOS\_Server インストールファイルを参照します。**【次へ】** をクリックします。
- f. 管理者ユーザー名とパスワードを入力し、ActiveVOS コンソールの管理者ユーザーを作成します。  
**重要:** このユーザー名とパスワードは、アプリケーションサーバーで作成した ActiveVOS コンソールのユーザー名とパスワードと同じにする必要があります。
- g. **【次へ】** をクリックします。

13. **【MDM 製品使用ツールキットを設定】** ウィンドウで、組織が属する業種と環境タイプを選択します。
14. プロキシサーバーを使用する場合は、**【はい】** を選択してプロキシサーバーの詳細情報を入力します。そうでない場合は、**【いいえ】** を選択します。

次のプロキシサーバーの詳細を入力できます。

- プロキシサーバーの名前/IP
- プロキシサーバーのポート
- プロキシサーバーのドメイン名。不要な場合は空白のままにします。
- プロキシサーバーのユーザー名。不要な場合は空白のままにします。
- プロキシサーバーのパスワード。不要な場合は空白のままにします。

15. **【次へ】** をクリックします。

**【Hub サーバーのデプロイ】** ページが表示されます。

16. Hub サーバー EAR ファイルのデプロイを自動にするか手動にするかを選択してから【次へ】をクリックします。

オプション	条件
スクリプトをこのインストール中に実行する	<p>インストールが正常終了したら Hub サーバー EAR ファイルが自動でデプロイされます。次の ActiveVOS シナリオのいずれかで、次のアプリケーションサーバー環境のいずれかを使用する場合は、このオプションを選択します。</p> <p>アプリケーションサーバー環境:</p> <ul style="list-style-type: none"> <li>JBoss スタンドアロン環境</li> <li>WebSphere スタンドアロン環境</li> </ul> <p>ActiveVOS シナリオ:</p> <ul style="list-style-type: none"> <li>ActiveVOS をインストールすることを選択し、ActiveVOS の他のバージョンはこの環境にインストールされていない。</li> <li>ActiveVOS をインストールすることを選択し、環境には ActiveVOS のサポート対象バージョンがある。サポート対象バージョンについては、PAM（製品可用性マトリックス）で確認してください。</li> <li>ActiveVOS をインストールしないことを選択する。</li> </ul> <p><b>重要:</b> ActiveVOS をインストールすることを選択しているが、環境には ActiveVOS のサポート対象外バージョンがある場合は、【後で実行する】を選択します。</p>
後で実行する	<p>Hub サーバー EAR ファイルを手動でパッケージ化してデプロイします。</p> <p>ActiveVOS の有無にかかわらず、次のアプリケーションサーバー環境のいずれかを使用する場合は、このオプションを選択します。</p> <ul style="list-style-type: none"> <li>Weblogic スタンドアロン環境</li> <li>Weblogic クラスタ化環境</li> <li>JBoss クラスタ化環境</li> <li>WebSphere クラスタ化環境</li> </ul> <p>ActiveVOS のサポート対象外バージョンがあるアプリケーションサーバー環境を使用している場合は、このオプションを選択します。ActiveVOS スキーマをサポート対象バージョンにアップグレードしてから手動でデプロイする必要があります。</p>

【サマリ】ウィンドウが表示されます。

17. いずれかのオプションを変更するには、【前へ】 ボタンをクリックして前の選択を変更します。
18. サマリウィンドウで設定内容を確認したら、【インストール】 をクリックしてインストールプロセスを開始します。

Hub サーバーのインストーラがシステムを設定する間、インストーラには【しばらくお待ちください】画面が表示されます。Hub サーバーのインストーラは重要なファイルを、MDM Hub インストールディレクトリのバックアップフォルダに格納されているアーカイブにバックアップします。アーカイブのファイル名には、次の例に示す形式が使用されます。

Informatica MDM Hub Server-2010-09-27\_12-13.jar

インストールが完了すると、【インストールの完了】ウィンドウが表示されます。

19. 【完了】 をクリックして Hub サーバーインストーラを終了します。
- 注:** アップグレードが正常に完了しなかった場合、アップグレードが失敗したことを示すウィンドウが表示され、エラーメッセージを含むログファイルの場所が示されます。
20. 【後で実行する】を選択した場合は、Hub Server EAR ファイルを再パッケージ化してデプロイします。
- a. 環境に ActiveVOS のサポート対象外バージョンがある場合は、ActiveVOS スキーマをサポート対象バージョンに更新します。

- b. 次のコマンドを実行して、EAR ファイルを再パッケージします。

UNIX の場合:

```
cd <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/bin  
./sip_ant.sh repack
```

Windows の場合:

```
cd <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\server\bin  
sip_ant.bat repack
```

- c. アプリケーションサーバーの管理コンソールで、Hub サーバーの EAR ファイルを手動でデプロイします。アプリケーションサーバーのドキュメントを参照してください。

21. アプリケーションサーバーを再起動します。

## コンソールモードでの Hub サーバーのアップグレード

UNIX では、Hub サーバーをコンソールモードでアップグレードできます。

**ヒント:** ビジネスプロセス管理に埋め込みの ActiveVOS を使用する場合、アップグレードプロセスで、更新されたバージョンの ActiveVOS を新しいディレクトリにインストールすることが求められます。既存の ActiveVOS インストールを上書きすることを選ぶ場合、アップグレードを開始する前に、`../hub/server/bin/build.properties` ファイルを開き、`activevos.install.dir = <AVOS installed path>` プロパティをコメントアウトします。

1. アプリケーションサーバーを起動します。
2. MDM Hub ディストリビューション内の次のディレクトリに移動します。  
UNIX の場合: `<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/<operating system name: オペレーティングシステム名>/mrmsvr`
3. コマンドプロンプトで次のコマンドを実行します。  
`./hub_install.bin -i console`
4. アップグレードで選択するロケールに対応する番号を入力し、**Enter** キーを押します。  
アップグレードに関する概要情報が表示されます。
5. **Enter** キーを押します。  
使用許諾契約が表示されます。
6. 使用許諾契約に目を通します。使用許諾契約の条項に同意する場合は **Y** と入力し、同意しない場合は **N** と入力してインストールプログラムを終了します。
7. **Enter** キーを押します。  
前の手順で **Y** と入力した場合、インストールフォルダに関する情報が表示されます。
8. Hub サーバーをインストールしたディレクトリを指定します。
  - デフォルトのフォルダを選択する場合は、**Enter** キーを押す。
  - パスを変更する場合は、インストールフォルダの絶対パスを入力し、**Enter** キーを押します。
9. インストールフォルダの場所を確認します。インストールフォルダを確認して **Y** と入力するか、または **N** と入力してインストールフォルダを変更します。

10. Customer 360 または Supplier 360 を使用する場合は、次のタスクを実行します。
  - a. すべてのドラフトレコードが Customer 360 または Supplier 360 で送信されていることを確認します。  
ドラフトレコードを送信しない場合は、それらを確認して削除するかアップグレードを続行することができます。
  - b. **【続行】** をクリックします。
11. **1** を入力して続行するか、**2** を入力してキャンセルします。デフォルトのオプションは **2** です。
12. **Enter** キーを押して、続行することを確認します。
13. Hub コンソール用にサーバーの詳細を設定します。
  - アプリケーションサーバーがバインドするサーバーの IP アドレスまたは完全修飾ホスト名 (FQDN)。
  - Hub コンソールが使用する必要のあるサーバーの HTTP ポート。  
アプリケーションサーバー用に HTTPS が有効な場合は、アップグレード後に、build.properties ファイルでプロパティを編集して Hub コンソールクライアントを設定します。
14. WebLogic 環境では、WebLogic パスワードを入力して **Enter** キーを押します。
15. ビジネスプロセス管理に埋め込みの ActiveVOS を使用する場合は、このバージョンの Multidomain MDM に推奨される埋め込みの ActiveVOS のバージョンをインストールします。

**重要:** スタンドアロン ActiveVOS はサポートされません。サポート対象バージョンの埋め込み ActiveVOS がすでにインストールされている場合、この手順のいくつかのサブ手順を省略できます。サブ手順 c. を実行してデータベース詳細を入力し、サブ手順 12f. に進み、ActiveVOS サーバー管理コンソールのユーザー名とパスワードを指定します。

  - a. [y=Yes] で **Enter** キーを押します。
  - b. ActiveVOS サーバーをインストールする場所を指定します。
  - c. ActiveVOS データベーススキーマを作成したときに指定したデータベース情報を入力し、**【次へ】** をクリックします。

**注:** スタンドアロン ActiveVOS から埋め込み ActiveVOS に移行する場合、スタンドアロン ActiveVOS スキーマの詳細を入力します。
  - d. MDM Web サービスおよび ActiveVOS Web サービスを呼び出すのに使用する URL を指定します。  
URL にアプリケーションサーバーに接続するための正しいポート番号が含まれていることを確認します。
  - e. [ActiveVOS インストーラ] ページで **【選択】** をクリックし、配布パッケージ内の ActiveVOS\_Server インストールファイルを指定します。
  - f. ユーザー名とパスワードを入力し、ActiveVOS サーバー管理コンソールで管理者ユーザーを作成します。

**重要:** このユーザー名とパスワードは、アプリケーションサーバーで作成した ActiveVOS コンソールのユーザー名とパスワードと同じにする必要があります。
16. **Enter** キーを押します。  
Informatica Platform のインストールプロンプトが表示されます。
17. Informatica Platform をインストールする場合、続行するには **Enter** キーを押します。インストールせずにキャンセルする場合は、**2** を入力して **Enter** を押します。  
Informatica Platform のインストール応答ファイルおよびアーカイブファイルの場所に関するプロンプトが表示されます。
18. Informatica Platform のインストール応答ファイルおよびアーカイブファイルの場所を入力して、**Enter** キーを押します。

19. 製品使用ツールキットオプションを指定します。
- 組織が属する業界を入力し、**Enter** キーを押します。
  - 環境タイプを入力します。Production に 1、Test/QA に 2、Development に 3 を入力してから **Enter** キーを押します。
20. プロキシサーバーがあるかどうかを選択します。ある場合は、**Enter** キーを押します。そうでない場合は、2 を入力して **Enter** キーを押します。
- 次のプロキシサーバーの詳細を入力できます。
- プロキシサーバーの名前/IP
  - プロキシサーバーのポート
  - プロキシサーバーのドメイン名。不要な場合は空白のままにします。
  - プロキシサーバーのユーザー名。不要な場合は空白のままにします。
  - プロキシサーバーのパスワード。不要な場合は空白のままにします。
- インストールの設定のサマリが表示されます。
21. インストール中に postInstallSetup を実行するか、後で手動で実行するか、いずれかの方法を選択します。
22. インストール中に postInstallSetup を実行するか、後で手動で実行するか、いずれかの方法を選択します。

オプション	条件
スクリプトをこのインストール中に実行する	<p>インストールが正常終了したら Hub サーバー EAR ファイルが自動でデプロイされます。次の ActiveVOS シナリオのいずれかで、次のアプリケーションサーバー環境のいずれかを使用する場合は、このオプションを選択します。</p> <p>アプリケーションサーバー環境:</p> <ul style="list-style-type: none"> <li>JBoss スタンドアロン環境</li> <li>WebSphere スタンドアロン環境</li> </ul> <p>ActiveVOS シナリオ:</p> <ul style="list-style-type: none"> <li>ActiveVOS をインストールすることを選択し、ActiveVOS の他のバージョンはこの環境にインストールされていない。</li> <li>ActiveVOS をインストールすることを選択し、環境には ActiveVOS のサポート対象バージョンがある。サポート対象バージョンについては、PAM（製品可用性マトリックス）で確認してください。</li> <li>ActiveVOS をインストールしないことを選択する。</li> </ul> <p><b>重要:</b> ActiveVOS をインストールすることを選択しているが、環境には ActiveVOS のサポート対象外バージョンがある場合は、<b>【後で実行する】</b>を選択します。</p>
後で実行する	<p>Hub サーバー EAR ファイルを手動でパッケージ化してデプロイします。</p> <p>ActiveVOS の有無にかかわらず、次のアプリケーションサーバー環境のいずれかを使用する場合は、このオプションを選択します。</p> <ul style="list-style-type: none"> <li>Weblogic スタンドアロン環境</li> <li>Weblogic クラスタ化環境</li> <li>JBoss クラスタ化環境</li> <li>WebSphere クラスタ化環境</li> </ul> <p>ActiveVOS のサポート対象外バージョンがあるアプリケーションサーバー環境を使用している場合は、このオプションを選択します。ActiveVOS スキーマをサポート対象バージョンにアップグレードしてから手動でデプロイする必要があります。</p>

23. **Enter** キーを押します。
- アップグレードの設定のサマリが表示されます。

24. アップグレード前のサマリの情報を確認します。情報が正しければ、**Enter** キーを押してアップグレードを開始します。変更する必要がある場合は、特定の情報に対して BACK と入力して変更を行います。  
プロセスが完了すると、アップグレードの完了に関する情報が表示されます。
25. **Enter** キーを押してインストーラを終了します。

## サイレントモードでの Hub サーバーのアップグレード

サイレントモードでは、ユーザーとの対話なしで Hub サーバーをアップグレードできます。複数のインストールがある場合や、マシンクラスでアップグレードする必要がある場合は、サイレントモードを実行できます。サイレントアップグレードでは、進捗やエラーメッセージは表示されません。

Hub サーバーのサイレントアップグレードを実行する前に、サイレントアップグレードのプロパティファイルを設定しておく必要があります。インストーラはファイルを読み込んでアップグレードオプションを確認します。サイレントアップグレードのプロセスは、不正なアプリケーションサーバーパスまたはポートなど、設定が正しくない場合でも正常に完了する場合があります。プロパティファイルに正しい設定がされていることを確認する必要があります。

Hub サーバーのインストール先となるマシンのハードディスクに、Hub サーバーアップグレードファイルをコピーします。サイレントモードでアップグレードするには、以下のタスクを実行します。

1. インストールプロパティファイルを設定し、そのプロパティファイル内でインストールオプションを指定する。
2. インストールプロパティファイルを使用してアップグレードを実行する。

### プロパティファイルの設定

サイレントアップグレードプロセスに影響を与えるプロパティファイルのパラメータの値を確認します。

1. Hub サーバーのインストール時に設定したプロパティファイルを検索します。
2. バンドルとしてライセンス供与された ActiveVOS サーバーのバージョンをインストールする場合は、ActiveVOS のプロパティをプロパティファイルに追加します。
  - a. このリリースに付属している silentInstallServer\_sample.properties ファイルを開きます。
  - b. ActiveVOS を検索します。
  - c. [ActiveVOS のインストール] セクションをプロパティファイルにコピーします。

```
#####  
##### ActiveVOS Server installation #####  
#####
```

```
## Do you want to install ActiveVOS (Yes/No)  
AVOS.INSTALL=Yes  
## Path to ActiveVOS Installer (ActiveVOS_Server_windows_9.2.4.3.exe for Windows or  
ActiveVOS_Server_unix_9.2.4.3.sh for Linux/UNIX)  
AVOS_INSTALLER_PATH=c:\ActiveVOS_Server_windows_9.2.4.3.exe  
## ActiveVOS server install directory  
AVOS_INSTALL_DIR=C:\infamd\avos\server
```

```
## Database type is the same as for HUB (There is no ability to set a different database type for  
ActiveVOS)
```

```
## Oracle connection data
## Connection Type SID or Service Name
AVOS.ORACLE.CONNECTION.TYPE="Service Name"
AVOS.DB.SERVER=localhost
AVOS.DB.PORT=1521
## Oracle SID name or service name
AVOS.DB.SID=orcl
AVOS.DB.SCHEMA_NAME=avos
AVOS.DB.PASSWORD=!!cmx!!
```

```
## DB2 connection data
AVOS.DB.SERVER=localhost
AVOS.DB.PORT=50000
AVOS.DB.DBNAME=AVOS
AVOS.DB.SCHEMA_NAME=AVOS
AVOS.DB.USER=avos
AVOS.DB.PASSWORD=!!cmx!!
```

```
## MSSQL connection data
AVOS.DB.SERVER=localhost
AVOS.DB.PORT=1433
AVOS.DB.DBNAME=avos
AVOS.DB.USER=avos
AVOS.DB.PASSWORD=!!cmx!!
```

##If you are moving from standalone ActiveVOS to embedded ActiveVOS, enter the details for the standalone ActiveVOS schema.

- d. プロパティファイルで、ActiveVOS データベースの情報を指定するとともに、サポートされている他のデータベースのプロパティを削除します。プロパティのサポートが必要な場合、自分の環境用の『*Multidomain MDM のインストールガイド*』を参照してください。

3. ライセンス供与された ActiveVOS サーバーのバージョンを使用する場合は、サイレントインストールのプロパティファイルに以下のプロパティを追加して設定します。

```
SIP.APPSERVER.WEB.URL=http://localhost:8080
## Avos console's administrator username
AVOS.CONSOLE.USER=aeadmin
## Avos console's administrator password
AVOS.CONSOLE.PASSWORD=admin
##The user name and password must be the same as the ActiveVOS Console user name and password
that was created in the application server during the pre-installation process.
```

## サイレントアップグレードの実行

プロパティファイルの設定後、サイレントアップグレードを開始できます。

1. アプリケーションサーバーが動作していることを確認します。
2. コマンドウィンドウを開きます。
3. 次のコマンドを実行します。

UNIX の場合: `./hub_install.bin -f <location of silent properties file for hub server: Hub サーバーのサイレントプロパティファイルの場所>`

Windows の場合: `hub_install.exe -f <location of silent properties file for hub server: Hub サーバーのサイレントプロパティファイルの場所>`

サイレントアップグレードがバックグラウンドで実行されます。プロセスにしばらく時間がかかる場合があります。Hub サーバーの `postInstallSetup` スクリプトをサイレントインストールの一部として実行した場合、`postinstallSetup.log` ファイルをチェックしてアップグレードが正常に行われたことを確認します。

ログファイルは、以下のディレクトリから利用できます。

UNIX の場合: `<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/logs/`

Windows の場合: `<MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\server\logs\`

# patchInstallSetup スクリプトの実行

Hub サーバーのインストール中に手動デプロイを選択した場合、patchInstallSetup スクリプトを実行する必要があります。

1. 次のディレクトリに移動します。 <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server
2. 次のコマンドを実行して、Hub サーバーアプリケーションをデプロイし、アプリケーションサーバーの設定に変更を適用します。

## UNIX の場合:

**注:** UNIX でパスワードに感嘆符 (!) を含める場合、感嘆符の前にバックスラッシュを付ける必要があります。例えば、パスワードが!!cmx!!の場合は、\\!\\cmx\\! \\!と入力します。

### WebLogic

```
patchInstallSetup.sh -Dweblogic.password=<WebLogic password> -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

**重要:** WebLogic 12.2.1.3 以降の環境で、ActiveVOS をインストールすることにした場合、または WebLogic T3S プロトコルを使用することにした場合は、決定に一致するオプションを追加します。

- インストールした ActiveVOS。-Dinstall.avos.patch=true
- 使用した T3S プロトコル。-Dweblogic.naming.protocol=t3s

### WebSphere (セキュリティ有効)

```
patchInstallSetup.sh -Dwebsphere.password=<WebSphere password> -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

### WebSphere (セキュリティ無効)

```
patchInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

### JBoss

```
patchInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

## Windows の場合:

### WebLogic

```
patchInstallSetup.bat -Dweblogic.password=<WebLogic password> -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

**重要:** WebLogic 12.2.1.3 以降の環境で、ActiveVOS をインストールすることにした場合、または WebLogic T3S プロトコルを使用することにした場合は、決定に一致するオプションを追加します。

- インストールした ActiveVOS。-Dinstall.avos.patch=true
- 使用した T3S プロトコル。-Dweblogic.naming.protocol=t3s

### WebSphere (セキュリティ有効)

```
patchInstallSetup.bat -Dwebsphere.password=<WebSphere password> -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

WebSphere（セキュリティ無効）

```
patchInstallSetup.bat -Ddatabase.password=<MDM Hub Master database password> -  
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -  
Davos.jdbc.database.password=<ActiveVOS database password>
```

JBoss

```
patchInstallSetup.bat -Ddatabase.password=<MDM Hub Master database password> -  
Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -  
Davos.jdbc.database.password=<ActiveVOS database password>
```

## アップグレードドキュメントフォルダへの Hub サーバーのログファイルのコピー

Hub サーバーのログファイルのコピーを保存します。これらのログファイルは、アップグレードのトラブルシューティングを行う場合に使用します。

Hub サーバーのログファイルを upgradedoc アップグレードドキュメントフォルダにコピーします。これらのファイルは hub\_server\_upgrade などの別のフォルダに保存してください。クラスタ内の複数の Hub サーバーをアップグレードした場合は、Hub サーバーの各インスタンスのファイルを個別のフォルダに保存します。

次の表に、コピーするログファイルを示します。

ファイル	説明
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/Infamdm_Hub_Server_InstallLog.xml	Hub サーバーのインストールに関するログメッセージを含む。
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/infamdm_installer_debug.txt	デバッグメッセージと、アップグレードプロセスの実行時に選択したすべてのオプションが含まれる。
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/logs/patchInstallSetup.log	patchInstallSetup スクリプトの実行結果が含まれる。
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/logs/cmserver.log	Hub サーバーログが含まれる。このファイルは、Hub サーバーの起動時に Hub サーバーにより作成される。
アプリケーションサーバーのログファイル。	ツリーの、アプリケーションサーバーのインストールディレクトリ内にあります。

## Hub Server のアップグレードの再適用（オプション）

Hub サーバーのアップグレードが完了した場合に、Hub サーバーのアップグレードを再適用することはアップグレードプロセスによって許可されません。Hub サーバーのアップグレードを再適用する例は、アップグレー

ドプロセス中にハードウェアが故障した場合などです。また、このプロシージャは、アップグレードのテストを行って、前のバージョンのソフトウェアに戻す場合にも実行できます。

1. 次のディレクトリにある `siperian-mrm.ear` ファイルをバックアップします。
  - UNIX の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server
  - Windows の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\server
2. アップグレード手順を繰り返します。インストールコマンドにパラメータ-  
`DSIPERIAN_FORCED_PATCH_INSTALL=true` を追加します。  
例えば、UNIX でアップグレードをグラフィカルモードで再適用する場合は、次のコマンドを実行します。  
`hub_install.bin -DSIPERIAN_FORCED_PATCH_INSTALL=true`

## 第 7 章

# プロセスサーバーのアップグレード ド（インプレースアップグレード）

この章では、以下の項目について説明します。

- [Process サーバーアップグレードの概要, 103 ページ](#)
- [グラフィカルモードでのプロセスサーバーのアップグレード, 104 ページ](#)
- [コンソールモードでのプロセスサーバーのアップグレード, 106 ページ](#)
- [サイレントモードでのプロセスサーバーのアップグレード, 107 ページ](#)
- [Informatica Address Verification 5 統合へのアップグレード手順, 109 ページ](#)
- [一致ポピュレーションの設定, 111 ページ](#)
- [アップグレードドキュメントディレクトリへのプロセスサーバーのログファイルのコピー, 113 ページ](#)
- [プロセスサーバーのアップグレードの再適用（オプション）, 113 ページ](#)

## Process サーバーアップグレードの概要

Process サーバーは、データクレンジング操作、一致操作、およびバッチジョブを取り扱うサプレットです。Process サーバーをアップグレードするには、配布内容に含まれている Process サーバーのインストーラを実行します。アップグレードプロセス時に既存の MDM Hub インストール先をターゲットの場所として選択すると、Process サーバーのインストーラにより、フルインストールでなくアップグレードと認識されます。既存の Process サーバーインストールを上書きする前に、Process サーバーインストーラは重要なファイルのバックアップを作成します。

**注:** 本章はインプレースアップグレードについてのみです。クリーンアップグレードについては、*Multidomain MDM のインストールガイド*の手順に従って、ご使用のアプリケーションサーバーおよびデータベース環境にプロセスサーバーをインストールしてください。

# グラフィカルモードでのプロセスサーバーのアップグレード

プロセスサーバーをグラフィカルモードでアップグレードするには、プロセスサーバーのインストーラを実行します。

1. Process サーバーのインストール時に使用したユーザー名でログインします。
2. Process サーバーがデプロイされているアプリケーションサーバーを起動します。
3. コマンドプロンプトを開き、配布ディレクトリ内のプロセスサーバーのインストーラに移動します。デフォルトでは、インストーラは以下のディレクトリにあります。
  - UNIX の場合:<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ><operating system name: オペレーティングシステム名>/mrncleanse
  - Windows の場合:<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>\windows\mrncleanse
4. 次のコマンドを実行します。
  - UNIX の場合:hub\_cleansse\_install.bin
  - Windows の場合:hub\_cleansse\_install.exe
5. **【概要】** ウィンドウで **【次へ】** をクリックします。  
**【使用許諾契約】** ウィンドウが表示されます。
6. **【使用許諾契約に同意する】** オプションを選択し、**【次へ】** をクリックします。  
**【インストールフォルダの選択】**ウィンドウが表示されます。
7. プロセスサーバーのインストール場所を選択します。プロセスサーバーのインストールフォルダには siperian-mrm-cleansse.ear ファイルが含まれます。
  - デフォルトの場所を選択するには、**【次へ】** をクリックします。
  - 別の場所を指定するには、**【選択】** をクリックして **【次へ】** をクリックします。**バージョン警告メッセージ**が表示されます。
8. **【OK】** をクリックして、続行することを確認します。  
**【ライセンスファイルの場所の入力】** ウィンドウが表示されます。
9. ライセンスファイルの場所を選択して、**【次へ】** をクリックします。
10. 以前のインストールで WebLogic をアプリケーションサーバーとして使用した場合、WebLogic Admin のパスワードを入力するように求めるメッセージが Process サーバーのインストーラに表示されます。WebLogic のパスワードを入力します。
11. **【製品使用ツールキット】** ページで **【環境タイプ】** を選択します。
12. プロキシサーバーがある場合は、**【はい】** を選択してプロキシサーバーの詳細を入力します。そうでない場合は、**【いいえ】** を選択し、**【次へ】** をクリックします。  
次のプロキシサーバーの詳細を入力できます。
  - プロキシサーバーの名前/IP
  - プロキシサーバーのポート
  - プロキシサーバーのドメイン名。不要な場合は空白のままにします。
  - プロキシサーバーのユーザー名。不要な場合は空白のままにします。
  - プロキシサーバーのパスワード。不要な場合は空白のままにします。

13. **【次へ】** をクリックします。  
[デプロイ] ページが表示されます。
14. 自動または手動のどちらでデプロイするかを選択します。自動的にデプロイするには **【はい】** をクリックし、手動でデプロイするには **【いいえ】** をクリックした後、**【次へ】** をクリックします。
  - WebSphere スタンドアロン環境または JBoss スタンドアロン環境では、自動的にデプロイするために **【はい】** をクリックして、**【次へ】** をクリックします。
  - WebLogic またはクラスタ環境では、手動でデプロイするために **【いいえ】** をクリックして、**【次へ】** をクリックします。

**【インストール前のサマリ】** ウィンドウが表示されます。
15. いずれかのオプションを変更するには、**【前へ】** ボタンをクリックして前の選択を変更します。
16. サマリウィンドウで設定内容を確認したら、**【インストール】** をクリックしてインストールプロセスを開始します。  
  
Process サーバーのインストーラがシステムを設定する間、インストーラには **【しばらくお待ちください】** 画面が表示されます。Process サーバーのインストーラは重要なファイルを、MDM Hub インストールディレクトリのバックアップフォルダに格納されているアーカイブにバックアップします。アーカイブのファイル名には、次の例に示す形式が使用されます。  
Siperian Hub Cleanse Match Server-2010-05-12\_18-09.jar  
インストールが完了すると、**【インストールの完了】** ウィンドウが表示されます。
17. **【完了】** をクリックして Process サーバーインストーラを終了します。  
**注:** アップグレードが正常に完了しなかった場合、アップグレードが失敗したことを示すウィンドウが表示され、エラーメッセージを含むログファイルの場所が示されます。
18. 手順 [14](#) で **【いいえ】** を選択した場合は、EAR ファイルを再パッケージして手動でデプロイします。
  - a. 次のコマンドを実行して、EAR ファイルを再パッケージします。  
UNIX の場合:  

```
cd <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse/bin
./sip_ant.sh repack
```

 Windows の場合:  

```
cd <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\cleanse\bin
sip_ant.bat repack
```
  - b. アプリケーションサーバーの管理コンソールでプロセスサーバーの EAR ファイルを手動でデプロイします。アプリケーションサーバーのドキュメントを参照してください。
19. SSA-Name3 ライブラリファイルを<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse/lib/upgrade/SSA から<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse/lib にコピーします。
20. アプリケーションサーバーを再起動します。

# コンソールモードでのプロセスサーバーのアップグレード

UNIX では、Process サーバーをコンソールモードでアップグレードできます。

**注:** RedHat Linux で Process サーバーをアップグレードする場合は、ルートユーザーを使用しないでください。ルートユーザーには、InstallAnywhere に必要な.profile がありません。代わりに、別のユーザープロファイルを作成して、Process サーバーをアップグレードします。

1. アプリケーションサーバーを起動します。
2. MDM Hub ディストリビューション内の次のディレクトリに移動します。  
Solaris の場合:<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/solaris/  
mrmcleanse  
HP-UX の場合:<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/hpux/mrmcleanse  
Linux の場合:<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/linux/mrmcleanse  
AIX の場合:<MDM Hub distribution directory: MDM Hub ディストリビューションディレクトリ>/aix/mrmcleanse
3. コマンドプロンプトで次のコマンドを実行します。  
./hub\_cleanse\_install.bin -i console
4. インストールで選択するロケールに対応する番号を入力し、**Enter** を押します。  
インストールに関する概要情報が表示されます。
5. **Enter** キーを押します。  
使用許諾契約が表示されます。
6. 使用許諾契約に目を通します。**Y** と入力して、使用許諾契約に同意するか、または使用許諾契約に同意しない場合は、**N** と入力してインストールプログラムを終了します。
7. **Enter** キーを押します。  
前の手順で **Y** と入力した場合、インストールフォルダに関する情報が表示されます。
8. プロセスサーバーをインストールしたディレクトリを指定します。
  - デフォルトの場所を選択する場合は、**Enter** キーを押します。
  - パスを変更する場合は、インストールフォルダの絶対パスを入力し、**Enter** キーを押します。
9. インストールフォルダの場所を確認します。インストールフォルダを確認して **Y** と入力するか、または **N** と入力してインストールフォルダを変更します。  
バージョン警告メッセージが表示されます。
10. **Enter** キーを押して、続行することを確認します。  
ライセンスファイルの場所を尋ねるプロンプトが表示されます。
11. ライセンスファイルの場所を絶対パスで入力し、**Enter** を押します。
12. WebLogic 環境では、WebLogic パスワードを入力して **Enter** キーを押します。
13. [製品使用ツールキット] オプションから、環境タイプを選択します。Production に 1、Test/QA に 2、Development に 3 を入力してから **Enter** キーを押します。
14. プロキシサーバーがあるかどうかを選択します。ある場合は、**Enter** キーを押します。そうでない場合は、2 を入力して **Enter** キーを押します。  
次のプロキシサーバーの詳細を入力できます。
  - プロキシサーバーの名前/IP

