



Informatica® Multidomain MDM
10.4

リポジトリマネージャガイド

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

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

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

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

発行日: 2020-06-05

目次

序文	7
Informatica のリソース	7
Informatica Network	7
Informatica ナレッジベース	7
Informatica マニュアル	7
Informatica 製品可用性マトリックス	8
Informatica Velocity	8
Informatica Marketplace	8
Informatica グローバルカスタマサポート	8
第 1 章 : 概要	9
概要	9
リポジトリマネージャツールについて	9
メタデータ管理の概念	10
メタデータ	10
デザインオブジェクト	10
リポジトリ	11
変更リスト	11
リアルタイムのメタデータ管理	12
メタデータのコピーに関する考慮事項	12
変更結果の監視	14
第 2 章 : リポジトリマネージャ入門ガイド	15
概要	15
リポジトリマネージャの開始	15
リポジトリマネージャインターフェースのコンポーネント	15
タブ	16
コマンドボタン	16
リポジトリリスト	16
リポジトリマネージャの操作	17
自動排他ロック	17
グラフィカルモデルビューでのスキーマの表示	17
第 3 章 : メタデータの検証	18
概要	18
メタデータ検証プロセスについて	18
論理モデルと物理スキーマの一致	18
メタデータ検証プロセス	19
メタデータ検証のスコープ	19
問題の重要度レベル	20

検証インジケータ.....	20
[検証] タブのコマンドボタン.....	20
オペレーショナルリファレンスストアのメタデータ検証.....	21
メタデータの検証.....	23
インポートされたスキーマの事前検証結果.....	24
フィルタリングの問題.....	24
検証結果の保存.....	25
階層履歴の表示.....	25
メタデータ修復プロセス.....	26
メタデータ修復結果.....	27
リポジトリ内のメタデータの修復.....	27
第4章: リポジトリ間での変更の昇格.....	28
リポジトリ間での変更の昇格の概要.....	28
昇格のシナリオ.....	28
昇格可能なデザインオブジェクト.....	29
オブジェクト昇格時の競合.....	29
昇格プロセスに関する考慮事項.....	31
変更の視覚的昇格.....	32
視覚的昇格タスクの概要.....	32
[昇格] / [視覚] タブへの移動.....	33
[視覚] タブのコマンドボタン.....	33
視覚的昇格のソースリポジトリの選択.....	34
視覚的昇格のターゲットリポジトリの選択.....	34
視覚的昇格でのデザインオブジェクト階層内の移動.....	34
変更のターゲットリポジトリへの視覚的昇格.....	36
マークアップを伴う表示.....	38
競合の検索.....	38
変更を元に戻す.....	38
比較変更リストファイルの変更の保存.....	39
ターゲットリポジトリへの変更の適用.....	39
変更リストを使用した変更の昇格.....	40
変更リスト昇格タスクの概要.....	40
[昇格] / [変更リスト] タブへの移動.....	41
[変更リスト] タブのコマンドボタン.....	41
変更リスト昇格のターゲットのリポジトリの選択.....	42
リポジトリの比較による比較変更リストの作成.....	42
比較変更リスト XML ファイルを開く.....	43
変更リスト内の移動.....	43
変更の簡単な説明の表示.....	43
変更の詳細な説明の表示.....	44
比較変更リスト XML ファイルの変更の保存.....	44
変更リストの適用シミュレーションの実行.....	45

ターゲットリポジトリへの変更リストの適用.	45
コマンドラインからの変更の昇格.	46
第 5 章 : デザインオブジェクトのインポート.	48
概要.	48
デザインオブジェクトのインポートについて.	48
インポートプロセス.	48
インポート可能なデザインオブジェクト.	49
インポートプロセスに関する考慮事項.	49
デザインオブジェクトのインポート.	50
インポートタスクの概要.	50
[インポート] タブのコマンドボタン.	50
インポートのソースリポジトリの選択.	50
インポートのターゲットリポジトリの選択.	51
階層内のデザインオブジェクトの表示および非表示.	51
インポートするデザインオブジェクトの選択.	52
デザインオブジェクトの名前変更.	52
選択したデザインオブジェクトのインポート.	53
リレーションベースオブジェクトの開始日と終了日の情報の更新.	54
第 6 章 : リポジトリのエクスポート.	55
概要.	55
リポジトリのエクスポートについて.	55
エクスポートについて.	55
エクスポート可能なデザインオブジェクト.	55
エクスポートした変更リスト XML ファイルの用途.	56
エクスポートプロセスに関する考慮事項.	56
[エクスポート] タブのコマンドボタン.	56
リポジトリのエクスポート.	57
デザインオブジェクトのサブセットのエクスポート.	57
第 7 章 : 共通ウェアハウスモデルのサポート.	59
概要.	59
[CWM ファイルからのインポート] タブ.	60
[CWM ファイルからのインポート] タブのコマンドボタン.	60
CWM ファイルからのデザインオブジェクトのインポート.	60
[CWM ファイルへのエクスポート] タブ.	62
[CWM ファイルへのエクスポート] タブのコマンドボタン.	62
CWM ファイルへのリポジトリのエクスポート.	63
付録 A : デザインオブジェクトリファレンス.	64
概要.	64
リポジトリマネージャでサポートされているデザインオブジェクト.	64

デザインオブジェクトの依存関係.....	66
名前の変更が可能なデザインオブジェクト.....	67
付録 B: 変更リストリファレンス.....	69
概要.....	69
変更リスト XSD ファイル.....	69
変更リスト XML ファイルのルートタグと属性.....	70
変更リスト XML ファイル内の変更のタイプ.....	70
変更リスト XML ファイル内のデザインオブジェクトと変更.....	70
付録 C: MetCommand リファレンス.....	76
概要.....	76
MetCommand について.....	76
作業を開始する前に.....	77
要件.....	77
接続の設定.....	77
使用方法.....	77
ヘルプの出力.....	77
コマンドラインの引数.....	78
XML Over HTTP.....	78
プロキシユーザーのアクセス.....	79
変更リスト適用時の変更のロールバック.....	79
例.....	79
メタデータの取得.....	79
変更リストの作成.....	79
変更リストの検証.....	79
変更リストの適用.....	80
RollbackToLast.....	80
メタデータの検証.....	80
戻りコード.....	80
スクリプトの実行.....	81
カスタムスクリプトの実行.....	81
スクリプトの例.....	81
MetCommand の拡張.....	81
索引.....	82

序文

MDM Hub コンソールのリポジトリマネージャツールについては、Informatica^(R) *Multidomain MDM Repository Manager* ガイドを参照してください。ツールを使用してリポジトリのメタデータを検証し、オブジェクトを 1 つのリポジトリから別のリポジトリにコピーし、リポジトリをエクスポートし、リポジトリスキーマを視覚化する方法を学習します。

Informatica のリソース

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

Informatica 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) 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) 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 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 からグローバルサポートセンターに連絡できます。

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 で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、eSupport オプションを選択します。

第 1 章

概要

この章では、以下の項目について説明します。

- [概要, 9 ページ](#)
- [リポジトリマネージャツールについて, 9 ページ](#)
- [メタデータ管理の概念, 10 ページ](#)

概要

この章では、リポジトリマネージャツールおよび関連する概念について説明します。

注: このドキュメントは、*Multidomain MDM の概要ガイド*をすでに読み、MDM Hub のアーキテクチャと主な概念を基本的に理解していることを前提として作成されています。

リポジトリマネージャツールについて

リポジトリマネージャは、以下の目的で使用する Hub コンソールツールです。

- Informatica MDM Hub の実装内でリポジトリのメタデータを**検証**する。

検証では、リポジトリを定義するメタデータの完全性と整合性が確認されます。検証プロセスでは、リポジトリの論理モデルと物理スキーマを比較します。問題が発生すると、リポジトリマネージャによって、重要度別に分類された、注意が必要な問題のリストが生成されます。特定の操作で、リポジトリマネージャは重要な問題のないリポジトリを必要とします。

- 以下の 2 つのうちのいずれかの方法で、リポジトリ間でデザインオブジェクトを**コピー**する。

方法	説明
昇格	新しいデザインオブジェクト、または既存のオブジェクトの変更（属性値の違いなど）を、ターゲットリポジトリに昇格することができます。昇格は、リポジトリ間で差分変更をコピーするために使用されます。
インポート	デザインライブラリから空のターゲットリポジトリにデザインオブジェクトをインポートすることができます。

- リポジトリを移植可能な XML ファイルに**エクスポート**する。この XML ファイルは、別のリポジトリにインポートまたは昇格したり、編集したり、またはアーカイブとしてソース制御システムに保存することができます。

リポジトリマネージャを使用して、リポジトリ全体を変更リスト XML ファイルにエクスポートすることができます。この XML ファイルを使用して、デザインオブジェクトをターゲットリポジトリにインポートしたり、アーカイブとしてソース制御システムに保存したりすることができます。

- リポジトリのグラフィカルモデルビューを使用して、スキーマを**視覚化**する。

関連項目：

- [「メタデータの検証」 \(ページ 18\)](#)
- [「リポジトリ間での変更の昇格」 \(ページ 28\)](#)
- [「デザインオブジェクトのインポート」 \(ページ 48\)](#)
- [「リポジトリのエクスポート」 \(ページ 55\)](#)
- [「グラフィカルモデルビューでのスキーマの表示」 \(ページ 17\)](#)
- [「リポジトリマネージャ入門ガイド」 \(ページ 15\)](#)

メタデータ管理の概念

この節では、リポジトリマネージャを効果的に使用するために理解する必要のある、メタデータ管理の概念について説明します。

メタデータ

メタデータは、他のデータを記述するために使用されるデータです。Informatica MDM Hub で、メタデータは、Informatica MDM Hub の実装内で使用されているスキーマ（データモデル）を説明するために、関連する構成設定とともに使用されます。

デザインオブジェクト

MDM Hub で、デザインオブジェクトは、実装のスキーマを定義するために使用されるメタデータです。デザインオブジェクトには、ベースオブジェクトとベースカラム、ランディングテーブルとステージングテーブル、カラム、インデックス、リレーション、マッピング、クレンジング関数、クエリとパッケージ、信頼関係の設定、検証ルールと一致ルール、Security Access Manager の定義、階層マネージャの定義、およびその他の設定が含まれます。

関連項目：

- [「リポジトリマネージャでサポートされているデザインオブジェクト」 \(ページ 64\)](#)

依存関係

リポジトリマネージャは、デザインオブジェクト間の依存関係を管理します。例えば、昇格またはインポートのためにベースオブジェクトを選択すると、リポジトリマネージャによって、関連付けられているすべての子オブジェクト（カラム、一致ルールなど）が自動的に選択されます。同様に、マッピングを選択すると、リポジトリマネージャによって、関連付けられているランディングテーブルとステージングテーブルが自動的に選択されます。リポジトリマネージャでは直接的な依存関係と間接的な依存関係の両方が対象となるため、関連

するすべてのデザインオブジェクトが選択されます。また、リポジトリマネージャは、昇格しようとしているデザインオブジェクトの依存関係の競合にフラグを設定します。

関連項目：

- [「依存関係の競合」 \(ページ 30\)](#)

リポジトリ間でオブジェクトをコピーしたときの競合

リポジトリ間でのデザインオブジェクトの昇格またはインポートを試みると、競合が発生する場合があります。競合は、名前が同じ 2 つのベースオブジェクトなど、ID が同じ 2 つのデザインオブジェクト間の相違によって生じます。一部の競合は自動的に解決できますが、人間による解釈や手動操作が必要な競合もあります。

関連項目：

- [「オブジェクト昇格時の競合」 \(ページ 29\)](#)

システムオブジェクト

新たに作成された ORS に事前定義されている特定のシステムオブジェクト（管理システム、システムクレンジング関数、階層マネージャの RBO オブジェクトなど）は、リポジトリマネージャによって管理されるデザインオブジェクトには含まれません。

リポジトリ

リポジトリマネージャは、リポジトリに格納されているデータを管理します。

Hub Store 内のメタデータ

メタデータは、Informatica MDM Hub 内の以下の 2 つの場所に格納されます。

- マスターデータベースには、Informatica MDM Hub の実装全体を説明するグローバルメタデータが格納されます。
- オペレーショナル参照ストア（ORS、*リポジトリ*とも呼ばれる）には、それ自体のスキーマについてのメタデータと他の構成設定が含まれます。

リポジトリマネージャでは、マスターデータベースではなく、リポジトリに格納されているメタデータが使用されます。マスターデータベースに格納されているメタデータ（ユーザーアカウントやメッセージキュー設定など）はこのドキュメントの対象外です。

Hub Store および Informatica MDM Hub の実装のスキーマの詳細については、*Multidomain MDM の設定ガイド*を参照してください。

ソースリポジトリとターゲットリポジトリ

リポジトリ間でメタデータをコピーする場合、コピーするデザインオブジェクトを含むソースリポジトリとデザインオブジェクトのコピー先であるターゲットリポジトリが常に必要です。