- プロキシサーバーのポート
- プロキシサーバーのドメイン名。不要な場合は空白のままにします。
- プロキシサーバーのユーザー名。不要な場合は空白のままにします。
- プロキシサーバーのパスワード。不要な場合は空白のままにします。

インストールの設定のサマリが表示されます。

15. インストール中に `postInstallSetup` を実行するか、後で手動で実行するか、いずれかの方法を選択します。
16. **Enter** キーを押します。  
アップグレードの設定のサマリが表示されます。
17. アップグレード前のサマリの情報を確認します。情報が正しければ、**Enter** キーを押してアップグレードを開始します。変更する必要がある場合は、特定の情報に対して `BACK` と入力して変更を行います。  
プロセスが完了すると、アップグレードの完了に関する情報が表示されます。
18. **Enter** キーを押してインストーラを終了します。

## サイレントモードでのプロセスサーバーのアップグレード

サイレントモードでは、ユーザーとの対話なしで Process サーバーをアップグレードできます。複数のインストールがある場合や、マシクラスタでアップグレードする必要がある場合は、サイレントモードを実行できます。サイレントアップグレードでは、進捗やエラーメッセージは表示されません。

Process サーバーのサイレントアップグレードを実行する前に、サイレントアップグレードのプロパティファイルを設定する必要があります。インストーラはファイルを読み込んでアップグレードオプションを確認します。サイレントアップグレードのプロセスは、不正なアプリケーションサーバーパスまたはポート設定など、設定が正しくない場合でも正常に完了する場合があります。プロパティファイルに正しい設定がされていることを確認する必要があります。

Process サーバーのアップグレード先となるマシンのハードディスクに、Process サーバーのアップグレードファイルをコピーします。サイレントモードでアップグレードするには、以下のタスクを実行します。

1. インストールプロパティファイルを設定し、そのプロパティファイル内でインストールオプションを指定する。
2. インストールプロパティファイルを使用してアップグレードを実行する。

### プロパティファイルの設定

サイレントアップグレードプロセスに影響を与えるプロパティファイルのパラメータの値を確認します。

1. プロセスサーバーのインストール時に設定したプロパティファイルを検索します。
2. テキストエディタを使用してファイルを開き、サイレントアップグレードプロセスに影響を与えるパラメータの値を確認します。

次の表で、確認するアップグレードパラメータについて説明します。

プロパティ名	説明
USER_INSTALL_DIR	プロセスサーバーをインストールしたディレクトリ。 例: C:\><MDM Hub installation directory: MDM Hub のインストールディレクトリ>\cleanse。 プロパティファイルでは、バックスラッシュ文字をエスケープする必要があります。インストールディレクトリパスを指定する際は、二重バックスラッシュを使用します。
SIP.APPSERVER.PASSWORD	WebLogic にアクセスするためのパスワード。WebLogic 環境用です。
RUN_DEPLOYMENT_FLAG	サイレントアップグレードの一部として、postInstallSetup スクリプトを実行します。サイレントアップグレードの最後に postInstallSetup を実行する場合は、1 を設定します。postInstallSetup を実行しない場合は、0 を設定します。

3. 次の製品使用ツールキットのプロパティをサイレントインストールのプロパティファイルに追加して設定します。

```
#Product Usage Toolkit Installation
#CSM_TYPE is the type of Product Usage Toolkit installation.
# valid values are:Production,Test,Development. Should not be blank.
CSM_TYPE=Production

# If the network has a proxy server, fill in the following parameters (leave empty if no proxy):
# proxy server host
CSM_HOST=
# proxy server port
CSM_PORT=
# Proxy server domain name (leave blank, if not applicable)
CSM_DOMAIN=
# Proxy server user name (leave blank, if not applicable)
CSM_PROXY_USER_NAME=
#Proxy server password (leave blank, if not applicable)
CSM_PROXY_PASSWORD=
```

## プロセスサーバーのサイレントアップグレードの実行

プロパティファイルの設定後、サイレントアップグレードを開始できます。

1. アプリケーションサーバーが動作していることを確認します。
2. コマンドウィンドウを開きます。
3. 次のコマンドを実行します。

UNIX の場合: ./hub\_cleanse\_install.bin -f <location of silent properties file for process server: プロセスサーバーのサイレントプロパティファイルの場所>

Windows の場合: hub\_cleanse\_install.exe -f <location of silent properties file for process server: プロセスサーバーのサイレントプロパティファイルの場所>

サイレントアップグレードがバックグラウンドで実行されます。プロセスにしばらく時間がかかる場合があります。Process サーバーのインストール後スクリプトをサイレントインストールの一部として実行した場合には、postinstallSetup.log ファイルをチェックしてアップグレードが正常に行われたことを確認します。

ログファイルは、以下のディレクトリから利用できます。

UNIX の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse/logs/

Windows の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\cleanse\logs\

# Informatica Address Verification 5 統合へのアップグレード手順

この節では、MDM Hub 実装で Informatica Address Verification 5 を使用するようにアップグレードする手順について説明します。

**注:** この節は、Informatica Address Verification を使用するライセンスを持つユーザーに適用されます。

Informatica Address Verification 5 統合にアップグレードするには、次の手順を実行する必要があります。

1. cmxcleanse.properties ファイルを開きます。このファイルは次の場所にあります。

**Windows:** <infamdm\_install\_directory>\hub\cleanse\resources

**UNIX:** <infamdm\_install\_directory>/hub/cleanse/resources

2. 次の Informatica Address Verification 5 のプロパティが cmxcleanse.properties ファイルに設定されていることを確認します。

**Windows:**

```
cleanse.library.addressDoctor.property.SetConfigFile=C:\infamdm\hub\cleanse\resources\AddressDoctor\5\SetConfig.xml
cleanse.library.addressDoctor.property.ParametersFile=C:\infamdm\hub\cleanse\resources\AddressDoctor\5\Parameters.xml
cleanse.library.addressDoctor.property.DefaultCorrectionType=PARAMETERS_DEFAULT
```

**UNIX:**

```
cleanse.library.addressDoctor.property.SetConfigFile=/u1/infamdm/hub/cleanse/resources/AddressDoctor/5/SetConfig.xml
cleanse.library.addressDoctor.property.ParametersFile=/u1/infamdm/hub/cleanse/resources/AddressDoctor/5/Parameters.xml
cleanse.library.addressDoctor.property.DefaultCorrectionType=PARAMETERS_DEFAULT
```

3. プロパティファイルを保存して閉じます。
4. SetConfig.xml および Parameters.xml を、cmxcleanse.properties ファイルで指定された場所にコピーします。

SetConfig.xml ファイルのサンプルは次のとおりです。

```
<!DOCTYPE SetConfig SYSTEM 'SetConfig.dtd'>
<SetConfig>
  <General WriteXMLEncoding="UTF-16LE" WriteXMLBOM="NEVER"
    MaxMemoryUsageMB="600" MaxAddressObjectCount="10" MaxThreadCount="10" />

  <UnlockCode>79FYLUAXAVSR0KLV1TDC6PAQVVC3KM14FZC</UnlockCode>

  <DataBase CountryISO3="ALL" Type="BATCH_INTERACTIVE" Path="c:\addressdoctor\5"
    PreloadingType="NONE" />

  <DataBase CountryISO3="ALL" Type="FASTCOMPLETION" Path="c:\addressdoctor\5"
    PreloadingType="NONE" />

  <DataBase CountryISO3="ALL" Type="CERTIFIED" Path="c:\addressdoctor\5"
    PreloadingType="NONE" />

  <DataBase CountryISO3="ALL" Type="GEOCODING" Path="c:\addressdoctor\5"
    PreloadingType="NONE" />

  <DataBase CountryISO3="ALL" Type="SUPPLEMENTARY" Path="c:\addressdoctor\5"
    PreloadingType="NONE" />
</SetConfig>
```

Parameters.xml ファイルのサンプルは次のとおりです。

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE Parameters SYSTEM 'Parameters.dtd'>
<Parameters
  WriteXMLEncoding="UTF-16LE"
```

```

WriteXMLBOM="NEVER">
<Process
  Mode="BATCH"
  EnrichmentGeoCoding="ON"
  EnrichmentCASS="ON"
  EnrichmentSERP="ON"
  EnrichmentSNA="ON"
  EnrichmentSupplementaryGB="ON"
  EnrichmentSupplementaryUS="ON" />
<Input
  Encoding="UTF-16LE"
  FormatType="ALL"
  FormatWithCountry="ON"
  FormatDelimiter="PIPE" />
<Result
  AddressElements="STANDARD"
  Encoding="UTF-16LE"
  CountryType="NAME_EN"
  FormatDelimiter="PIPE" />
</Parameters>

```

5. 構成ファイル SetConfig.xml で、Informatica Address Verification 5 のロック解除コードを指定します。

SetConfig.xml ファイルおよび Parameters.xml ファイルの詳細については、Informatica Address Verification 5 のドキュメントを参照してください。

6. 次の場所から、Informatica Address Verification 5 のライブラリをコピーします。

**Windows:** <infadm\_install\_directory>\hub\cleanse\lib\upgrade\AddressDoctor

**UNIX:** <infadm\_install\_directory>/hub/cleanse/lib/upgrade/AddressDoctor

7. JADE.dll（または同等の Informatica Address Verification 4 ライブラリ）を、次の場所にある Informatica Address Verification 5 ライブラリに置き換えます。

**Windows:** <infadm\_install\_directory>\hub\cleanse\lib

**UNIX:** <infadm\_install\_directory>/hub/cleanse/lib

詳細については、以下の場所にある libupdate\_readme.txt ドキュメントを参照してください。

**Windows:** <infadm\_install\_directory>\hub\cleanse\lib\upgrade

**UNIX:** <infadm\_install\_directory>/hub/cleanse/lib/upgrade

8. アプリケーションサーバーを再起動します。

現在実行しているアプリケーションサーバーと同じユーザー名でログインし、アプリケーションサーバー起動中に例外が発生しないことを確認します。

9. Process サーバーを再起動します。

Process サーバーの初期化中に、ターミナルコンソールに次のようなメッセージが表示されます。

[INFO ] com.siperian.mrm.cleanse.addressDoctor.Library: Initializing AddressDoctor5

10. クレンジング関数というツールを起動します。

11. 書き込みロックを取得します（[書き込みロック] > [ロックの取得]）。

12. Informatica Address Verification クレンジング関数を選択します。

13. **[更新]** ボタンをクリックします。

Informatica Address Verification 5 クレンジング関数が Informatica Address Verification クレンジング関数ノードに追加されます。

# 一致ポピュレーションの設定

一致ポピュレーションには、マッチプロセスに使用する標準ポピュレーションセットが含まれます。サポートされている国、言語、またはポピュレーションごとに標準ポピュレーションセットがあります。マッチルールに使用する一致ポピュレーションを有効にする必要があります。

一致ポピュレーションは、Informatica MDM Hub のインストールの *population.ysp* ファイルとして用意されています。ポピュレーション名は ysp ファイル名と同じです。Japanese ポピュレーションを追加する際に、Person\_Name\_Kanji 一致フィールドを使用する場合は、\_Kanji をポピュレーション名に追加します。例えば、Japan\_Kanji または Japan\_i\_Kanji になります。この場合、標準の Person\_Name 一致フィールドは使用できません。

使用するポピュレーションには、SSA-Name3 バージョンの MDM Hub との互換性が必要です。追加のポピュレーションファイルが必要な場合、または更新されたポピュレーションファイルを新しいバージョンにアップグレードする必要がある場合は、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。この製品で要求する最初のポピュレーションファイルは無料です。他の国用のポピュレーションファイルが必要な場合や、MDM Hub の最新バージョンにアップグレードするために更新されたポピュレーションファイルが必要な場合があります。

## カスタムポピュレーションファイルのアップグレード

アップグレードプロセス中に、ポピュレーションファイルがアップグレードされます。ポピュレーションファイルをカスタマイズした場合は、Informatica グローバルカスタマサポートに問い合わせ、アップグレードされたプロパティファイルにカスタマイズを適用します。

## 一致ポピュレーションの有効化

マッチルールに使用する一致ポピュレーションを有効にする必要があります。

1. <population: ポピュレーション>.ysp ファイルを以下の場所にコピーします。  
UNIX の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse/resources/match  
Windows の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\cleanse\resources\match
2. C\_REPOS\_SSA\_POPULATION メタデータテーブルで、ポピュレーションが登録されていることを確認します。  
MDM Hub インストールのシードデータベースには、C\_REPOS\_SSA\_POPULATION テーブルに登録されたいくつかのポピュレーションがありますが、有効にはなっていません。
3. C\_REPOS\_SSA\_POPULATION テーブルにポピュレーションが含まれていない場合、それをテーブルに追加して有効にします。  
ポピュレーション名は ysp ファイル名と同じです。例えば、ysp ファイルの名前が US.ysp なら、ポピュレーション名は US です。  
ポピュレーションをオペレーショナルリファレンスストアに追加するには、以下の手順を実行します。  
IBM Db2 または Oracle の場合:
  - a. ポピュレーションを追加するオペレーショナルリファレンスストアスキーマに接続します。
  - b. SQL\*Plus で、次のディレクトリにあるスクリプト (add\_std\_ssa\_population.sql) を実行します。  
UNIX の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/server/resources/database/custom\_scripts/oracle

Windows の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\server  
\resources\database\custom\_scripts\oracle

- c. 以下の表に説明する内容に従ってプロンプトに回答します。

プロンプト	説明
追加するポピュレーションを入力します	ポピュレーションの名前。
ROWID_SSA_POP の値を入力します (例: INFA.0001)。デフォルトは [INFA.0001]	C_REPOS_SSA_POPULATION メタデータテーブルの ROWID_SSA_POP カラムの一意の値。デフォルトは INFA.0001 です。

ポピュレーションは C\_REPOS\_SSA\_POPULATION テーブルに登録されています。

- d. 次のコマンドを実行して、ポピュレーションを有効にします。

```
UPDATE c_repos_ssa_population SET enabled_ind = 1 WHERE population_name = '<Your Population>';
COMMIT;
```

Microsoft SQL Server の場合:

- a. 次のディレクトリにある add\_std\_ssa\_population.bat スクリプトを実行します。

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>\server\resources\database  
\custom\_scripts\MSSQL

- b. 以下の表に説明する内容に従ってプロンプトに回答します。

プロンプト	説明
CMX_ORS DB ("localhost") の MSSQL インスタンスを持つホスト名	Microsoft SQL Server インスタンスのホストの名前。
cmx_ors ユーザー名 ("cmx_ors")	オペレーショナル参照ストアのユーザー名。
cmx_ors ユーザーパスワード	オペレーショナル参照ストアのパスワード。
ポピュレーション名を入力します (注: Japan ポピュレーションまたは Japan_i ポピュレーションに Person_Name_Kanji を使用する場合は、接尾辞 「_Kanji」 を名前の最後に追加します)。デフォルトは (" ")	ポピュレーションの名前。
ROWID_SSA_POP の値を入力します (例: INFA.0001)。デフォルトは (INFA.0001)	C_REPOS_SSA_POPULATION メタデータテーブルの ROWID_SSA_POP カラムの一意の値。

ポピュレーションは C\_REPOS\_SSA\_POPULATION テーブルに登録されています。

- c. 次のコマンドを実行して、ポピュレーションを有効にします。

```
USE <Operational Reference Store user: オペレーショナル参照ストアユーザー>
GO
UPDATE [dbo].[C_REPOS_SSA_POPULATION] SET ENABLED_IND = 1 WHERE POPULATION_NAME = '<population: ポピュレーション>'
```

4. Process サーバーを再起動します。

- Hub コンソールにログインし、ポピュレーションが有効になっていることを確認します。  
ポピュレーションは、ベースオブジェクトの【一致/マージ設定】ユーザーインタフェースに表示されます。

## アップグレードドキュメントディレクトリへのプロセスサーバーのログファイルのコピー

Hub サーバーのログファイルのコピーを保存します。これらのログファイルは、アップグレードのトラブルシューティングを行う場合に使用します。

プロセスサーバーのログファイルをアップグレードドキュメントフォルダにコピーします。これらのファイルは cleanse\_match\_server\_upgrade などの別のフォルダに保存してください。クラスタ内の複数のプロセスサーバーをアップグレードした場合は、プロセスサーバーの各インスタンスのファイルを個別のフォルダに保存します。

次の表に、コピーするログファイルを示します。

ファイル	説明
<cleanse installation directory: クレンジングインストールディレクトリ>/hub/cleanse/Infamdm_Hub_Cleanse_Match_Server_InstallLog.xml	Process サーバーインストールのログファイルが含まれる。
<cleanse installation directory: クレンジングインストールディレクトリ>/hub/cleanse/infamdm_installer_debug.txt	デバッグメッセージと、アップグレードプロセスの実行時に選択したすべてのオプションが含まれる。
<cleanse installation directory: クレンジングインストールディレクトリ>/hub/cleanse/logs/patchInstallSetup.log	patchInstallSetup スクリプトの実行結果が含まれる。
<cleanse installation directory: クレンジングインストールディレクトリ>/hub/cleanse/logs/cmserver.log	Process サーバーのログが含まれる。
アプリケーションサーバーのログファイル。	ツリーの、アプリケーションサーバーのインストールディレクトリ内にあります。

## プロセスサーバーのアップグレードの再適用（オプション）

プロセスサーバーのアップグレードが完了した場合に、プロセスサーバーのアップグレードを再適用することはアップグレードプロセスによって許可されません。プロセスサーバーのアップグレードを再適用する例は、アップグレードプロセス中にハードウェアが故障した場合などです。また、このプロシージャは、アップグレードのテストを行って、前のバージョンのソフトウェアに戻す場合にも実行できます。

- 次のディレクトリにある siperian-mrm.ear ファイルをバックアップします。
  - UNIX の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse

- Windows の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\cleanse
2. アップグレード手順を繰り返します。インストールコマンドにパラメータ-DSIPERIAN\_FORCED\_PATCH\_INSTALL=true を追加します。  
例えば、UNIX でアップグレードをグラフィカルモードで再適用する場合は、次のコマンドを実行します。  
hub\_cleanse\_install.bin -DSIPERIAN\_FORCED\_PATCH\_INSTALL=true

## 第 8 章

# アップグレード後のタスク

この章では、以下の項目について説明します。

- [アップグレード後のタスク, 115 ページ](#)
- [Microsoft SQL Server 2017 用 JDBC ドライバの設定, 116 ページ](#)
- [更新プロパティ, 116 ページ](#)
- [JBoss のアップグレード後の作業, 117 ページ](#)
- [インブレースアップグレードのアップグレード後のタスクを実行する, 117 ページ](#)
- [オブジェクト、カラム、および廃止されたオブジェクトへの参照の削除, 118 ページ](#)
- [Hub サーバーをデプロイするための PostInstall スクリプトの実行（条件による）, 118 ページ](#)
- [Hub コンソールクライアントの設定, 119 ページ](#)
- [WebSphere 管理セキュリティの設定, 120 ページ](#)
- [クラスローダーの設定 WebSphere で, 127 ページ](#)
- [オペレーショナルリファレンスストアの登録, 128 ページ](#)
- [アップグレードされたメタデータの検証, 132 ページ](#)
- [コンテンツセキュリティポリシーのカスタマイズ, 134 ページ](#)
- [クリーンアップグレードのアップグレード後のタスクを実行する, 135 ページ](#)
- [プラットフォームトランスフォーメーションに対するクレンジング関数の設定, 137 ページ](#)
- [MDM Hub 環境レポートの確認, 138 ページ](#)
- [外部呼び出しとアプリケーションのアップグレード, 138 ページ](#)
- [EJB プロトコル用の SiperianClient ライブラリクラスのアップグレード, 140 ページ](#)
- [MDM Hub メタデータを準備する, 140 ページ](#)
- [アップグレードテスト, 141 ページ](#)
- [Hub サーバーの全般的なプロパティの設定, 143 ページ](#)
- [Data Director および Hub サーバーのプロパティ, 143 ページ](#)
- [Data Director のグローバルプロパティ, 144 ページ](#)
- [ビジネスエンティティスキーマの生成, 144 ページ](#)

## アップグレード後のタスク

クリーンアップグレードまたはインブレースアップグレードを実行するかどうかにかかわらず、環境が適切に構成されていることを確認するため、アップグレード後のタスクを実行します。

# Microsoft SQL Server 2017 用 JDBC ドライバの設定

Microsoft SQL Server 2017 を使用する環境に Hub サーバーをインストールした場合、正しいバージョンの JDBC ドライバファイルを Hub サーバーの lib ディレクトリにコピーします。

1. Microsoft SQL Server がインストールされているマシンの Binn ディレクトリで、サポートされている最新バージョンの Microsoft JDBC ドライバを見つけます。  
インストール前のタスクを実行するときに、ドライバを Microsoft Web サイトからダウンロードして Binn ディレクトリにコピーします。
2. Microsoft JDBC JAR ファイルを次のディレクトリにコピーします。  
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/lib
3. アプリケーションサーバーを再起動します。

## 更新プロパティ

アップグレードタスク	詳細
プロパティファイルのアプリケーションサーバー設定の更新	<p>アプリケーションサーバーをアップグレードする場合、プロパティファイルのアプリケーションサーバー設定を手動で更新する必要があります。</p> <p>&lt;MDM Hub installation directory: MDM Hub のインストールディレクトリ&gt;\hub\server\bin にある build.properties ファイルで、次の設定を更新します。</p> <ul style="list-style-type: none"><li>- SIP.AS.HOME</li><li>- SIP.AS.SERVER_FOLDER</li><li>- SIP.AS.DEPLOY_FOLDER</li></ul> <p>&lt;MDM Hub installation directory: MDM Hub のインストールディレクトリ&gt;\hub\server にある setSiperianEnv.bat ファイルで、次の設定を更新します。</p> <ul style="list-style-type: none"><li>- SET JBS_HOME</li><li>- SET JBS_SERVER_DIR</li><li>- SET JBS_DEPLOY_DIR</li><li>- SET JBS_CLIENT_CLASSPATH</li></ul> <p>&lt;MDM Hub installation directory: MDM Hub のインストールディレクトリ&gt;\hub\server\resources にある cmxserver.properties ファイルで、次の設定を更新します。</p> <ul style="list-style-type: none"><li>- cmx.appserver.version</li></ul> <p>ポート番号など、アプリケーションサーバーの他の構成設定を変更していた場合は、cmxserver.properties の設定も更新する必要があります。</p>

# JBoss のアップグレード後の作業

JBoss 環境で、次のタスクを実行します。

タスク	アクション
odjbc6.jar ファイルの削除	JAR ファイルを次の場所から削除します。  <JBoss install location>\modules\com\activevos\main <JBoss install location>\modules\com\informatica\mdm\jdbc\main <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\server\lib
必要に応じて、JDBC ドライバを sqljdbc42.jar に更新します。	1. JDBC ドライバをコアモジュールとしてインストールします。手順については、JBoss のマニュアルを参照してください。 2. JBoss standalone-full.xml ファイルの datasource 定義を、このモジュールを使用するように更新します。

## インプレースアップグレードのアップグレード後のタスクを実行する

インプレースアップグレード後、アップグレード後のタスクを実行する必要があります。

次の表に、アップグレード後のタスクの詳細を示します。

アップグレードタスク	詳細
Java のキャッシュをクリアする	1. Java のキャッシュをクリアします。手順については、Java のマニュアルを参照してください。 2. Hub コンソールを起動します。
ロギングの設定	Data Director のパフォーマンスを最適化するには、log4j.xml ファイルを編集することでロギングを設定します。 1. 以下のディレクトリにある log4j.xml を開きます。 <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/conf 2. 次のロギング設定を追加します。 <category name="org.eclipse.persistence.sdo"> <priority value="WARN"/> </category> <category name="org.eclipse.persistence.default"> <priority value="WARN"/> </category> 3. log4j.xml ファイルを保存して閉じます。

# オブジェクト、カラム、および廃止されたオブジェクトへの参照の削除

アップグレードタスク	詳細
REL_START_DATE、REL_END_DATE、および HUID テーブルへの参照の更新	REL_START_DATE および REL_END_DATE システムカラムならびに HUID テーブルの参照を更新します。アップグレードプロセスによって、階層マネージャベースオブジェクトで使用するパッケージおよびビューに含まれる REL_START_DATE および REL_END_DATE カラムの参照が削除されます。REL_START_DATE の参照に代わって、PERIOD_START_DATE の参照が作成されます。また、REL_END_DATE の参照に代わって、PERIOD_END_DATE の参照が作成されます。

## Hub サーバーをデプロイするための PostInstall スクリプトの実行（条件による）

インストール時に postInstallSetup スクリプトをスキップした場合は、このスクリプトを実行します。インストール後のプロセスにより、Hub サーバアプリケーションがデプロイされ、データソースが作成され、JMS メッセージキューが設定されます。

管理対象サーバーを含む WebLogic Server 環境を使用し、管理サーバーと管理対象サーバーが異なるマシンにある場合は、すべてのデプロイメントファイルを管理サーバーの MDM Hub インストールディレクトリにコピーします。詳細については、『*Multidomain MDM のインストールガイド*』を参照してください。

- ▶ コマンドプロンプトを開き、postInstallSetup スクリプトをディレクトリ<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server で実行します。

**注:** MDM Hub インストーラにバンドルされている ActiveVOS をインストールしなかった場合は、コマンドに ActiveVOS のユーザー名とパスワードを含めないでください。UNIX でパスワードに感嘆符 (!) を含める場合、感嘆符の前にバックスラッシュを付ける必要があります。例えば、パスワードが!! cmx!! の場合、次のパスワードを入力します: \! \!cmx\! \!

WebSphere（セキュリティ無効）の場合:

```
./postInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password>
-Davos.username=<ActiveVOS Console username>
-Davos.password=<ActiveVOS Console password>
-Davos.jdbc.database.username=<ActiveVOS database username>
-Davos.jdbc.database.password=<ActiveVOS database password>
```

WebSphere（セキュリティ有効）の場合:

```
./postInstallSetup.sh -Dwebsphere.password=<WebSphere password>
-Ddatabase.password=<MDM Hub Master database password>
-Davos.username=<ActiveVOS Console username>
-Davos.password=<ActiveVOS Console password>
-Davos.jdbc.database.username=<ActiveVOS database username>
-Davos.jdbc.database.password=<ActiveVOS database password>
```

JBoss の場合:

```
./postInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password>
-Davos.username=<ActiveVOS Console username>
-Davos.password=<ActiveVOS Console password>
```

```
-Davos.jdbc.database.username=<ActiveVOS database username>
-Davos.jdbc.database.password=<ActiveVOS database password>
```

WebLogic の場合:

```
./postInstallSetup.sh -Dweblogic.password=<WebLogic password>
-Ddatabase.password=<MDM Hub Master database password>
-Davos.username=<ActiveVOS Console username>
-Davos.password=<ActiveVOS Console password>
-Davos.jdbc.database.username=<ActiveVOS database username>
-Davos.jdbc.database.password=<ActiveVOS database password>
```

**重要:** WebLogic 12.2.1.3 以降の環境で、ActiveVOS をインストールすることにした場合、または WebLogic T3S プロトコルを使用することにした場合は、決定に一致するオプションを追加します。

- インストールした ActiveVOS。-Dinstall.avos.patch=true
- 使用した T3S プロトコル。-Dweblogic.naming.protocol=t3s

ActiveVOS コンソールの資格情報は、アプリケーションサーバーの管理者ユーザーの資格情報と同じです。

ActiveVOS データベースの資格情報は、create\_bpm スクリプトを実行するために使用した資格情報と同じです。

管理対象サーバーがある WebLogic 環境にデプロイする場合、WebLogic Server 管理コンソールでデプロイメントのターゲットとしてすべての管理対象サーバーを指定するようにします。

詳細については、『*Multidomain MDM のインストールガイド*』と WebLogic のマニュアルを参照してください。

## Hub コンソールクライアントの設定

Hub サーバマシンに接続するように Hub コンソールクライアントを設定するには、cmxserver.properties ファイルを編集して、Hub サーバアプリケーションを再パッケージ化します。

**注:** Hub コンソールを起動すると、ホスト名とポート番号が上書きされます。

1. 次のディレクトリにある cmxserver.properties ファイルで cmx.appserver.console.mode プロパティに関連する値を設定します。

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/resources  
使用する通信プロトコル (HTTP または HTTPS) に値を設定します。

2. 次のディレクトリに移動します。

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/bin

3. 次のコマンドを実行します。

UNIX の場合:

```
./sip_ant.sh repack -Dconsole.hostname=<host name> -Dconsole.webport=<port>
```

Windows の場合:

```
sip_ant.bat repack -Dconsole.hostname=<host name> -Dconsole.webport=<port>
```

host name はアプリケーションサーバーのバインド先のサーバーの IP アドレス、または公的にアクセス可能なホスト名で、port は Hub コンソールが使用する必要のある現在のノードの HTTP または HTTPS ポートです。

# WebSphere 管理セキュリティの設定

WebSphere 管理セキュリティを設定して、WebSphere 管理コンソールへの MDM Hub のアクセスを制御できます。

WebSphere 管理セキュリティを設定するには、次の手順を実行します。

1. オペレーショナルリファレンスストア（ORS）を登録解除します。
2. EAR ファイルをアンインストールし、WebSphere からデータソースを削除します。
3. WebSphere 管理コンソールで WebSphere 管理セキュリティを有効にします。
4. Hub サーバーおよびプロセスサーバーのプロパティを設定します。
5. Hub サーバーおよびプロセスサーバーの PostInstallSetup スクリプトを実行します。
6. ORS を登録します。

ORS の登録時には WebSphere 資格情報を入力します。ORS の確認後は、資格情報を入力する必要はありません。

## オペレーショナルリファレンスストアの登録解除

オペレーショナルリファレンスストア（ORS）を登録解除するには、MDM Hub コンソールでデータベースツールを使用します。

1. MDM Hub コンソールで、**書き込みロック** > **ロックの取得** の順にクリックします。
2. **設定** ワークベンチで、**データベース** ツールを選択します。  
**データベース情報** ページが表示されます。
3. データベースのリストから、登録を解除する ORS を選択します。
4. **データベースの登録解除** をクリックします。  
データベースツールに、ORS の登録解除を確認するメッセージが表示されます。
5. **はい** をクリックします。

## EAR ファイルのアンインストールおよびデータソースの削除

EAR ファイルをアンインストールしてデータソースを削除するには、WebSphere 管理コンソールを使用します。

1. WebSphere 管理コンソールを使用して、次のデプロイメントファイルのデプロイを解除します。

デプロイメントファイル名	説明
siperian-mrm.ear	必須。Hub サーバーアプリケーション。
provisioning-ear.ear	必須。プロビジョニングツールアプリケーション。
entity360view-ear.ear	オプション。エンティティ 360 フレームワーク。

2. WebSphere 管理コンソールを使用して、MDM Hub マスターデータベースのすべてのデータソースとオペレーショナルリファレンスストアを削除します。
3. アプリケーションサーバーを再起動します。

詳細については、WebSphere のマニュアルを参照してください。

## WebSphere 管理コンソールでの WebSphere 管理セキュリティの有効化

WebSphere 管理コンソールで WebSphere 管理セキュリティを有効にする必要があります。WebSphere 管理セキュリティを有効する場合は、WebSphere アプリケーションセキュリティを無効にします。

詳細については、WebSphere のマニュアルを参照してください。

## Hub サーバーとプロセスサーバーのプロパティの設定

Hub サーバーとプロセスサーバーのプロパティファイルを設定し、WebSphere 管理セキュリティを有効にする必要があります。

1. アプリケーションサーバーを停止します。
2. Hub サーバーで WebSphere セキュリティを有効にします。
  - a. 次のディレクトリで、cmxserver.properties を開きます。  
UNIX の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/resources  
Windows の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\server\resources
  - b. cmx.websphere.security.enabled を true に設定します。
3. プロセスサーバーで WebSphere セキュリティを有効にします。
  - a. 次のディレクトリで cmxcleanse.properties を開きます。  
UNIX の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse/resources  
Windows の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\cleanse\resources
  - b. cmx.websphere.security.enabled を true に設定します。
4. Hub サーバーで WebSphere ユーザー名を設定します。
  - a. 次のディレクトリで build.properties を開きます。  
UNIX の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/bin  
Windows の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\server\bin
  - b. websphere.username を WebSphere 管理ユーザー名に設定します。
5. プロセスサーバーで WebSphere ユーザー名を設定します。
  - a. 次のディレクトリで build.properties を開きます。  
UNIX の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse/bin  
Windows の場合:<MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\cleanse\bin
  - b. websphere.username を WebSphere 管理ユーザー名に設定します。
6. SQL\* Plus で、次のコマンドを実行して c\_repos\_cleanse\_match\_server.is\_secured を 1 に設定します。  
UPDATE c\_repos\_cleanse\_match\_server set is\_secured = 1 where rowid\_cleanse\_match\_server='<Insert value here>';  
COMMIT;
7. アプリケーションサーバーを起動します。

## Hub サーバーの PostInstallSetup スクリプトの手動実行

Hub サーバーの PostInstallSetup スクリプトを実行する必要があります。

1. コマンドプロンプトを開きます。
2. 次のディレクトリにある PostInstallSetup スクリプトに移動します。  
UNIX の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server  
Windows の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\server
3. 次のコマンドを実行します。  
UNIX の場合: `postinstallsetup.sh -Ddatabase.password=<MDM Hub マスターデータベース> -Dwebsphere.password=<WebSphere administrative user password>`  
Windows の場合: `postinstallsetup.bat -Ddatabase.password=<MDM Hub マスターデータベース> -Dwebsphere.password=<WebSphere administrative user password>`
4. アプリケーションサーバーを再起動します。

## プロセスサーバーの PostInstallSetup スクリプトの実行

プロセスサーバーの PostInstallSetup スクリプトを実行する必要があります。

1. コマンドプロンプトを開きます。
2. 次のディレクトリにある PostInstallSetup スクリプトに移動します。  
UNIX の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse  
Windows の場合: <MDM Hub installation directory: MDM Hub のインストールディレクトリ>\hub\cleanse
3. 次のコマンドを実行します。  
UNIX の場合: `postinstallsetup.sh -Dwebsphere.password=<websphere administrative user password>`  
Windows の場合: `postinstallsetup.bat -Dwebsphere.password=<websphere administrative user password>`
4. アプリケーションサーバーを再起動します。

## オペレーショナルリファレンスストアの登録

オペレーショナルリファレンスストアを登録するには、Hub コンソールを使用します。

1. 設定ワークベンチにあるデータベースツールを起動します。
2. **【書き込みロック】** > **【ロックの取得】** の順にクリックします。
3. **【データベースの登録】** をクリックします。  
**Informatica MDM Hub 接続ウィザード**が表示され、データベースタイプの選択が求められます。
4. Microsoft SQL Server、Oracle、または IBM Db2 を選択して **【次へ】** をクリックします。
5. Microsoft SQL Server で、データベースの接続プロパティを設定します。
  - a. **【接続プロパティ】** ページで、接続プロパティを指定し、**【次へ】** をクリックします。

以下の表に、接続プロパティの種類と説明を示します。

プロパティ	説明
データベース表示名	Hub コンソールに表示する必要があるオペレーショナルリファレンスストアの名前。
マシン識別子	Hub ストアインスタンスからのレコードを一意に識別するためにキーに割り当てられるプレフィックス。
データベースホスト名	Microsoft SQL Server データベースをホストするサーバーの IP アドレスまたは名前。
ポート	Microsoft SQL Server データベースのポート。デフォルトは 1433。
スキーマ名	オペレーショナル参照ストアの名前。
パスワード	オペレーショナル参照ストアのユーザー名に関連付けられているパスワード。
Dynamic Data Masking ホスト	Dynamic Data Masking をホストするサーバーの IP アドレスまたは名前。Dynamic Data Masking を使用しない場合は、空のままにする。
DDM 接続 URL	オプション。Dynamic Data Masking サーバーの URL。Dynamic Data Masking を使用しない場合は、空のままにする。

- b. [接続プロパティ] ページで、接続プロパティを指定し、[次へ] をクリックします。
  - c. サマリを確認し、追加の接続プロパティを指定します。
- 以下の表に、設定可能な追加の接続プロパティを示します。

プロパティ	説明
接続 URL	接続 URL。接続ウィザードでは、デフォルトで接続 URL が生成されます。
登録後にデータソースを作成する	登録後にアプリケーションサーバーのデータソースを作成する場合は選択します。

6. Oracle 環境で、データベースの接続プロパティを設定します。
  - a. Oracle の接続方式を選択して、[次へ] をクリックします。
- 次の表に、選択できる Oracle の接続方式を示します。

接続方式	説明
サービス	サービス名を使用して Oracle に接続します。
SID	Oracle のシステム ID を使用して Oracle に接続します。

サービス名と SID 名の詳細については、Oracle のマニュアルを参照してください。  
**[接続プロパティ]** ページが表示されます。

- b. 選択する接続タイプの接続プロパティを指定し、[次へ] をクリックします。

以下の表に、接続プロパティの種類と説明を示します。

プロパティ	説明
データベース表示名	Hub コンソールに表示する必要があるオペレーショナルリファレンスストアの名前。
マシン識別子	Hub ストアインスタンスからのレコードを一意に識別するためにキーに割り当てられるプレフィックス。
データベースホスト名	Oracle データベースをホストするサーバーの IP アドレスまたは名前。
SID	サーバー上で実行される Oracle データベースのインスタンスを参照する Oracle システム識別子。[SID] フィールドは、[SID] 接続タイプを選択した場合に表示される。
サービス	Oracle データベースへの接続に使用する Oracle サービスの名前。[サービス] フィールドは、[サービス] 接続タイプを選択した場合に表示される。
ポート	Oracle データベースサーバー上で実行される Oracle リスナの TCP ポート。デフォルトは 1521。
Oracle TNS 名	ネットワーク上で認識されているデータベースの名前（アプリケーションサーバーの TNSNAMES.ORA ファイルで定義）。 例: mydatabase.mycompany.com Oracle TNS 名は、Oracle データベースのインストール時に設定します。Oracle TNS 名の詳細については、Oracle のマニュアルを参照してください。
スキーマ名	オペレーショナルリファレンスストアの名前。
ユーザー名	オペレーショナル参照ストアのユーザー名。デフォルトでは、これがオペレーショナル参照ストアを作成するために使用するスクリプトで指定するユーザー名です。このユーザーが、Hub ストア内のすべてのオペレーショナルリファレンスストアデータベースオブジェクトの所有者となる。 オペレーショナル参照ストアでプロキシユーザーが設定されている場合は、代わりにプロキシユーザーを指定できます。
パスワード	オペレーショナルリファレンスストアのユーザー名に関連付けられているパスワード。 Oracle の場合、パスワードでは大文字と小文字の区別がない。 デフォルトでは、これがオペレーショナル参照ストアを作成するときに指定するパスワードです。 オペレーショナル参照ストアでプロキシユーザーが設定されている場合は、代わりにプロキシユーザーのパスワードを指定できます。
Dynamic Data Masking ホスト	Dynamic Data Masking をホストするサーバーの IP アドレスまたは名前。 Dynamic Data Masking を使用しない場合は、空のままにする。
DDM 接続 URL	オプション。Dynamic Data Masking サーバーの URL。Dynamic Data Masking を使用しない場合は、空のままにする。

**注: スキーマ名とユーザー名**は、どちらもオペレーショナル参照ストアを作成するときに指定したオペレーショナル参照ストアの名前です。この情報が必要な場合はデータベース管理者に問い合わせます。

サマリページが表示されます。

- c. サマリを確認し、追加の接続プロパティを指定します。  
以下の表に、設定可能な追加の接続プロパティを示します。

プロパティ	説明
接続 URL	<p>接続 URL。接続ウィザードでは、デフォルトで接続 URL が生成されます。次のリストに、Oracle 接続タイプの接続 URL のフォーマットを示します。</p> <p><b>サービス接続タイプ</b></p> <p><code>jdbc:oracle:thin:@//database_host:port/service_name</code></p> <p><b>SID 接続タイプ</b></p> <p><code>jdbc:oracle:thin:@//database_host:port:sid</code></p> <p>サービス接続タイプの場合のみ、別の接続 URL をカスタマイズして後でテストできます。</p>
登録後にデータソースを作成する	<p>登録後にアプリケーションサーバーのデータソースを作成する場合は選択します。</p> <p><b>注:</b> このオプションを選択しない場合、手動でデータソースを設定する必要があります。</p>

- d. サービス接続タイプには、デフォルトの URL を変更する場合、**【編集】** ボタンをクリックし、URL を指定して **【OK】** をクリックします。
7. IBM Db2 環境で、データベースの接続プロパティを設定します。
    - a. 接続プロパティを指定して **【次へ】** をクリックします。  
以下の表に、接続プロパティの種類と説明を示します。

プロパティ	説明
データベース表示名	Hub コンソールに表示する必要があるオペレーショナルリファレンスストアの名前。
マシン識別子	Hub ストアインスタンスからのレコードを一意に識別するためにキーに割り当てられるプレフィックス。
データベースサーバー名	IBM Db2 データベースをホストするサーバーの IP アドレスまたは名前。
データベース名	作成するデータベースの名前
データベースホスト名	IBM Db2 データベースをホストするサーバーの IP アドレスまたは名前。
スキーマ名	オペレーショナルリファレンスストアの名前。

プロパティ	説明
ユーザー名	オペレーショナルリファレンスストアのユーザー名。デフォルトでは、これがオペレーショナルリファレンスストアを作成するために使用するスクリプトで指定するユーザー名です。このユーザーが、Hub Store 内のすべてのオペレーショナルリファレンスストアデータベースオブジェクトの所有者となります。 オペレーショナルリファレンスストアでプロキシユーザーが設定されている場合は、代わりにプロキシユーザーを指定できます。
パスワード	オペレーショナルリファレンスストアのユーザー名に関連付けられているパスワード。 IBM Db2 の場合、パスワードでは大文字と小文字が区別される。 デフォルトでは、これがオペレーショナルリファレンスストアを作成するときに指定するパスワードです。 オペレーショナルリファレンスストアでプロキシユーザーが設定されている場合は、代わりにプロキシユーザーのパスワードを指定できます。
Dynamic Data Masking ホスト	Dynamic Data Masking をホストするサーバーの IP アドレスまたは名前。 Dynamic Data Masking を使用しない場合は、空のままにする。
DDM 接続 URL	オプション。Dynamic Data Masking サーバーの URL。Dynamic Data Masking を使用しない場合は、空のままにする。

**注:** スキーマ名とユーザー名は、どちらもオペレーショナル参照ストアを作成するときに指定したオペレーショナル参照ストアの名前です。この情報が必要な場合はデータベース管理者に問い合わせます。

サマリページが表示されます。

- b. サマリを確認し、追加の接続プロパティを指定します。

以下の表に、設定可能な追加の接続プロパティを示します。

プロパティ	説明
接続 URL	接続 URL。接続ウィザードでは、デフォルトで接続 URL が生成されます。次の例は、接続 URL のフォーマットを示しています。 <code>jdbc:db2://database_host:port/db_name</code>
登録後にデータソースを作成する	登録後にアプリケーションサーバーのデータソースを作成する場合は選択します。 <b>注:</b> このオプションを選択しない場合、手動でデータソースを設定する必要があります。

8. **【完了】** をクリックします。  
**【データベースの登録】** ダイアログボックスが表示されます。
9. **【OK】** をクリックします。  
**【アプリケーションサーバーログイン】** ダイアログボックスが表示されます。
10. WebSphere 管理ユーザー名とパスワードを入力します。
11. **【OK】** をクリックします。  
MDM Hub で ORS が登録されます。
12. アプリケーションサーバーを再起動します。

13. 登録したオペレーショナルリファレンスストアを選択し、**【データベース接続のテスト】** ボタンをクリックしてデータベース設定をテストします。  
[データベースのテスト] ダイアログに、データベース接続テストの結果が表示されます。
14. **【OK】** をクリックします。  
ORS が登録され、データベースへの接続がテストされます。

## クラスローダーの設定 WebSphere で

必要な PostInstallSetup スクリプトを実行した後、WebSphere デプロイメントマネージャを使用して、Hub サーバーアプリケーションおよびプロセスサーバーアプリケーションのクラスローダーを設定します。

1. 次のアプリケーションのクラスローダーを設定します。siperian-mrm.ear、provisioning-ear.ear、entity360view-ear.ear、および siperian-mrm-cleanse.ear。
  - a. **【アプリケーション】 > 【アプリケーションタイプ】 > 【WebSphere エンタープライズアプリケーション】** を選択します。
  - b. **【エンタープライズアプリケーション】** ページで、いずれかのアプリケーションをクリックします。
  - c. アプリケーションの設定ページで、**【クラスロードおよび更新の検出】** リンクをクリックします。
  - d. クラスローダーの設定ページで、**【最初にローカルクラスローダーでロードしたクラス（親が最後）】** クラスローダーの順序オプションを選択します。
  - e. **【適用】** をクリックしてから、**【OK】** をクリックします。
2. 次のアプリケーション EAR ファイルの Web モジュールのクラスローダーを設定します。

アプリケーション EAR ファイル	Web モジュール	クラスローダーの順序
siperian-mrm.ear	zds-gui.war	最初にローカルクラスローダーで読み込むクラス（親が最後）
provisioning-ear.ear	provisioning.war	最初にローカルクラスローダーで読み込むクラス（親が最後）
siperian-mrm-cleanse.ear	siperian-mrm-cleanse.war	最初にローカルクラスローダーで読み込むクラス（親が最後）

- a. **【アプリケーション】 > 【アプリケーションタイプ】 > 【WebSphere エンタープライズアプリケーション】** を選択します。
  - b. **【エンタープライズアプリケーション】** ページで、いずれかのアプリケーション EAR ファイルの名前をクリックします。
  - c. アプリケーションの設定ページで、**【モジュールの管理】** リンクをクリックします。
  - d. モジュールのリストで、Web モジュールのリンクをクリックします。
  - e. Web モジュールの設定ページで、クラスローダーの順序を選択します。
  - f. **【適用】** をクリックしてから、**【OK】** をクリックします。
3. WebSphere を再起動して、Hub サーバーアプリケーションおよびプロセスサーバーアプリケーションを起動します。

# オペレーショナルリファレンスストアの登録

オペレーショナルリファレンスストアを登録するには、Hub コンソールを使用します。

1. 設定ワークベンチにあるデータベースツールを起動します。
2. **【書き込みロック】** > **【ロックの取得】** の順にクリックします。
3. **【データベースの登録】** をクリックします。

**Informatica MDM Hub 接続ウィザード**が表示され、データベースタイプの選択が求められます。

4. Microsoft SQL Server、Oracle、または IBM Db2 を選択して **【次へ】** をクリックします。
5. Microsoft SQL Server で、データベースの接続プロパティを設定します。
  - a. **【接続プロパティ】** ページで、接続プロパティを指定し、**【次へ】** をクリックします。

以下の表に、接続プロパティの種類と説明を示します。

プロパティ	説明
データベース表示名	Hub コンソールに表示する必要があるオペレーショナルリファレンスストアの名前。
マシン識別子	Hub ストアインスタンスからのレコードを一意に識別するためにキーに割り当てられるプレフィックス。
データベースホスト名	Microsoft SQL Server データベースをホストするサーバーの IP アドレスまたは名前。
ポート	Microsoft SQL Server データベースのポート。デフォルトは 1433。
スキーマ名	オペレーショナル参照ストアの名前。
パスワード	オペレーショナル参照ストアのユーザー名に関連付けられているパスワード。
Dynamic Data Masking ホスト	Dynamic Data Masking をホストするサーバーの IP アドレスまたは名前。Dynamic Data Masking を使用しない場合は、空のままにする。
DDM 接続 URL	オプション。Dynamic Data Masking サーバーの URL。Dynamic Data Masking を使用しない場合は、空のままにする。

- b. **【接続プロパティ】** ページで、接続プロパティを指定し、**【次へ】** をクリックします。
- c. サマリを確認し、追加の接続プロパティを指定します。

以下の表に、設定可能な追加の接続プロパティを示します。

プロパティ	説明
接続 URL	接続 URL。接続ウィザードでは、デフォルトで接続 URL が生成されます。
登録後にデータソースを作成する	登録後にアプリケーションサーバーのデータソースを作成する場合は選択します。

6. Oracle 環境で、データベースの接続プロパティを設定します。
- a. Oracle の接続方式を選択して、**[次へ]** をクリックします。
- 次の表に、選択できる Oracle の接続方式を示します。

接続方式	説明
サービス	サービス名を使用して Oracle に接続します。
SID	Oracle のシステム ID を使用して Oracle に接続します。

サービス名と SID 名の詳細については、Oracle のマニュアルを参照してください。

**[接続プロパティ]** ページが表示されます。

- b. 選択する接続タイプの接続プロパティを指定し、**[次へ]** をクリックします。
- 以下の表に、接続プロパティの種類と説明を示します。

プロパティ	説明
データベース表示名	Hub コンソールに表示する必要があるオペレーショナルリファレンスストアの名前。
マシン識別子	Hub ストアインスタンスからのレコードを一意に識別するためにキーに割り当てられるプレフィックス。
データベースホスト名	Oracle データベースをホストするサーバーの IP アドレスまたは名前。
SID	サーバー上で実行される Oracle データベースのインスタンスを参照する Oracle システム識別子。[SID] フィールドは、[SID] 接続タイプを選択した場合に表示される。
サービス	Oracle データベースへの接続に使用する Oracle サービスの名前。 <b>[サービス]</b> フィールドは、 <b>[サービス]</b> 接続タイプを選択した場合に表示される。
ポート	Oracle データベースサーバー上で実行される Oracle リスナの TCP ポート。デフォルトは 1521。
Oracle TNS 名	ネットワーク上で認識されているデータベースの名前（アプリケーションサーバーの TNSNAMES.ORA ファイルで定義）。 例: mydatabase.mycompany.com Oracle TNS 名は、Oracle データベースのインストール時に設定します。Oracle TNS 名の詳細については、Oracle のマニュアルを参照してください。
スキーマ名	オペレーショナルリファレンスストアの名前。
ユーザー名	オペレーショナル参照ストアのユーザー名。デフォルトでは、これがオペレーショナル参照ストアを作成するために使用するスクリプトで指定するユーザー名です。このユーザーが、Hub ストア内のすべてのオペレーショナルリファレンスストアデータベースオブジェクトの所有者となる。 オペレーショナル参照ストアでプロキシユーザーが設定されている場合は、代わりにプロキシユーザーを指定できます。

プロパティ	説明
パスワード	オペレーショナルリファレンスストアのユーザー名に関連付けられているパスワード。 Oracle の場合、パスワードでは大文字と小文字の区別がない。 デフォルトでは、これがオペレーショナル参照ストアを作成するときに指定するパスワードです。 オペレーショナル参照ストアでプロキシユーザーが設定されている場合は、代わりにプロキシユーザーのパスワードを指定できます。
Dynamic Data Masking ホスト	Dynamic Data Masking をホストするサーバーの IP アドレスまたは名前。 Dynamic Data Masking を使用しない場合は、空のままにする。
DDM 接続 URL	オプション。Dynamic Data Masking サーバーの URL。Dynamic Data Masking を使用しない場合は、空のままにする。

**注: スキーマ名とユーザー名は、どちらもオペレーショナル参照ストアを作成するときに指定したオペレーショナル参照ストアの名前です。この情報が必要な場合はデータベース管理者に問い合わせます。**

サマリページが表示されます。

- c. サマリを確認し、追加の接続プロパティを指定します。  
以下の表に、設定可能な追加の接続プロパティを示します。

プロパティ	説明
接続 URL	接続 URL。接続ウィザードでは、デフォルトで接続 URL が生成されます。次のリストに、Oracle 接続タイプの接続 URL のフォーマットを示します。 <b>サービス接続タイプ</b> <code>jdbc:oracle:thin:@//database_host:port/service_name</code> <b>SID 接続タイプ</b> <code>jdbc:oracle:thin:@//database_host:port:sid</code> サービス接続タイプの場合のみ、別の接続 URL をカスタマイズした後でテストできます。
登録後にデータソースを作成する	登録後にアプリケーションサーバーのデータソースを作成する場合は選択します。 <b>注:</b> このオプションを選択しない場合、手動でデータソースを設定する必要があります。

- d. サービス接続タイプには、デフォルトの URL を変更する場合、**【編集】** ボタンをクリックし、URL を指定して **【OK】** をクリックします。
7. IBM Db2 環境で、データベースの接続プロパティを設定します。
- a. 接続プロパティを指定して **【次へ】** をクリックします。

以下の表に、接続プロパティの種類と説明を示します。

プロパティ	説明
データベース表示名	Hub コンソールに表示する必要があるオペレーショナルリファレンスストアの名前。
マシン識別子	Hub ストアインスタンスからのレコードを一意に識別するためにキーに割り当てられるプレフィックス。
データベースサーバー名	IBM Db2 データベースをホストするサーバーの IP アドレスまたは名前。
データベース名	作成するデータベースの名前
データベースホスト名	IBM Db2 データベースをホストするサーバーの IP アドレスまたは名前。
スキーマ名	オペレーショナルリファレンスストアの名前。
ユーザー名	オペレーショナルリファレンスストアのユーザー名。デフォルトでは、これがオペレーショナルリファレンスストアを作成するために使用するスクリプトで指定するユーザー名です。このユーザーが、Hub Store 内のすべてのオペレーショナルリファレンスストアデータベースオブジェクトの所有者となります。 オペレーショナルリファレンスストアでプロキシユーザーが設定されている場合は、代わりにプロキシユーザーを指定できます。
パスワード	オペレーショナルリファレンスストアのユーザー名に関連付けられているパスワード。 IBM Db2 の場合、パスワードでは大文字と小文字が区別される。 デフォルトでは、これがオペレーショナルリファレンスストアを作成するときに指定するパスワードです。 オペレーショナルリファレンスストアでプロキシユーザーが設定されている場合は、代わりにプロキシユーザーのパスワードを指定できます。
Dynamic Data Masking ホスト	Dynamic Data Masking をホストするサーバーの IP アドレスまたは名前。 Dynamic Data Masking を使用しない場合は、空のままにする。
DDM 接続 URL	オプション。Dynamic Data Masking サーバーの URL。Dynamic Data Masking を使用しない場合は、空のままにする。

**注:** スキーマ名とユーザー名は、どちらもオペレーショナル参照ストアを作成するときに指定したオペレーショナル参照ストアの名前です。この情報が必要な場合はデータベース管理者に問い合わせます。

サマリページが表示されます。

- b. サマリを確認し、追加の接続プロパティを指定します。

以下の表に、設定可能な追加の接続プロパティを示します。

プロパティ	説明
接続 URL	接続 URL。接続ウィザードでは、デフォルトで接続 URL が生成されます。次の例は、接続 URL のフォーマットを示しています。 jdbc:db2://database_host:port/db_name
登録後にデータソースを作成する	登録後にアプリケーションサーバーのデータソースを作成する場合は選択します。 <b>注:</b> このオプションを選択しない場合、手動でデータソースを設定する必要があります。

8. **【完了】** をクリックします。  
**【データベースの登録】** ダイアログボックスが表示されます。
9. **【OK】** をクリックします。  
**【アプリケーションサーバーログイン】** ダイアログボックスが表示されます。
10. WebSphere 管理ユーザー名とパスワードを入力します。
11. **【OK】** をクリックします。  
MDM Hub で ORS が登録されます。
12. アプリケーションサーバーを再起動します。
13. 登録したオペレーショナルリファレンスストアを選択し、**【データベース接続のテスト】** ボタンをクリックしてデータベース設定をテストします。  
**【データベースのテスト】** ダイアログに、データベース接続テストの結果が表示されます。
14. **【OK】** をクリックします。  
ORS が登録され、データベースへの接続がテストされます。

## アップグレードされたメタデータの検証

オペレーショナルリファレンスストア（ORS）に検証エラーがないことを確認します。この結果を、アップグレード前に得られた検証結果と比較します。メタデータを検証するため、Hub コンソールでリポジトリマネージャを使用します。

**注:** 前バージョンの MDM Hub からのアップグレード後も、マッピングのないステー징テーブルが含まれていた古いデータベースの検証エラーが発生する場合があります。Hub コンソールのリポジトリマネージャで、**【修復】** ボタンをクリックしてこのような修復可能な問題を修正します。

### メタデータの検証

オペレーショナル参照ストア（ORS）のメタデータを検証するには、Hub コンソールで Repository Manager ツールを使用します。

1. **【設定】** ワークベンチで、**【Repository Manager】** をクリックします。
2. **【検証】** タブで、検証するリポジトリを選択します。
3. **【検証】** をクリックします。

【検証チェックの選択】 ダイアログボックスが表示されます。

4. 実行する検証チェックを選択し、【OK】をクリックします。  
リポジトリマネージャツールにより、リポジトリが検証され、すべての問題が【検出された問題】 ペインに表示されます。
5. 修復可能な問題を修正するには、【修理】をクリックします。
6. Customer 360 または Supplier 360 リポジトリを検証する場合は、【復元】をクリックして、【ドメインの結果】 タブに表示されたエラーを解決します。
7. 修正後も ORS が【不明】 状態の場合は、アプリケーションサーバーのシステムクロックとデータベースマシンを同期します。

## 検証結果の保存

検証プロセスを実行したら、検証結果を HTML ファイルとして保存できます。

1. Hub コンソールのリポジトリマネージャツールから、【検証】 タブを選択します。
2. 【保存】 ボタンをクリックします。
3. 【保存】 ダイアログボックスから、検証結果を保存するディレクトリに移動します。
4. わかりやすい HTML ファイル名を指定します。【保存】 をクリックします。  
リポジトリマネージャによって、指定した場所に検証結果が HTML ファイルとして保存されます。

## メタデータ検証メッセージの解決

検証ツールを実行した後に、検証メッセージが表示されることがあります。

次のエラーメッセージは、最も一般的な検証メッセージです。

警告 SIP-PV-10703 パッケージ'EMPLOYEE\_DETAILS\_PKG'がデータベースビューと同期されていません。

データベースビューと同期化するには、リポジトリマネージャから修復プロセスを実行します。

ビュー'C\_EMPLOYEE\_DETAILS\_MTIPTIP' - プロキシユーザーロールの SELECT 特権が付与されていません。またはルート MTIP の SIP-MV-11410- SQL が正しくありません。

MTIP ビューを再作成します。

1. Hub コンソールで、設定ワークベンチを開き、エンタープライズマネージャをクリックします。
2. 書き込みロックを取得します。
3. 【ORS データベース】 タブを選択します。
4. データベースを選択します。
5. 【プロパティ】 タブを選択します。
6. 【MTIP の再生成が必要】 というプロパティを見つけて、【MTIP の再生成】 ボタンをクリックします。

ビュー'EMPLOYEE\_DETAILS\_PKG' - プロキシユーザーロールの SELECT 特権が付与されていません。

この特権が含まれるようにプロキシユーザーロールを更新します。

1. Hub コンソールで、セキュリティアクセススマネージャワークベンチを開き、【ロール】 をクリックします。
2. 書き込みロックを取得します。
3. プロキシユーザーロールを選択します。
4. 【リソース特権】 タブをクリックします。

5. メッセージで参照されているパッケージまたはテーブルを特定します。
6. **【読み取り】** チェックボックスをオンにします。

SIP-PV-11105 - ビューに対するプロキシユーザーロールの SELECT 特権が付与されていません。

ビュー 'C\_REPOS\_USER\_GROUP\_ALL' - プロキシユーザーロールの SELECT 特権が付与されていません。

データベース移行スクリプトにより、プロキシユーザーロールが作成されましたが、プロキシユーザーにはリポジトリビューに対する特権が付与されていません。データベースから、リポジトリビューに対するプロキシユーザーの SELECT 特権を付与します。

## ローカライズされたスキーマの更新

アップグレード前のスキーマにローカライズされたルックアップテーブルが含まれる場合、アップグレード後に検証エラーが表示されることがあります。

バージョン 10.2.x でローカライズされたルックアップテーブルの場合、メタデータの検証プロセスは次のタイプのエラーを生成します。

SIP-MV-22000 The name attribute for DB bundle [<lookup table name>.dbBundleMapping] is not defined in the configuration

このエラーを修正するには、次の手順を実行します。

1. dbBundleConfig.xml ファイルを C\_REPOS\_CO\_CS\_CONFIG リポジトリテーブルからダウンロードします。
2. dbBundleConfig.xml ファイルで、name=<lookup table name> パラメータを bundle 要素から mapping 要素にコピーします。

例えば、次のスニペットでは編集された mapping 要素を示します。

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dbBundleConfiguration xmlns="http://www.example.com/mdm/db-bundle-configuration">
<bundle name="LUCountry" hubObject="C_LU_COUNTRY_LCL">
<mapping name="LUCountry" keyColumn="COUNTRY_CD" countryColumn="COUNTRY_CODE"
languageColumn="LANGUAGE_CODE" valueColumn="LOCALIZED_STRING"/>
</bundle>
<bundle name="LUState" hubObject="C_LU_STATE_LCL">
<mapping name="LUState" keyColumn="STATE_CD" countryColumn="COUNTRY_CODE" languageColumn="LANGUAGE_CODE"
valueColumn="LOCALIZED_STRING"/>
</bundle>
<bundle name="LUCountry.LUState" hubObject="C_LU_STATE_LCL">
<mapping name="LUCountry.LUState" keyColumn="STATE_CD" countryColumn="COUNTRY_CODE"
languageColumn="LANGUAGE_CODE" valueColumn="LOCALIZED_STRING"/>
</bundle>
</dbBundleConfiguration>
```