リポジトリ間でメタデータをコピーするときは、どのリポジトリがソースでどのリポジトリがターゲットかをリポジトリマネージャで明示的に特定します。

変更リスト

変更リストは、ターゲットリポジトリに対する変更のリストです。変更リストにおける**変更**とは、ターゲットリポジトリに対して実行される操作（ベースオブジェクトの追加や一致ルールのプロパティの更新など）のことです。

変更リストのタイプ

リポジトリマネージャでは、以下の 2 種類の変更リストが使用されます。

- 作成変更リストは、リポジトリの内容をエクスポートした結果です。作成変更リストはリポジトリ全体を表し、インポート操作と昇格操作のソースとして使用されます。
- 比較変更リストは、2 つのリポジトリの内容を比較してターゲットリポジトリに対して行う変更をまとめたものです。比較変更リストは、デザインオブジェクトを昇格するときに使用されます。

関連項目：

- [「リポジトリのエクスポート」 \(ページ 55\)](#)

変更リスト XML ファイル

変更リストは拡張子が *.change.xml の XML 形式で格納されます。生成された XML ファイルは、その後、確認したり、編集したり、ターゲットリポジトリに適用したりすることができます。変更リスト XML ファイルは、設定のバックアップとしてアーカイブしたり、設定変更を管理するためにソース制御システムに格納したりすることもできます。

関連項目：

- [「変更リストリファレンス」 \(ページ 69\)](#)

リアルタイムのメタデータ管理

外部アプリケーションは、以下の Services Integration Framework (SIF) 要求を使用してメタデータを管理できます。

タスク	メソッド	説明
検証	validateMetadata	指定されたリポジトリのメタデータを検証します。
エクスポート	getOrsMetadata	指定されたリポジトリのメタデータを取得します。
変更リスト管理	applyChangeList	指定された変更リストを指定されたターゲットリポジトリに適用します。
	createChangeList	2 つのリポジトリを比較し、比較変更リスト XML ファイルを作成します。
	validateChangeList	指定された変更リストを指定されたターゲットリポジトリに適用するシミュレーションを実行します。

詳細については、*Multidomain MDM サービスの統合フレームワークガイド*および *Multidomain MDM Javadoc* を参照してください。

メタデータのコピーに関する考慮事項

デザインオブジェクトのインポートおよび昇格時には、以下の問題を考慮します。

大きなリポジトリの移行

リポジトリマネージャ内で大きなリポジトリを移行するときのパフォーマンスを向上させるには、アプリケーションサーバーが実行されているホスト以外のマシンで Hub コンソールを起動します。移行処理は大きなリ

ポジトリでは大量のメモリを消費する可能性があり、アプリケーションサーバーマシンとクライアントマシン両方のメモリが 1GB（またはそれ以上）あることが理想的です。Informatica MDM Hub のシステム要件については、『*Informatica MDM Hub* リリースノート』を参照してください。

階層マネージャの要件

ソースリポジトリとターゲットリポジトリの両方を、階層マネージャのライセンスが同一の Informatica MDM Hub 環境で作成する必要があります。

- 一方のリポジトリは階層マネージャのライセンスのある Informatica MDM Hub 環境で作成され、他方のリポジトリは階層マネージャのライセンスのない Informatica MDM Hub 環境で作成された場合、リポジトリ間で変更をコピーすることはできません。両方の環境のライセンスは同じである必要があります。つまり、両方に階層マネージャのライセンスがあるか、どちらにも階層マネージャのライセンスがないかです。
- リポジトリ間で階層マネージャのメタデータをコピーするには、両方のリポジトリが階層マネージャのライセンスのある Informatica MDM Hub 環境で作成されている必要があります。

これらのライセンス要件に加えて、ソースリポジトリとターゲットリポジトリの階層マネージャが有効になっている必要があります。階層マネージャを有効にするには、Hub コンソールで階層ツールを開き、リポジトリを選択して、リポジトリベースオブジェクト（RBO テーブル）およびそれらに関連付けられているクエリと HM パッケージの作成を求めるメッセージに従います。詳細については、*Multidomain MDM* の設定ガイドを参照してください。

注: リポジトリマネージャでは、リポジトリベースオブジェクト（RBO テーブル）の昇格、エクスポート、またはインポートは行われません。ソース ORS データベースとターゲット ORS データベースに異なるテーブルスペースで作成された RBO テーブルがある場合、リポジトリマネージャに競合が表示されます。ソース RBO テーブルとターゲット RBO テーブルが同じインデックスとデータスペース名で作成されていることを確認してください。

Java クレンジングアダプタ

ファイルシステムに格納されている Java クレンジングアダプタは、ターゲットリポジトリに物理的にコピーされるわけではありません。メタデータ内の Java クレンジングライブラリへの参照のみがコピーされます。ただし、クレンジングアダプタは動的なライブラリであるため、メタデータは変更になる可能性があります。原因としては、外部コンフィギュレーションファイルが変更された場合や、ユーザーがクレンジングアダプタの新機能にアクセスした場合などが考えられます。

インポート処理および昇格処理では、ソース環境とターゲット環境に同じクレンジングアダプタが設定されているものと仮定されます。

ユーザーイグジット

データベースに保存されているユーザーイグジットは、物理的にターゲットリポジトリにコピーされません。

ユーザー情報

ユーザー情報はコピーされません。ユーザー情報には次が含まれます。

- ユーザーアカウント（ユーザーツール）
- ユーザーグループとユーザーアカウントメンバーシップ（ユーザーとグループツール）
- データベースへのユーザーアカウントの割り当て（ユーザーとグループツール）
- ユーザー/ロールの割り当て（ロールツール）

詳細については、*Multidomain MDM* の設定ガイドを参照してください。

一致用のポピュレーション

Informatica MDM Hub の実装において、実装用に US 以外のポピュレーションが複数使用されている場合、ソースリポジトリとターゲットリポジトリの間でデザインオブジェクトのコピーを試みる前に、必要なすべてのポピュレーションをソースリポジトリとターゲットリポジトリの両方で有効にしておく必要があります。詳細については、*Multidomain MDM の設定ガイド*を参照してください。

プロキシユーザーが登録されているリポジトリ

ORS にプロキシユーザーが登録されている場合、（昇格またはインポートによって）変更リストの変更を適用するときには、ORS スキーマ所有者のパスワードが必要です。

この場合、リポジトリマネージャにより、要求した操作を続行する前に所有者パスワードの入力を求めるメッセージが表示されます。

変更結果の監視

デザインオブジェクトをインポートしたり、ターゲットリポジトリに変更の昇格を適用したりすると、リポジトリマネージャによって Hub Store 内の以下のログテーブルに変更リストの実行結果が格納されます。

Hub Store のテーブル	説明
C_REPOS_MET_CHANGE_EXEC	変更リストの実行ごとのログ。結果の実行ステータスコードやエラーの説明など（該当する場合）。
C_REPOS_MET_CHANGE_EXEC_ITEM	変更リストで実行された個別のアクションごとのログ。関係する項目や、その項目に対して取られたアクションなど。C_REPOS_MET_CHANGE_EXEC の子テーブル。

変更リストの実行中にエラーが発生した場合、処理は停止し、最後に発生した問題が報告されます。問題を修正し、再度変更リストを適用してください。

第 2 章

リポジトリマネージャ入門ガイド

この章では、以下の項目について説明します。

- [概要, 15 ページ](#)
- [リポジトリマネージャの開始, 15 ページ](#)
- [リポジトリマネージャインターフェースのコンポーネント, 15 ページ](#)
- [リポジトリマネージャの操作, 17 ページ](#)

概要

この章では、Hub コンソールのリポジトリマネージャツールを開始および操作する方法について説明します。

リポジトリマネージャの開始

リポジトリマネージャを開始するには、以下の手順を実行します。

1. *Multidomain MDM の設定ガイド*に記載されている「Hub コンソールの基本操作」の説明に従って、Hub コンソールを起動します。
2. Hub コンソールで、マスターデータベースに接続します。
3. Hub コンソールで設定ワークベンチを展開し、**リポジトリマネージャ**をクリックします。
Hub コンソールにリポジトリマネージャツールが表示されます。

リポジトリマネージャインターフェースのコンポーネント

この節では、リポジトリマネージャインターフェースのコンポーネントについて説明します。

タブ

リポジトリマネージャツールには、以下のタブがあります。

タブ	説明
検証	Informatica MDM Hub の実装内でリポジトリのメタデータを検証するために使用します。メタデータの検証プロセスは、リポジトリを説明するメタデータの完全性と整合性を検証することを目的としています。
昇格	リポジトリ間でデザインオブジェクトを昇格するために使用します。
インポート	デザインオブジェクトを空のリポジトリにインポートするために使用します。
エクスポート	リポジトリをエクスポートするために使用します。

関連項目：

- [「メタデータの検証」 \(ページ 18\)](#)
- [「リポジトリ間での変更の昇格」 \(ページ 28\)](#)
- [「デザインオブジェクトのインポート」 \(ページ 48\)](#)
- [「リポジトリのエクスポート」 \(ページ 55\)](#)

コマンドボタン

コマンドボタンは、現在のタブで適用可能な操作を実行するために使用します。タブごとのボタンのリストについては、以下を参照してください。

タブ	説明
検証	「[検証] タブのコマンドボタン」 (ページ 20)
昇格	「[視覚] タブのコマンドボタン」 (ページ 33) 「[変更リスト] タブのコマンドボタン」 (ページ 41)
インポート	「[インポート] タブのコマンドボタン」 (ページ 50)
エクスポート	「[エクスポート] タブのコマンドボタン」 (ページ 56)

リポジトリリスト

リポジトリマネージャのすべてのタブには、Informatica MDM Hub のこの実装のマスターデータベースで定義された、リポジトリ（ORS データベース）のリストがあります。ORS に対して以前に検証プロセスが実行されている場合、ORS 名の横にあるアイコンにそのリポジトリの検証の有無が示され、検証されている場合は、最新の検証結果（成功したか失敗したか）が示されます。

関連項目：

- [「検証インジケータ」 \(ページ 20\)](#)

リポジトリマネージャの操作

この節では、リポジトリマネージャの操作方法について説明します。

自動排他ロック

リポジトリマネージャは、昇格またはインポート中にターゲットリポジトリに変更を適用する前に、他のユーザーがターゲットリポジトリに対してかけているロックを自動的に解除し、変更の適用中にはターゲットリポジトリに排他ロックをかけます。メタデータの変更中に *排他ロック* がかかると、Hub コンソールで他のすべてのユーザーがターゲットリポジトリに変更を加えることができなくなります。

変更が完了すると、リポジトリマネージャはターゲットリポジトリに対する排他ロックを自動的に解除します。Hub コンソールでのロックの詳細については、*Multidomain MDM の設定ガイド* の「Hub コンソールの基本操作」を参照してください。

注: プロダクションモードでない（データベースツールの「データベースのプロパティ」タブでこの ORS に対して「プロダクションモード」チェックボックスが選択されていない）ORS に書き込みロックを設定することはできません。詳細については、*Multidomain MDM の設定ガイド* の「オペレーショナル参照ストアとデータソースの設定」を参照してください。

グラフィカルモデルビューでのスキーマの表示

スキーマビューアツールを使用すると、ORS であっても、XML ファイルからロードされたものであっても、あらゆるリポジトリのデータモデルのグラフィカルビューを表示できます。このツールには、リポジトリマネージャのさまざまなタブからアクセスできます。スキーマビューアは、複雑なスキーマを視覚化したり、スキーマ内のベースオブジェクトや外部キー関係をグラフィカルに表示したりする際に特に役立ちます。スキーマビューアでは、既存のスキーマに変更を適用したり、既存のスキーマのメタデータを変更リストにエクスポートしたりする前に、既存のスキーマを視覚的に確認することができます。

スキーマビューアを起動する手順

- ▶ 「検証」、[昇格]、または「エクスポート」タブで、**「スキーマビューア」** ボタンをクリックします。Hub コンソールにスキーマビューアが表示されます。
このツールの使用方法の詳細については、*Multidomain MDM の設定ガイド* を参照してください。

第 3 章

メタデータの検証

この章では、以下の項目について説明します。

- [概要, 18 ページ](#)
- [メタデータ検証プロセスについて, 18 ページ](#)
- [オペレーショナルリファレンスストアのメタデータ検証, 21 ページ](#)
- [メタデータ修復プロセス, 26 ページ](#)
- [リポジトリ内のメタデータの修復, 27 ページ](#)

概要

この章では、Hub コンソール内のリポジトリマネージャを使用して、Informatica MDM Hub の実装内でリポジトリのメタデータを検証する方法について説明します。

メタデータ検証プロセスについて

メタデータの検証プロセスは、リポジトリを説明するメタデータの完全性と整合性を検証することを目的としています。

論理モデルと物理スキーマの一致

ORS では、メタデータ（論理モデル）が物理スキーマ（テーブルおよびカラム）と正確に一致する必要があります。メタデータ定義ごとに対応する物理コンポーネントが必要であり、物理コンポーネントごとに対応するメタデータ定義が必要です。

ただし、特定のイベントによって、メタデータと物理スキーマの間に不一致が生じる場合があります。例えば、データベース管理者が、Hub コンソールのスキーママネージャを使用せずに、直接データベースに変更を加えた場合などです。同様に、停電やハードウェア障害が原因でデータベースが破損する場合があります。リポジトリマネージャは、このような不一致が Informatica MDM Hub の操作にあまり重要でないか、大きな混乱を引き起こす可能性があるかを判断する手助けとなります。

メタデータ検証プロセス

メタデータ検証プロセスは、以下のタスクを実行します。

- 想定されるメタデータテーブルがすべてスキーマにあり、想定される宣言（カラムのデータ型、精度、およびスケール）がそれらのテーブルに含まれているかどうかを判断する
- 物理スキーマがメタデータと同期しているかどうかを判断する
- メタデータのみをチェックし、メタデータ内で整合性が取れているかどうかを判断する
- 注意が必要な問題（ある場合）のリストを生成する

注: リポジトリマネージャの昇格、インポート、およびエクスポート操作では、重要な問題のない検証済みのリポジトリを必要とします。

関連項目：

- [「問題の重要度レベル」](#)（ページ 20）

メタデータ検証のスコープ

メタデータ検証プロセスは、リポジトリに対して以下の種類のチェックを実行します。

チェック のタイプ	説明
システム	すべてのリポジトリテーブル（C_REPOS_*）、カラム、およびシステムレベルのビューをチェックします。制約（プライマリキー、外部キー、およびインデックス）、シーケンス、およびトリガ（宣言のみ）が対象です。
物理	リポジトリ構造をデータベースのメタデータと比較します。例えば、ベースオブジェクトテーブルの場合、リポジトリメタデータは Oracle の物理メタデータ（テーブル、制約、ビュー、およびシーケンス）と一致している必要があります。
リポジトリ	リポジトリに格納されているメタデータ。

注: リポジトリマネージャでは、検証範囲を絞り込むことができます。

また、リポジトリマネージャによって、インストールされている Informatica MDM Hub ソフトウェアと ORS のバージョンが一致しているかどうかも確認されます。バージョンが一致していない場合は検証を進めることができません。

注: Informatica MDM Hub のサービス統合フレームワーク（SIF）と関連付けられているデザインオブジェクトを検証するには、Hub コンソールの SIF Manager ツールを代わりに使用します。

関連項目：

- [「リポジトリマネージャでサポートされているデザインオブジェクト」](#)（ページ 64）
- [「メタデータの検証」](#)（ページ 23）
- [「問題の重要度レベル」](#)（ページ 20）
- [「階層履歴の表示」](#)（ページ 25）

問題の重要度レベル





検証プロセスでは、リポジトリマネージャによって問題ごとに重要度レベルが割り当てられます。問題は、次の表に示す問題の重要度カテゴリタブの「検出された問題」ペインに表示されます。

重要度	説明
情報	注意を求める情報のみのメッセージ 例:パッケージに基本テーブルのレコードの一部が含まれていない。
警告	Informatica MDM Hub の操作に悪影響を与えない不整合
エラー	通常の昇格、インポート、およびエクスポート操作を正常に完了できなくなる問題。
重大 (FATAL)	リポジトリをロードできないか、すぐに修正しないと他のエラーが発生する可能性のある深刻なエラー。 例:不明なデータ型。

リポジトリマネージャの昇格、インポート、およびエクスポート操作では、エラーまたは重大 (FATAL) な問題のないリポジトリを必要とします。




検証インジケータ










Hub コンソール全体を通して、ORS の横のアイコンは、その ORS が検証済みかどうか、検証済みの場合は最新の検証に問題が発生しているかどうかを示します。

画像	意味
	不明。最初に作成されたとき、または前回リポジトリが変更されたとき以降、ORS が検証されていません。ORS のメタデータで何かに変更されると、その検証インジケータは「不明」に戻ります。
	ORS は問題なく検証済みです。検証プロセスの作成後、リポジトリは変更されていません。
	ORS は検証済みで、警告が発生しています。
	ORS が検証に失敗し、エラーが検出されました。

「検証」タブのコマンドボタン

「検証」タブには以下のコマンドボタンがあります。

ボタン	説明
	選択されたリポジトリを検証します。
	選択されたリポジトリの検証結果を HTML ファイルに保存します。
	選択されたリポジトリの検証履歴を表示します。

ボタン	説明
	選択したリポジトリに対して、スキーマビューを起動します。
	[検出された問題] ペインに修復可能なメタデータエラーを示します。
	[検出された問題] ペインの修復可能なもののフィルタリングを停止します。
	[検出された問題] ペインのすべてのノードを展開します。
	[検出された問題] ペインのすべてのノードを縮小します。
	選択された修復可能なメタデータエラーを修復します。
	検索フィルタフィールドで指定された検索条件に一致しないメタデータエラーを除外し、非表示にします。
	すべてのメタデータエラーを表示します。検索条件に一致するメタデータエラーが強調表示されます。
	検索フィルタフィールドをクリアします。

関連項目：

- [「メタデータの検証」 \(ページ 23\)](#)
- [「検証結果の保存」 \(ページ 25\)](#)
- [「グラフィカルモデルビューでのスキーマの表示」 \(ページ 17\)](#)

オペレーショナルリファレンスストアのメタデータ検証

Hub コンソールのリポジトリマネージャツールを使用、実行して、オペレーショナルリファレンスストア (ORS) のメタデータを検証できます。

昇格操作、インポート操作、またはエクスポート操作を実行する場合は、事前に ORS のメタデータを検証し、すべてのエラーおよび致命的な問題を解決しておく必要があります。また、アップグレードを行う前にも、メタデータを検証し、すべてのエラーおよび致命的な問題を解決しておく必要があります。

デフォルトでは、すべての検証チェックが有効化されていますが、検証対象としてリポジトリの特定の領域を選択することもできます。

以下の表に選択可能な検証チェックを示します。

検証チェック	説明
すべてのシステムチェック	以下のシステムチェックを実行する。 <ul style="list-style-type: none"> - リポジトリテーブル - システムカラム - システムビュー - プライマリキー制約 - 外部キー制約 - インデックスの制約 - シーケンス - トリガシングニチャ - パッケージ
すべての物理チェック	リポジトリ構造をデータベースのメタデータと比較する。例えば、ベースオブジェクトテーブルの場合、リポジトリメタデータはテーブル、制約、ビュー、およびシーケンスの物理データベースメタデータと一致している必要がある。
すべてのリポジトリチェック	リポジトリに格納されているメタデータを検証する

以下の表に選択可能なリポジトリチェックをそれぞれ示します。

検証チェック	説明
リポジトリ	リポジトリを検証する
マッピング	ランディングテーブルとステージングテーブルの間のマッピングを検証する
クレンジング	プロセスサーバー、クレンジング関数、およびクレンジングリストを検証する
クエリ	クエリを検証する
スキーマ	以下のデータモデルデザインオブジェクトを検証する。 <ul style="list-style-type: none"> - ベースオブジェクトテーブル - ベースオブジェクトのカラム - ランディングテーブル - 一致設定 - 外部マッチテーブル - 検証ルール - メッセージキュー - リレーション - ステージングテーブル - カスタムインデックス
信頼	ソースシステムの信頼設定および信頼カラムを検証する
セキュリティアクセスマネージャ	リソースグループ、セキュアリソース、およびロールを検証する
階層マネージャ	エンティティタイプ、リレーションタイプ、階層、パッケージ、プロファイル、サンドボックス、および階層マネージャのメタデータを検証する
その他	サポートされているその他すべてのデザインオブジェクトを検証する

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）リポジトリのメタデータを検証するには、Hub コンソールでリポジトリマネージャツールを使用します。

1. Hub コンソールの**設定**ワークベンチから、**リポジトリマネージャ**ツールを選択します。
2. **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。
3. **【検証するリポジトリを選択】** リストから、リポジトリを選択します。
4. **【検証】** ボタンをクリックします。
5. **【検証チェックの選択】** ダイアログボックスから、実行する検証チェックを選択します。**【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。
6. **【修復】** ボタンをクリックして、修復可能な問題を修正します。
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。

関連項目：

- [「問題の重要度レベル」](#)（ページ 20）
- [「リポジトリマネージャの開始」](#)（ページ 15）
- [「メタデータ検証のスコープ」](#)（ページ 19）
- [「リポジトリマネージャでサポートされているデザインオブジェクト」](#)（ページ 64）

【情報】 ペイン

【情報】 ペインには、リポジトリマネージャによって以下の情報が表示されます。

フィールド	説明
バージョン	Informatica MDM Hub のバージョン。リポジトリのバージョンは、インストールされている Informatica MDM Hub ソフトウェアのバージョンと一致する必要があります。
日付	検証プロセスが開始された日付/時刻。
検証スコープ	検証のスコープ。【検証チェックの選択】ですべてのオプションが選択されている（すべて）か、少なくとも 1 つのオプションが選択されていない（一部）かによって決まります。
サマリ	検証結果のサマリ。

プロパティペイン

リポジトリマネージャのプロパティペインには、検出された問題についてのスクロール可能なレポートが表示されます。このレポートには、以下のカラムが含まれます。

カラム	説明
#	問題の連続番号。
重要度	問題の重要度。
メッセージ	問題の説明。以下の内容が含まれます。 - エラーコード。例:SIP-PV-10312 - 診断テキスト。例:テーブル'C_RBO_HIERARCHY_XREF'のインデックス'SVR1_AF9'が、メタデータにはありますが、データベースにはありません。 注: SIP-PV-10000 代のエラーが発生した場合、リポジトリマネージャがメタデータをロードできていない（例えば、データベースに破損箇所があるなど）ため、リポジトリチェックは実行されていません。このエラーは、システムチェックが失敗したことを意味します。

インポートされたスキーマの事前検証結果

Oracle のエクスポート（ダンプ）ファイルが検証プロセスを経た ORS に対して作成されていて、ダンプファイルに以前の検証結果が含まれている場合に、そのダンプファイルをその後新しい ORS にインポートする（*Multidomain MDM* のインストールガイドの説明に従って、`setup.sql` または `setup_ors.sql` を実行する）と、新しい ORS にはエクスポートされたダンプファイルの前回の検証結果が含まれます。

新しい ORS を検証すると、リポジトリマネージャによって、2 つのシステムテーブル（`C_REPOS_MET_VALID_MSG` と `C_REPOS_MET_VALID_RESULT`）に新しい結果が付加されます。[履歴] ボタンをクリックすると、過去のすべての検証結果（インポートされたダンプファイルの検証結果を含む）が表示されます。

関連項目：

- [「階層履歴の表示」](#)（ページ 25）

フィルタリングの問題

メタデータの検証結果をフィルタ処理する手順（問題が検出された場合）

1. メタデータの検証プロセスを実行します。
[検出された問題] ペインの [重大 (FATAL)]、[エラー]、[警告]、および [情報] の問題カテゴリタブに、メタデータエラーが表示されます。
2. 問題のタブをクリックして、[重大 (FATAL)]、[エラー]、[警告]、[情報] などの問題カテゴリを選択します。
選択した問題カテゴリの検出された問題が表示されます。
3. 選択した問題カテゴリの検証エラーメッセージをフィルタリングするために、フィルタフィールドに文字列のフィルタ条件を入力します。
フィルタリングされたエラーメッセージは [検出された問題] ペインで強調表示されます。
4. **[非表示]** ボタンをクリックして、フィルタ条件に一致しないエラーメッセージを非表示にします。
[非表示] ボタンと [修復可能なもののフィルタリングを停止] または [修復可能なもののみ表示] は同時に使用できます。2 つのボタンを同時にクリックすると、それらは AND 演算として機能します。

関連項目：

- [「メタデータの検証」 \(ページ 23\)](#)

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。

関連項目：

- [「メタデータの検証」 \(ページ 23\)](#)

階層履歴の表示

検証履歴を表示する手順

1. メタデータの検証プロセスを実行します。
2. **[履歴]** ボタンをクリックします。

リポジトリマネージャに、**[検証履歴]** ウィンドウが表示されます。

[検証履歴] ウィンドウの各行は、1 回の検証プロセスの実行結果を表しています。**[検証履歴]** ウィンドウには、以下のカラムが表示されます。

カラム	説明
日付	検証プロセスが実行された日付と時刻。
ユーザー	検証プロセスを開始したユーザー。
致命的なエラーの数	検証中に発生した、重要度レベルが [致命的] のエラーの数。
エラーの数	検証中に発生した、重要度レベルが [エラー] のエラーの数。
警告の数	検証中に発生した、重要度レベルが [警告] のエラーの数。
情報の数	検証中に発生した、重要度レベルが [情報] のエラーの数。
スコープ	検証のスコープが [すべて] であるか [一部] であるかを指定します。