3. 編集された dbBundleConfig.xml ファイルを C\_REPOS\_CO\_CS\_CONFIG リポジトリテーブルにアップロードします。
4. メタデータの検証を実行します。

## コンテンツセキュリティポリシーのカスタマイズ

コンテンツセキュリティポリシー (CSP) は、クロスサイトスクリプティングなどのコードインジェクション攻撃を防ぐための標準です。Web サイトは、ブラウザが Web サイトのコンテンツを表示するために読み込むことができる、コンテンツの承認済みの発信元を宣言します。アップグレードプロセスでは、登録済みのオペレーショナル参照ストア (ORS) データベース内のカスタムユーザーインタフェースコンポーネントがチェッ

クされます。コンポーネントが見つかった場合は、適切なコンテンツセキュリティポリシーが適用され、カスタムインタフェースコンポーネントがブロックされないようにします。

カスタムユーザーインタフェースコンポーネントが見つかった場合、コンテンツセキュリティポリシーは、Multidomain MDM が機能するために必要なデフォルトに設定されます。アップグレードプロセスでは、次のルールが追加されます: script-src \*; font-src \*; style-src \*; frame-src \*; image-src \*; default-src \*;

アップグレード後に、コンテンツセキュリティポリシーをカスタマイズして、システムを保護し、コードインジェクション攻撃を防ぎます。コンテンツセキュリティポリシーの設定の詳細については、『*Multidomain MDM プロビジョニングツールガイド*』を参照してください。

## クリーンアップグレードのアップグレード後のタスクを実行する

クリーンアップグレード後、Hub コンソールの起動時に復号化のエラーが発生した場合、アップグレード後のタスクを実行する必要があります。

次のアップグレード後のタスクを実行します。

1. スキーマのパスワードの暗号化。
2. スキーマのパスワードの更新。
3. オペレーショナル参照ストア接続のテストおよび更新。
4. ActiveVOS 接続のテストおよび更新。
5. クレンジング関数のテストおよびプロセスサーバーの追加。

### スキーマのパスワードの暗号化

データベーススキーマのパスワードを保護するために暗号化します。

- ▶ データベーススキーマパスワードを暗号化するには、コマンドプロンプトで以下のコマンドを実行します。

```
java -classpath siperian-api.jar;siperian-common.jar;siperian-server.jar  
com.delos.util.PublicKeyBasedEncryptionHelper <plain text password> <Hub Server installation directory>
```

結果は端末のウィンドウにエコーされます。

Plaintext Password: *password*

Encrypted Password: *encrypted password*

### スキーマのパスワードの更新

MDM Hub マスタデータベースのパスワードまたはオペレーショナル参照ストアのパスワードを更新することができます。

1. マスタデータベースのパスワードまたはオペレーショナルリファレンスストアのパスワードを更新するには、cmx\_system ユーザーとして接続して次の文を実行します。

Oracle および IBM Db2 の場合

```
UPDATE C_REPOS_DATABASE SET PASSWORD = '<new_password>' WHERE USER_NAME = <user_name>;  
COMMIT;
```

Microsoft SQL Server の場合

```
UPDATE [dbo].[C_REPOS_DATABASE] SET PASSWORD = '<new_password>' WHERE USER_NAME = <user_name>
```

2. アプリケーションサーバを再起動します。

## オペレーショナル参照ストア接続のテストおよび更新

Operational Reference Store: オペレーショナル参照ストアへの接続をテストします。接続に失敗する場合は、Hub コンソールで Operational Reference Store: オペレーショナル参照ストアのパスワードを編集します。

1. [設定] ワークベンチで、[データベース] をクリックします。
2. 書き込みロックを取得します。
3. テストする Operational Reference Store: オペレーショナル参照ストアを選択します。
4. [データベース接続のテスト] ボタンをクリックします。  
[データベースのテスト] ダイアログボックスが表示されます。
5. [OK] をクリックします。
6. 接続テストに失敗する場合は、**データベース接続プロパティの編集**アイコンをクリックします。  
選択した Operational Reference Store: オペレーショナル参照ストアの [データベースの登録] ダイアログボックスが表示されます。
7. Operational Reference Store: オペレーショナル参照ストアに関連付けられているユーザー名のパスワードを編集し、[OK] をクリックします。

## ActiveVOS 接続のテストおよび更新

ActiveVOS ワークフローエンジンから MDM Hub への接続をテストします。接続に失敗する場合は、Hub コンソールで ActiveVOS の信頼されたユーザーのパスワードを編集します。

1. 設定ワークベンチで、[Workflow Manager] をクリックします。
2. 書き込みロックを取得します。
3. [ワークフローエンジン] タブをクリックし、ActiveVOS ワークフローエンジンを選択します。
4. ワークフローエンジン接続をテストするには、[テスト] をクリックします。
5. ワークフローエンジン接続のテストに失敗する場合は、[編集] をクリックします。
6. [ワークフローの編集] ダイアログボックスで、ActiveVOS の信頼されたユーザーのパスワードを編集し、[OK] をクリックします。

## プロセスサーバーのテストおよび追加

文字列関数などの組み込みクレンジング関数をテストします。復号化のエラーでテストに失敗する場合は、Hub コンソールでプロセスサーバーを削除して追加します。

1. 組み込みクレンジング関数をテストします。
  - a. モデルワークベンチで、[クレンジング関数] をクリックします。
  - b. 書き込みロックを取得します。
  - c. テストするクレンジング関数を選択します。
  - d. [テスト] タブをクリックします。
  - e. 入力値を入力し、[テスト] をクリックします。
2. テストに失敗する場合は、cmxserver.log ファイルで「SIP-09131: 一般的な復号化エラー」エラーを確認します。

cmxserver.log ファイルは以下のディレクトリにあります。

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/logs

3. エラーログに「SIP-09131」エラーが含まれる場合は、プロセスサーバーを削除して追加します。
  - a. [ユーティリティ] ワークベンチで、[プロセスサーバー] をクリックします。
  - b. 書き込みロックを取得します。
  - c. 削除するプロセスサーバーを選択し、**プロセスサーバーの削除**アイコンをクリックします。
  - d. 削除を確認し、[OK] をクリックします。
  - e. **プロセスサーバーの追加**アイコンをクリックします。
  - f. [プロセスサーバーの追加/編集] ダイアログボックスで、プロセスサーバーのプロパティを設定します。
  - g. 設定を保存します。

## プラットフォームトランスフォーメーションに対するクレンジング関数の設定

設定したプラットフォームトランスフォーメーションを使用する場合は、クレンジング関数ツールに IDQ ライブラリを追加します。これにより、プラットフォームトランスフォーメーションの代わりに、ライブラリ内のクレンジング関数を使用できます。Multidomain MDM 10.5 には、Apache Axis2 のバージョン 1.8.0 へのアップグレードが含まれています。クレンジング関数のカスタム コードを更新して、すべての Apache Axis2 ライブラリがバージョン 1.8.0 となるようにします。次の手順 1 から 3 を実行して、既存の SOAP クレンジング関数を更新します。

**注:** Multidomain MDM 10.5 には、Apache Axis2 のバージョン 1.8.0 へのアップグレードが含まれています。既存の SOAP クレンジング関数を更新して、すべての Apache Axis2 ライブラリがバージョン 1.8.0 となるように、クレンジング関数のカスタム コードを更新します。

1. Hub コンソールを起動して、**クレンジング関数ツール**を開始します。
2. 書き込みロックを取得します。
3. [クレンジング関数] を右クリックして、[IDQ ライブラリの追加] をクリックします。  
[IDQ ライブラリの追加] ダイアログボックスが表示されます。
4. 以下のプロパティを指定します。

プロパティ	説明
ライブラリ名	IDQ ライブラリの名前。名前は、クレンジング関数リストでフォルダ名として表示されます。
IDQ WSDL URI	プラットフォームトランスフォーメーションに関連付けられた WSDL の URI。
IDQ WSDL サービス	プラットフォームトランスフォーメーションに関連付けられた WSDL のサービス。

プロパティ	説明
IDQ WSDL ポート	プラットフォームトランスフォーメーションに関連付けられた WSDL のポート。
説明	クレンジング関数ツールに表示するライブラリの説明テキスト。

5. **[OK]** をクリックします。  
IDQ ライブラリが **[クレンジング関数]** ナビゲータに表示されます。
6. **[更新]** をクリックして、IDQ ライブラリを生成します。  
クレンジング関数ツールでは、プラットフォームトランスフォーメーションに関連付けられた WSDL を取得し、IDQ ライブラリを生成し、クレンジング関数リストに利用可能なクレンジング関数を表示します。
7. クレンジング関数をテストします。  
プラットフォームトランスフォーメーションの代わりに、クレンジング関数を使用できるようになりました。IDQ ライブラリ内のクレンジング関数は、プラットフォームトランスフォーメーションに関連付けられた Web サービスを呼び出すことができます。

## MDM Hub 環境レポートの確認

Hub コンソールのエンタープライズマネージャツールを使用して、Hub サーバー、プロセスサーバー、MDM Hub マスターデータベース、およびオペレーショナルリファレンスストアのデータベースに対する MDM Hub の現在の設定を確認します。コンポーネントのバージョン履歴を確認してください。

環境レポートのコピーをアップグレードドキュメントフォルダ upgradedoc に保存します。

## MDM Hub 環境レポートの保存

MDM Hub 環境レポートを保存するには、Hub コンソールでエンタープライズマネージャツールを使用します。

1. Hub コンソールの **[設定]** ワークベンチで **エンタープライズマネージャツール** を選択します。
2. **エンタープライズマネージャツール** で **[環境レポート]** タブを選択します。
3. **[保存]** をクリックします。
4. **[Hub 環境レポートの保存]** ダイアログボックスで、環境レポートを保存するディレクトリに移動します。
5. **[保存]** をクリックします。

## 外部呼び出しとアプリケーションのアップグレード

バージョン 10.4 では、Multidomain MDM は証明書ベースの認証を使用して、外部呼び出しとアプリケーションを認証します。外部呼び出しとカスタムアプリケーションを使用するには、信頼されたアプリケーションユ

ーザーを設定する必要があります。また、外部呼び出しでは EJB はサポートされていません。代わりに、HTTP 通信プロトコルを使用する必要があります。

ビジネスエンティティサービスとカスタムアプリケーションで、10.4 より以前のバージョンのリソースキットに含まれる BESExternalCall サンプルコードとライブラリを使用する場合は、次のアップグレード手順を実行します。

1. カスタムアプリケーションに信頼されたアプリケーションユーザーを設定します。
2. bes-client.properties ファイルで次の接続プロパティを設定します。

接続プロパティ	説明
siperian-client.protocol	使用する通信プロトコル。デフォルトは HTTP です。デフォルト値は変更しないでください。
bes-client.http.url	MDM に接続するカスタムアプリケーションの URL。次の構文を使用します: http://<MDM ホスト>:<ポート番号>/cmx デフォルトは http://localhost:8080/cmx です。

サンプルの bes-client.properties ファイルは次のディレクトリにあります。  
<リソースキットのインストールディレクトリ>/samples/BESExternalCall/source/resources

3. BESExternalCall サンプルの CustomLogicService.java ファイルの CustomLogicService クラスを確認し、信頼されたアプリケーションユーザーの使用を実装します。

次のコードサンプルは、CustomLogicService クラスを示しています。

```
public class CustomLogicService implements Provider<Source> {

    @Override
    public Source invoke(Source request) {

        CompositeServiceClient compositeServiceClient = createCompositeServiceClient();
        CustomLogicFactory customLogicFactory = new CustomLogicFactoryImpl(compositeServiceClient);
        String appName = "<trusted application user>";
        ExternalCallProcessor externalCallProcessor =
            new ExternalCallProcessor(compositeServiceClient, appName, customLogicFactory);

        return externalCallProcessor.invoke(request);
    }

    private static CompositeServiceClient createCompositeServiceClient() {
        InputStream resourceAsStream = CustomLogicService.class.getResourceAsStream("/bes-
client.properties");
        Properties config = new Properties();
        try {
            config.load(resourceAsStream);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
        return CompositeServiceClient.newCompositeServiceClient(config);
    }
}
```

4. 更新された次の JAR ファイルを使用するカスタムアプリケーションを作成します。

- siperian-api.jar
- siperian-common.jar
- mdm-spi.jar

アプリケーションの作成後、外部呼び出しとアプリケーションでは HTTP 通信プロトコルと証明書ベースの認証が使用されます。

5. カスタムアプリケーションをアプリケーションサーバーに再デプロイします。

# EJB プロトコル用の SiperianClient ライブラリクラス のアップグレード

EJB プロトコルを使用して、サービス統合フレームワーク（SIF）要求を通じて MDM Hub と通信する場合は、最新バージョンの SiperianClient ライブラリクラスを使用する必要があります。カスタムの JNDI ルックアップメソッドを使用する場合は、EJB3 の規則に準拠するようにそのルックアップメソッドを更新してください。

1. 既存の SiperianClient ライブラリクラスを最新バージョンの SiperianClient ライブラリクラスに置き換えます。

SiperianClient ライブラリクラスは、以下のディレクトリにある siperian-api.jar ファイルに含まれています。

- <リソースキットのインストールディレクトリ>\sdk\sifsdk\lib
- <MDM Hub のインストールディレクトリ>\hub\server\lib

2. カスタムの JNDI ルックアップメソッドを使用する場合は、EJB3 の規則に準拠するようにそのルックアップメソッドを更新してください。

## MDM Hub メタデータを準備する

アップグレードタスク	詳細
一致トークンの再生成	ベースオブジェクトごとに一致トークンの生成バッチジョブを実行します。一致トークンの生成バッチジョブは、プロセスサーバーのアップグレード時に更新する SSA-Name3 ライブラリファイルに基づいて、一致トークンを生成します。
検索データを再インデックス処理します。	検索データにアクセント記号付き文字（â, î など）が含まれている場合は、検索データの初期インデックス処理バッチジョブを実行して、データの再インデックス処理を行うことができます。データの再インデックス処理を行うと、検索要求で、アクセント記号付き文字が含まれるレコードを返すことができるようになります。
メタデータキャッシュの構成（省略可能）	バージョン 10.1 以前では、MDM Hub は、メタデータキャッシュに JBoss Cache を使用していました。これらのバージョンのいずれかからアップグレードすると、MDM Hub サーバーは、JBoss Cache 構成ファイルではなく、Infinispan 構成ファイルを使用します。JBoss Cache と同様の結果を得るには、場合によっては Infinispan キャッシュを設定する必要があります。 詳細については、「 <a href="#">メタデータキャッシュの構成（オプション）</a> 」（ <a href="#">ページ 197</a> ）を参照してください。
カスタムインデックスを再登録します。	移行後にカスタムインデックスを再登録する必要があります。カスタムインデックスを再登録するには、registerCustomIndex SIF API を使用します。 RegisterCustomIndex SIF API の詳細については、『 <i>Multidomain MDM サービスの統合フレームワークガイド</i> 』を参照してください。 registerCustomIndex SIF API を実行する SOAP および Java のコードサンプルについては、KB 500116（ <a href="https://kb.informatica.com/howto/6/Pages/19/500116.aspx?myk=500116">https://kb.informatica.com/howto/6/Pages/19/500116.aspx?myk=500116</a> ）を参照してください。

# アップグレードテスト

アップグレードされた MDM Hub をテストします。Multidomain MDM の各実装は固有のもので、テスト要件は開発、テスト、およびプロダクション環境によって異なります。提案されたアップグレードテストがお使いの環境に適さない場合は、独自のテストを設計できます。テスト作業は、実装固有の要件を満たすように設計してください。

## MDM Hub アップグレードテスト

お使いの環境に対し、Hub コンソールのアップグレードの以下のテストを実行します。

1. Hub コンソールを起動します。
2. **【設定】** ワークベンチの中の **【ユーザー】** ツールを選択し、既存のユーザーのプロパティを表示します。
3. **【モデル】** ワークベンチの中の **【スキーマビューア】** ツールを選択し、オペレーショナルリファレンスストアに接続します。 **【スキーマビューア】** でスキーマを確認します。
4. **【モデル】** ワークベンチの中の **【スキーマ】** ツールを選択し、ベースオブジェクトの **【一致/マージ設定】** を表示します。
5. **【ユーティリティ】** ワークベンチで **【バッチビューア】** ツールを選択します。可能な場合は、ステージバッチジョブ、ロードバッチジョブ、マッチバッチジョブ、およびマージバッチジョブのテストバッチジョブを実行します。
6. **【ユーティリティ】** ワークベンチで Process サーバーツールを選択します。登録されている Process サーバーへの接続をテストします。
7. **【モデル】** ワークベンチで **【クレンジング関数】** ツールを選択します。外部クレンジングエンジンごとにテスト用クレンジング関数を実行します。
8. **【データスチュワード】** ワークベンチで **【データマネージャ】** ツールを選択します。一致する 2 つのテストレコードを作成します。
9. **【データスチュワード】** ワークベンチで **【マージマネージャ】** ツールを選択します。2 つのテストレコードを探し、マージして、その後、テストレコードのマージ解除を行います。

## カスタムコードのアップグレードのテスト

カスタムクライアントアプリケーションなどのカスタムコードがある場合は、それが予期したとおりに動作することを確認するテストを実行します。

## プロビジョニングツールアップグレードテスト

プロビジョニングツールにログインします。このツールでは、ビジネスエンティティ、参照エンティティ、アプリケーション、カスタムビュー、タスクなどのための設定が保存された XML ファイルが検証されます。

検証プロセスが正常に実行された場合、次のアップグレードテストに進みます。必要に応じて、設定を検証するために設定をスポットチェックできます。

検証プロセスでエラーが検出された場合、エラーと提案された修正のリストを確認します。修正に、いくつかの設定を削除することが含まれる場合があります。すべての修正を受け入れるか、変更せずにキャンセルするかを選択できます。キャンセルすることを選択した場合、XML ファイル内のエラーを自分で解決する必要があります。XML ファイルは C\_REPOS\_CO\_CS\_CONFIG と C\_REPOS\_COMPONENT\_INSTANCE の各リポジトリテーブルに保存されます。

**注意:** エラーを修正せずに終了した場合、プロビジョニングツールからロックアウトされる可能性があります。

1. プロビジョニングツールにログインします。

2. オペレーショナル参照ストアデータベースを選択します。
3. 検証エラーが表示された場合、提案された修正を確認します。
  - 修正を適用するには、**[適用]** をクリックします。
  - 修正を適用せずに終了するには、**[キャンセル]** をクリックします。  
プロビジョニングツールからログアウトします。XML ファイルを開き、エラーを修正します。
4. 必要に応じて、XML が有効になった後で、設定を検証できます。

## ビジネスエンティティを使用する Data Director のアップグレードテスト

Data Director をビジネスエンティティと使用する場合は、このアプリケーションを開いてテストします。

お使いの環境に適用したアップグレードの以下のテストを実行します。

1. Data Director にログインします。
2. 複数の検索を実行します。
3. 複数のタスクを作成、処理します。
4. テストレコードを挿入します。
5. テストレコードをコピーし、もう 1 つのテストレコードを作成します。
6. 検索を実行して、これら 2 つのレコードを見つけます。
7. 2 つのテストレコードをマージ、アンマージします。

## サブジェクト領域を使用する Data Director のアップグレードテスト

Data Director をサブジェクト領域と使用する場合、テストを開始する前に、このアプリケーションをデプロイする必要があります。

お使いの環境に適用したアップグレードの以下のテストを実行します。

1. Data Director コンフィギュレーションマネージャを起動し、Data Director アプリケーションインスタンスをデプロイします。
2. Data Director にログインします。
3. 複数の検索を実行します。
4. 複数のタスクを作成、処理します。
5. テストレコードを挿入します。
6. テストレコードをコピーし、もう 1 つのテストレコードを作成します。
7. 検索を実行して、これら 2 つのレコードを見つけます。
8. 2 つのテストレコードをマージ、アンマージします。

# Hub サーバーの全般的なプロパティの設定

アップグレードプロセスでは、Hub サーバーのプロパティ値が保持されます。以前のバージョンからアップグレードする場合、cmxserver.properties ファイルには新しいプロパティおよび、そのリリースに追加されたプロパティデフォルトへの変更は含まれません。

JBoss 環境にインストールした場合、cmx.jboss7.management.port プロパティの値を 9999 から 9990 に変更します。

そのリリースに追加された新しいプロパティを必ず構成してください。詳細については、『*Multidomain MDM のリリースガイド*』および『*Multidomain MDM の設定ガイド*』を参照してください。

## Data Director および Hub サーバーのプロパティ

アップグレードプロセスでは、Data Director に影響を与える Hub サーバーのプロパティ値が保持されます。

以前のバージョンからアップグレードする場合、アップグレード前の cmxserver.properties ファイルには、バージョン 10.0.0 以降で追加されたいくつかのプロパティが含まれていません。アップグレードプロセスによって、プロパティが cmxserver.properties ファイルに追加されたことを確認します。必要に応じて、足りないプロパティを<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/resources/cmxserver.properties ファイルの末尾に追加します。

プロパティのデフォルト値によって、Data Director アプリケーションの現在の動作が維持されます。プロパティをカスタマイズする前に、『*Multidomain MDM の設定ガイド*』の Hub サーバーのプロパティに関する説明を参照してください。

### バージョン 10.3 で追加されたプロパティ

新しいプロパティが、ファイルのアップロード、タスクマネージャ、ワークフロー図をサポートするために追加されました。

```
# File upload properties
# -----
# Maximum upload size.
cmx.file.max_file_size_mb=20
# Maximum number of concurrent uploads.
cmx.file.max_concurrent_uploads=20
# Type of files that can be uploaded.
cmx.file.allowed_file_extensions=pdf,jpg
# Number of minutes until an uploaded file expires.
# To avoid expiration, set to 0.
cmx.server.attachment.temp.ttl_minutes=60

# Task Manager property
# -----
# Set to true to display the Task Manager tab in applications
# that use subject areas.
cmx.dataview.taskmanager.enabled=true

# Workflow diagram properties
# -----
# Set to true to display the workflow diagram associated
# with the tasks in the Task Manager for the users with
# the ActiveVOS abAdmin role.
cmx.e360.BPMProcess.view.enabled=false
cmx.e360.BPMProcess.view.autologout.seconds=30
```

**注:** Data Director アプリケーション内でのフルテキスト検索に Elasticsearch を使用する場合、Elasticsearch プロパティを手動で追加します。詳細については、[「検索設定のアップグレードの概要」](#) (ページ 145) を参照してください。

### バージョン 10.2 以前で追加されたプロパティ

プロパティが、データタブ、検索、エンティティ 360 フレームワークをサポートするために追加されました。デフォルト値を追加し、必要に応じてそれらを変更します。

```
# View properties
# -----
# Show or hide the views for subject areas.
cmx.dataview.enabled=true
# Show or hide the views for business entities.
cmx.e360.view.enabled=false
# Show or hide the Cross-reference view and Match view.
cmx.e360.match_xref.view.enabled=false

# Search with Solr (formerly Smart Search) properties
# -----
# Set to true to use Solr for search.
cmx.ss.enabled=false
```

## Data Director のグローバルプロパティ

Data Director 環境に、カラムの表示または非表示に関するユーザー設定が含まれていた場合、このバージョンで暗号化ハッシュアルゴリズムが変更されているため、アップグレードするとその設定は失われます。アップグレード後、C\_REPOS\_DS\_PREF\_DETAIL テーブルをクリアし、ユーザー設定を再作成します。

ユーザー設定などの Data Director のグローバルプロパティの更新方法の詳細については、『*Multidomain MDM Data Director の実装ガイド*』を参照してください。

## ビジネスエンティティスキーマの生成

ビジネスエンティティサービスを使用する場合、アップグレード後に、ビジネスエンティティスキーマを生成する必要があります。ビジネスエンティティスキーマを生成するには、Informatica Data Director コンフィギュレーションマネージャを使用します。

ビジネスエンティティスキーマを生成する前に、カスタマイズされたビジネスエンティティまたはビジネスエンティティサービスをバックアップします。

Informatica Data Director コンフィギュレーションマネージャからビジネスエンティティスキーマを生成するには、[アプリケーション] 画面で **【ビジネスエンティティスキーマの生成】** をクリックします。

## 第 9 章

# 検索設定のアップグレード

この章では、以下の項目について説明します。

- [検索設定のアップグレードの概要, 145 ページ](#)
- [手順 1.Elasticsearch のインストールとセットアップ, 145 ページ](#)
- [手順 2。検索のための MDM Hub プロパティの設定, 150 ページ](#)
- [手順 3。プロビジョニングツールを使用した検索の設定, 152 ページ](#)
- [手順 4.オペレーショナル参照ストアの検証, 154 ページ](#)
- [手順 5。検索データのインデックス処理, 155 ページ](#)
- [Elasticsearch バージョン 7.17.0 へのアップグレード（オプション）, 155 ページ](#)

## 検索設定のアップグレードの概要

Data Director アプリケーションまたはカスタムアプリケーションを使用して、特定のビジネスエンティティ内のデータを検索できます。以前は、検索処理に Solr を設定していましたが、現在はサポートされていません。検索操作用に Elasticsearch を設定する必要があります。

Elasticsearch を使用するための検索設定をアップグレードするには、次のタスクを実行します。

1. Elasticsearch をインストールしてセットアップします。
2. 検索用の MDM Hub のプロパティを設定します。
3. プロビジョニングツールを使用して検索を設定します。
4. オペレーショナルリファレンスストア（ORS）を検証します。
5. 検索データをインデックス処理します。

## 手順 1.Elasticsearch のインストールとセットアップ

検索を設定するには、Elasticsearch をインストールしてセットアップする必要があります。

Elasticsearch をセットアップするには、次のタスクを実行します。

1. インストール前作業を完了します。
2. Elasticsearch をインストールします。
3. Elasticsearch Java 仮想マシン（JVM）を設定します。

4. Elasticsearch プロパティファイルを設定します。
5. Elasticsearch を保護します。
6. 分析プラグインをインストールします。
7. ストップワード、シノニム、文字マッピングの設定
8. Elasticsearch を起動します。

## インストール前のタスクの完了

Elasticsearch クラスタをインストールおよび設定する前に、環境を準備し、高可用性を設定するかどうかを決定します。

### すべての環境でのタスク

次のタスクを実行して、インストール環境を準備します。

- 各マシンが Elasticsearch のサポートしているバージョンのハードウェア要件を満たしていることを確認します。ハードウェアの詳細については、Elasticsearch のマニュアルを参照してください。
- 各マシンが Elasticsearch のサポートしているバージョンのソフトウェア要件（サポートしているオペレーティングシステムおよび Java バージョンなど）を満たしていることを確認します。ソフトウェア要件の詳細については *Elasticsearch のサポートマトリックス* を参照してください。
- スワップ、ファイル記述子および仮想メモリなどの重要なシステム構成を完了します。重要なシステム構成の詳細については、Elasticsearch のマニュアルを参照してください。

### UNIX 環境でのタスク

UNIX 環境では、次のタスクを実行します。

- ファイル記述子の数の不足によるデータ損失を避けるために、ファイル記述子の数は 65536 以上に設定します。
- メモリのスワップを防止するには、スワップを防止するようにシステムを設定します。mlockall を使用して、メモリ内のヒープをロックするように Java 仮想マシン (JVM) を設定できます。

### 高可用性の要件

インデックス処理および検索するデータが大量にある場合、可用性の高い Elasticsearch クラスタを実装することをお勧めします。高可用性クラスタは複数のノードを持ち、ノード間で負荷を分散できます。プロダクション環境で 1 つのノードに障害が発生した場合、クラスタは別のノードに負荷を分散します。

インストール前のタスクとして、可用性の高い Elasticsearch クラスタを実装するかどうかを決定します。実装する場合、Elasticsearch クラスタを通常どおりに構成しますが、次の追加の要件を満たすことを確認します。

- Elasticsearch クラスタ内に 3 つ以上のノードがある。  
**ヒント:** 最初は小さなクラスタを設定して、必要に応じて拡大縮小することができます。負荷を分析し、ノードの障害を処理するための十分な容量があることを確認します。
- 各ノードは個別の専用マシンに構成される。
- 安定性とパフォーマンスを確保するために、少なくとも 3 つのノードをマスタノードにする。  
Elasticsearch では、奇数のマスタノードが推奨されることに注意してください。
  - クラスタ内のノードが 3 つのみである場合、すべてのノードをマスタノードとして構成します。
  - クラスタ内に 3 つよりも多くのノードがある場合、3 ノードをマスタノードとして構成し、残りのノードをデータノードとして構成します。
- Elasticsearch クラスタサイズに基づき、レプリカの数を決める。プロビジョニングツールを使用して Elasticsearch インデックスを構成する場合、使用するレプリカ数を指定できます。

- ノードごとに、elasticsearch.yml 構成ファイルに次の追加のプロパティを設定します。
  - discovery.zen.minimum\_master\_nodes
  - discovery.zen.ping.unicast.hosts

ハードウェア要件、システム構成、およびプロパティ値を含む、高可用性クラスタの詳細については、Elasticsearch のマニュアルを参照してください。

## Elasticsearch のインストール

Hub サーバーとプロセスサーバーのインストール後、検索を設定するには、Elasticsearch クラスタをインストールしてセットアップします。

Elasticsearch インストールにサポートされているオペレーティングシステムと Java バージョンを使用していることを確認します。詳細については、Elasticsearch のサポートマトリックスを参照してください。