関連項目：

- [「メタデータ検証のスコープ」 \(ページ 19\)](#)
- [「メタデータの検証」 \(ページ 23\)](#)
- [「問題の重要度レベル」 \(ページ 20\)](#)

検証結果の表示

[検証履歴] ウィンドウから検証結果を表示する手順

1. [検証履歴] ウィンドウで、表示する検証結果を選択します。
2. **【表示】** ボタンをクリックします。
リポジトリマネージャに、[検証結果の表示] ウィンドウが表示されます。
3. 必要に応じて、以下のことが可能です。
 - 結果をフィルタ処理します。
 - **【保存】** ボタンをクリックして結果を保存します。
 - リストから問題を選択し、**【推奨事項】** ボタンをクリックして、その問題に対する推奨事項を確認します。
4. **【閉じる】** をクリックします。

関連項目：

- [「検証結果の保存」 \(ページ 25\)](#)

検証履歴ログ内のエントリの削除

[検証履歴] ウィンドウから検証結果を削除する手順

1. [検証履歴] ウィンドウで、表示する検証結果を選択します。
2. **【削除】** ボタンをクリックします。
リポジトリマネージャから、削除の確定を求められます。
3. **【はい】** をクリックします。
4. **【閉じる】** をクリックします。

メタデータ修復プロセス

この節では、Informatica MDM Hub の実装でリポジトリ（ORS）のメタデータを修復する方法について説明します。

MDM Hub には、リポジトリマネージャによって報告される一部の検証エラーの自動修復の内部フレームワークが用意されています。修復可能なメタデータ検証エラーのバッチ修復または選択的修復を実行できます。メタデータエラーは、それに依存するメタデータの検証を妨げたり、非表示のメタデータ検証エラーが発生したりする可能性があります。非表示のメタデータ検証エラーは、それらに関連するメタデータエラーが修復された後で明らかになります。

注:

- 修復できるのは、修復アイコンの付いたメタデータ検証エラーのみです。
- 修復機能を使用できるのは、管理者ロールのあるユーザーのみです。

メタデータ修復結果

リポジトリのメタデータを修復すると、リポジトリマネージャによって、C_REPOS_MET_VALID_MSG システムテーブルに新しい結果が付加されます。

メタデータの修復の結果は、C_REPOS_MET_VALID_MSG システムテーブルの次のカラムに格納されます。

- REPAIR_IND
- REPAIR_FAIL_MESSAGE

リポジトリ内のメタデータの修復

メタデータエラーを修復するには、次の手順を使用する必要があります。

1. メタデータを検証します。
問題が検出されると、[検出された問題] ペインに表示されます。修復可能なエラーは、[重大 (FATAL)]、[エラー]、[警告]、および [情報] の問題カテゴリタブに、修復アイコンとともに表示されます。
2. 問題のタブをクリックして、[重大 (FATAL)]、[エラー]、[警告]、[情報] などの問題カテゴリを選択します。
選択した問題カテゴリの検出された問題が表示されます。
3. **【修復可能なもののみ表示】** ボタンをクリックして、修復可能なメタデータエラーを表示します。
修復できないメタデータ検証エラーのヘルプについては、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。
4. [検出された問題] ペインからエラーを選択します。
 - 修復可能な検証エラーのバッチ修復を行う場合は、検証エラーコードを選択します。
 - 修復可能な検証エラーの選択的修復を行う場合は、検証エラーコードの下に一覧表示されている検証エラーから一度に 1 つずつエラーを選択します。
5. **【修復】** ボタンをクリックします。
実行されるメタデータ修復操作の警告とともに、修復ダイアログが表示されます。
6. 求められたメタデータ修復操作を実行する必要がある場合は、**【OK】** をクリックします。
メタデータは自動的に修復されます。
求められたメタデータ修復操作を実行しない場合は、**【キャンセル】** をクリックします。
修復が成功すると、修復された検証エラーは修復されたものとして特徴のあるアイコンでマークが付けられます。
7. 修復が失敗した場合は、**【修復】** ボタンをクリックしてもう一度実行します。
注: 修復は、他の問題によってブロックされる可能性があるため、失敗する場合があります。修復をブロックしている問題を見つけて修正し、もう一度実行する必要があります。

関連項目：

- [「メタデータの検証」 \(ページ 23\)](#)

第 4 章

リポジトリ間での変更の昇格

この章では、以下の項目について説明します。

- [リポジトリ間での変更の昇格の概要, 28 ページ](#)
- [変更の視覚的昇格, 32 ページ](#)
- [変更リストを使用した変更の昇格, 40 ページ](#)
- [コマンドラインからの変更の昇格, 46 ページ](#)

リポジトリ間での変更の昇格の概要

Hub コンソールのリポジトリマネージャを使用して、MDM Hub の実装内のリポジトリ間で差分変更を移動できます。

リポジトリマネージャでは、[昇格] タブでリポジトリ間の変更を昇格できます。昇格によって、差分変更がリポジトリ間でコピーされます。差分変更は次の種類の操作を伴う場合があります。

新しいデザインオブジェクトの挿入

新しいデザインオブジェクトを空のターゲットリポジトリに挿入するには、[第 5 章, 「デザインオブジェクトのインポート」 \(ページ 48\)](#)に説明されているインポートプロセスを実行します。

ソースリポジトリとターゲットリポジトリでの同名のデザインオブジェクトに対するメタデータの更新

ただし、ターゲットリポジトリにデザインオブジェクトがすでに含まれている場合、リポジトリ間の変更を昇格する必要があります。リポジトリマネージャを使用して、属性値の相違など、デザインオブジェクトへの変更をソースリポジトリからターゲットリポジトリに昇格できます。例えば、本番環境のリポジトリの Party ベースオブジェクトを、開発環境のリポジトリの変更を使用して更新できます。変更には、ベースオブジェクトのプロパティ設定、カラム定義、またはマッピングの差分が含まれる場合があります。

変更の昇格の前後にリポジトリマネージャによってデータの整合性の検証が実行されるかどうかを選択できます。データの整合性の検証では、一意制約と外部キーがチェックされます。

注: Oracle リポジトリと IBM DB2 リポジトリの間など、異種データベースのリポジトリ間で変更を昇格することはできません。

昇格のシナリオ

この節では、昇格が使用される一般的なシナリオについて説明します。

同期化昇格

同期化昇格は、開発用リポジトリとテスト用リポジトリ間、テスト用リポジトリとプロダクションリポジトリ間など、リポジトリ間でデザインオブジェクトを同期するために使用されます。

同期化昇格は、同一のリポジトリを作成するためにも使用できます。このシナリオの同期化昇格では、変更リストの中間 XML ファイルが使用されます。変更リスト XML ファイルには、ターゲットリポジトリに適用する変更のリストが含まれます。変更リスト XML ファイル内のすべての変更がターゲットリポジトリに適用されます。このシナリオでは、リポジトリマネージャの「変更リスト」タブを使用します。

関連項目：

- [「変更リスト」 \(ページ 11\)](#)
- [「変更リストを使用した変更の昇格」 \(ページ 40\)](#)

選択的昇格

分散開発環境では、開発者は、リポジトリマネージャツールを使用して、別個のものであるが類似した実装リポジトリ間でデザインオブジェクトを共有したり再利用したりすることができます。例えば、開発者は、データモデリング、SQL コーディング、およびアプリケーション統合に別個のリポジトリを使用できます。

選択した変更のリストは、事前に変更を確認、編集、および承認できるように、適用前に比較変更リスト XML ファイルに保存できます。承認された変更は、中央のマスタリポジトリにプロパゲートできます。このシナリオでは、リポジトリマネージャの「視覚」タブを使用します。

関連項目：

- [「変更の視覚的昇格」 \(ページ 32\)](#)

選択的昇格と同期化昇格の違い

選択的昇格は同期化昇格よりも細かな設定が可能で、昇格させるデザインオブジェクトを個別に選択できます。同期化昇格では、変更リスト XML 全体をターゲットリポジトリに適用する必要があります。ただし、必要に応じて事前に変更リスト XML ファイルを編集し、変更を編集または削除することができます。

昇格可能なデザインオブジェクト

昇格可能なデザインオブジェクトの完全なリストについては、[「リポジトリマネージャでサポートされているデザインオブジェクト」 \(ページ 64\)](#)を参照してください。

注: サポートテーブルで作成したカスタムインデックスは、移行中にターゲット環境に昇格されません。

オブジェクト昇格時の競合

リポジトリ間でのオブジェクトの昇格を試みると、競合が発生する場合があります。

プロパティの競合

プロパティの競合は、ソースリポジトリとターゲットリポジトリの両方にある同じデザインオブジェクトが異なるプロパティ値を持つ場合の昇格プロセスで発生します。例えば、[履歴を有効にする] プロパティがソースベースオブジェクトとターゲットベースオブジェクト間で異なる顧客ベースオブジェクトで変更を昇格しようとしていたとします。

この場合、リポジトリマネージャでリポジトリを比較し、競合リストを作成することができます。プロパティの各競合に対し、リポジトリマネージャによって競合が強調表示され、保持する値（ターゲットリポジトリ内の既存のプロパティ、または昇格させようとしてるベースオブジェクト内のプロパティ）の選択など、適切な

アクションを求めるメッセージが表示されます。リポジトリマネージャによってプロパティパネルが表示され、プロパティ値の比較が横に並べて示されます。

関連項目：

- [「選択したデザインオブジェクトのプロパティパネル」](#) (ページ 36)

依存関係の競合

依存関係の競合は、ソースデザインオブジェクトとターゲットデザインオブジェクトが異なる子デザインオブジェクトの集まりを持つ場合の昇格プロセスで発生します。

例えば、Person ベースオブジェクトの変更を昇格しようとしていて、ソースリポジトリとターゲットリポジトリでカラム定義が異なる場合に、依存関係の競合が発生する可能性があります。

この場合、リポジトリマネージャで2つのリポジトリを分析し、競合リストを作成することができます。変更を昇格しようとする、リポジトリマネージャによって依存関係の競合にフラグが設定され、適切なアクションを求めるメッセージが各競合に対して表示されます。

関連項目：

- [「デザインオブジェクトの依存関係」](#) (ページ 66)
- [「リポジトリ間でオブジェクトをコピーしたときの競合」](#) (ページ 11)

競合を解決するためのアクション

変更をターゲットリポジトリに適用するプロセスを開始すると、リポジトリマネージャによって、プロパティの競合と依存関係の競合が発生した場合にその処理方法を指定するように求めるメッセージが表示されます。

以下のいずれかのアクションを選択します。

- **マージ**

- 2つのデザインオブジェクトを手動でマージし、ソースデザインオブジェクトとターゲットデザインオブジェクトの両方を組み合わせたものを作成する

注: 手動マージは、ターゲットオブジェクトに、一部のプロパティ値はソースから、他のプロパティ値はターゲットから取得したい場合に便利です。例えば、開発環境からプロダクション環境にベースオブジェクトを昇格する場合に、プロダクション環境のパフォーマンスに影響を与える可能性のある特定のプロパティ設定（バッチサイズなど）を昇格から除外したいという場合があります。

- ソースデザインオブジェクトを保持し、ターゲットデザインオブジェクトを破棄する
- ターゲットデザインオブジェクトを保持し、ソースデザインオブジェクトを破棄する

- **置換**

- ターゲットデザインオブジェクトをソースデザインオブジェクトで完全に置換する（ターゲットを上書きする）

注:

- [置換] オプションはプロパティの競合とオブジェクトの競合の両方に適用されるのに対して、[マージ] オプションはプロパティの競合のみに適用されます。
- 削除されたオブジェクト（ベースオブジェクト、ステー징テーブル、カラムなど）を昇格させるときに競合を解決するには、[ターゲットオブジェクトをソースオブジェクトで完全に置換する] オプションを選択します。

またリポジトリマネージャは、ランディングテーブルやステー징テーブル内のカラムに依存するマッピングなど、より複雑な依存関係の関連するデザインオブジェクトにフラグを設定します。

関連項目：

- [「ターゲットリポジトリへの変更の適用」 \(ページ 39\)](#)
- [「ターゲットリポジトリへの変更リストの適用」 \(ページ 45\)](#)