Elasticsearch をインストールし、クラスタをセットアップする方法の詳細については、Elasticsearch のドキュメントを参照してください。

1. Elastic Web サイトから、サポートされているバージョンの Elasticsearch アーカイブファイルをダウンロードします。  
サポートされているバージョンの詳細については、Product Availability Matrix (PAM) を参照してください。PAM には <https://network.informatica.com/community/informatica-network/product-availability-matrices> からアクセスできます。
2. Elasticsearch アーカイブファイルを抽出します。

## Elasticsearch Java 仮想マシン (JVM) の設定

マシン上で使用可能な RAM の量に基づいてヒープサイズを使用するように、Elasticsearch Java 仮想マシン (JVM) を設定します。JVM を設定するには、jvm.options ファイルを編集します。

1. 次のディレクトリにある jvm.options ファイルを検索します。  
<elasticsearch installation directory>/config
2. テキストエディタを使用してファイルを開き、以下のプロパティを編集します。

プロパティ	説明
-Xms	最小ヒープサイズ。デフォルトは 1 GB です。
-Xmx	最大ヒープサイズ。デフォルトは 1 GB です。
-XX:HeapDumpPath	ヒープダンプパス。デフォルトは/var/lib/elasticsearch です。マルチクラスタ環境では、このプロパティを別のパスに設定する必要があります。

**注:** 最小ヒープサイズ (Xms) と最大ヒープサイズ (Xmx) を同じ値に設定します。その他のプロパティについては、デフォルト設定を使用します。

## Elasticsearch プロパティファイルの設定

Informatica は、サンプルの Elasticsearch プロパティファイルを提供しています。Elasticsearch を設定するには、そのプロパティファイルを編集します。

1. 次のディレクトリにある elasticsearch.yml ファイルを検索します。  
<elasticsearch installation directory>/config
2. テキストエディタを使用してファイルを開き、以下のプロパティを編集します。

プロパティ	説明
bootstrap.memory_lock	メモリのロックを設定します。Elasticsearch メモリがスワップされて消されることがないようにするには、true を設定します。デフォルトは true です。
cluster.name	Elasticsearch クラスタに一意の名前を指定します。複数のクラスタがある場合、各クラスタの名前が一意であることを確認します。クラスタに複数のノードがある場合、そのクラスタ内の各ノードで、同じクラスタ名が指定されていることを確認します。
discovery.zen.minimum_master_nodes	データの損失を防ぎ、クラスタの安定性を維持するために、複数ノードクラスタに必要です。値を(マスタに適格なノード数 / 2) + 1 に設定します。 例えば、クラスタに 3 つのノードがあり、それらすべてがマスタに適格なノードであり、データを格納できる場合、このプロパティを (3 / 2) + 1 に設定します。この値は、2 に丸められます。
discovery.zen.ping.unicast.hosts	複数ノードのクラスタに必要です。このプロパティは、検出設定の指定に使用されます。この設定は、クラスタ内のノードの IP アドレスとポートポートのリストです。このプロパティを設定するには、["host1:port1","host2:port2","host3:port3"] の形式を使用します。
http.port	HTTP 要求のポートです。デフォルトは 9200 です。
network.host	ホストの IP アドレスは、バインドアドレスとして使用されます。
node.data	CRUD や検索などのデータ関連操作を実行するためのデータノードとして、ノードを有効にします。デフォルトは true です。
node.ingest	インデックス処理の前に、データを変換してエンリッチ化する取り込みデータノードとして、ノードを有効にします。デフォルトは true です。
node.master	クラスタを制御するマスタノードとして、ノードを有効にします。クラスタに複数のノードがある場合、1 つ以上のノードをマスタノードとして有効にします。高可用性のためには、複数のノードをマスタノードとして設定します。デフォルトは true です。
node.name	ノードの一意の名前を指定します。

プロパティ	説明
path.data	データを保存するディレクトリのパスを指定します。複数のデータディレクトリを設定できます。複数のデータディレクトリの設定に関する詳細については、Elasticsearch のマニュアルを参照してください。
path.logs	ログファイルのパスを指定します。
transport.tcp.port	TCP バインドポートを指定します。デフォルトは 9300 です。

3. 同じ名前 (elasticsearch.yml) でプロパティファイルを保存します。

## Elasticsearch の保護 (オプション)

Elasticsearch のインストール後、MDM Hub と Elasticsearch 間の通信を保護します。また、Elasticsearch クラスタも保護します。

Elasticsearch の保護の詳細については、Elasticsearch のセキュリティに関するドキュメントを参照してください。

## 分析プラグインのインストール

音標および日本語 (kuromoji) 分析プラグインをインストールします。これは、新しいアナライザ、トークナイザ、トークンフィルタおよび文字フィルタを追加することで Elasticsearch を拡張します。音標分析プラグインはトークンを分析し、それに相当する音標に変換します。日本語 (kuromoji) 分析プラグインは Kuromoji アナライザを使用することで日本語を分析します。

1. 音標および日本語 (kuromoji) 分析プラグインを Elastic Web サイトからダウンロードします。
2. 次のコマンドを実行することで、各クラスタに音標分析プラグインをインストールします。  

```
sudo bin/elasticsearch-plugin install analysis-phonetic
```
3. 次のコマンドを実行することで、各クラスタに日本語 (kuromoji) 分析プラグインをインストールします。  

```
sudo bin/elasticsearch-plugin install analysis-kuromoji
```
4. 各クラスタノードを再起動します。

## ストップワード、シノニム、文字マッピングの設定

検索の実行時に、MDM では "and"、"an"、"is" などの一般的な語を無視できます。また、検索文字列のシノニムを検索することもできます。例えば、"William" を検索するときに、検索結果に "Will" や "Willy" などのシノニムを含めることができます。

一般的な語を無視したり、検索結果にシノニムを含めたりするために、Informatica ではストップワードやシノニムを含むテキストファイルを用意しています。また、ユーザーが独自にファイルを設定することもできます。

中国語、日本語、韓国語などの言語でデフォルトの Elasticsearch アナライザを使用するために、mapping-FoldToASCII.txt というマッピングファイルも提供しています。デフォルトアナライザの文字フィルタは、マッピングファイルを使用して Unicode の基本ラテン文字ブロックに含まれていない英字、数字、シンボリック文字を対応する ASCII 文字に変換します。

stopwords.txt、synonyms.txt、stopwords\_ja.txt、および mapping-FoldToASCII.txt ファイルを必要とされる場合は、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。

ストップワード、シノニム、文字マッピングを設定するには、次の手順を実行します。

1. 次の場所に analysis ディレクトリを作成します。  
`<elasticsearch installation directory>/config`
2. stopwords.txt ファイルと synonyms.txt ファイルを analysis ディレクトリにコピーします。
3. 日本語など他言語のストップワードを設定するには、次の場所に lang ディレクトリを作成します。  
`<elasticsearch installation directory>/config/analysis`
4. stopwords\_ja.txt などの他言語のストップワードファイル、および mapping-FoldToASCII.txt を lang ディレクトリにコピーします。

## Elasticsearch の起動

Elasticsearch のセットアップ後、変更内容が有効になるように、Elasticsearch クラスタ内の各ノードを起動します。

**ヒント:** Elasticsearch を起動するときに、メモリのロックの問題が発生する場合、soft memlock unlimited と hard memlock unlimited を設定する必要がある可能性があります。

1. コマンドプロンプトを開き、次のディレクトリに移動します。  
`<elasticsearch installation directory>/bin`
2. 次のコマンドを実行します。  
UNIX の場合: `elasticsearch.sh`  
Windows の場合: `elasticsearch.bat`

## 手順 2。検索のための MDM Hub プロパティの設定

MDM Hub プロパティを設定するには、Hub コンソール、プロセスサーバーのプロパティファイル、Hub サーバーのプロパティファイルを使用します。

1. プロセスサーバーのプロパティを設定します。
2. Hub サーバーのプロパティを設定します。

## Hub サーバーでの検索の設定

検索を有効にするすべての Hub サーバーインスタンスを設定する必要があります。Hub コンソールの Hub サーバーツールおよび `cmxserver.properties` ファイルを使用して、検索の Hub サーバープロパティを設定します。

1. テキストエディタを使用して、`<MDM Hub インストールディレクトリ>\hub\server\resources\cmxserver.properties` にある `cmxserver.properties` ファイルを開きます。

## 2. 検索用の次のプロパティを設定します。

cmx.ss.engine

Elasticsearch エンジンを検索に使用する場合は必須です。このプロパティは手動で追加して es に設定します。

ex.max.conn.per.host

ホストに接続する Elasticsearch ノードの最大数を設定します。ホスト上の Elasticsearch クラスタノードの数に設定します。

ex.max.threads

Apache Elasticsearch クラスタでノードごとに非同期ノンブロッキングレシーバを使用するスレッドの最大数を設定します。デフォルトは1です。

この値を変更するのは、Informatica グローバルカスタマサポートから提案された場合のみにしてください。

es.index.refresh.interval

「スマート検索データの初期インデックス処理」バッチジョブが実行された後に Elasticsearch がデータへの変更をコミットする間隔（秒）を設定します。そのデータを検索できるようになるのは、この時間間隔の後です。デフォルトは30です。

このプロパティは、初回のインデックス処理でインデックス処理量が大きくなった場合に影響します。この値を変更するのは、Informatica グローバルカスタマサポートから提案された場合のみにしてください。

ssl.keyStore

アプリケーションサーバーの HTTPS ポートを使用して Hub サーバーを設定する場合は必須です。プロパティを手動で追加します。キーストアファイルの絶対パスおよびファイル名。

ssl.keyStore.password

アプリケーションサーバーの HTTPS ポートを使用して Hub サーバーを設定する場合は必須です。プロパティを手動で追加します。キーストアファイルのプレーンテキストパスワード。

ssl.trustStore

アプリケーションサーバーの HTTPS ポートを使用して Hub サーバーを設定する場合は必須です。プロパティを手動で追加します。トラストストアファイルの絶対パスおよびファイル名。

ssl.trustStore.password

アプリケーションサーバーの HTTPS ポートを使用して Hub サーバーを設定する場合は必須です。プロパティを手動で追加します。トラストストアファイルのプレーンテキストパスワード。

Hub サーバーのプロパティを更新したら、オペレーショナルリファレンスストア（ORS）を検証して、Hub コンソールを再起動する必要があります。

## プロセスサーバーでの検索の設定

Elasticsearch を使用した検索を設定するときは、すべてのプロセスサーバーインスタンスで検索を有効にします。Hub コンソールのプロセスサーバーツールおよび cmxcleanse.properties ファイルを使用して、検索のプロセスサーバープロパティを設定します。

1. ノードの Hub コンソールでプロセスサーバーツールを起動します。
2. **【書き込みロック】** > **【ロックの取得】** の順にクリックします。
3. プロセスサーバーツールの右側のペインで、**【プロセスサーバーの追加】** ボタンをクリックします。  
**【プロセスサーバーの追加/編集】** ダイアログボックスが表示されます。
4. 検索を有効にするには、プロセスサーバーの **【検索処理を有効にする】** プロパティを選択します。

5. **[OK]** をクリックしてから **[保存]** をクリックします。
6. cmxcleanse.properties ファイルを編集します。  
cmxcleanse.properties ファイルは、<MDM Hub Installation Directory>\hub\cleanse\resources にあります。
  - a. 検索について次のプロパティを設定します。  
cmx.ss.engine  
Elasticsearch エンジンを検索に使用する場合は必須です。このプロパティは手動で追加して es に設定します。  
  
ex.max.conn.per.host  
ホストに接続する Elasticsearch ノードの最大数を設定します。ホスト上の Elasticsearch クラスターノードの数に設定します。  
  
ex.max.threads  
Apache Elasticsearch クラスターでノードごとに非同期ノンブロッキングレシーバを使用するスレッドの最大数を設定します。デフォルトは1です。  
この値を変更するのは、Informatica グローバルカスタマサポートから提案された場合のみにしてください。
  - b. Solr を使用する検索に対して設定された次のプロパティを削除します。
    - solr.allowAdminConsole
    - zookeeper.tickTime
    - pingSolrOnStartup
7. cmxcleanse.properties ファイルを保存します。
8. アプリケーションサーバーを再起動します。

## 手順 3。プロビジョニングツールを使用した検索の設定

Elasticsearch をセットアップし、MDM Hub プロパティを設定したら、プロビジョニングツールを使用して、検索環境を設定します。

1. Elasticsearch クラスターを設定します。
2. 検索結果ビューを設定します。

### Elasticsearch クラスターの設定

プロビジョニングツールを使用して、MDM アプリケーション向けに Elasticsearch クラスターを設定します。検索 API でこの設定が使用されます。Data Director アプリケーションと任意のカスタムアプリケーションが、この検索 API を使用します。

**注:** Elasticsearch クラスターを設定するときは、クラスターのマスタノードのみを指定する必要があります。

1. サポートされているブラウザを開いて、次の URL を入力します。  
`https://<MDM Hub Server host name>:<MDM Hub Server port number>/provisioning/`  
**[ログイン]** ページが表示されます。
2. ユーザー名とパスワードを入力し、**[ログイン]** をクリックします。

3. **【データベース】** リストから、Elasticsearch クラスタを設定するデータベースを選択します。
4. **【設定】** > **【インフラストラクチャの設定】** をクリックします。  
**【インフラストラクチャの設定】** ページが表示されます。
5. リストから **【Elasticsearch クラスタ】** を選択してから **【ESCluster】** をクリックします。  
ツリービューのパネルに **【ESCluster】** が表示されます。
6. Elasticsearch クラスタノードを設定するには、ツリービューのパネルで **【esNode】** を選択してから **【作成】** をクリックします。
7. 設定した Elasticsearch クラスタの次のプロパティを指定します。

プロパティ	説明
名前	Elasticsearch クラスタ内のマスタノードの名前を指定します。
URL	Elasticsearch クラスタ内のマスタノードの URL を指定します。URL の形式は <code>https://&lt;host name&gt;:&lt;port&gt;</code> です。
エラスティック検索ノードの更新	<p>既存のノードが引き続き検索要求を管理する場合に、インデックスの詳細でノードを更新するかどうかを示します。スマート検索データの初期インデックスバッチジョブを手動で実行し、インデックスが新しいノードに追加された後に既存のノードを削除できます。次に、チェックボックスをオフにして、新しいノードが検索要求を管理できるようにします。</p> <p><b>注:</b> エラスティック検索バージョン 6 からバージョン 7 にアップグレードする場合に適用されます。エラスティック検索認証サポートは、バージョン 7.x でのみ有効です。</p> <p>エラスティック検索のインストールと設定の詳細については、『<i>Multidomain MDM 設定ガイド</i>』を参照してください。</p>

8. **【適用】** をクリックします。
9. 追加のマスタノードを作成する場合、手順 6 から 8 を繰り返します。
10. 変更内容を MDM Hub にパブリッシュします。
  - a. **【パブリッシュ】** をクリックします。  
確認ダイアログボックスが表示され、変更内容をパブリッシュまたは確認するように求められます。
  - b. 変更内容を確認するか、確認せずにパブリッシュします。
    - 確認せずにパブリッシュする場合は、**【パブリッシュ】** をクリックします。
    - 確認後にパブリッシュするには、**【変更の確認】** をクリックし、画面に表示される指示に従います。

## 検索またはクエリ結果の表示の設定

プロビジョニングツールを使用して、検索に使用するビジネスエンティティビューを設定します。検索結果には、検索結果用に設定するビジネスエンティティビューに属するフィールドのみが含まれます。また、検索フィルタの表示順序を変えることもできます。

検索可能なビューを設定する前に、検索結果に使用するビジネスエンティティビューを作成します。

**注:** 検索結果にビジネスエンティティの子レコードフィールドを表示するには、ビジネスエンティティから変換したビジネスエンティティビューを使用します。ビューのルートレコードレベルに、子レコードのフィールドが含まれていることを確認します。

1. プロビジョニングツールにログインします。
2. **【データベース】** リストから、アプリケーションが関連付けられているデータベースを選択します。
3. **【設定】** > **【アプリケーションエディタ】** をクリックします。  
**【アプリケーション】** ページが表示されます。
4. **【アプリケーション】** リストから、検索を設定するアプリケーションを選択します。  
アプリケーションがない場合、検索を設定する前に、アプリケーションを作成します。
5. **【ツリービュー】** パネルで **【検索設定】** を選択し、**【作成】** をクリックします。
6. **【プロパティ】** パネルでビジネスエンティティと、検索またはクエリ結果を表示するのに使用するビジネスエンティティビューを選択します。  
ビジネスエンティティビューを選択しないと、検索およびクエリ結果にはすべてのビジネスエンティティフィールドが含まれます。
7. 検索を設定した場合、必要に応じて、フィルタを選択し、検索フィルタの表示順序を設定します。
  - a. **【フィルタ表示順序】** の横の **【編集】** アイコンをクリックします。  
**【フィルタ表示順序の編集】** ダイアログボックスが表示されます。このダイアログボックスには、ビジネスエンティティモデルでフィルタ可能として設定されているフィールドであるフィルタが含まれます。
  - b. **【使用可能なフィルタ】** セクションから **【選択したフィルタ】** セクションへフィルタをドラッグします。
  - c. 順序を設定するには、フィルタをドラッグして上下に移動します。
  - d. **【OK】** をクリックします。
8. **【適用】** をクリックします。  
検索設定は、一時的なワークスペースに保存されます。
9. 変更内容を MDM Hub にパブリッシュします。
  - a. **【パブリッシュ】** をクリックします。  
確認ダイアログボックスが表示され、変更内容をパブリッシュまたは確認するように求められます。
  - b. 変更内容を確認するか、確認せずにパブリッシュします。
    - 確認せずにパブリッシュする場合は、**【パブリッシュ】** をクリックします。
    - 確認後にパブリッシュするには、**【変更の確認】** をクリックし、画面に表示される指示に従います。

## 手順 4. オペレーショナル参照ストアの検証

Elasticsearch 構成の影響を受けるオペレーショナル参照ストア（ORS）のメタデータを検証するには、Hub コンソールで Repository Manager ツールを使用します。

1. Hub コンソールを起動し、MDM Hub マスタデータベースに接続します。
2. **【設定】** ワークベンチを展開し、**【Repository Manager】** をクリックします。  
Repository Manager が表示されます。

3. **【検証】** タブをクリックし、検証するリポジトリを選択します。
4. **【検証】** をクリックします。  
**【検証チェックの選択】** ダイアログボックスが表示されます。
5. 実行する検証チェックを選択します。
6. **【OK】** をクリックします。  
Repository Manager により、リポジトリが検証され、すべての問題が **【検出された問題】** ペインに表示されます。
7. 問題を修復するには、**【修復】** をクリックします。

## 手順 5。検索データのインデックス処理

環境内にデータが含まれる場合、「スマート検索データの初期インデックス処理」バッチジョブを手動で実行し、データをインデックス処理します。環境内にデータがない場合、「スマート検索データの初期インデックス処理」ジョブを実行する必要はありません。ロードバッチジョブを実行してデータをロードすると、ロードバッチジョブで「スマート検索データの初期インデックス処理」バッチジョブが自動的に実行され、データがインデックス処理されます。検索要求では、レコードの検索のためにインデックスが使用されます。

「スマート検索データの初期インデックス処理」バッチジョブは、ビジネスエンティティに関係するすべてのベースオブジェクトに対して実行します。「スマート検索データの初期インデックス処理」バッチジョブをベースオブジェクトに対して実行すると、Elasticsearch サーバーによって検索可能なフィールド内のデータがインデックス処理されます。次に、このジョブによって、検索可能なフィールドが属するビジネスエンティティを表すすべてのコレクションに対して、インデックス処理されたデータが追加されます。コレクションが大きすぎる場合、コレクションを 1 つまたは複数のシャードに分割できます。シャードは、複数のノードにわたって分割されるコレクションの論理的な断片です。検索を実行すると、Elasticsearch サーバーがコレクションを読み取って一致するフィールドを返します。

「スマート検索データの初期インデックス処理」バッチジョブは、すべてのレコードのインデックス処理要求をキューに入れた後、レコードを非同期的にインデックス処理して、正常に完了したことを報告します。検索要求は、インデックス要求が正常に完了した後にのみ、インデックス処理されたレコードを表示できます。この処理には数分かかる場合があります。

**重要:** データのインデックス処理後にフィールドの検索可能なプロパティを更新した場合、インデックスは削除されます。データをインデックス処理するには、「スマート検索データの初期インデックス処理」バッチジョブを実行する必要があります。さらに、インデックス処理はリソースを大量に消費する処理であるため、「スマート検索データの初期インデックス処理」バッチジョブは同時に複数実行しないでください。

## Elasticsearch バージョン 7.17.0 へのアップグレード (オプション)

既存の MDM 環境で Elasticsearch バージョン 6.2.3 または 6.8.6 と Search Guard を使用している場合は、X-Pack コミュニティエディションを使用する Elasticsearch バージョン 7.17.0 にアップグレードすることをお勧めします。Elasticsearch バージョン 7.17.0 は下位互換性があります。

## Elasticsearch バージョン 7.17.0 へのアップグレードの前提条件

Elasticsearch バージョン 7.17.0 にアップグレードするには、既存の Elasticsearch バージョン 6.2.3 または 6.8.6 の `elasticsearch.yml` ファイルに次のプロパティが含まれていることを確認しておく必要があります。

- `cluster.name:<user defined value>`
- `node.name:<user defined value>`

**注:** これらのプロパティを既存の Elasticsearch バージョンの設定ディレクトリから Elasticsearch 7.17.0 設定ディレクトリに移行する必要があります。移行手順は、記載されているとおりの順番で実行してください。

既存の Elasticsearch バージョンにこれらの設定プロパティが含まれていない場合は、既存の Elasticsearch バージョンに互換性がないため、Elasticsearch バージョン 7.17.0 にアップグレードしないでください。

## Elasticsearch プロパティファイルの設定

Elasticsearch 7.17.0 を設定するには、このプロパティファイルを編集します。

1. 次のディレクトリに移動します。  
`<elasticsearch 7.17.0 installation directory>/config`
2. テキストエディタで `elasticsearch.yml` ファイルを開き、以下のプロパティを編集します。

プロパティ	説明
<code>bootstrap.memory_lock</code>	メモリをロックするには、 <code>true</code> に設定します。デフォルトは <code>true</code> です。
<code>cluster.name</code>	Elasticsearch クラスターの一意の名前です。複数のクラスタがある場合、各クラスタの名前が一意であることを確認します。クラスタに複数のノードがある場合、そのクラスタ内の各ノードで、同じクラスタ名が指定されていることを確認します。
<code>http.port</code>	HTTP 要求のポートです。Elasticsearch 7.17.0 の HTTP ポートが有効になっており、ポート番号が Elasticsearch バージョン 6.2.3 または 6.8.6 のポート番号と競合していないことを確認してください。デフォルトは 9200 です。
<code>network.host</code>	ホストの IP アドレスは、バインドアドレスとして使用されます。
<code>node.data</code>	CRUD や検索などのデータ関連操作を実行するデータノードとしてノードを使用するかどうかを指定します。 <code>true</code> に設定します。デフォルトは <code>true</code> です。
<code>node.ingest</code>	インデックス処理の前にデータを変換してエンリッチ化する取り込みノードとしてノードを使用するかどうかを指定します。 <code>true</code> に設定します。デフォルトは <code>true</code> です。
<code>node.master</code>	クラスタを制御するマスタノードとしてノードを使用するかどうかを指定します。クラスタに複数のノードがある場合、1 つ以上のノードをマスタノードとして有効にします。高可用性のためには、複数のノードをマスタノードとして設定します。 <code>true</code> に設定します。デフォルトは <code>true</code> です。

プロパティ	説明
node.name	ノードの一意の名前を指定します。
transport.tcp.port	TCP バインドポートを指定します。Elasticsearch 7.17.0 の TCP バインドポートが有効になっており、ポート番号が Elasticsearch バージョン 6.2.3 または 6.8.6 のポート番号と競合していないことを確認してください。デフォルトは 9300 です。

3. elasticsearch.yml ファイルを保存します。
4. 分析プラグインをインストールします。分析プラグインのインストール方法の詳細については、[「分析プラグインのインストール」 \(ページ 149\)](#)のトピックを参照してください。

## Elasticsearch インデックスのアップグレード

既存の Elasticsearch バージョン 6.2.3 または 6.8.6 のインデックスプロパティを Elasticsearch バージョン 7.17.0 のインデックスプロパティにアップグレードします。インデックスをアップグレードするには、Elasticsearch 設定プロパティファイルを編集します。

1. 次のディレクトリに移動します。  
`<elasticsearch>6.2.3 or 6.8.6> installation directory>/config`
2. analysis フォルダをコピーします。
3. analysis フォルダを次のディレクトリに貼り付けます。  
`<elasticsearch 7.17.0 installation directory>/config`
4. Elasticsearch バージョン 7.17.0 の設定ディレクトリ内の elasticsearch.yml ファイルで、以下のプロパティを編集します。
5. テキストエディタを使用してファイルを開き、以下のプロパティを編集します。

プロパティ	説明
xpack.security.transport.ssl.enabled	REST レイヤのトランスポートレイヤセキュリティ (TLS) を有効にするかどうかを指定します。true に設定します。デフォルトは true です。
xpack.security.transport.ssl.verification_mode	証明書を検証するモードを指定します。サポートされているモードは certificate です。値として certificate を入力します。
xpack.security.transport.ssl.keystore.path	キーストアファイルの相対パスを指定します。キーストアを作成するために sip_ant スクリプトを使用した場合、デフォルトのパスを使用します。デフォルトは MDM_ESKEYSTORE_FILE_JKS.keystore です。
xpack.security.transport.ssl.truststore.path	トラストストアファイルの相対パスを指定します。トラストストアを作成するために sip_ant スクリプトを使用した場合、デフォルトのパスを使用します。デフォルトは MDM_TRUST_JKS.keystore です。
xpack.security.transport.ssl.keystore.password	キーストアのパスワードを指定します。デフォルトは infa@es です。

プロパティ	説明
xpack.security.transport.ssl.truststore.password	トラストストアのパスワードを指定します。デフォルトは infa@es です。
xpack.security.http.ssl.enabled	X-Pack を有効にするかどうかを指定します。true に設定します。
xpack.security.http.ssl.verification_mode	証明書を検証するモードを指定します。サポートされているモードは certificate です。値として certificate を入力します。
xpack.security.http.ssl.client_authentication	REST レイヤに TLS クライアント認証モードを使用するかどうかを指定します。REQUIRED に設定します。
xpack.security.http.ssl.keystore.path	キーストアファイルの相対パスを指定します。キーストアを作成するために sip_ant スクリプトを使用した場合、デフォルトのパスを使用します。デフォルトは MDM_ESCLIENT_FILE_JKS.keystore です。
xpack.security.http.ssl.truststore.path	トラストストアファイルの相対パスを指定します。トラストストアを作成するために sip_ant スクリプトを使用した場合、デフォルトのパスを使用します。デフォルトは MDM_TRUST_JKS.keystore です。
xpack.security.http.ssl.truststore.password	トラストストアのパスワードを指定します。デフォルトは infa@es です。
xpack.security.http.ssl.keystore.password	キーストアのパスワードを指定します。デフォルトは infa@es です。
xpack.security.authc.token.enabled	認証トークンを有効にするかどうかを指定します。値が false に設定されていることを確認します。
xpack.security.authc.api_key.enabled	API キートークンを有効にするかどうかを指定します。値が false に設定されていることを確認します。

- Elasticsearch バージョン 7.17.0 設定ディレクトリ内の roles.yml ファイルを開き、次のコードを入力してユーザーロールを定義します。

```

user:
  cluster: [ 'monitoring_user', 'transport_client', 'kibana_user', 'snapshot_user' ]
  indices:
    - names: [ '*' ]
      privileges: [ 'all' ]

```

- Elasticsearch バージョン 7.17.0 設定ディレクトリ内の elasticsearch.yml ファイルを開き、次のコードを入力します。

```

xpack.security.authc:
  anonymous:
    username:
    roles:
    authz_exception:

```

- Elasticsearch を起動します。Elasticsearch の起動方法の詳細については、[「Elasticsearch の起動」](#) (ページ 150)のトピックを参照してください。

9. Elasticsearch 健全性 API を実行して、Elasticsearch サーバーで HTTPS プロトコルが確実に使用されるようにします。Elasticsearch API のテスト方法の詳細については、Elasticsearch のドキュメントを参照してください。
10. 既存の Elasticsearch バージョン 6.2.3 または 6.8.6 の環境、MDM Hub Server サービス、およびクレンジングサーバーサービスを停止します。
11. 次のディレクトリに移動します。  
`<elasticsearch 7.17.0 installation directory>/config`
12. `elasticsearch.yml` ファイルに、次のプロパティを設定します。

プロパティ	説明
<code>path.data</code>	<p>既存のインデックスを移行する場合は、既存のバージョンの Elasticsearch データディレクトリを指します。複数のデータディレクトリを設定できます。複数のデータディレクトリの設定に関する詳細については、Elasticsearch のドキュメントを参照してください。</p> <p><b>注:</b> Elasticsearch バージョン 7.17.0 を移行するには、既存の Elasticsearch バージョン 6.2.3 または 6.8.6 の設定プロパティに <code>cluster.name</code> と <code>node.name</code> プロパティが含まれている必要があります。これらのプロパティが存在しない場合は、既存のインデックスを移行できません。</p>
<code>path.logs</code>	ログファイルのパスを指定します。
<code>http.port</code>	HTTP 要求のポートを指定します。既存の Elasticsearch バージョンの環境で設定したポート番号を入力します。
<code>transport.tcp.port</code>	TCP バインドポートを指定します。既存の Elasticsearch バージョンの環境で設定した TCP バインドポート番号を入力します。

13. Elasticsearch バージョン 7.17.0 環境を起動します。
14. Elasticsearch API をテストして、インデックスが開いていること、およびエラーが表示されないことを確認します。Elasticsearch API のテスト方法の詳細については、Elasticsearch のドキュメントを参照してください。
15. Hub サーバー、プロセスサーバー、および Elasticsearch サービスを起動します。  
 既存の MDM 環境に Elasticsearch 7.17.0 が設定されました。

## 第 10 章

# 階層のアップグレード

この章では、以下の項目について説明します。

- [階層のアップグレードの概要, 160 ページ](#)
- [階層リレーションおよびネットワークリレーションを理解する, 161 ページ](#)
- [階層マネージャリレーションのコピーおよび階層リレーションの作成, 163 ページ](#)
- [階層マネージャリレーションのコピーおよびネットワークリレーションの作成, 166 ページ](#)

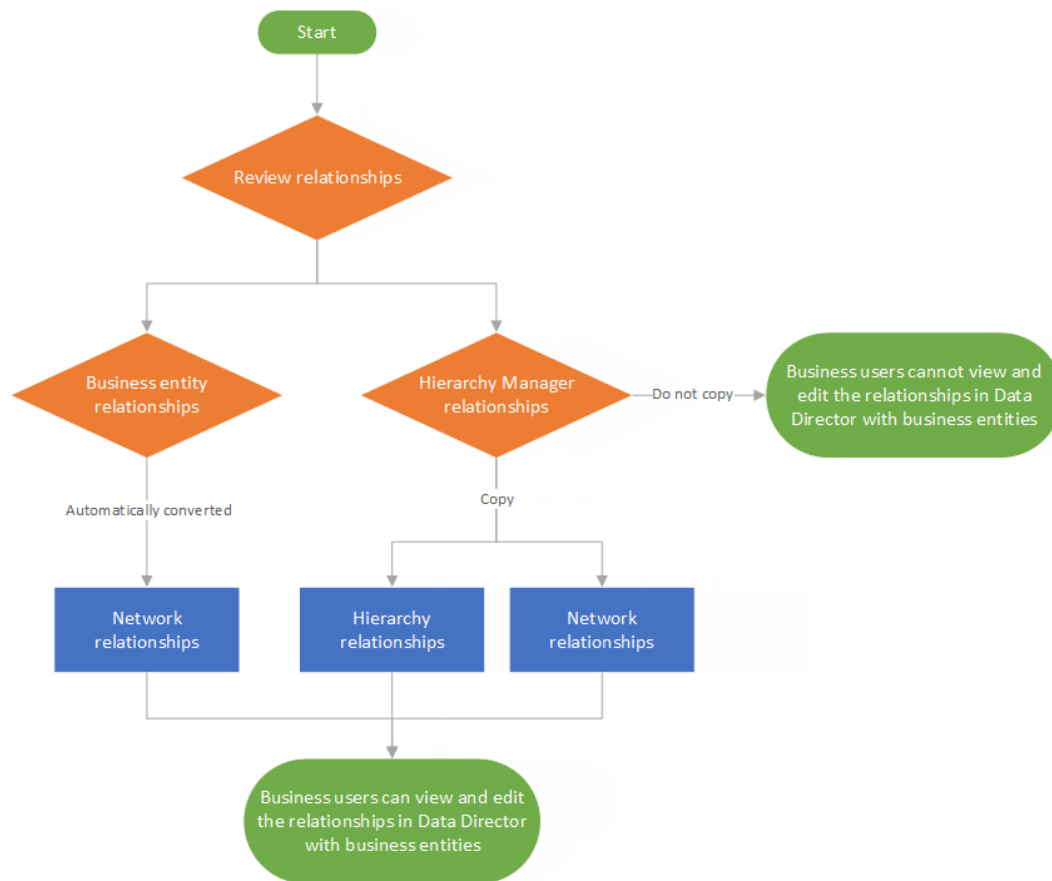
## 階層のアップグレードの概要

プロビジョニングツールを使用して、ビジネスエンティティ間の階層リレーションおよびネットワークリレーションを作成します。次に Data Director で、ビジネスユーザーは【階層】ビューおよび【ネットワーク】ビューを使用して、レコード間のリレーションを作成および管理できます。

10.4 より前のバージョンからアップグレードする場合、プロビジョニングツールにビジネスエンティティのリレーション、または階層マネージャに階層リレーションがある可能性があります。10.4 にアップグレードするとき、プロビジョニングツールはビジネスエンティティのリレーションをネットワークリレーションに自動的に変換します。

階層マネージャにリレーションがある場合は、そのリレーションを確認して、どのリレーションの使用を継続するか決定します。次に、要件に基づき、コピーして階層リレーションまたはネットワークリレーションとして作成するリレーションを決定します。プロビジョニングツールを使用して、階層マネージャのリレーションをコピーし、それらを階層リレーションまたはネットワークリレーションとして作成します。

手順を以下の図に示します。



## 階層リレーションおよびネットワークリレーションを理解する

プロビジョニングツールを使用して、階層マネージャのリレーションをコピーし、それらを階層リレーションまたはネットワークリレーションとして作成します。

ビジネスエンティティ間の階層リレーションを定義する場合、リレーションをコピーし、それらを階層リレーションとして作成します。関連ビジネスエンティティ間の接続を作成する場合、リレーションをコピーし、それらをネットワークリレーションとして作成します。

階層マネージャでリレーションをコピーする場合、作成する新しいリレーションは Hub コンソールの同じリレーションベースオブジェクトを参照します。使用するリレーションのコピーを完了した後で、リレーションベースオブジェクトをベースオブジェクトに戻す必要があります。リレーションベースオブジェクトをベースオブジェクトに戻すことで、オブジェクトから階層マネージャメタデータが削除され、プロビジョニングツールで階層およびネットワークリレーションを作成および更新できます。

## 階層関係

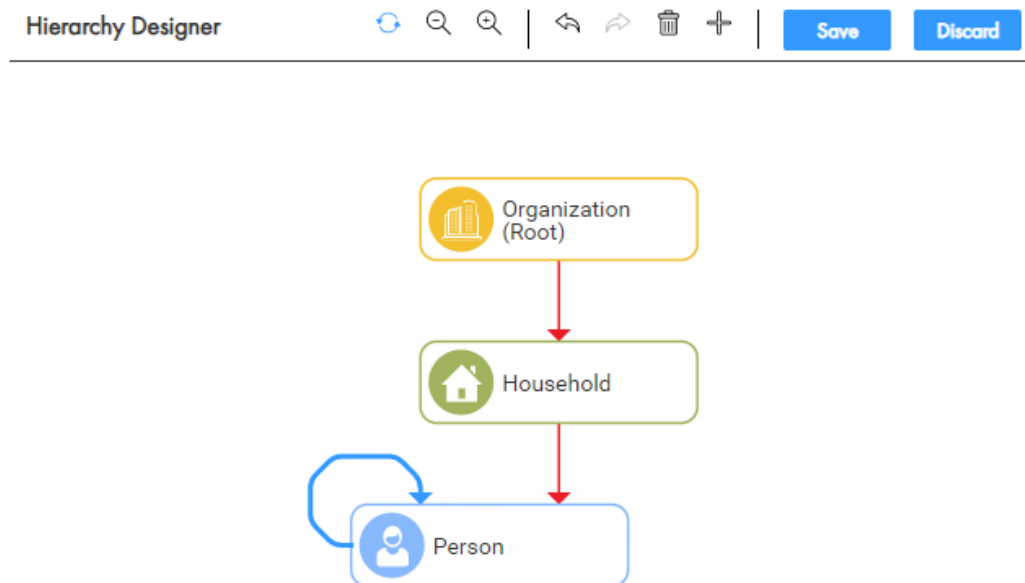
階層リレーションは、ビジネスエンティティ間の親子関係です。階層には、ビジネスエンティティ間の階層リレーションの集合が含まれます。複数の階層を作成して、組織にとって重要な階層リレーションを定義できます。

階層リレーションを作成する場合、以下のガイドラインを用いてください。

- ビジネスエンティティをルートビジネスエンティティとして指定する必要があります。
- ルートビジネスエンティティと階層内の各ビジネスエンティティ間の直接または間接のリレーションを定義する必要があります。
- 1つのビジネスエンティティから同じビジネスエンティティへのリレーションのループを作成できます。例えば、Person ビジネスエンティティから Person ビジネスエンティティへのリレーションを作成できます。

**注:** プロビジョニングツールはルートビジネスエンティティのリレーションのループをサポートしますが、Data Director は、ルートビジネスエンティティレベルでデータのループがあるレコードの階層を表示できません。例えば、プロビジョニングツールで、Organization ルートビジネスエンティティにリレーションのループを作成します。Data Director で、Informatica レコードに Informatica レコードとのリレーションがある場合、Data Director は、Informatica レコードの階層を表示できません。

次の図に、プロビジョニングツールのサンプル階層を示します。



## ネットワークリレーション

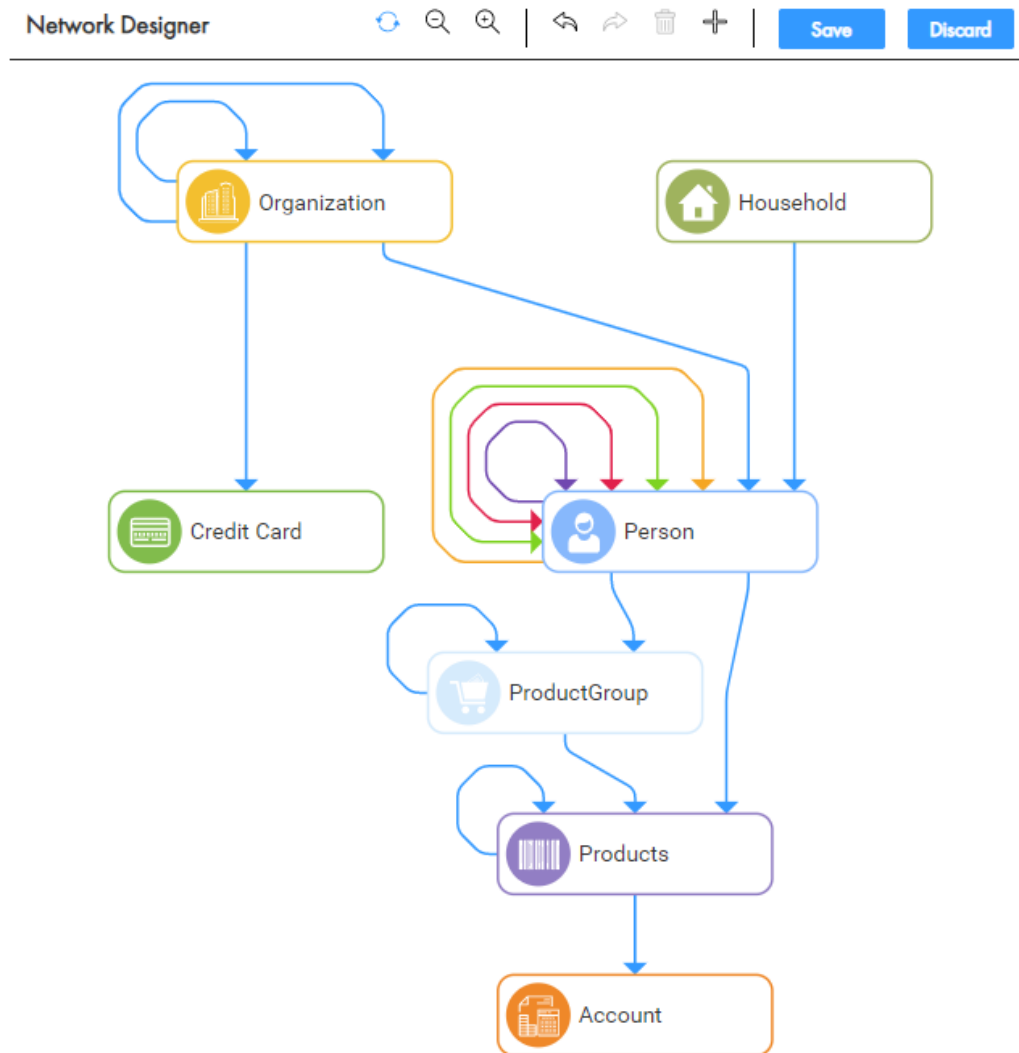
ネットワークリレーションは、関連ビジネスエンティティ間の接続です。作成した各ネットワークリレーションは、ネットワークの一部になります。

ネットワークはビジネスエンティティ間のネットワークリレーションの集合です。ネットワークリレーションを追加するのは1つのネットワークのみです。

ネットワークリレーションを作成する場合、以下のガイドラインを用いてください。

- ルートビジネスエンティティを指定する必要はありません。
- 1つのビジネスエンティティから同じビジネスエンティティへのリレーションのループを作成できます。

次の図に、プロビジョニングツールのサンプルネットワークを示します。



## 階層マネージャリレーションのコピーおよび階層リレーションの作成

プロビジョニングツールを使用して、階層マネージャのリレーションをコピーし、階層を作成します。階層マネージャの追加のリレーションをコピーすることで、複数の階層を作成できます。

リレーションのコピーにより階層を作成する場合、プロビジョニングツールは対応するビジネスエンティティを検索し、ルートビジネスエンティティノードとして設定しようとしています。選択したルート HM エンティティに基づきルートビジネスエンティティを検索しようとしています。対応するビジネスエンティティが見つからない場合、ビジネスエンティティを作成します。次にプロビジョニングツールはコピーされたリレーションに基づき残りの階層モデルを作成します。このプロセスで作成された階層および階層リレーションを確認する必要があります。

階層マネージャのリレーションをコピーしてから階層を作成するには、次のタスクを実行します。

1. 階層マネージャのリレーションをコピーして階層を作成する。
2. 階層 REST API をサポートするように Hub サーバーを構成する。
3. 階層へのアクセスの設定。
4. リレーションベースオブジェクトをベースオブジェクトに戻す。

## 階層マネージャのリレーションのコピーおよび階層の作成

階層マネージャのリレーションをコピーし、階層リレーションを含む階層を作成します。階層マネージャから追加の階層リレーションをコピーすることで、追加の階層を作成できます。

**重要:** リレーションのコピーにより階層を作成する場合、プロビジョニングツールは対応するビジネスエンティティを検索し、ルートビジネスエンティティノードとして設定しようとします。選択したルート HM エンティティに基づきルートビジネスエンティティを検索しようとします。対応するビジネスエンティティが見つからない場合、ビジネスエンティティを作成します。次にプロビジョニングツールはコピーされたリレーションに基づき残りの階層モデルを作成します。リレーションのコピーにより作成された階層および階層リレーションを確認する必要があります。

1. **【ビジネスエンティティ】 > 【モデリング】** をクリックしてから、**【階層】** を選択します。
2. **【作成】 > 【階層マネージャからコピーされた階層】** をクリックします。  
**【階層マネージャからコピーされた階層の作成】** ウィンドウが表示されます。
3. **【既存の階層の選択】** カラムで、コピーする階層を選択します。
4. **【ルートエンティティを選択】** カラムで、次のルートエンティティ設定を構成します。
  - a. **【ルート HM エンティティ】** フィールドで、ルートノードとして使用する HM エンティティを選択します。  
プロビジョニングツールはルート HM エンティティを使用して階層のルートビジネスエンティティを決定します。
  - b. **【新しい階層名】** フィールドに、階層の名前を入力します。
5. **【新しい階層の作成】** をクリックします。  
階層デザインが開きます。
6. ビジネスエンティティとリレーションを確認して、必要に応じて問題を解決します。
  - ルートビジネスエンティティにリレーションのループがあった場合、リレーションを削除するか階層を編集してそのリレーションのループを削除し、新しいビジネスエンティティをルートビジネスエンティティとして指定します。  
**警告:** プロビジョニングツールはルートビジネスエンティティのリレーションのループをサポートしますが、Data Director は、ルートビジネスエンティティレベルでデータのループがあるレコードの階層を表示できません。階層を作成するためのガイドラインの詳細については、[「階層関係」 \(ページ 162\)](#) を参照してください。
  - リレーションにエラーがある場合、そのリレーションを選択し、リレーションのプロパティを確認します。エラーのあるリレーションが赤い線で表示されます。
  - エラーの発生したビジネスエンティティノードがある場合、そのノードを選択し、ビジネスエンティティを確認します。エラーの発生したビジネスエンティティは赤い線で表示されます。
7. **【保存】** をクリックします。

## 階層用の Hub サーバーの設定

Hub コンソールの Hub サーバーツールおよび `cmxserver.properties` ファイルを使用して、階層 REST API をサポートする Hub サーバーのプロパティを設定します。

階層 REST API を使用して、階層を処理できます。階層 REST API については、*Multidomain MDM* ビジネスエンティティのサービスガイドを参照してください。

1. テキストエディタを使用して、<MDM Hub インストールディレクトリ>\hub\server\resources\cmxserver.properties にある `cmxserver.properties` ファイルを開きます。
2. 次のプロパティを手動で追加および設定します。

`cmx.server.hierarchy.max-search-depth`

階層 REST API を使用して階層パスを検索するときに検索される最大深度。デフォルト値は 100 です。

`cmx.server.hierarchy.max-search-width`

階層 REST API を使用してエクスポートするときに含める検索される階層の最大幅。デフォルト値は 1000000 です。

## リレーションベースオブジェクトをベースオブジェクトに戻す

リレーションベースオブジェクトをベースオブジェクトに戻し、リレーションオブジェクトから階層マネージャのメタデータを削除します。リレーションオブジェクトは、ベースオブジェクトとして残されますが、階層マネージャではこのベースオブジェクトは表示されません。

戻すリレーションタイプカラムがルックアップ用のステー징テーブルにある場合、リレーションベースオブジェクトを戻す前に、このステー징テーブルカラムを空にする必要があります。

階層マネージャリレーションをアップグレードする場合は、リレーションベースオブジェクトを戻す前に、階層マネージャのリレーションをプロビジョニングツールにコピーしてください。

1. 階層ツールで、書き込みロックを取得します。
2. リレーションベースオブジェクトで右クリックして、**【エンティティ/リレーションオブジェクトを BO に戻す】** を選択します。
3. **【エンティティ/リレーションオブジェクトを戻す】** ダイアログボックスで **【OK】** をクリックします。  
エンティティが戻ると、ダイアログボックスが表示されます。

## 階層へのアクセスの設定

階層をセキュアリソースとして設定し、階層に MDM ロールアクセスを付与します。

1. Hub コンソールを開き、管理するデータベースを選択します。
2. **【書き込みロック】** > **【ロックの取得】** の順にクリックします。
3. 階層をセキュアとして設定します。
  - a. **【ワークベンチ】** パネルで、**【セキュリティアクセスマネージャ】** の下の **【セキュアリソース】** をクリックします。  
**【セキュアリソース】** パネルが表示されます。
  - b. **【リソース】** タブで、**【ビジネスエンティティ階層】** ノードを展開します。
  - c. セキュアとして設定するビジネスエンティティ階層を選択してから、**【リソースのステータスをセキュアに変更】** をクリックします。
  - d. **【保存】** アイコンをクリックします。

4. 階層にMDMロールアクセスを付与します。
  - a. **【ワークベンチ】** パネルで、**【セキュリティアクセスマネージャ】** の下の **【ロール】** をクリックします。  
**【ロール】** パネルが表示されます。
  - b. **【DataSteward】** を選択します。
  - c. **【リソース特権】** タブで、**【ビジネスエンティティ階層】** ノードを展開します。
  - d. 各階層に対し、**【読み取り】**、**【作成】** および **【更新】** チェックボックスを選択します。
  - e. Manager および SrManager ロールについて、**c** から **d** の手順を繰り返します。
  - f. **【保存】** アイコンをクリックします。
5. リポジトリを検証します。
  - a. **【ワークベンチ】** パネルで、**【設定】** の下の **【リポジトリマネージャ】** をクリックします。
  - b. **【マスタデータベースに接続】** をクリックします。
  - c. **【検証】** タブで、階層アクセスを設定するデータベースを選択します。
  - d. **【検証】** アイコンをクリックします。
  - e. **【OK】** をクリックします。

## 階層マネージャリレーションのコピーおよびネットワークリレーションの作成

プロビジョニングツールを使用して、階層マネージャのリレーションをコピーし、それらをネットワークリレーションとして作成します。

階層マネージャのリレーションをコピーし、ネットワークリレーションとしての作成を追加するには、次のタスクを実行します。

1. 階層マネージャのリレーションをコピーしてネットワークリレーションを作成する。
2. リレーションベースオブジェクトをベースオブジェクトに戻す。

## 階層マネージャでリレーションをコピーすることによるネットワークリレーションの追加

階層マネージャでリレーションをコピーし、ネットワークリレーションとしてリレーションを追加します。階層マネージャでその他の階層のリレーションをコピーすることで、追加のネットワークリレーションを追加できます。

1. **【ビジネスエンティティ】** > **【モデリング】** をクリックしてから、**【ネットワーク】** を選択します。
2. **【作成】** > **【階層マネージャからコピーされたリレーション】** をクリックします。  
**【階層マネージャからコピーされたリレーションの追加】** ウィンドウが表示されます。
3. **【既存の階層を選択】** カラムで、リレーションのコピー元の階層を選択します。
4. **【リレーションを選択】** カラムで、コピーするリレーションを選択します。
5. **【選択したものをネットワークに追加】** をクリックします。  
ネットワークデザイナーが開きます。

6. ネットワークのリレーションを確認します。リレーションにエラーがある場合、そのリレーションを選択し、リレーションのプロパティを確認します。  
エラーのあるリレーションが赤い線で表示されます。
7. **【保存】** をクリックします。

## リレーションベースオブジェクトをベースオブジェクトに戻す

リレーションベースオブジェクトをベースオブジェクトに戻し、リレーションオブジェクトから階層マネージャのメタデータを削除します。リレーションオブジェクトは、ベースオブジェクトとして残されますが、階層マネージャではこのベースオブジェクトは表示されません。

戻すリレーションタイプカラムがルックアップ用のステー징テーブルにある場合、リレーションベースオブジェクトを戻す前に、このステー징テーブルカラムを空にする必要があります。

階層マネージャリレーションをアップグレードする場合は、リレーションベースオブジェクトを戻す前に、階層マネージャのリレーションをプロビジョニングツールにコピーしてください。

1. 階層ツールで、書き込みロックを取得します。
2. リレーションベースオブジェクトで右クリックして、**【エンティティ/リレーションオブジェクトを BO に戻す】** を選択します。
3. **【エンティティ/リレーションオブジェクトを戻す】** ダイアログボックスで **【OK】** をクリックします。  
エンティティが戻ると、ダイアログボックスが表示されます。

## 第 11 章

# アプリケーションサーバーに対する ActiveVOS のインストール後のタスク

- [アプリケーションサーバーに対する ActiveVOS のインストール後のタスク, 168 ページ](#)

## アプリケーションサーバーに対する ActiveVOS のインストール後のタスク

クリーンアップグレードまたはインプレースアップグレードを実行するかどうかにかかわらず、環境が適切に構成されていることを確認するため、アプリケーションサーバーに対して ActiveVOS のインストール後のタスクを実行します。

アプリケーションサーバーに対する ActiveVOS のインストール後のタスクについては、ご使用の環境に対応した *Multidomain MDM* のインストールガイドの「アプリケーションサーバーに対する ActiveVOS のインストール後のタスク」の章を参照してください。

## 第 12 章

# ビジネスエンティティアダプタ用の ActiveVOS アップグレード後タスク

この章では、以下の項目について説明します。

- [ビジネスエンティティアダプタに対する ActiveVOS のアップグレード後のタスク, 169 ページ](#)
- [ビジネスエンティティワークフローアダプタの ActiveVOS URN の設定, 170 ページ](#)
- [ActiveVOS プロトコルの HTTPS への設定, 170 ページ](#)
- [ビジネスエンティティ用にカスタマイズしたワークフローの更新, 171 ページ](#)
- [ActiveVOS 用の MDM Identity Service の設定, 174 ページ](#)
- [カスタム BeMDMWorkflow プロジェクト（インプレースアップグレード）, 175 ページ](#)
- [マージ解除ワークフローおよびマージワークフローのトリガの設定（インプレースアップグレード）, 175 ページ](#)
- [エンティティ 360 フレームワークタスクマネージャの追加, 176 ページ](#)

## ビジネスエンティティアダプタに対する ActiveVOS のアップグレード後のタスク

クリーンアップグレードまたはインプレースアップグレードを実行するかどうかにかかわらず、ビジネスエンティティに基づいて ActiveVOS ワークフローアダプタを使用する場合は、環境が適切に構成されていることを確認するため、ビジネスエンティティアダプタに対して ActiveVOS のアップグレード後のタスクを実行します。

# ビジネスエンティティワークフローアダプタの ActiveVOS URN の設定

ActiveVOS サーバーには、内部的に使用する 2 つの定義済みの Uniform Resource Name (URN) があります。URN マッピングの URL は、ActiveVOS サーバーが動作しているホスト名とポート番号に変更する必要があります。

1. ActiveVOS コンソールを起動します。ブラウザで、正しいホスト名とポート番号に置き換えて、以下の URL を入力します。  
暗号化接続: `https://[host]:[port]/activevos`  
非暗号化接続: `http://[host]:[port]/activevos`
2. ActiveVOS コンソールのホームページで、**[Administration] > [Configure Server] > [URN Mappings]** をクリックします。
3. 以下の URN について、ActiveVOS サーバーのホスト名とポート番号を反映するようにパスを更新します。

URN	URL パス
ae:internal-reporting	暗号化接続: <code>https://[host]:[port]/activevos/internalreports</code> 非暗号化接続: <code>http://[host]:[port]/activevos/internalreports</code>
ae:task-inbox	暗号化接続: <code>https://[host]:[port]/activevos-central/avc</code> 非暗号化接続: <code>http://[host]:[port]/activevos-central/avc</code>

4. **urn:mdm:service** が MDM Hub サーバーのホスト名とポート番号にマッピングされていることを確認します。  
暗号化接続: `https://[host]:[port]/cmx/services/BEServices`  
非暗号化接続: `http://[host]:[port]/cmx/services/BEServices`

## ActiveVOS プロトコルの HTTPS への設定

ActiveVOS と MDM Hub の間の安全な通信を有効にするには、Hub コンソールの Workflow Manager で HTTPS にプロトコルを設定します。

最初に、HTTPS 通信用のアプリケーションサーバーを設定する必要があります。

1. Hub コンソールを起動します。
2. 書き込みロックを取得します。
3. 設定ワークベンチの **[Workflow Manager]** をクリックします。
4. Workflow Manager で **[ワークフローエンジン]** タブをクリックします。
5. ActiveVOS ワークフローエンジンを選択し、**[編集]** ボタンをクリックします。
6. **[ワークフローの編集]** ダイアログボックスで、プロトコルを HTTPS に設定します。
7. WebLogic 環境では、**[ワークフローの編集]** ダイアログボックスで、abAdmin ロールに属するユーザーのユーザー名およびパスワードを入力します。

# ビジネスエンティティ用にカスタマイズしたワークフローの更新

ビジネスエンティティアダプタを操作するようにカスタマイズしたワークフローがある場合、変更内容のリストを確認し、必要に応じてカスタマイズしたワークフローに更新を加えます。

次の表に、リリース別のデフォルトワークフローへの変更内容の一覧を示します。

リリース	変更	手順
10 HotFix 1	パスワード暗号化を有効にします。	各ワークフローについて、sif:encrypted を true に設定します。
10 HotFix 1	タスクインボックスのプレゼンテーションパラメータを更新します。	<a href="#">「ビジネスエンティティ用のワークフロー内のプレゼンテーションパラメータの更新」</a> (ページ 171)
10.3	ファイル添付機能を有効にします。	<a href="#">「ビジネスエンティティのためのワークフロー内でのファイル添付の有効化」</a> (ページ 173)

## ビジネスエンティティ用のワークフロー内のプレゼンテーションパラメータの更新

ビジネスエンティティに基づいてワークフローをカスタマイズした場合は、.bpel ファイルを開き、プレゼンテーションパラメータを次のパラメータと比較します。必要に応じて各タスクのプレゼンテーションパラメータを追加または更新します。

式にはパラメータの絶対参照を含めることができますが、カスタマイズしたワークフローでのベストプラクティスは、タスクの getInput() メソッドを参照することです。絶対参照を getInput() メソッドの呼び出しを使用したプレゼンテーションパラメータに置換することを検討してください。例えば、\$ProcessTaskRequest/mdmavxsd:INFATask/ を htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ に置換します。次の表の式は、入力メソッドを呼び出します。

**ヒント:** 名前空間のプレフィックスとして mdmavxsd を使用します。カスタムワークフローを更新した後で問題が発生した場合は、mdmavxsd が urn:informatica.mdm.av.xsd のプレフィックスとして定義されていることを確認してください。

次の表にプレゼンテーションパラメータと式を示します。

パラメータ	式
hubUsername	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:hubUsername/text()
hubPassword	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:hubPassword/text()
securityPayload	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:securityPayload/text()
orsId	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:orsId/text()

パラメータ	式
taskTypeName	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:name/text()
taskTypeDisplayName	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:displayName/text()
taskTypeDescription	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:description/text()
pendingBVT	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:pendingBVT
taskTypeDataUpdateType	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:dataUpdateType/text()
taskTypeDisplayType	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:displayType/text()
defaultApproval	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:defaultApproval
taskDataTaskId	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:taskId/text()
taskDataOwnerUID	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:ownerUID/text()
taskDataGroups	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:groups/mdmavxsd:groups/text()
dueDate	let \$in := htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:dueDate/text() let \$out :=
status	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:status/text()
taskDataPriority	length(\$out > 0))), string-length(\$out) + (string-length(\$in)) * xsd:int((string- length(\$out) = 0)))
taskDataSubjectAreaUID	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:subjectAreaUID/text()
taskDataTitle	let \$in := htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:title/text() let \$out :=
taskDataComments	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:comments/text()
taskDataInteractionId	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:interactionId/text()
taskDataCreator	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:creator/text()

パラメータ	式
createDate	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:createDate
taskDataUpdatedBy	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:updatedBy/text()
lastUpdateDate	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:lastUpdateDate
workflowVersion	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:workflowVersion/text()
beRowId	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:taskRecords/ mdmavxsd:INFARecordKey[1]/mdmavxsd:rowId/text()
bePkeySrcObject	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:taskRecords/ mdmavxsd:INFARecordKey[1]/mdmavxsd:pkeySrcObject/text()
beSystem	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:taskRecords/ mdmavxsd:INFARecordKey[1]/mdmavxsd:system/text()
beRowidXref	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:taskRecords/ mdmavxsd:INFARecordKey[1]/mdmavxsd:rowidXref/text()
beTableUID	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskData/mdmavxsd:taskRecords/ mdmavxsd:INFARecordKey[1]/mdmavxsd:tableUID/text()
taskTypeCreationType	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/ mdmavxsd:taskType/mdmavxsd:creationType/text()

.bpel ファイルの更新の詳細については、ActiveVOS のマニュアルを参照してください。

## ビジネスエンティティのためのワークフロー内でのファイル添付の有効化

バージョン 10.3 以降、カスタマイズしたワークフローを更新し、ユーザーが自分のタスクにファイルを添付できるようにしました。カスタマイズしたワークフロー内でこの機能を有効にするには、データを初期化するプロセスを更新し、すべてのユーザープロセスを更新します。