昇格プロセスに関する考慮事項

変更を昇格する前に、以下の事項を考慮します。

- 変更を昇格する前に、ターゲットリポジトリのバックアップを作成することを検討します。
- 昇格は、ターゲットリポジトリの変更のみを伴います。ソースリポジトリは変更されません。
- リポジトリマネージャの昇格操作では、エラーまたは重大な問題のない検証済みのリポジトリを必要とします。
- ユーザーカラム（システムカラムではなく）として定義されている `deleted_*` カラムは、他のユーザーカラムと同様に昇格することができます。
- 昇格プロセスでカラムの長さを短くすることはできません。例えば、ソースリポジトリとターゲットリポジトリの両方に `varchar(50)` カラムがある場合、ソースリポジトリでカラムを `varchar(20)` にし、その変更をターゲットリポジトリに昇格しようとする、リポジトリマネージャによってエラーが生成されます。
- `ROWID_SYSTEM` の値がソースリポジトリとターゲットリポジトリで一致している必要があります。これらの値が一致していない場合は、昇格操作を試みる前に、ソースリポジトリとターゲットリポジトリの `C_REPOS_SYSTEM` 内のシステム行 ID を同期させる必要があります。ソースリポジトリの行 ID を使用して、同じシステムのソースとターゲットが同じ行 ID になるようにターゲットリポジトリを更新します。
- リポジトリ間の変更を昇格する前に、ソースリポジトリとターゲットリポジトリの両方に、必要なすべてのテーブルスペースに対して十分な特権があることを確認します。
- リポジトリマネージャでは、ORS でのデザインオブジェクトの移行はサポートされますが、マスターデータベースでのデザインオブジェクトの移行はサポートされません。これにはユーザーアカウント、ユーザーグループ、ユーザーアカウントのデータベースへの割り当て、およびユーザーまたはロールの割り当てが含まれます。ターゲットスキーマで、ユーザー情報を手動で同期する必要があります。
- カスタムクエリに基づくパッケージを昇格する場合、リポジトリマネージャでは基になるカスタムクエリの正確さは保証されません。昇格後にパッケージが無効の場合、Hub コンソールのパッケージツールを使用してパッケージを再度保存する必要があります。
- カスタムクエリでカスタムテーブルが使用される場合、そのカスタムテーブルを昇格することはできません。カスタムクエリの移行後、ターゲットリポジトリにカスタムテーブルを作成する必要があります。
- リポジトリ間で、ビジネスエンティティに関連付けられている保存済みクエリを昇格すると、リポジトリマネージャはクエリのみを昇格します。クエリに関連付けられているユーザー情報は、ターゲットリポジトリに昇格されません。昇格後に、ターゲットリポジトリ内の保存済みクエリに、ユーザーを再割り当てする必要があります。
- Elasticsearch が設定された環境では、データへのインデックス付けが終了した後に、あるフィールドの検索可能なプロパティへの変更を昇格した場合、インデックスは削除されます。データを再び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。
- リポジトリ間で一致ルールの変更を昇格すると、関連付けられている一致テーブルをリセットする必要がある場合があることを示す変更警告がリポジトリマネージャによって生成されます。
- 無効な設定を昇格しようとする、リポジトリマネージャによってフラグが設定される場合があります。例えば、信頼設定がターゲットリポジトリのカラムに対して設定されているが、信頼がソースリポジトリのそのカラムに対して無効な場合、リポジトリマネージャによって無効のフラグが設定されます。変更リストを適用すると、このオブジェクトは昇格されません。視覚的に昇格する場合は、**「ターゲットオブジェクトをソースオブジェクトで完全に置換する」** オプションを使用して昇格することができます。

- 既定の日付の昇格が成功するようにするには、日付カラムの既定の日付の値を設定します。クレンジング関数を使用して、ステージプロセス中にデフォルト値を割り当てることをお勧めします。
- 一意のカラムを移行するには、デフォルト値を指定するようにカラムを設定する必要があります。詳細については、『*Multidomain MDM の設定ガイド*』を参照してください。
- ORS に対してプロダクションモードが有効になっている場合、遷移モードを有効にする必要があります。遷移モードでは、リポジトリマネージャの昇格アクションを実行できます。詳細については、『*Multidomain MDM の設定ガイド*』を参照してください。
- Oracle リポジトリで昇格が成功するようにするには、Unicode 対応データベースを使用します。データベースの文字セットを確認するには、次のコマンドを入力します。

```
SQL> select * from v$nls_parameters where parameter Like '%CHARACTERSET';
```

階層マネージャのメタデータなど、その他の考慮事項については、[「メタデータのコピーに関する考慮事項」\(ページ 12\)](#)を参照してください。

関連項目：

- [「問題の重要度レベル」\(ページ 20\)](#)
- [「メタデータのコピーに関する考慮事項」\(ページ 12\)](#)
- [「リポジトリのエクスポート」\(ページ 57\)](#)

カラム長の短縮

カラムの長さを短くする場合は、カラムを削除した上で、変更リストを使用して再作成する必要があります。

1. ソースリポジトリで、長すぎるカラムを削除します。
2. 変更リストを作成します。
3. ターゲットリポジトリに変更リストを適用します。
4. ソースリポジトリで、短いカラムを追加します。
5. 変更リストを作成します。
6. ターゲットリポジトリに変更リストを適用します。

変更の視覚的昇格

この節では、リポジトリ間で視覚的に昇格する方法について説明します。

関連項目：

- [「選択的昇格」\(ページ 29\)](#)

視覚的昇格タスクの概要

注: 変更を昇格する前に、ターゲットリポジトリのバックアップを作成します。

変更を視覚的に昇格する手順

1. **【昇格】 / 【視覚】** タブに移動します。
2. ソースリポジトリを選択します。

3. ターゲットリポジトリを選択します。
4. ソースリポジトリから昇格するデザインオブジェクトを選択して昇格し、必要に応じて競合を解決します。
5. 必要であれば、昇格した変更を変更リスト XML ファイルに保存できます。
6. ターゲットリポジトリに変更を適用します。

関連項目：

- [「リポジトリのエクスポート」](#) (ページ 57)
- [「\[昇格\] / \[視覚\] タブへの移動」](#) (ページ 33)
- [「視覚的昇格のターゲットリポジトリの選択」](#) (ページ 34)
- [「変更のターゲットリポジトリへの視覚的昇格」](#) (ページ 36)
- [「比較変更リストファイルの変更の保存」](#) (ページ 39)
- [「ターゲットリポジトリへの変更の適用」](#) (ページ 39)

「昇格」 / 「視覚」 タブへの移動

変更を視覚的に昇格する手順








1. リポジトリマネージャツールを起動します。
 2. 「昇格」タブをクリックします。
- この節の残りのタスクを完了します。

関連項目：

- [「リポジトリマネージャの開始」](#) (ページ 15)

「視覚」 タブのコマンドボタン

「視覚」タブには以下のコマンドボタンがあります。

ボタン	説明
	ソースリポジトリ内の選択したデザインオブジェクトをターゲットリポジトリに昇格します。
	デザインオブジェクト階層をマークアップとともに表示します。
	デザインオブジェクト階層内の前の競合に移動します。
	デザインオブジェクト階層内の次の競合に移動します。
	変更リストを保存します。
	ターゲットリポジトリに変更を適用します。
	選択したリポジトリに対して、スキーマビューアを起動します。

視覚的昇格のソースリポジトリの選択

ソースリポジトリには、データベース（ORS）または変更リストの XML ファイルのいずれかを選択できます。

昇格のソースリポジトリを選択する手順

1. [ソース] ドロップダウンリストの横にある **【選択】** ボタンをクリックします。
リポジトリマネージャに、[リポジトリを開く] ウィンドウが表示されます。
2. リストからソースリポジトリを選択します。
 - データベースリポジトリの場合は、リストからソースリポジトリを選択します。
まだ検証されていないデータベースリポジトリを選択した場合は、**【検証】** ボタンをクリックし、検証プロセスを完了します。
注: リポジトリマネージャでは、エラーまたは重大（FATAL）な問題のないことが検証済みのリポジトリのみを使用できます。
【OK】 をクリックします。
リポジトリマネージャによってソースリポジトリがロードされます。
 - 変更リスト XML ファイルの場合は、**【ファイルリポジトリ】** タブをクリックします。
【開く】 ボタンをクリックします。
変更リスト XML ファイルを選択（必要に応じてフォルダーに移動）し、**【開く】** をクリックします。
注: 比較の基準として完全なスキーマを表す作成変更リストを選択する必要があります。リポジトリマネージャでは、比較変更リストは選択できません。
リポジトリマネージャによって、選択したファイルからリポジトリがロードおよび検証されます。
【OK】 を選択します。
3. リポジトリマネージャによって階層ツリー（デザインオブジェクト階層）として表示された、ロードされたソースリポジトリを確認します。

関連項目：

- [「オペレーショナルリファレンスストアのメタデータ検証」](#)（ページ 21）
- [「問題の重要度レベル」](#)（ページ 20）

視覚的昇格のターゲットリポジトリの選択

昇格のターゲットリポジトリを選択する手順

1. **【ターゲット】** ドロップダウンリストをクリックします。
2. リストからリポジトリを選択します。
注: リポジトリマネージャでは、致命的なエラーのないことが検証済みのリポジトリのみを使用できます。
リポジトリマネージャによってターゲットリポジトリがロードされます。
リポジトリマネージャに、ロードされたターゲットリポジトリが表示されます。

関連項目：



- [「オペレーショナルリファレンスストアのメタデータ検証」](#)（ページ 21）
- [「問題の重要度レベル」](#)（ページ 20）

視覚的昇格でのデザインオブジェクト階層内の移動

ソースリポジトリとターゲットリポジトリのどちらのデザインオブジェクト階層にも、リポジトリで定義されたデザインオブジェクトの集まりが含まれています。階層の最上位は、各デザインオブジェクトタイプです。



デザインオブジェクト階層の展開と縮小

次のボタンをクリックすると、デザインオブジェクト階層のレベルを展開または縮できます。

ボタン	説明
	デザインオブジェクト階層内のツリーを展開します。
	デザインオブジェクト階層内のツリーを縮小します。

ソースリポジトリ内の競合アイコン

ソースデザインオブジェクトの階層にあるアイコンは、その横のデザインオブジェクトまたは関連付けられている一連の子オブジェクトに関する追加情報を表しています。ソースデザインオブジェクトの階層には、以下のアイコンが表示される可能性があります。

アイコン	説明
	<p>このデザインオブジェクトまたは関連付けられている一連の子オブジェクトのソースリポジトリとターゲットリポジトリの間で相違があります。以下に例を示します。</p> <p>関連付けられている一連の子オブジェクトに、新しいデザインオブジェクトまたは異なるデザインオブジェクトが含まれています。</p> <p>デザインオブジェクトはソースリポジトリとターゲットリポジトリの両方に存在しますが、1つ以上のプロパティがソースリポジトリとターゲットリポジトリで異なります。</p>
	<p>デザインオブジェクトは一方のリポジトリには存在しますが、他方のリポジトリには存在しません。</p> <p>このアイコンがソースリポジトリ内のデザインオブジェクトの横に表示された場合、そのデザインオブジェクトはターゲットリポジトリには存在しません。デザインオブジェクトが昇格されて変更が適用されると、デザインオブジェクトはターゲットリポジトリに追加されます。</p> <p>このアイコンがターゲットリポジトリ内のデザインオブジェクトの横に表示された場合、そのデザインオブジェクトはソースリポジトリには存在しません。デザインオブジェクトをそのままにしておくか、削除するかを決定できます。</p>

マークアップモードではこの他のアイコンも表示される場合があります。

関連項目：

- [「マークアップを伴う表示」 \(ページ 38\)](#)
- [「リポジトリ間でオブジェクトをコピーしたときの競合」 \(ページ 11\)](#)

選択したデザインオブジェクトのプロパティパネル

デザインオブジェクト階層でデザインオブジェクトを選択すると、リポジトリマネージャのプロパティパネルに、そのデザインオブジェクトと関連付けられたプロパティが表示されます。プロパティパネルには、以下のカラムが含まれます。

カラム	説明
プロパティ	選択したプロパティの名前。
ソースの値	選択したプロパティのソースリポジトリでの値。
ターゲットの値	選択したプロパティのターゲットリポジトリでの値。
最終結果	変更が適用された後にターゲットリポジトリに格納される値。

注: 階層マネージャで使用されるカスタムクレンジング Java ライブラリまたはアイコンなどのバイナリファイル内の違いは、リポジトリマネージャでは検出できません。これらのオブジェクトについては、ユーザー自身の判断でデザインオブジェクトを移行してください。

コンテキストメニュー

デザインオブジェクト階層内のデザインオブジェクトを右クリックすると、利用可能な操作のポップアップメニューが表示されます。ソースリポジトリでは、**【昇格】** を選択できます。

マークアップモードでは、ターゲットリポジトリで **【元に戻す】** を選択でき、前に昇格した値を前の値に戻すことができます。

複数のデザインオブジェクトの選択

デザインオブジェクト階層内で複数のデザインオブジェクトを選択して、1 回の操作で移動することができます。

デザインオブジェクト階層内の親デザインオブジェクト（ベースオブジェクトなど）を選択すると、そのすべての子オブジェクト（カラム、一致設定、ステー징テーブルなど）が自動的に選択されます。