**注:** 一部のタスクに対して添付ファイルを追加する機能を制限する場合、プロビジョニングツールでタスク設定を設定できます。タスク設定の更新の詳細については、『*Multidomain MDM のプロビジョニングツールガイド*』を参照してください。

1. ActiveVOS デザイナーで、カスタマイズしたワークフローの.bpel ファイルを開きます。  
ワークフローがキャンバスに表示されます。

2. データを初期化するプロセスを更新します。
  - a. **【データの初期化】** プロセスを選択します。  
**【操作】** ダイアログボックスが表示されます。
  - b. **【新しいスクリプト】** をクリックします。  
**【スクリプトビルダ】** ダイアログボックスが表示されます。
  - c. **【スクリプト】** ボックスで、次の文を追加します。  
`abx:copyAllAttachments('StartRequest', '_peopleActivityAttachments')`  
`[StartRequest]` を、ワークフロー内でデータの初期化に使用する変数名で置き換えます。
  - d. **【OK】** をクリックします。  
**【スクリプトビルダ】** ダイアログボックスが閉じ、スクリプトが **【操作】** リストに表示されます。
  - e. **【OK】** をクリックします。  
**【操作】** ダイアログボックスが閉じます。
3. すべてのユーザープロセスを更新します。
  - a. ユーザープロセスを選択します。
  - b. **【プロパティ】** タブをクリックします。
  - c. **【すべて】** タブをクリックします。
  - d. **【添付ファイルのプロパゲート】** を展開します。
  - e. **【送信元プロセス】** を **【すべて】** に設定します。
  - f. **【送信先プロセス】** を **【newOnly】** に設定します。
4. .bpel ファイルを保存します。

## ActiveVOS 用の MDM Identity Service の設定

埋め込み ActiveVOS を使用している場合は、MDM Identity Service を使用するように ActiveVOS を設定してください。ActiveVOS 用の MDM Identity Service を設定するには、ActiveVOS コンソールを使用して、MDM Hub ワークフローエンジンユーザーのパスワードを Identity Service のパスワードに設定します。

1. ActiveVOS コンソールで、**【管理】** > **【サービスの設定】** > **【Identity Service】** を選択します。
2. **【プロバイダ設定】** セクションで、**【有効】** チェックボックスを有効にして、**【プロバイダタイプ】** から **【MDM】** を選択します。
3. **【接続】** タブで、ユーザー名を admin として MDM Hub ユーザーのパスワードを入力します。  
**注:** 後で管理者ユーザーのパスワードを変更する場合、ActiveVOS ID サービス設定で新しいパスワードを入力する必要があります。
4. **【更新】** をクリックします。
5. ActiveVOS が admin ユーザーとして MDM Hub にログインできるかどうか、**【テストのユーザー】** として指定したユーザーのロールのリストを ActiveVOS が取得できるかどうかをテストします。
  - a. **【テスト】** タブを選択します。
  - b. **【テストのユーザー】** フィールドに、ロールに割り当てられた MDM Hub ユーザーを入力します。

- c. **【テスト設定】** をクリックします。

**注:** オペレーショナル参照ストアが設定されていない場合、このテストは失敗し、テストのユーザーはロールに属さなくなる、またはそのロール名にスペースが格納されます。

## カスタム BeMDMWorkflow プロジェクト（インプレースアップグレード）

BeMDMWorkflow プロジェクトのワークフローをカスタマイズした場合は、プロジェクトを更新してから再デプロイすることを確認します。

1. Informatica ActiveVOS Designer でプロジェクトエクスプローラを使用して、**BeCommonMDM > wsdl > cs.wsdl** に移動します。
2. **cs.wsdl** を右クリックして、**【アプリケーションから開く】 > 【テキストエディタ】** を選択します。
3. テキストエディタで次のコードに移動します。

```
<xsd:complexType name="TaskFilter">
  <xsd:sequence>
```
4. 次のコードを追加します。

```
<xsd:element minOccurs="0" name="overdueOnly" type="xsd:boolean" />
```
5. Informatica ActiveVOS Designer で、エクスポートする BeCommonMDM プロジェクトを開きます。
6. **【ファイル】 > 【エクスポート】** をクリックします。  
**【エクスポート】** ダイアログボックスが開きます。
7. **【Orchestration】** の下で、**【コントリビューション-ビジネスプロセスアーカイブ】** を選択します。**【次へ】** をクリックします。
8. **【デプロイメント URL】** フィールド（**【サーバーデプロイメントオプション】** の下）に、ActiveVOS インスタンスの URL を入力します。**【完了】** をクリックします。
9. **【デプロイメントの完了】** ダイアログボックスで、**【OK】** をクリックします。

## マージ解除ワークフローおよびマージワークフローのトリガの設定（インプレースアップグレード）

Multidomain MDM バージョン 10.2 で導入された、マージ解除ワークフローおよびマージワークフローのトリガを設定する必要があります。マージ解除ワークフローおよびマージワークフローのトリガを設定するには、プロビジョニングツールを使用します。

詳細については、『*Multidomain MDM のプロビジョニングツールガイド*』を参照してください。

# エンティティ 360 フレームワークタスクマネージャ の追加

ビジネスエンティティの ActiveVOS ワークフローアダプタを使用する場合、エンティティ 360 フレームワークタスクマネージャとエンティティ 360 フレームワークタスクインボックスを使用します。

Informatica Data Director 設定を更新して、従来のタスクインボックスを置換します。エンティティ 360 フレームワークタスクインボックスを【ホーム】ページに追加できます。Informatica Data Director ユーザーインターフェースの設計に関する詳細については、『*Multidomain MDM のプロビジョニングツールガイド*』を参照してください。

## 第 13 章

# サブジェクト領域アダプタ用の ActiveVOS アップグレード後タスク

この章では、以下の項目について説明します。

- [サブジェクト領域アダプタに対する ActiveVOS のアップグレード後のタスク, 177 ページ](#)
- [ActiveVOS URN の更新, 178 ページ](#)
- [ActiveVOS の信頼されたユーザーの確認, 178 ページ](#)
- [サブジェクト領域に基づく ActiveVOS ワークフローの Informatica Data Director タスク設定の更新, 179 ページ](#)
- [サブジェクト領域用にカスタマイズしたワークフローの更新, 181 ページ](#)
- [サブジェクト領域に基づく ActiveVOS ワークフローの再デプロイ, 183 ページ](#)
- [ビジネスエンティティおよびビジネスエンティティサービス構成ファイルの生成, 184 ページ](#)

## サブジェクト領域アダプタに対する ActiveVOS のアップグレード後のタスク

クリーンアップグレードまたはインプレースアップグレードを実行するかどうかにかかわらずサブジェクト領域に基づいて ActiveVOS ワークフローアダプタを使用する場合は、環境が適切に構成されていることを確認するため、サブジェクト領域アダプタに対して ActiveVOS のアップグレード後のタスクを実行します。

# ActiveVOS URN の更新

HTTP Secure (HTTPS) プロトコルを使用して MDM Hub と ActiveVOS の間の通信を保護するには、URN パスの URL を http から https に変更します。

1. ActiveVOS コンソールを起動します。ブラウザで、正しいホスト名とポート番号に置き換えて、以下の URL を入力します。

JBoss バージョン 7.2 または 7.1	URL パス
保護された接続	https://<host>:<port>/activevos
保護されていない接続	http://<host>:<port>/activevos

2. ActiveVOS コンソールのホームページで、**[Administration] > [Configure Server] > [URN Mappings]** をクリックします。
3. 以下の URN について、ActiveVOS サーバーのホスト名とポート番号を反映するようにパスを更新します。

URN	URL パス
ae:internal-reporting	保護されている接続。https://<host>:<port>/activevos/internalreports 保護されていない接続。http://<host>:<port>/activevos/internalreports
ae:task-inbox	保護されている接続。https://<host>:<port>/activevos-central/avc 保護されていない接続。http://<host>:<port>/activevos-central/avc

4. **MDMHost:InfMDM** が MDM Hub サーバーのホスト名とポート番号にマップされていることを確認します。

保護されている接続。https://<host>:<port>/cmx/services/SifService

保護されていない接続。http://<host>:<port>/cmx/services/SifService

## ActiveVOS の信頼されたユーザーの確認

Hub コンソールで、ActiveVOS ワークフローエンジン設定に信頼されたユーザーが指定されていることを確認します。

1. Hub コンソールの設定ワークベンチで **[Workflow Manager]** をクリックします。
2. **[ワークフローエンジン]** タブを選択します。
3. 書き込みロックを取得します。
4. **[ActiveVOS]** を選択し、**[編集]** ボタンをクリックします。
5. **[ワークフローの編集]** ダイアログボックスで、信頼されたユーザーのユーザー名とパスワードを入力します。
6. **[OK]** をクリックします。

# サブジェクト領域に基づく ActiveVOS ワークフローの Informatica Data Director タスク設定の更新

タスクマネージャでサブジェクト領域ベースの ActiveVOS ワークフローアダプタを使用するには、Informatica Data Director の構成ファイルを更新する必要があります。サブジェクト領域に基づいた ActiveVOS ワークフローを使用している場合、ビジネスエンティティに基づいた ActiveVOS ワークフローに移行することはできません。

Informatica Data Director の構成ファイルで以下のタスクパラメータを設定できます。

taskType

タスクタイプを示します。

taskTypeID

プロセス名です。

name

taskType の名前です。この名前は、ActiveVOS ワークフロー設定内のタスクの名前と同じにする必要があります。

## サブジェクト領域ベースの ActiveVOS アダプタの IDD 設定の更新

ビジネスエンティティベースのタスクマネージャでサブジェクト領域ベースの ActiveVOS ワークフローアダプタを使用するには、Data Director 設定ファイルを更新します。Data Director 設定ファイルを更新しないと、タスクの作成にタスクマネージャを使用できません。

次のコードサンプルは、サブジェクト領域ベースの ActiveVOS タスクを Multidomain MDM で提供されるワークフローの Data Director 設定ファイルに設定する方法を示します。

```
<tasks includeUnassignedTasks="true">
<!-- Task Definitions -->
<taskType taskTypeId="IDDMergeTask" name="AVOSMerge" displayName="Merge" creationType="MERGE"
displayName="MERGE">
  <description>Merge two records together.</description>
</taskType>

<taskType taskTypeId="IDDUntermergeTask" name="AVOSUnmerge" displayName="Unmerge" creationType="UNMERGE"
displayName="UNMERGE">
  <description>Unmerge an XREF record from a Base Object record.
  </description>
</taskType>

<taskType taskTypeId="IDDOneStepApprovalTask" name="AVOSFinalReview" displayName="Final
review" creationType="NONE" pendingBVT="true">
  <description>Update a record and require the user to go through an approval process before completing the
task.
  </description>
</taskType>

<taskType name="Notification" displayName="Notification" creationType="NONE" displayName="NORMAL">
  <description>Notification step in the workflow</description>
</taskType>

<taskType taskTypeId="IDDTwoStepApprovalTask" name="AVOSReviewNoApprove"
displayName="NORMAL" displayName="Review no approve" creationType="NONE"
defaultApproval="true" pendingBVT="true">
  <description>Update a record and require the user to go through an approval process
before completing the task.
  </description>
</taskType>

<taskType taskTypeId="IDDUntermergeTask" name="Update" displayName="NORMAL" displayName="Update"
```

```

creationType="CREATE" pendingBVT="true">
  <description>Update a record and do not require the user to go through an approval
    process before completing the task. The approval step is optional.
  </description>
</taskType>

</tasks>

```

## サブジェクト領域ワークフローアダプタのタスクトリガの設定

タスクマネージャでサブジェクト領域に基づいた ActiveVOS ワークフローを使用するには、タスクトリガを設定する必要があります。タスクトリガを設定しない場合は、タスクがタスクマネージャに表示されません。

トリガを設定するには、プロビジョニングツールを使用して [詳細設定] ページからタスク構成ファイルを編集します。詳細については、『*Multidomain MDM のプロビジョニングツールガイド*』を参照してください。

以下の startWorkflow 属性を設定すると、タスクトリガを設定できます。

### プロセス

ActiveVOS ワークフロープロセスの名前。

taskKind

プロセスに必要なユーザーインタフェースのタイプを定義します。REVIEW、MERGE、または UNMERGE を定義できます。taskKind は、ActiveVOS ワークフローエンジンによって返されます。

taskTemplate

使用するタスクテンプレートの名前。

firstTask Type

ワークフローの最初のタスク。オプション。このパラメータにより、タスク作成時にそのタスクを割り当てられます。

### 2 段階承認のコードサンプル

次のコードサンプルは、2 段階承認タスクのためのサブジェクト領域に基づいた ActiveVOS アダプタの startWorkflow 要素設定を示します。

```

<trigger name="DefaultApproval">
  <startWorkflow process="IDDTwoStepApprovalTask" taskKind="REVIEW" taskTemplate="DefaultApproval"
    firstTaskType="AVOSReviewNoApprove"/>
  <event name="CreateBE"/>
  <event name="UpdateBE"/>
  <role name="*/>
</trigger>

```

### 1 段階承認のコードサンプル

次のコードサンプルは、1 段階承認タスクのためのサブジェクト領域に基づいた ActiveVOS アダプタの startWorkflow 要素設定を示します。

```

  <trigger name="DefaultApproval">
    <startWorkflow process="IDDOneStepApprovalTask" taskKind="REVIEW" taskTemplate="DefaultApproval"
      firstTaskType="AVOSFinalReview"/>
    <event name="CreateBE"/>
    <event name="UpdateBE"/>
    <role name="*/>
  </trigger>

```

### 承認を伴う更新のコードサンプル

次のコードサンプルは、承認を伴う更新タスクのためのサブジェクト領域に基づいた ActiveVOS アダプタの startWorkflow 要素設定を示します。

```
<trigger name="DefaultApproval">
<startWorkflow process="IDDUpdateWithApprovalTask" taskKind="REVIEW" taskTemplate="DefaultApproval"
firstTaskType="Update"/>
<event name="CreateBE"/>
<event name="UpdateBE"/>
<role name="*/>
</trigger>
```

### マージのコードサンプル

次のコードサンプルは、マージタスクのためのサブジェクト領域に基づいた ActiveVOS アダプタの startWorkflow 要素設定を示します。

```
<trigger name="Matched">
<startWorkflow process="IDDMergeTask" taskKind="MERGE" taskTemplate="MergeTaskGenerator"
firstTaskType="AVOSMerge"/>
<event name="MatchedBE"/>
<role name="SYSTEM"/>
</trigger>
```

## サブジェクト領域用にカスタマイズしたワークフローの更新

サブジェクト領域アダプタを操作するようにカスタマイズしたワークフローがある場合、変更内容のリストを確認し、必要に応じて、カスタマイズしたワークフローに更新を加えます。

次の表に、リリース別のデフォルトワークフローへの変更内容の一覧を示します。

リリース	変更	手順
10 HotFix 1	パスワード暗号化を有効にします。	各ワークフローについて、sif:encrypted を true に設定します。
10 HotFix 1	タスクインボックスのプレゼンテーションパラメータを更新します。	<a href="#">「サブジェクト領域用のワークフロー内のプレゼンテーションパラメータの更新」 (ページ 181)</a>
10.3	ファイル添付機能を有効にします。	<a href="#">「サブジェクト領域のためのワークフロー内での添付の有効化」 (ページ 183)</a>

## サブジェクト領域用のワークフロー内のプレゼンテーションパラメータの更新

サブジェクト領域に基づいてワークフローをカスタマイズした場合は、.bpel ファイルを開き、プレゼンテーションパラメータを次のパラメータと比較します。必要に応じて各タスクのプレゼンテーションパラメータを追加または更新します。

式にはパラメータの絶対参照を含めることができますが、カスタマイズしたワークフローでのベストプラクティスは、タスクの getInput() メソッドを参照することです。絶対参照を getInput() メソッドの呼び出しを使用したプレゼンテーションパラメータに置換することを検討してください。例えば、

htd:getInput('processTaskRequest')/mdmavxsd:INFATask/を htd:getInput('processTaskRequest')/mdmavxsd:INFATask/に置換します。次の表の式は、入力メソッドを呼び出します。

**ヒント:** 名前空間のプレフィックスとして mdmavxsd を使用します。カスタムワークフローを更新した後で問題が発生した場合は、mdmavxsd が urn:informatica.mdm.av.xsd のプレフィックスとして定義されていることを確認してください。

次の表にプレゼンテーションパラメータと式を示します。

パラメータ	タイプ	式
subjectareaid	文字列	\$InfaTask/mdmavxsd:taskData/mdmavxsd:subjectAreaUID
title	文字列	\$InfaTask/mdmavxsd:taskData/mdmavxsd:title
creator	文字列	\$InfaTask/mdmavxsd:taskData/mdmavxsd:creator
mdmtasktype	文字列	\$InfaTask/mdmavxsd:taskType/mdmavxsd:name
orsId	文字列	\$InfaTask/mdmavxsd:orsId
duedate	文字列	\$InfaTask/mdmavxsd:taskData/mdmavxsd:dueDate
tasktypename	文字列	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:name
taskTypeDisplayName	文字列	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:displayName
taskTypeDescription	文字列	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:description
taskTypePendingBVT	ブール	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:pendingBVT
taskTypeDataUpdateType	文字列	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:dataUpdateType
taskTypeDisplayType	文字列	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskType/mdmavxsd:displayType
priorityOut	文字列	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:taskData/mdmavxsd:priority
workflowVersion	文字列	htd:getInput('processTaskRequest')/mdmavxsd:INFATask/mdmavxsd:workflowVersion/text()

.bpel ファイルの更新の詳細については、ActiveVOS のマニュアルを参照してください。

## サブジェクト領域のためのワークフロー内での添付の有効化

バージョン 10.3 以降、カスタマイズしたワークフローを更新し、ユーザーが自分のタスクにファイルを添付できるようになりました。カスタマイズしたワークフロー内でこの機能を有効にするには、データを初期化するプロセスを更新し、すべてのユーザープロセスを更新します。

**注:** 一部のタスクに対して添付ファイルを追加する機能を制限する場合、プロビジョニングツールでタスク設定を設定できます。タスク設定の更新の詳細については、『*Multidomain MDM のプロビジョニングツールガイド*』を参照してください。

1. ActiveVOS デザイナーで、カスタマイズしたワークフローの.bpel ファイルを開きます。  
ワークフローがキャンバスに表示されます。
2. データを初期化するプロセスを更新します。
  - a. **【データの初期化】** プロセスを選択します。  
**【操作】** ダイアログボックスが表示されます。
  - b. **【新しいスクリプト】** をクリックします。  
**【スクリプトビルダ】** ダイアログボックスが表示されます。
  - c. **【スクリプト】** ボックスで、次の文を追加します。  
`abx:copyAllAttachments('StartRequest', '_peopleActivityAttachments')`  
`[StartRequest]` を、ワークフロー内でデータの初期化に使用する変数名で置き換えます。
  - d. **【OK】** をクリックします。  
**【スクリプトビルダ】** ダイアログボックスが閉じ、スクリプトが **【操作】** リストに表示されます。
  - e. **【OK】** をクリックします。  
**【操作】** ダイアログボックスが閉じます。
3. すべてのユーザープロセスを更新します。
  - a. ユーザープロセスを選択します。
  - b. **【プロパティ】** タブをクリックします。
  - c. **【すべて】** タブをクリックします。
  - d. **【添付ファイルのプロパゲート】** を展開します。
  - e. **【送信元プロセス】** を **【すべて】** に設定します。
  - f. **【送信先プロセス】** を **【newOnly】** に設定します。
4. .bpel ファイルを保存します。

## サブジェクト領域に基づく ActiveVOS ワークフローの再デプロイ

バージョン 10.0 HotFix 2 では、サブジェクト領域に基づいたワークフローアダプタ用の ActiveVOS ワークフローが変更されました。サブジェクト領域に基づいたワークフローアダプタを使用する場合、リソースキットで提供されるデフォルトワークフローを再デプロイする必要があります。タスクワークフローが含まれる Informatica ActiveVOS プロジェクトを MDM Hub サーバーにデプロイするには、まず CommonMDM プロジェクトをエクスポートし、次に MDMWorkflow プロジェクトをエクスポートします。

1. Informatica ActiveVOS Designer で、エクスポートする BeCommonMDM プロジェクトを開きます。

2. **【ファイル】 > 【エクスポート】** をクリックします。  
**【エクスポート】** ダイアログボックスが開きます。
3. **【Orchestration】** の下で、**【コントリビューション-ビジネスプロセスアーカイブ】** を選択します。**【次へ】** をクリックします。
4. **【デプロイメント URL】** フィールド (**【サーバーデプロイメントオプション】** の下) に、ActiveVOS インスタンスの URL を入力します。**【完了】** をクリックします。
5. **【デプロイメントの完了】** ダイアログボックスで、**【OK】** をクリックします。
6. BeMDMWorkflow プロジェクトについてすべての手順を繰り返します。  
BeMDMWorkflow プロジェクトをエクスポートする前に、BeCommonMDM プロジェクトをエクスポートする必要があります。

## ビジネスエンティティおよびビジネスエンティティサービス構成ファイルの生成

ビジネスエンティティおよびビジネスエンティティサービス構成ファイルを生成するには、Informatica Data Director コンフィギュレーションマネージャを使用します。

1. コンフィギュレーションマネージャの **【アプリケーション】** ペインで、ビジネスエンティティおよびビジネスエンティティサービス構成内に構成を生成する Informatica Data Director アプリケーションを選択します。
2. **【ビジネスエンティティスキーマの生成】** をクリックします。  
コンフィギュレーションマネージャによって、ビジネスエンティティとビジネスエンティティサービスの設定が生成されます。
3. コンフィギュレーションマネージャに、ビジネスエンティティおよびビジネスエンティティサービス構成の生成中に発生した問題のメッセージが表示されます。メッセージは、コンフィギュレーションマネージャが生成処理中に問題を解決したかどうかを示し、および問題解決のためにコンフィギュレーションマネージャが行った変更について説明します。コンフィギュレーションマネージャが問題を解決しなかった場合は、問題と、問題解決のために行える推奨アクションを書き留めます。

## 付録 A

# アップグレードプロセスのトラブルシューティング

アップグレードに失敗したか、アップグレード中に問題が発生した場合は、次の情報を利用して問題のトラブルシューティングを行います。

### EAR ファイルが JBoss 環境で許可される時間内にデプロイされない。

オペレーションリファレンスストアの数が増えると、EAR ファイルのデプロイメント時間も長くなります。EAR ファイルのデプロイメント時間が、JBoss 環境で許可される時間を超えると、アップグレードは失敗します。

この問題を解決するには、EAR ファイルのデプロイメント時間に対応するように、許可されるデプロイメント時間を増やします。許可される時間のデフォルトは 600 秒です。

1. `deploy.wait.time` プロパティ（次のディレクトリの `build.properties` ファイル内）の値を増やします。  
`<infamdm installation directory>/hub/server/bin`
2. 次のディレクトリに移動します。<JBoss installation directory>/standalone/configuration
3. `standalone-full.xml` ファイルに次のコードを設定してタイムアウト値を増やします。  

```
<subsystem xmlns="urn:jboss:domain:deployment-scanner:1.1">
  <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-interval="5000"
    deployment-timeout="1200"/>
</subsystem>
```

### Hub サーバーのアップグレードが失敗しました。

この問題を解決するには、EAR ファイルを再デプロイして Hub サーバーのアップグレードを再試行します。

**注:** JBoss 環境で JBoss の実行中に `standalone-full.xml` ファイルの中のデータソースの設定を手動で変更すると、`patchInstallSetup` スクリプトを実行したときに設定の変更が失われます。

1. 次のディレクトリに移動します。<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server
2. 次のコマンドを実行して、Hub サーバーアプリケーションをデプロイし、アプリケーションサーバーの設定に変更を適用します。

**注:** 使用する環境に埋め込み ActiveVOS がない場合は、コマンドに ActiveVOS のユーザー名とパスワードを含める必要はありません。

## UNIX の場合:

### WebLogic

```
patchInstallSetup.sh -Dweblogic.password=<WebLogic password> -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

### WebSphere (セキュリティ有効)

```
patchInstallSetup.sh -Dwebsphere.password=<WebSphere password> -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

### WebSphere (セキュリティ無効)

```
patchInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

### JBoss

```
patchInstallSetup.sh -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

**注:** UNIX でパスワードに感嘆符 (!) を含める場合、感嘆符の前にバックスラッシュを付ける必要があります。例えば、パスワードが!!cmx!! の場合は、\! \!cmx\! \!と入力します。

## Windows の場合:

### WebLogic

```
patchInstallSetup.bat -Dweblogic.password=<WebLogic password> -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

### WebSphere (セキュリティ有効)

```
patchInstallSetup.bat -Dwebsphere.password=<WebSphere password> -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

### WebSphere (セキュリティ無効)

```
patchInstallSetup.bat -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

### JBoss

```
patchInstallSetup.bat -Ddatabase.password=<MDM Hub Master database password> -Davos.username=<ActiveVOS Console username> -Davos.password=<ActiveVOS Console password> -Davos.jdbc.database.password=<ActiveVOS database password>
```

ActiveVOS コンソールの資格情報は、アプリケーションサーバーの管理者ユーザーの資格情報と同じです。

ActiveVOS データベースの資格情報は、create\_bpm スクリプトを実行するために使用した資格情報と同じです。

## MDM Hub のクリーンアップグレードが失敗する。

バージョン 10.4 にアップグレードした後、クリーンアップグレードを実行するときに既存のデータベースを使用すると、アップグレードに失敗します。この問題は、クリーンアップグレードを実行するたびに公開キーと秘密キーの新しいペアが生成されるために発生します。既存データベースにアクセスするためにこの新しいキーペアを使用することはできません。

この問題を解決するには、次の手順を実行します。

1. Hub サーバー用の公開キーと秘密キーの新しいペアを古い公開鍵と秘密鍵のペアに置換します。  
Hub サーバー用の公開鍵と秘密鍵のペアは、次のディレクトリに格納されます。  
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/resources/certificates
2. postInstallSetup スクリプトを実行して、Hub サーバーアプリケーションをデプロイし、アプリケーションサーバーの設定に変更を適用します。  
postInstallSetup スクリプトの実行の詳細については、『*Multidomain MDM のインストールガイド*』の章「Hub サーバーのインストール後のタスク」を参照してください。
3. プロセスサーバー用の公開キーと秘密キーの新しいペアを古い公開鍵と秘密鍵のペアに置換します。  
プロセスサーバー用の公開鍵と秘密鍵のペアは、次のディレクトリに格納されます。  
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse/resources/certificates
4. postInstallSetup スクリプトを実行して、プロセスサーバーアプリケーションをデプロイし、アプリケーションサーバーの設定に変更を適用します。  
postInstallSetup スクリプトの実行の詳細については、『*Multidomain MDM のインストールガイド*』の章「プロセスサーバーのインストール後のタスク」を参照してください。

### WebLogic 環境でプロセスサーバーのアップグレードが失敗しました。

WebLogic 環境でプロセスサーバーをアップグレードする場合、次のエラーでアップグレードが失敗することがあります。

```
Unable to start application, deployment error msg: weblogic.management.ManagementException:  
[Deployer:149196]Rejecting start request for application siperian-mrm-cleanse.ear because stop request is  
running for the application.
```

問題を解決するには、WebLogic 管理コンソールを使用して siperian-mrm-cleanse.ear ファイルを手動でデプロイしてから、アプリケーションサーバーを再起動します。

### WebSphere 環境でプロセスサーバーのアップグレードが失敗しました。

WebSphere 環境でプロセスサーバーをアップグレードする場合、アップグレードが失敗することがあります。patchInstallSetup.log を開くと、次のエラーが表示されます。

```
Failed to load webapp: com/delos/cmx/server/Initialization.getCleanseLoggingConfiguration;loaded from  
file:/E:/IBM/WAS9053/AppServer/profiles/AppSrv01/installedApps/MDMWSDB2W16001Node02Cell/siperian-mrm-  
cleanse.ear.ear/lib/siperian-server
```

この問題を解決するには、WebSphere 管理コンソールを使用して siperian-mrm-cleanse.ear ファイルを手動でデプロイ解除してから、patchInstallSetup スクリプトを実行します。

patchInstallSetup スクリプトは次のディレクトリにあります。

<MDM Hub インストールディレクトリ>/hub/cleanse/

### プロセスサーバーのアップグレードが失敗しました。

この問題を解決するには、EAR ファイルを再デプロイしてプロセスサーバーのアップグレードを再試行します。

**注:** JBoss の実行中に standalone-full.xml の中のデータソースの設定を手動で変更すると、patchInstallSetup スクリプトを実行したときに設定の変更が失われます。

1. 次のディレクトリに移動します。<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/cleanse

2. 次のコマンドを実行して、プロセスサーバーアプリケーションをデプロイし、アプリケーションサーバーの設定に変更を適用します。

#### UNIX の場合:

WebLogic

```
patchInstallSetup.sh -Dweblogic.password=<WebLogic password: WebLogic パスワード>
```

WebSphere

```
patchInstallSetup.sh -Dwebsphere.password= <WebSphere password>
```

JBoss

```
patchInstallSetup.sh
```

#### Windows の場合:

WebLogic

```
patchInstallSetup.bat -Dweblogic.password=<WebLogic password: WebLogic パスワード>
```

WebSphere

```
patchInstallSetup.bat -Dwebsphere.password= <WebSphere password>
```

JBoss

```
patchInstallSetup.bat
```

**注:** UNIX でパスワードに感嘆符 (!) を含める場合、感嘆符の前にバックスラッシュを付ける必要があります。例えば、パスワードが!!cmx!! の場合は、\! \!cmx\! \!と入力します。

### オペレーショナル参照ストアのアップグレードで ORA-00955 エラーが発生する。

オペレーショナル参照ストアをアップグレードすると、アップグレードは成功するが、次のエラーが sip\_ant ログに表示されます。