関連するデザインオブジェクト

デザインオブジェクトをターゲットリポジトリに昇格するときに、関連するデザインオブジェクトも昇格したい場合があります。これにはデザインオブジェクト階層内の子オブジェクトや、デザインオブジェクトツリー内の他で関連付けられているオブジェクトが考えられ、デザインオブジェクトのタイプが異なる場合もあります。例えば、マッピングはランディングテーブルとステー징テーブルに依存します。以下の例は、ベースオブジェクトの LAST_NAME カラムに対して信頼関係が設定されていることを示しています。

変更のターゲットリポジトリへの視覚的昇格

ソースリポジトリからターゲットリポジトリに変更を昇格する手順

1. ソースリストで、昇格するデザインオブジェクトまたはプロパティを選択します。
2. 次のいずれかを実行します。
 - **【昇格】** ボタンをクリックします。

- 選択したデザインオブジェクトをソースリポジトリからドラッグし、ターゲットリポジトリの任意の場所にドロップします。
- ソースリポジトリを右クリックし、ポップアップメニューから**【昇格】**を選択します。

3. リポジトリマネージャによってターゲットシステムで競合が検出されると、アクションを選択するように求めるメッセージが表示されます。

どのような結果が求められているかを基準にして、対象となるメタデータに最適なアクションを選択します。一部の競合は、ソースとターゲットどちらの値を昇格後の最終結果とするかに基づいて自動的に解決することができます。その他の競合は、選択的な手動修正が必要になる場合があります。例えば、ソースから一部の値を取得し、ターゲットからその他の値を取得する場合などです。

注: プロパティの競合は、変更をターゲットリポジトリに適用する前に自動または手動で解決する必要があります。これに対して、依存関係の競合は変更を適用する前に解決する必要がありません。例えば、ソースリポジトリのベースオブジェクトに追加のカラムがある場合、選択したアクションに関係なくカラムが追加されます。

4. 以下のいずれかのアクションを選択します。

オプション	説明
マージ	
競合を手動でマージする	ソースとターゲットのデザインオブジェクトの値を比較しながら、ターゲットデザインオブジェクトで使用する値を手動で（個別に）選択できます。 「手動競合解決」 （ページ 37）に進んでください。
ソースの値を競合の最終結果として使用する	ターゲットデザインオブジェクトの値をソースデザインオブジェクトの値で上書きします。 「自動競合解決」 （ページ 38）に進んでください。
ターゲットの値を競合の最終結果として使用する	ターゲットデザインオブジェクトの値を保持します。ターゲットデザインオブジェクトは変更されません。 「自動競合解決」 （ページ 38）に進んでください。
置換	
ターゲットオブジェクトをソースオブジェクトで置換する	ターゲットデザインオブジェクトをソースデザインオブジェクトで完全に置換します。ソースデザインオブジェクトに新しいバージョンのデザインオブジェクトが含まれている場合に使用します。 「自動競合解決」 （ページ 38）に進んでください。

関連項目：

- [「手動競合解決」](#)（ページ 37）
- [「自動競合解決」](#)（ページ 38）

手動競合解決

競合を手動でマージすることを選択した場合は、リポジトリマネージャでソースカラムとターゲットカラムの競合の横にチェックボックスが追加されます。

プロパティの競合ごとに、昇格後に維持する値の横にあるチェックボックスを選択してください。

自動競合解決





特定のルール（ソースの値、ターゲットの値、またはターゲットをソースで置換する）に基づいて競合を自動的に解決することを選択した場合は、リポジトリマネージャに Impact Analyzer が表示されて、選択内容による影響が示されます。

影響を許容できる場合は、[OK] をクリックします。

マークアップを伴う表示



[マークアップ] ボタンをクリックすると、ターゲットリポジトリで提案された変更を示すアイコンの表示/非表示が切り替わります。

マークアップモードでは、以下のアイコンがターゲットデザインオブジェクトの階層に表示される可能性があります。

アイコン	説明
	デザインオブジェクトが追加されました。
	デザインオブジェクトが変更されます。
	デザインオブジェクトが削除されました。
	デザインオブジェクトが変更されました。

競合の検索

昇格対象として選択したオブジェクトのデザインオブジェクト階層内で競合に移動するには、以下のボタンを使用します。

ボタン	説明
	デザインオブジェクト階層内の前の競合に移動します。
	解決する必要がある、デザインオブジェクト階層内の次の競合に移動します。

変更を元に戻す

変更をターゲットリポジトリに適用する前であれば、加えた変更を個別に選択して元に戻すことができます。

変更を元に戻す手順

- ▶ ターゲットリポジトリで、リストアする変更に関連付けられているオブジェクトを右クリックし、ポップアップメニューから **【元に戻す】** を選択します。
リポジトリマネージャにより変更が元に戻され、ターゲットリポジトリ内の元の値で置換されます。

比較変更リストファイルの変更の保存

リポジトリマネージャでは、提案された変更を比較変更リスト XML ファイルに保存できます。例えば、次のような目的で、適用前に変更を保存できます。

- 最初に変更を確認および承認するため
- 適用する変更のサブセットを選択するなど、XML ファイル内で変更を手動で編集するため
- 将来参照できるように、変更のログを保存するため
- 同じ変更を複数のターゲットリポジトリに適用するため

比較変更リスト XML ファイルの変更を保存する手順

1. **【保存】** ボタンをクリックします。

リポジトリマネージャでは、変更の処理中に進行状況バーが表示されます。

リポジトリマネージャに、[変更リストの保存] ダイアログが表示されます。

2. 変更リスト XML ファイルを保存するターゲットディレクトリに移動します。
3. リポジトリに関する以下の情報を指定します。

フィールド	説明
名前	変更リストファイルの論理名。この値は変更リスト XML ファイルの<name>タグに格納されます。
説明	この変更リストファイルの説明。この値は変更リスト XML ファイルの<description>タグに格納されます。
ファイル名	保存する変更リストファイルの名前。リポジトリマネージャにより、このファイル名に拡張子が付加されます (*.change.xml)。

4. **【保存】** をクリックします。

リポジトリマネージャにより、指定した変更リストファイルが保存され、ターゲットの場所への書き込み中は進行状況バーが表示されます。

5. エディタで変更リスト XML ファイルを開いて内容を確認します。

特定の名前と説明が、タイムスタンプ、ファイル名、その他の情報とともに、<changelist>要素の属性として保存されます。

ターゲットリポジトリへの変更の適用

ターゲットリポジトリに変更を適用する手順

1. **【適用】** ボタンをクリックします。

インポートプロセスで障害が発生した場合は、リポジトリマネージャによって、ロールバックストラテジを選択するように求めるメッセージが表示されます。

2. 次のいずれかのオプションを選択します。

ストラテジ	説明
完全ロールバック	昇格を実行中にターゲットリポジトリに適用された変更をすべてロールバックします。
最後の変更へのロールバック	昇格のターゲットリポジトリへの適用を中断する前に、最後に成功した変更までロールバックします。例えば、ベースオブジェクトのカラム A とカラム B の変更はターゲットリポジトリに正常に適用され、カラム C の変更がターゲットリポジトリに適用される前に昇格の適用プロセスが失敗した場合、変更はカラム B までロールバックされます。 このオプションは、多数の変更を昇格していて、障害が発生した場合、適用に成功した変更はそのまま維持したい場合に使用します。

各デザインオブジェクトのロールバックでは、リポジトリの物理的な変更（例えば、物理的なカラムの削除など）とデザインオブジェクトに関するメタデータ（カラムのメタデータ記述子）の 2 つの変更がロールバックされます。

3. **[OK]** を選択します。

ターゲットリポジトリに現在プロキシユーザーが登録されている場合、リポジトリマネージャによって、このリポジトリの所有者パスワードを入力するように求めるメッセージが表示されます。

4. メッセージが表示されたら、ORS スキーマの所有者のパスワードを入力します。

5. データの整合性の検証は、ユーザーの選択に基づいて実行されます。ダイアログボックスに、変更（変更リスト昇格または視覚的昇格のいずれか）を昇格する前にデータの整合性の検証を実行するかどうかを選択するように求めるメッセージが表示されます。

リポジトリマネージャに、進行状況バーが表示されます。

完了すると、リポジトリマネージャに昇格プロセスが成功したかどうかを示すメッセージが表示されます。

関連項目：

- [「プロキシユーザーが登録されているリポジトリ」](#)（ページ 14）
- [「変更結果の監視」](#)（ページ 14）

変更リストを使用した変更の昇格

この節では、変更リストを使用して変更を昇格する方法について説明します。

変更リスト昇格タスクの概要

注: 変更を昇格する前に、ターゲットリポジトリのバックアップを作成します。

変更リストを使用して変更を昇格する手順

1. **[昇格]** / **[変更リスト]** タブに移動します。
2. ターゲットリポジトリを選択します。
3. ターゲットリポジトリに適用する変更が含まれる、変更リスト XML ファイルを開きます。
4. 必要であれば、変更リスト XML ファイルで提案された変更を確認します。
5. 必要であれば、提案された変更を変更リスト XML ファイルに保存します。

6. 変更を適用する前にシミュレーションを実行してテストします。
7. ターゲットリポジトリに変更を適用します。

関連項目：

- [「リポジトリのエクスポート」](#) (ページ 57)
- [「\[昇格\] / \[変更リスト\] タブへの移動」](#) (ページ 41)
- [「変更リスト昇格のターゲットのリポジトリの選択」](#) (ページ 42)
- [「比較変更リスト XML ファイルを開く」](#) (ページ 43)
- [「変更リスト内の移動」](#) (ページ 43)
- [「変更の簡単な説明の表示」](#) (ページ 43)
- [「変更の詳細な説明の表示」](#) (ページ 44)
- [「比較変更リスト XML ファイルの変更の保存」](#) (ページ 44)
- [「変更リストの適用シミュレーションの実行」](#) (ページ 45)
- [「ターゲットリポジトリへの変更リストの適用」](#) (ページ 45)







[昇格] / [変更リスト] タブへの移動



[昇格] タブの [変更リスト] タブに移動する手順

1. リポジトリマネージャツールを起動します。
 2. **[昇格]** タブをクリックします。
 3. **[変更リスト]** タブをクリックします。
- この節の残りのタスクを完了します。

[変更リスト] タブのコマンドボタン

[変更リスト] タブには以下のコマンドボタンがあります。

ボタン	説明
	2 つのリポジトリを比較することによって新しい変更リストを作成します。
	変更リストを開きます。
	変更リストを保存します。
	ターゲットリポジトリへの変更リストの適用をシミュレートします。
	変更リストをターゲットリポジトリに適用します。
	変更の簡単な説明を表示します。

ボタン	説明
	変更の詳細な説明を表示します。
	選択したリポジトリに対して、スキーマビューアを起動します。

変更リスト昇格のターゲットのリポジトリの選択

ターゲットリポジトリを選択する手順

1. [ターゲット] ドロップダウンリストをクリックします。
2. リストからリポジトリを選択します。

注: リポジトリマネージャでは、エラーまたは重大（FATAL）な問題のないことが検証済みのリポジトリのみを使用できます。

関連項目：

- [「問題の重要度レベル」](#)（ページ 20）
- [「オペレーショナルリファレンスストアのメタデータ検証」](#)（ページ 21）

リポジトリの比較による比較変更リストの作成

選択されているターゲットリポジトリを選択したソースリポジトリと比較することによって、新しい変更リストを作成できます。

1. **【変更リストの作成】** ボタンをクリックします。
リポジトリマネージャに、[変更リストの保存] ダイアログが表示されます。
2. 変更リストに使用するソースリポジトリを選択します。データベース（ORS）または作成変更リスト XML ファイルのいずれかを選択できます。
 - データベースリポジトリの場合は、ソースリポジトリリストからソースリポジトリを選択します。
まだ検証されていないデータベースリポジトリを選択した場合は、**【検証】** ボタンをクリックし、検証プロセスを完了します。
注: リポジトリマネージャでは、エラーまたは重大（FATAL）な問題のないことが検証済みのリポジトリのみを使用できます。
【OK】 をクリックします。
リポジトリマネージャによってソースリポジトリがロードされます。
 - 変更リスト XML ファイルの場合は、**【ファイルリポジトリ】** タブをクリックします。
【開く】 ボタンをクリックします。
変更リスト XML ファイルを選択（必要に応じてフォルダーに移動）し、**【開く】** をクリックします。
注: 比較の基準として完全なスキーマを表す作成変更リストを選択する必要があります。リポジトリマネージャでは、比較変更リストは選択できません。
リポジトリマネージャによって、選択したファイルからリポジトリがロードおよび検証されます。
【OK】 をクリックします。
3. **【OK】** をクリックします。
リポジトリマネージャによって 2 つのリポジトリが比較され、変更リストが生成されます。

関連項目：

- [「変更リスト内の移動」](#)（ページ 43）
- [「問題の重要度レベル」](#)（ページ 20）
- [「オペレーショナルリファレンスストアのメタデータ検証」](#)（ページ 21）

比較変更リスト XML ファイルを開く

選択したターゲットリポジトリに適用する変更が含まれる、比較変更リストを開くことができます。変更をターゲットリポジトリに適用する前に確認できます。

比較変更リスト XML ファイルを開く手順

1. **【開く】** ボタンをクリックします。
2. 開く変更リスト XML ファイルの名前を指定します。必要に応じて、ファイルが格納されているフォルダーに移動します。
3. **【開く】** をクリックします。

リポジトリマネージャによって、指定した変更リストファイルがロードされ、そのファイルに含まれる変更リストが表示されます。



関連項目：

- [「変更リスト内の移動」](#)（ページ 43）

変更リスト内の移動

開いている変更リスト XML ファイル内の特定のテキストを検索する手順

- ▶ 検索する文字列を **【検索】** フィールドに入力します。
リポジトリマネージャにより、指定した文字列を含む最初の変更が選択されます。
変更リスト内の他の一致を見つけるには、次のボタンを使用します。

ボタン	説明
	文字列の前のインスタンスに移動します。
	文字列の次のインスタンスに移動します。

変更の簡単な説明の表示

変更リスト内の変更の簡単な説明を表示する手順

1. 変更リストから変更を選択します。
2. **【説明を表示】** ボタンをクリックします。
リポジトリマネージャに変更の説明が表示されます。
3. **【閉じる】** をクリックします。

変更の詳細な説明の表示

変更リスト内の変更の詳細な説明を表示する手順

1. 変更リストから変更を選択します。
2. **【詳細を表示】** ボタンをクリックします。
リポジトリマネージャに変更の詳細な説明が表示されます。
3. **【閉じる】** をクリックします。

比較変更リスト XML ファイルの変更の保存

リポジトリマネージャでは、提案された変更を変更リスト XML ファイルに保存できます。例えば、次のような目的で、適用前に変更を保存できます。

- 最初に変更を確認および承認するため
- 適用する変更のサブセットを選択するなど、XML ファイル内で変更を手動で編集するため
- 将来参照できるように、変更のログを保存するため
- 同じ変更を複数のターゲットリポジトリに適用するため

このプロセスにより、比較変更リスト XML ファイルが生成されます。

比較変更リスト XML ファイルに変更を保存する手順

1. **【保存】** ボタンをクリックします。
リポジトリマネージャでは、変更の処理中に進行状況バーが表示されます。
リポジトリマネージャに、[変更リストの保存] ダイアログが表示されます。
2. 変更リスト XML ファイルを保存するターゲットディレクトリに移動します。
3. リポジトリに関する以下の情報を指定します。


フィールド	説明
名前	変更リストファイルの論理名。この値は変更リスト XML ファイルの<name>タグに格納されます。
説明	この変更リストファイルの説明。この値は変更リスト XML ファイルの<description>タグに格納されます。
ファイル名	保存する変更リストファイルの名前。リポジトリマネージャにより、このファイル名に拡張子が付加されます (*.change.xml)。

4. **【保存】** をクリックします。
リポジトリマネージャにより、指定した変更リストファイルが保存され、ターゲットの場所への書き込み中は進行状況バーが表示されます。
5. エディタで変更リスト XML ファイルを開いて内容を確認します。
特定の名前と説明が、タイムスタンプ、ファイル名、その他の情報とともに、<changelist>要素の属性として保存されます。

変更リストの適用シミュレーションの実行

変更リストをターゲットリポジトリに適用するプロセスをシミュレートし、変更を実際に適用する前に結果を確認してエラーを修正することができます。

変更リストの適用をシミュレートする手順

1. ボタン  **【シミュレート】** ボタンをクリックします。
リポジトリマネージャによって、シミュレーション実行の確認を求めるメッセージが表示されます。
2. **【はい】** をクリックして確認します。
ターゲットリポジトリに現在プロキシユーザーが登録されている場合、リポジトリマネージャによって、このリポジトリの所有者パスワードを入力するように求めるメッセージが表示されます。
3. メッセージが表示されたら、ORS スキーマの所有者のパスワードを入力します。
4. データの整合性の検証は、ユーザーの選択に基づいて実行されます。ダイアログボックスに、変更（変更リスト昇格または視覚的昇格のいずれか）を昇格する前にデータの整合性の検証を実行するかどうかを選択するように求めるメッセージが表示されます。
リポジトリマネージャによってシミュレーションが実行され、結果についてのメッセージが表示されます。
5. **【OK】** をクリックします。
問題を示すメッセージが表示された場合は、実際に変更をターゲットリポジトリに適用する前に、問題を修正して再度シミュレーションを実行する必要があります。 そうしないと、変更の適用プロセスが失敗する可能性があります。

関連項目：

- [「プロキシユーザーが登録されているリポジトリ」](#)（ページ 14）

ターゲットリポジトリへの変更リストの適用

ターゲットリポジトリに変更を適用する手順

1. **【適用】** ボタンをクリックします。
インポートプロセスで障害が発生した場合は、リポジトリマネージャによって、ロールバックストラテジを選択するように求めるメッセージが表示されます。
2. 次のいずれかのオプションを選択します。

ストラテジ	説明
完全ロールバック	昇格プロセス中に適用された変更をすべてロールバックします。
最後の変更へのロールバック	昇格の適用プロセスを中断する前に、最後に成功した変更までロールバックします。例えば、ベースオブジェクトのカラム A とカラム B の変更はターゲットリポジトリに正常に適用され、カラム C の変更がターゲットリポジトリに適用される前に昇格の適用プロセスが失敗した場合、変更はカラム B までロールバックされます。このオプションは、多数の変更を昇格していて、障害が発生した場合、適用に成功した変更はそのまま維持したい場合に使用します。

各デザインオブジェクトのロールバックでは、リポジトリの物理的な変更（例えば、物理的なカラムの削除など）とデザインオブジェクトに関するメタデータ（カラムのメタデータ記述子）の 2 つの変更がロールバックされます。

3. **【OK】** を選択します。

ターゲットリポジトリに現在プロキシユーザーが登録されている場合、リポジトリマネージャによって、このリポジトリの所有者パスワードを入力するように求めるメッセージが表示されます。

4. メッセージが表示されたら、ORS スキーマの所有者のパスワードを入力します。
5. データの整合性の検証は、ユーザーの選択に基づいて実行されます。ダイアログボックスに、変更（変更リスト昇格または視覚的昇格のいずれか）を昇格する前にデータの整合性の検証を実行するかどうかを選択するように求めるメッセージが表示されます。

リポジトリマネージャに、進行状況バーが表示されます。

完了すると、リポジトリマネージャに昇格プロセスが成功したかどうかを示すメッセージが表示されます。

関連項目：

- [「プロキシユーザーが登録されているリポジトリ」](#)（ページ 14）
- [「変更結果の監視」](#)（ページ 14）

コマンドラインからの変更の昇格

コマンドラインからリポジトリに変更を昇格するには、MDM Hub リソースキットに含まれる MetCommand ユーティリティを使用します。リポジトリは、オペレーショナルリファレンスストア（ORS）としても知られます。

変更をターゲットリポジトリに昇格する前に、変更リスト XML ファイルにソースリポジトリをエクスポートします。また、MDM Hub 環境にリソースキットがインストールされていることも確認します。コマンドラインから、リポジトリに変更を昇格するためのコマンドを実行します。

1. 以下の場所にある SiperianConnection.properties ファイルを編集します。

```
<Resource Kit installation directory>/samples/MetCommand/source/resources/properties
```

アプリケーションサーバーの接続プロパティ値が、アプリケーションサーバーの設定と一致することを確認します。

2. コマンドラインから MetCommand ディレクトリに移動します。これは、次の場所にあります。

```
<Resource Kit installation directory>/samples/MetCommand
```

3. 変更リストを作成するには、次のコマンドを実行します。

```
metcommand -createChangelist
-sourceXmlFilename "<Source repository change list XML file name>"
-targetOrsId "<Target repository ID>"
-outputFilename "<Change list XML file name>"
-propertiesFilename "<Resource Kit installation directory>/samples/MetCommand/source/resources/properties/SiperianConnection.properties"
```

変更リストを作成するプロセスでは、ソースリポジトリの変更リストが、ターゲットリポジトリのメタデータと比較され、ターゲットリポジトリの変更リストが生成されます。

4. 変更リストを検証するには、次のコマンドを実行します。

```
metcommand -validateChangelist
-sourceXmlFilename "<Change list XML file name>"
-targetOrsId "<Target repository ID>"
-propertiesFilename "<Resource Kit installation directory>/samples/MetCommand/source/resources/properties/SiperianConnection.properties"
```

5. 変更リストをターゲットリポジトリに適用するには、次のコマンドを実行します。

```
metcommand -applyChangelist
-sourceXmlFilename "<Change list XML file name>"
-targetOrsId "<Target repository ID>"
-propertiesFilename "<Resource Kit installation directory>/samples/MetCommand/source/resources/properties/SiperianConnection.properties"
```

6. ターゲットリポジトリのメタデータを検証するには、次のコマンドを実行します。

```
metcommand -validateMetadata  
-targetOrsId "<Target repository ID>"  
-propertiesFilename "<Resource Kit installation directory>/samples/MetCommand/source/resources/  
properties/SiperianConnection.properties"
```

第 5 章

デザインオブジェクトのインポート

この章では、以下の項目について説明します。

- [概要, 48 ページ](#)
- [デザインオブジェクトのインポートについて, 48 ページ](#)
- [デザインオブジェクトのインポート, 50 ページ](#)

概要

この章では、Hub コンソールのリポジトリマネージャを使用して新しいデザインオブジェクトをリポジトリにインポートする方法について説明します。

デザインオブジェクトのインポートについて

この節では、デザインオブジェクトを空のリポジトリにインポートする前に理解しておく必要のある概念について説明します。

インポートプロセス

リポジトリマネージャでは、ソースリポジトリまたは変更リストから空のターゲットリポジトリにデザインオブジェクトを選択的にインポートすることができます。インポート処理では、ターゲットリポジトリにまだ定義されていない、デザインオブジェクトに関するメタデータが挿入されます。例えば、標準的なデザインオブジェクトのテンプレートを新しいリポジトリに追加できます。

注: あるデータベースタイプで作成された変更リストを、他のデータベースタイプのリポジトリにインポートすることはできません。例えば、Oracle で作成された変更リストを、Microsoft SQL Server のリポジトリにインポートすることはできません。

デザインオブジェクトのインポートは、デザインオブジェクトがターゲットリポジトリにすでに存在している昇格操作とは対照的な操作です。メタデータが含まれているリポジトリにデザインオブジェクトを追加する場合は、[昇格] タブの機能を使用します。

関連項目：

- [「リポジトリ間での変更の昇格」 \(ページ 28\)](#)

インポート可能なデザインオブジェクト

インポート可能なデザインオブジェクトの完全なリストについては、[「リポジトリマネージャでサポートされているデザインオブジェクト」 \(ページ 64\)](#)を参照してください。

インポートプロセスに関する考慮事項

インポートする前に、以下の情報を検討します。

- デザインオブジェクトをインポートする前に、ターゲットリポジトリのバックアップを作成することを確認します。
- インポート（および名前変更）ではターゲットリポジトリのみが変更され、ソースリポジトリは変更されません。
- リポジトリマネージャのインポート操作では、エラーまたは重大（FATAL）な問題のない検証済みのリポジトリを必要とします。
- Informatica MDM Hub バージョン 9.5 では、Period_Start_Date と Period_End_Date カラムが Rel_Start_Date カラムと Rel_End_Date カラムの代わりとなります。9.5 より前の変更リストを 9.5 の新規インストールにインポートする場合は、変更リストをインポートして 9.5 より前のスキーマを更新した後、スクリプト migrate_hm_rel_start_end_dates.sql を実行する必要があります。[「リレーションベースオブジェクトの開始日と終了日の情報の更新」 \(ページ 54\)](#)を参照してください。
- リポジトリマネージャのエクスポートおよびインポートプロセスでは、ROWID_OBJECT の値はデザインオブジェクトレコードに保持されません。デザインオブジェクトをエクスポートするとき、ソース ORS の ROWID_OBJECT の値はエクスポートファイルに保持されません。これらのデザインオブジェクトをターゲット ORS にインポートすると、インポートプロセスによって新しい ROWID_OBJECT の値が割り当てられます。この値は、ソース ORS の対応するレコードと異なる場合があります。ROWID_OBJECT の値に基づいてデザインオブジェクトを一意に識別すると、問題が発生する場合があります。例えば、ソーススキーマの C_RBO_BO_CLASS テーブルへの参照として ROWID_BO_CLASS を使用するように階層マネージャが設定されている場合があります。SIF クライアントアプリケーションがこの設定に依存していた場合、ソース ORS では正しく動作しますが、ターゲット ORS では必ずしも正しく動作しません。このため、ROWID_OBJECT の値ではなく、エンティティの場合は BO_CLASS_CODE、リレーションタイプの場合は REL_TYPE_CODE、階層の場合は HIERARCHY_CODE など、特定のデザインオブジェクトを識別する一意の識別子を使用してください。
- カスタム一致ポピュレーションは、変更リストからインポートされません。変更リスト XML ファイルからカスタム一致ポピュレーション要素を削除した上で、ターゲットスキーマを手動で変更するか、またはデータベースのインポート/変更リストオプションを直接使用します。
- リポジトリにインポートする前に、ソースリポジトリとターゲットリポジトリの両方に、必要なすべてのテーブルスペースに対して十分な特権があることを確認します。
- ソースリポジトリで、管理システムが状態管理オーバーライドシステムとして有効にされている場合は、ターゲットリポジトリで管理システムをインポートしても、状態管理オーバーライドシステムインジケータは更新されません。手動により、ターゲットリポジトリで管理システムを状態管理オーバーライドシステムとして有効にする必要があります。