```
[exec] CREATE SEQUENCE "C_REPOS_ZDT_EVENT_SEQ" MINVALUE 1 MAXVALUE 999999999999
[exec] INCREMENT BY 1 START WITH 1 CACHE 20 NOORDER CYCLE
[exec] *
[exec] ERROR at line 1:
[exec] ORA-00955: name is already used by an existing object
[exec]
```

このエラーは無視しても問題ありません。

### 変更リストを空のオペレーショナル参照ストアに昇格させると ORA-00910 エラーが発生する。

変更リストを空のオペレーショナル参照ストアに昇格させると、マッチカラムの合計の長さが 4000 を超える場合、次のエラーが発生します。

ORA-00910: specified length too long for its datatype

変更リストを空のオペレーショナル参照ストアに昇格させるには、MDM Hub によって外部マッチ入力テーブルに追加されるマッチカラムの長さを 4000 以下にしてください。マッチカラムの長さは、マッチカラムのソースであるすべてのベースオブジェクトのカラムの長さとのソースカラム数の合計です。

### Oracle 環境においてオペレーショナル参照ストアのアップグレードで ORA-20005 エラーが発生する。

sip\_ant updateorsdatabase を実行するとエラー ORA-20005 が発生する場合は、以下の手順を実行します。

1. 次のコマンドを実行し、必要な権限を付与します。

```
exec dbms_java.grant_permission(upper('ORS_USER'),'SYS:java.net.SocketPermission','*',
'connect,resolve');
```

2. 次のコマンドを実行して、Oracle に Java クラスがロードされていることを確認します。  

```
select dbms_java.longname(object_name), status from user_objects where object_type='JAVA CLASS';
```
3. これらのクラスがロードされていない場合は、次のコマンドを実行してこれらのクラスを再ロードします。  

```
loadjava -verbose -force -resolve -oracleresolver -user &ors_name/&ors_passwd@&tns_name siperian-cleansecaller.jar  
loadjava -verbose -force -resolve -oracleresolver -user &ors_name/&ors_passwd@&tns_name siperian-dbutil.jar
```

### Hub ストアのアップグレードが失敗しました。

部分的にアップグレードされたスキーマで Hub ストアのアップグレードを再実行することはできません。アップグレードが失敗した場合は、フルバックアップからデータベースをリストアし、Hub ストアのアップグレードを再度実行します。

カラム名に予約語が含まれていることが原因で Hub ストアのアップグレードが失敗する場合は、名前を変更したカラムにデータを移行するスクリプトについて、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。

### 英語以外のロケールを使用してアップグレードすると、一部のテーブルは英語で表示され、一部のテーブルはロケールの言語で表示される。

Hub ストアデータベース環境を英語以外のロケールに設定する場合は、必ず文字セットを Unicode に変更したうえで、アップグレードスクリプトを実行して MDM Hub マスターデータベースとオペレーショナル参照ストアをアップグレードします。アップグレード時に、テーブルのメタデータはすべて翻訳キーを使用して英語に翻訳されます。Unicode 文字セットを選択していない場合は、一部のテーブルのみが翻訳されます。

### Hub コンソールが起動に失敗する

Hub コンソール用にサポートされている Java Runtime Environment (JRE) を使用していることを確認します。システム要件については、本バージョンの Multidomain MDM の「製品可用性マトリックス」を Informatica Network (<https://network.informatica.com/community/informatica-network/product-availability-matrices/overview>) で参照してください。

### Hub コンソールが JBoss 環境で起動できない

JBoss 環境では、JBoss アプリケーションサーバーが再起動しない場合、Hub コンソールを起動することはできません。MDM Hub により、リポジトリレイヤが初期化されなかったことを示すエラーが生成されます。

問題を解決するには、バッチファイルで以下のコードを実行して JBoss を再起動します。

```
rmmdir C:\<JBoss installation directory>\standalone\tmp /s /q  
mkdir C:\<JBoss installation directory>\standalone\tmp  
C:\<JBoss installation directory>\bin\standalone.bat -c standalone-full.xml -b 0.0.0.0
```

### Hub コンソールが Db2 環境で起動できない

Db2 データソースを含む MDM Hub 環境で Hub コンソールが起動できず、以下のエラーが表示されたとします。

SIP-09070: SIP-10318: データアクセスエラーにより、ユーザーを取得できませんでした。

SIP-10324: データオブジェクトをロードしようとしたときに、予期せぬ例外が発生しました。  
java.lang.NullPointerException

この問題は、MDM Hub の管理者ユーザー名とアプリケーションサーバーの管理者ユーザー名における大文字と小文字の不一致によって発生します。例えば、MDM Hub の管理者ユーザーが DB2ADMIN (大文字) であるのに対し、アプリケーションサーバーの管理者ユーザーが db2admin (小文字) である場合がこれに該当します。

この問題を解決するには、アプリケーションサーバーのユーザー名を MDM Hub のユーザー名と厳密に一致させる必要があります。

**注:** 大文字と小文字の区別に関連する問題を回避するには、Db2 用のユーザー名を定義する際にすべて大文字を使用することをお勧めします。

例えば、WebSphere を使用する場合には、ユーザー名を WebSphere コンソールで設定します。

1. WebSphere コンソールを開きます。
2. [リソース] > [データソース] > [siperian-cmx\_system-ds] > [カスタムプロパティ] .
3. [ユーザー] フィールドに DB2ADMIN と大文字で入力します。
4. [パスワード] フィールドにこのユーザーのパスワードを入力します。
5. [適用] をクリックしてから [保存] をクリックします。
6. WebSphere を再起動します。
7. Hub コンソールを起動し、ログインします。

### Db2 環境でプロビジョニングツールから加えた変更が適用できない。

Db2 環境で、オペレーショナル参照ストアが大規模である場合に、プロビジョニングツールで変更を適用すると、次のエラーメッセージが表示されます。

Failed to set user workspace configuration.

この問題を解決するには、オペレーショナル参照ストアで次の Db2 コマンドを実行して、ユーザーワークスペーステーブルのカラムの長さを増やします。

```
UPDATE C_REPOS_COLUMN SET DATA_LENGTH = 50000000 WHERE TABLE_NAME = 'C_REPOS_USER_WORKSPACE' AND COLUMN_NAME = 'WORKSPACE_DATA'  
ALTER TABLE C_REPOS_USER_WORKSPACE ALTER COLUMN WORKSPACE_DATA SET DATA TYPE BLOB(50M)  
REORG TABLE C_REPOS_USER_WORKSPACE  
COMMIT
```

### IDD では、従来のデータビューを使用して、サブジェクト領域に基づくレコードを表示できない。

IDD のレコードを表示するデフォルトのページは、ビジネスエンティティに基づくエンティティビューです。

従来のデータビューを使用するには、cmxserver.properties ファイルで dataview.enabled を true に設定します。

詳細については、次のハウツー記事「*Migrating IDD Applications to the Business Entity Data Model*」を参照してください。

### IDD がエラー SIP-BV-11500 で失敗する。

次のエラーが発生すると、IDD が失敗することがあります。SIP-BV-11500 Fatal Error Operational Reference Store localhost-orcl-MDM\_SAMPLE does not have a workflow engine configured.Each Operational Reference Store must have a workflow engine configured for use with the IDD even if workflow will not be used.

この問題を解決するには、プライマリワークフローアダプタが構成されていることを確認します。

詳細については、次の KB の記事を参照してください。

<https://kb.informatica.com/solution/23/Pages/55/381456.aspx?myk=381456>

### メタデータを検証するときに、オブジェクトがデータベースではなくメタデータに存在することを示すエラーが表示される。

リポトリマネージャを使用して問題を修正すると、次のエラーが発生します。ORA-00955 既存のオブジェクトによって名前がすでに使用されています。

この問題を解決するには、エラーが発生したテーブルに対して、プロキシロールの正しい権限が付与されていることを確認します。必要なアクセス許可の一覧を取得するには、エラーが発生しないテーブルを参照してください。

### Windows で、一致トークンの生成時に、エラーが発生する。

Generate Match Tokens プロセスで、クラス `ssa.ssaname3.jssan3cl` が初期化できないというエラーが返される。

1. SSA-NAME3 の DLL (dynamic linked library) ファイルが含まれるディレクトリ<MDM installation directory>/hub/cleanse/lib へのパスが PATH 環境変数に含まれているかを確認します。
2. MDM Hub の検索および一致を実行するプロセスサーバーに Visual Studio 2019 の Microsoft Visual C++再頒布可能パッケージがインストールされているかを確認します。
3. Visual Studio 2019 の Microsoft Visual C++再頒布可能パッケージがインストールされている場合は、依存性チェッカー (Dependency Walker (depends.exe) など) を使用して `jssan3cl.dll` をロードし、Visual C++再頒布可能パッケージが正常に適用されていることを確認します。

**ヒント:** Visual Studio 2019 の Visual C++再頒布可能パッケージには、Windows Server にオペレーティングシステムパッチがインストールされていることが必要です。Visual C++再頒布可能パッケージをインストールする前に、オペレーティングシステム要件を確認します。たとえば、Windows Server 2012 のベースラインバージョン以降、Visual C++再頒布可能パッケージを正常にインストールするには、オペレーティングシステムに約 100 個のパッチ (合計約 2 GB) を適用する必要があります。

### WebLogic アプリケーションサーバーの Microsoft SQL Server 環境でアップグレードを行った後に、Hub コンソールにログインできない。

Hub コンソールにログインすると、NULL ポインタ例外が発生します。

この問題を解決するには、<Microsoft SQL Server installation directory>\sqljdbc\_4.0\enu\xa にある `xa_install.sql` スクリプトで、削除コマンド、スキーマの作成コマンド、およびロールコマンドをコメントアウトします。スクリプトを実行して、アプリケーションサーバーを再起動します。

### WebSphere アプリケーションサーバーに Hub サーバーをインストールすると、アップグレードコンポーネント `patchInstallSetup` が失敗する。

この問題を解決するには、ファイル<WebSphere profile home>/properties/soap.client.props を開いて `com.ibm.SOAP.requestTimeout` を大きくしてから、WebSphere サーバードプロファイルを再起動します。`patchInstallSetup.bat` を再度実行します。

### IBM AIX 環境で Hub サーバーをアップグレードするとき、`entity360view.ear` ファイルがデプロイに失敗する。

この問題を解決するには、`patchInstallSetup.sh` スクリプトを実行します。

## 付録 B

# FAQ（よくある質問）

アップグレードするには新しいライセンスファイルが必要ですか？

はい。Multidomain MDM バージョン 10.1.x 以前からアップグレードする場合、新しいライセンスファイルが必要です。

10.0 より前のバージョンのデータベースのユーザーイグジットを使用できますか？

いいえ。データベースレイヤで実行されるデータベースのユーザーイグジットは、バージョン 10.0 以降では廃止されています。

アップグレードプロセス中に DBA のユーザー名とパスワードを提供する必要があるのはなぜですか？

アップグレードプロセスでは、権限の付与やシーケンスの作成など、DBA レベルの権限を必要とするアクションを実行します。アップグレードプロセスでこれらのアクションを実行できるようにするには、DBA 資格情報が必要です。

アップグレードプロセス中に、ActiveVOS の既存のバージョンはどうなりますか？

ActiveVOS がインストールされている環境でインプレースアップグレードを実行すると、アップグレードプロセスによって、環境に最新バージョンの ActiveVOS がインストールされます（まだインストールされていない場合）。必要な ActiveVOS のバージョンを確認するには、Informatica Network で製品可用性マトリックス (<https://network.informatica.com/community/informatica-network/product-availability-matrices/overview>) を参照してください。

ActiveVOS がインストールされていない環境では、MDM アップグレードプロセス中に ActiveVOS をインストールすることは必須ですか？

いいえ。アップグレードプロセス中に、ActiveVOS をインストールするかどうかを選択するように求められます。

このバージョンでは、Informatica Data Director の推奨画面解像度に変更されていますか？

いいえ。推奨画面解像度は変更されていません。Informatica Data Director の推奨画面解像度は 1280 x 1024 です。

Multidomain MDM バージョン 10.2 以降にアップグレードするときに、Informatica Data Quality をアップグレードする必要がありますか？

はい。環境内で Informatica Data Quality (IDQ) を使用する場合は、バージョン 10.1 の IDQ にアップグレードする必要があります。システム要件については、Informatica Network で製品可用性マトリックス (<https://network.informatica.com/community/informatica-network/product-availability-matrices/overview>) を参照してください。

### MDM Hub のセキュリティをカスタマイズする方法は？

MDM Hub のセキュリティのカスタマイズの詳細については、『*Multidomain MDM のセキュリティガイド*』と、*MDM Multidomain* 内でのセキュリティ構成ユーティリティの使用に関するハウツー記事を参照してください。

### Java 8 はサポートされていますか？

はい。このバージョンの Multidomain MDM は Java 8 をサポートしています。システム要件については、Informatica Network で製品可用性マトリックス (<https://network.informatica.com/community/informatica-network/product-availability-matrices/overview>) を参照してください。

**注:** ActiveVOS には Java 7 が必要です。

## 付録 C

# 既存の ActiveVOS タスクの処理

この付録では、以下の項目について説明します。

- [既存の ActiveVOS タスクの処理の概要, 194 ページ](#)
- [移行プロパティ, 194 ページ](#)
- [移行スクリプトの実行プロパティファイルを指定, 195 ページ](#)
- [移行スクリプトの実行プロパティをコマンドラインに指定, 196 ページ](#)

## 既存の ActiveVOS タスクの処理の概要

Multidomain MDM バージョン 10.1 より前に作成された ActiveVOS タスクを使用するには、移行スクリプトを実行して必要なプレゼンテーションパラメータをタスクに入力します。移行スクリプトを実行しないと、タスクがタスクマネージャに表示されません。バージョン 10.1 にアップグレードする前に作成されたすべてのタスクが処理されるまで、移行スクリプトを実行します。

移行スクリプトには、いくつかのプロパティを設定する必要があります。プロパティは、ビルドファイルに追加することも、コマンドラインに追加することもできます。

## 移行プロパティ

次の表に、移行プロパティを示します。

プロパティ	説明
avos.jdbc.database.driver.jar	ActiveVOS データベースの JDBC ドライバが含まれる JAR ファイルへのパス。 このパラメーターは、Hub サーバーのインストール時に<infamdm installation directory>\conf\avos.install.properties に avos プレフィックスなしで入力されます。
avos.jdbc.database.driver.class	ActiveVOS データベースの JDBC ドライバクラス。 このパラメーターは、Hub サーバーのインストール時に<infamdm installation directory>\conf\avos.install.properties に avos プレフィックスなしで入力されます。

プロパティ	説明
avos.jdbc.database.url	ActiveVOS データベースの接続 URL。 このパラメーターは、Hub サーバーのインストール時に<infamdm installation directory>\conf\avos.install.properties に avos プレフィックスなしで入力されます。
avos.jdbc.database.username	ActiveVOS データベースのユーザー名。 このパラメーターは、Hub サーバーのインストール時に<infamdm installation directory>\conf\avos.install.properties に avos プレフィックスなしで入力されます。
avos.jdbc.database.password	ActiveVOS データベースのパスワード。
avos.ws.protocol	ActiveVOS サーバー接続のプロトコル。http または https になります。
avos.ws.host	ActiveVOS が実行されるアプリケーションサーバーのホスト名。
avos.ws.port	アプリケーションサーバー接続のポート番号。
avos.ws.trusted.username	信頼されたユーザーのユーザー名。 <b>注:</b> 信頼されたユーザーは、Multidomain MDM のインストールおよびアップグレードプロセスの一環として作成されます。
avos.ws.trusted.password	信頼されたユーザーのパスワード。 <b>注:</b> 信頼されたユーザーは、Multidomain MDM のインストールおよびアップグレードプロセスの一環として作成されます。
avos.hub.username	すべてのタスク管理ロールに属する MDM Hub ユーザー。
avos.ws.pagesize	1 つのデータベーストランザクションで処理され、ActiveVOS からバッチロードされるタスクの数。
avos.ws.statuses	オプション。処理する ActiveVOS タスクステータスのカンマ区切りリスト。例えば、READY や IN_PROGRESS などです。デフォルトではすべてのタスクが処理されます。

## 移行スクリプトの実行プロパティファイルを指定

ワークフローに関連するロールに属する MDM ユーザーを作成します。移行プロパティを Hub サーバーの build.properties ファイルに追加してから、MDM Hub マスターデータベースを移行スクリプトコマンドで実行します。移行後も、タスク所有者は自身が割り当てたタスクでのアクションを継続できます。

移行スクリプトは定期的に行われます。サブジェクト領域のワークフローアダプタについてのタスクがすべて処理されたら、スクリプトを実行する必要がなくなり、この目的のために作成した MDM ユーザーを削除できます。

- MDM Hub ユーザーを作成し、そのユーザーにワークフローに関与するすべてのロールを割り当てます。たとえば、事前定義済みワークフローが使用するロールは、DataSteward、Manager、および SrManager です。
- テキストエディタで次のファイルを開きます。

<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/bin/build.properties

3. 移行プロパティを build.properties ファイルに追加します。プロパティのリストについては、[「移行プロパティ」 \(ページ 194\)](#)を参照してください。
4. コマンドプロンプトを開きます。
5. 次のディレクトリに移動します。  
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/bin
6. 移行スクリプトコマンドを含むアップグレードスクリプト MDM Hub マスターデータベースを実行します。
  - UNIX の場合: sip\_ant.sh migrate-avos-sa-tasks
  - Windows の場合: sip\_ant.bat migrate-avos-sa-tasks

## 移行スクリプトの実行プロパティをコマンドラインに指定

ワークフローに関連するロールに属する MDM ユーザーを作成します。コマンドプロンプトを開き、目的のプロパティを指定してスクリプトを実行します。移行後も、タスク所有者は自身が割り当てたタスクでのアクションを継続できます。

移行スクリプトは定期的に実行します。サブジェクト領域のワークフローアダプタについてのタスクがすべて処理されたら、スクリプトを実行する必要がなくなり、この目的のために作成した MDM ユーザーを削除できます。

1. MDM Hub ユーザーを作成し、そのユーザーにワークフローに関与するすべてのロールを割り当てます。  
たとえば、事前定義済みワークフローが使用するロールは、DataSteward、Manager、および SrManager です。
2. コマンドプロンプトを開きます。
3. 次のディレクトリに移動します。  
<MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/bin
4. 移行スクリプトコマンドと移行プロパティを含むアップグレードスクリプト MDM Hub マスターデータベースを実行します。プロパティのリストについては、[「移行プロパティ」 \(ページ 194\)](#)を参照してください。  
たとえば、次のコマンドのいずれか 1 つを実行できます。
  - UNIX の場合:
 

```
sip_ant.sh migrate-avos-sa-tasks -Davos.jdbc.database.password=!!cmx!!  
-Davos.ws.protocol=http -Davos.ws.host=localhost -Davos.ws.port=8080 -Davos.ws.pagesize=100  
-Davos.ws.trusted.username=avos -Davos.ws.trusted.password=avos -Davos.hub.username=admin
```
  - Windows の場合:
 

```
sip_ant.bat migrate-avos-sa-tasks -Davos.jdbc.database.password=!!cmx!!  
-Davos.ws.protocol=http -Davos.ws.host=localhost -Davos.ws.port=8080 -Davos.ws.pagesize=100  
-Davos.ws.trusted.username=avos -Davos.ws.trusted.password=avos -Davos.hub.username=admin
```

# メタデータキャッシュの設定

- [メタデータキャッシュの構成（オプション）](#) , 197 ページ

## メタデータキャッシュの構成（オプション）

メタデータキャッシュは、データオブジェクト、リポジトリオブジェクト、検索トークンなどのアイテムを管理します。MDM Hub は、Infinispan を使用してメタデータキャッシュを実行します。Infinispan は、Hub サーバーと共にインストールされます。Hub サーバーによって使用されるキャッシュの場合、Infinispan 構成ファイルにはデフォルトの属性値が含まれます。

バージョン 10.1 以前では、MDM Hub は、メタデータキャッシュに JBoss Cache を使用していました。これらのバージョンのいずれかからアップグレードすると、MDM Hub サーバーは、JBoss Cache 構成ファイルではなく、Infinispan 構成ファイルを使用します。

JBoss Cache 構成ファイルが以前のバージョンの MDM Hub で編集されていた場合は、Infinispan 構成ファイルを編集する必要がある場合があります。ファイルが編集された理由によって異なります。

### ネットワークポリシー

組織のネットワークポリシーに対応するために JBoss Cache ファイルを編集した場合は、Infinispan ファイルと jgroups\* ファイルを同じポリシー変更で更新します。

### パフォーマンス

キャッシュのパフォーマンスを向上させるために JBoss Cache ファイルを編集した場合は、まず、Infinispan のデフォルト値で MDM Hub を実行してみます。パフォーマンスの問題が発生した場合は、変更された値を JBoss Cache 構成ファイルから Infinispan 構成ファイルにコピーします。それでもパフォーマンスの問題が発生する場合は、Infinispan をよく理解し、環境により適した値に調整します。

## Infinispan 属性

次の表は、デフォルトの Infinispan 属性値をまとめたもので、属性が以前の JBoss 属性にどのようにマップされるかを示しています。

Infinispan 要素と属性	デフォルト 値	説明	JBoss 属性
locking acquire-timeout	60000	Hub サーバーがロックの取得を試みる最大時間。	lockAcquisitionTimeout
transaction stop-timeout	30000	キャッシュが停止すると、この属性は、Hub サーバーがリモートおよびローカルのトランザクションを完了するまでに Infinispan が待機する最大時間を設定します。	sync replTimeout
transport cluster	infinispan-cluster	基になるグループ通信クラスタの名前。	clustering
transport stack	UDP	構成の種類: UDP または TCP。構成は、jgroups-udp.xml ファイルおよび jgroups-tcp.xml ファイルで定義されます。	jgroupsConfig
transport node-name	\$node\$	現在のノードの名前。Hub サーバーがこの属性を設定します。 node-name のデフォルトはホスト名と乱数の組み合わせです。この番号により、同じホスト上の複数のノードを区別します。	--
transport machine	\$machine\$	ノードが実行されるコンピュータの ID。Hub サーバーがこの属性を設定します。	--
expiration lifespan	--	キャッシュエントリの最大寿命（ミリ秒単位）。キャッシュエントリが寿命を超えると、クラスタ内のエントリは有効期限切れになります。パフォーマンスを最適化する場合がある場合は、DISABLE_WHEN_LOCK、DATA_OBJECTS、および REPOS_OBJECTS の各キャッシュの寿命を長くします。 例えば、寿命を 1 時間（3600000）から 1 日（86400000）に増やすことができます。 各キャッシュには、この属性の独自のデフォルト値があります。デフォルト値を検索するには、infinispanConfig.xml ファイルを開きます。	eviction timeToLive
expiration interval	--	寿命を確認するための最大間隔。パフォーマンスを最適化する場合がある場合は、DISABLE_WHEN_LOCK、DATA_OBJECTS、および REPOS_OBJECTS の各キャッシュの間隔を長くします。 例えば、間隔を 5 秒（5000）から 5 分（300000）に増やすことができます。 各キャッシュには、この属性の独自のデフォルト値があります。デフォルト値を検索するには、infinispanConfig.xml ファイルを開きます。	eviction timeToLive

## Infinispan 属性の編集

メタデータキャッシュ属性を設定するには、Hub サーバーの `infinispanConfig.xml` ファイルを編集します。Infinispan 設定のヘルプについては、Infinispan のマニュアルを参照してください。

**注:** プロセスサーバーには、Infinispan 設定ファイルもあります。デフォルトの属性値は十分に機能するはずですが、プロセスサーバーのパフォーマンスに問題があることに気付く場合は、属性値を微調整できます。

1. 次のディレクトリに移動します。 <MDM Hub installation directory: MDM Hub のインストールディレクトリ>/hub/server/resources
2. 次のファイルのバックアップコピーを作成します。 `infinispanConfig.xml`
3. `infinispanConfig.xml` ファイルを開き、Infinispan バージョン番号を見つけます。これは `xsi:schemaLocation` 属性内にあります。
4. その Infinispan バージョンのマニュアルを確認します。

**注:** 次の URL で、パスに # が含まれている場合は、バージョン番号を置き換えます。#.

- 設定スキーマを表示するには、ファイルの `xsi:schemaLocation` 属性に含まれている URL に移動します。
  - 属性の詳細を知るには、<https://docs.jboss.org/infinispan/<#.x>/configdocs/> に移動してください。
  - Infinispan の詳細を知るには、<http://infinispan.org/docs/<#.x>/> に移動し、「Frequently Asked Questions」リンクを選択します。
5. ファイルを編集して保存します。

# 索引

## A

ActiveVOS

URN、設定 [170](#), [178](#)

自動インストーラのプロパティ [98](#)

接続テスト [136](#)

接続の更新 [136](#)

ActiveVOS コンソールの管理者ユーザー

abAdmin ロール [56](#), [65](#), [75](#)

作成 [56](#), [65](#), [75](#)

## C

cmxserver.log ファイル [101](#), [113](#)

## D

Data Director

アップグレードテスト [142](#)

サブジェクト領域のアップグレードテスト [142](#)

## E

Elasticsearch

高可用性 [146](#)

elasticsearch アーカイブ

抽出 [147](#)

Elasticsearch のインストール

インストール前のタスク [146](#)

前提条件 [146](#)

## H

hub\_cleanse\_install.bin [104](#)

hub\_cleanse\_install.exe [104](#)

hub\_install.bin [92](#)

hub\_install.exe [92](#)

Hub サーバー

アップグレードの再適用 [102](#)

グラフィカルモードでのアップグレード [92](#)

サイレントアップグレード [98](#), [99](#)

サイレントプロパティファイル [98](#)

Hub サーバーのアップグレード

コンソールモード [95](#)

ログファイル [101](#)

概要 [91](#)

Hub ストア

テーブルスペース、作成 [35](#), [43](#)

Hub ストアのアップグレード

オペレーショナル参照ストア

アップグレード [85](#)

マスターデータベース [83](#)

Hub ストアのアップグレード (続く)

概要 [82](#)

## I

IBM Db2 の設定

MDM Hub 用 [43](#)

infamdm\_installer\_debug.txt ファイル [101](#), [113](#)

Infinispan

設定 [199](#)

Informatica ActiveVOS

スキーマの作成 [38](#), [48](#)

データベースの作成 [42](#)

## J

Java のオプション

設定 [71](#)

JBoss クラスタードでの開始

スタンドアロンインスタンスでの開始 [59](#)

JBoss

JBoss クラスタードでの開始 [60](#)

スタンドアロンの JBoss インスタンスでの開始 [59](#)

JBoss の設定

MDM Hub 用 [49](#)

JVM パラメータ

設定 [71](#)

## M

MDM Hub

アップグレードテスト [141](#)

MDM Hub マスターデータベースのアップグレード

Verbose モード [83](#)

サイレントモード [84](#)

Microsoft SQL Server

ODBC ドライバ [41](#)

unixODBC ドライバ [41](#)

インストール [40](#)

データファイルストア [41](#)

設定 [40](#)

分散トランザクション [40](#)

Microsoft SQL Server の設定

MDM Hub 用 [40](#)

## O

Oracle の設定

init.ora パラメータの設定 [35](#)

MDM Hub 用 [35](#)

Oracle リサイクルビンの無効化 [35](#)

## P

postInstallSetup.log ファイル [101](#), [113](#)

## S

stopwords.txt ファイル

設定 [149](#)

synonyms.txt ファイル

設定 [149](#)

## T

TNS 名

追加 [35](#)

## U

URN

ActiveVOS の設定 [170](#), [178](#)

## W

WebLogic

構成 [60](#)

WebSphere

構成 [70](#)

WebSphere 管理セキュリティ

EAR ファイルのアンインストール [120](#)

Hub サーバーの PostInstallSetup スクリプトの実行 [122](#)

プロセスサーバーの PostInstallSetup スクリプトの実行 [122](#)

WebSphere セキュリティ

ORS の登録解除 [120](#)

WebSphere の設定

Informatica Data Director 用の [81](#)

## あ

アップグレード

ガイドライン [13](#)

マスターデータベース [83](#)

概要 [12](#), [13](#)

アップグレードテスト

Hub コンソールのツール [141](#)

サブジェクト領域を使用する Data Director [142](#)

ビジネスエンティティを使用する Data Director [142](#)

プロビジョニングツール [141](#)

概要 [141](#)

アップグレードプロセス

インプレースアップグレード [14](#)

クリーンアップグレード [13](#)

## い

一致バリュレーション

有効化 [111](#)

## お

オペレーショナルリファレンスストア

登録 [23](#)

オペレーショナル参照ストア (ORS)

接続テスト [136](#)

接続の更新 [136](#)

オペレーショナル参照ストアのアップグレード

Verbose モード [85](#)

サイレントモード [88](#)

## か

カスタムコード、テスト [141](#)

環境レポート

確認 [138](#)

保存 [138](#)

## く

クレンジング関数

テスト [136](#)

## け

検証結果

保存 [133](#)

## さ

サーバーのプロパティ 設定 [55](#)

サイレントアップグレード

Hub サーバーの [99](#)

プロセスサーバーのサイレントアップグレードの実行 [108](#)

サブジェクト領域のワークフロー

カスタム、ファイル添付の有効化 [183](#)

## す

スキーマのパスワード

暗号化 [135](#)

更新 [135](#)

## た

タイムアウトプロパティの設定 [57](#)

## て

データベース

手動作成 [44](#)

接続テスト [35](#)

テーブルスペース

作成 [35](#), [43](#)

テスト

アップグレードテスト [141](#)

カスタムコード [141](#)

## と

トラブルシューティング

インストール後のプロセス [185](#)

## ふ

ファイル添付

カスタムワークフロー内での有効化 [173](#)

有効化、サブジェクト領域のためのワークフロー [183](#)

プロセスサーバー

アップグレードの再適用 [113](#)

グラフィカルモードでのアップグレード [104](#)

コンソールモードでのアップグレード [106](#)

サイレントアップグレード [107](#)

サイレントプロパティファイル [107](#)

削除して追加 [136](#)

登録 [136](#)

プロセスサーバーのアップグレード

ログファイル [113](#)

概要 [103](#)

## へ

ベースオブジェクト

リレーションベースオブジェクトから戻す [165](#), [167](#)

## め

メタデータの検証

検証チェック [132](#)

メタデータ

検証 [132](#)

メタデータ (続く)

検証メッセージ、解決 [133](#)

## り

リレーションベースオブジェクト

ベースオブジェクトに戻す [165](#), [167](#)

## ろ

ログファイル

Cleanse Match Server のアップグレード [113](#)

cmxserver.log ファイル [101](#), [113](#)

Hub サーバーのアップグレード [101](#)

Infamdm\_Cleanse\_Match\_Server\_InstallLog.xml ファイル [113](#)

infamdm\_installer\_debug.txt ファイル [101](#), [113](#)

Infamdm\_Server\_InstallLog.xml ファイル [101](#)

postInstallSetup.log ファイル [101](#), [113](#)

アプリケーションサーバーのログファイル [101](#), [113](#)

## わ

ワークフロー

カスタム、ファイル添付の有効化 [173](#)