関連項目：

- [「問題の重要度レベル」 \(ページ 20\)](#)
- [「メタデータのコピーに関する考慮事項」 \(ページ 12\)](#)
- [「リポジトリのエクスポート」 \(ページ 57\)](#)

デザインオブジェクトのインポート

この節では、リポジトリマネージャでデザインオブジェクトをインポートする方法について説明します。

インポートタスクの概要

注: デザインオブジェクトをインポートする前に、ターゲットリポジトリのバックアップを作成します。

デザインオブジェクトをリポジトリにインポートする手順





1. **【インポート】** タブに移動します。
2. ソースリポジトリを選択します。
3. ターゲットリポジトリを選択します。
4. ソースリポジトリからインポートするデザインオブジェクトを選択します。
5. 必要に応じて、インポートするオブジェクトの名前を変更します。
6. インポートプロセスを実行します。

関連項目：

- [「インポートするデザインオブジェクトの選択」 \(ページ 52\)](#)
- [「デザインオブジェクトの名前変更」 \(ページ 52\)](#)
- [「リポジトリのエクスポート」 \(ページ 57\)](#)
- [「インポートのソースリポジトリの選択」 \(ページ 50\)](#)
- [「インポートのターゲットリポジトリの選択」 \(ページ 51\)](#)
- [「選択したデザインオブジェクトのインポート」 \(ページ 53\)](#)

【インポート】 タブのコマンドボタン

【インポート】 タブには以下のコマンドボタンがあります。

ボタン	説明
	変更リストをターゲットリポジトリに適用することによってオブジェクトをインポートします。
	選択したデザインオブジェクトの名前を変更します。
	ノードを縮小します。
	ノードを展開します。

インポートのソースリポジトリの選択

ソースリポジトリには、データベース（ORS）または変更リストのXML ファイルのいずれかを選択できます。

インポートするソースリポジトリを選択する手順:

1. ソースフィールドの横にある **【選択】** ボタンをクリックします。
リポジトリマネージャに、[リポジトリを開く] ウィンドウが表示されます。
2. リストからソースリポジトリを選択します。
 - データベースリポジトリの場合は、リストからソースリポジトリを選択します。
まだ検証されていないデータベースリポジトリを選択した場合は、**【検証】** ボタンをクリックし、検証プロセスを完了します。
注: リポジトリマネージャでは、エラーまたは重大 (FATAL) な問題のないことが検証済みのリポジトリのみをインポートできます。
【OK】 をクリックします。
リポジトリマネージャによってソースリポジトリがロードされます。
 - 変更リスト XML ファイルの場合は、**【ファイルリポジトリ】** タブをクリックします。
【開く】 ボタンをクリックします。
変更リスト XML ファイルを選択 (必要に応じてフォルダーに移動) し、**【開く】** をクリックします。
注: 作成変更リストを選択する必要があります。リポジトリマネージャでは、インポートするデザインオブジェクトのソースとして、比較変更リストは選択できません。
リポジトリマネージャによって、選択したファイルからリポジトリがロードおよび検証されます。
【OK】 を選択します。
3. リポジトリマネージャにデザインオブジェクト階層として表示される、ロードされたソースリポジトリを確認します。

関連項目:

- [「問題の重要度レベル」 \(ページ 20\)](#)
- [「オペレーショナルリファレンスストアのメタデータ検証」 \(ページ 21\)](#)



インポートのターゲットリポジトリの選択

ターゲットリポジトリを選択する手順

1. [ターゲット] ドロップダウンリストをクリックします。
2. リストからリポジトリを選択します。
リポジトリマネージャによってターゲットリポジトリがロードされ、2 つのリポジトリが比較されます。

階層内のデザインオブジェクトの表示および非表示

次のボタンをクリックすると、デザインオブジェクト階層が展開または縮小されます。

ボタン	説明
	ノードを展開します。
	ノードを縮小します。

ノードを展開すると、リポジトリマネージャによって、選択したノードに関連付けられた子オブジェクトのツリー全体が展開されます。

インポートするデザインオブジェクトの選択

ソースリポジトリとターゲットリポジトリを選択したら、デザインオブジェクト階層をナビゲートして、インポートするデザインオブジェクトを選択することができます。デザインオブジェクトを選択すると、リポジトリマネージャによってその依存関係が分析され、関連付けられているデザインオブジェクトがすべて自動的に選択されます。例えば、パッケージを選択すると、パッケージに基づくすべてのテーブルとクエリが自動的に選択（チェック）されます。同様に、クエリを削除（選択解除）すると、そのクエリに基づくパッケージもすべて削除されます。インポートから除外するオブジェクトを個別に削除（選択解除）することができます。

通常は、必要なデザインオブジェクトが誤って除外されないように、サブセットを個別にインポートするのではなく、デザインオブジェクト全体を一度にインポートすることを検討してください。例えば、ランディングテーブル、ステージングテーブル、マッピング、およびパッケージを個別にインポートするのではなく、依存デザインオブジェクトがすべてインポートされるよう、ベースオブジェクト全体をインポートすることを検討してください。

デザインオブジェクトの選択

デザインオブジェクトを選択する手順

- ▶ デザインオブジェクトの横にあるチェックボックスをクリックします。
例えば、ベースオブジェクトを選択した場合、それに基づくクエリとパッケージはオプションです。

オブジェクトタイプの選択

特定のオブジェクトタイプのデザインオブジェクトをすべて選択する手順

- ▶ オブジェクトタイプの横にあるチェックボックスを選択します。
リポジトリマネージャによって、そのタイプのデザインオブジェクトがすべて自動的に選択されます。

すべてのデザインオブジェクトの選択

ソースリポジトリ内のデザインオブジェクトをすべて選択する手順

- ▶ 階層の上部にある【リポジトリ】チェックボックスを選択します。

デザインオブジェクトの名前変更

デザインオブジェクトの名前は変更することができます。例えば、デザインオブジェクトのテンプレートをインポートする前に、組織の命名規則に合わせてオブジェクトの名前を変更できます。

関連項目：

- [「名前の変更が可能なデザインオブジェクト」](#)（ページ 67）

グローバル一意名のデザインオブジェクト

特定のデザインオブジェクト（ステージングテーブルおよび一致パスコンポーネント）には、グローバルに一意な名前があります。それ以外のデザインオブジェクトの名前は、ベースオブジェクトまたはカラムに関連付けられています。リポジトリマネージャでは、親デザインオブジェクトの名前変更が子デザインオブジェクトに適切に実装されます。例えば、ベースオブジェクトの名前を変更した場合、それに従ってそのベースオブジェクトを参照するクエリも更新されます。

選択したオブジェクトの名前変更

選択したデザインオブジェクトの名前を変更する手順

- 次のいずれかを実行します。
 - オブジェクトを選択し、**【名前の変更】** ボタンをクリックします。
 - または、デザインオブジェクトを右クリックします。
リポジトリマネージャによって、新しい名前を指定するように求めるメッセージが表示されます。
- 次の情報を入力します。

ストラテジ	説明
新しい名前	デザインオブジェクトの物理名。新しい名前は、そのオブジェクトタイプに対する Informatica MDM Hub の命名規則に従う必要があります。
新しい表示名	Hub コンソール内のこのデザインオブジェクトの表示に使用される名前。

- 【OK】** を選択します。
リポジトリマネージャによって、名前が変更されたデザインオブジェクトが階層内に表示されます。
注: 新しい名前は、ソースリポジトリのデザインオブジェクト階層には表示されますが、ターゲットリポジトリに適用されるのはインポートプロセスの完了後です。ソースリポジトリ内のデザインオブジェクトの名前は変更されません。

選択したデザインオブジェクトのインポート

選択したオブジェクトをインポートする手順

- 【適用】** ボタンをクリックします。
インポートプロセスで障害が発生した場合は、リポジトリマネージャによって、ロールバックストラテジを選択するように求めるメッセージが表示されます。
- 次のいずれかのオプションを選択します。

オプション	説明
完全ロールバック	インポートプロセス中に適用された変更をすべてロールバックします。
最後の変更へのロールバック	インポートプロセスを中断する前に、最後に成功した変更までロールバックします。例えば、ベースオブジェクト A とベースオブジェクト B が正常にインポートされたが、カラム C がターゲットリポジトリに正常に作成される前にインポートプロセスが失敗した場合、変更はカラム B にロールバックされます。 このオプションは、多数のデザインオブジェクトをインポートしていて、障害が発生した場合に正常に適用された変更を保持する場合に使用します。

各デザインオブジェクトのロールバックでは、リポジトリの物理的な変更（例えば、物理的なカラムの削除など）とデザインオブジェクトに関するメタデータ（カラムのメタデータ記述子）の 2 つの変更がロールバックされます。

- 【OK】** を選択します。
ターゲットリポジトリに現在プロキシユーザーが登録されている場合、リポジトリマネージャによって、このリポジトリの所有者パスワードを入力するように求めるメッセージが表示されます。

4. メッセージが表示されたら、ORS スキーマの所有者のパスワードを入力します。
5. データの整合性の検証は、ユーザーの選択に基づいて実行されます。ダイアログボックスに、変更（変更リスト昇格または視覚的昇格のいずれか）を昇格する前にデータの整合性の検証を実行するかどうかを選択するように求めるメッセージが表示されます。

リポジトリマネージャに、進行状況バーが表示されます。

完了すると、リポジトリマネージャにインポートプロセスが成功したかどうかを示すメッセージが表示されます。

関連項目：

- [「プロキシユーザーが登録されているリポジトリ」](#)（ページ 14）
- [「変更結果の監視」](#)（ページ 14）

リレーションベースオブジェクトの開始日と終了日の情報の更新

Informatica MDM Hub バージョン 9.5 をインストールして 9.5 より前の変更リストをインポートすることによって 9.5 にアップグレードする場合は、次の変更が必要です。

- Rel_Start_Date カラムと Rel_End_Date カラムを Null 可能にする必要があります。
- Rel_Start_Date カラムと Rel_End_Date カラムへのマッピングを、Period_Start_Date カラムと Period_End_Date カラムに再マッピングする必要があります。
- Rel_Start_Date カラムと Rel_End_Date カラムの参照を、階層マネージャのリレーションパッケージから削除する必要があります。

変更リストのインポート後、必要な変更を加えるために次の手順を実行します。

1. SQL*Plus で、次の場所にある migrate_hm_start_end_dates.sql スクリプトを実行します。
 - Windows:<infamdm_install_directory>\server\resource\database\migration_readiness\oracle
 - UNIX:<infamdm_install_directory>/server/resource/database/migration_readiness/oracleスクリプトが実行され、必要な変更が加えられます。
2. アプリケーションサーバーを再起動します。
Period_Start_Date カラムと Period_End_Date カラムへのマッピングがアクティブになります。

第 6 章

リポジトリのエクスポート

この章では、以下の項目について説明します。

- [概要, 55 ページ](#)
- [リポジトリのエクスポートについて, 55 ページ](#)
- [リポジトリのエクスポート, 57 ページ](#)
- [デザインオブジェクトのサブセットのエクスポート, 57 ページ](#)

概要

この章では、Hub コンソールのリポジトリマネージャを使用して、Informatica MDM Hub 実装の変更リスト XML ファイルにリポジトリをエクスポートする方法について説明します。

リポジトリのエクスポートについて

この節では、リポジトリマネージャを使用してリポジトリをエクスポートする前に理解しておく必要のあることについて説明します。

エクスポートについて

リポジトリマネージャを使用して、リポジトリ全体を変更リスト XML ファイルにエクスポートすることができます。この XML ファイルを使用して、デザインオブジェクトを別のリポジトリにインポートしたり、アーカイブとしてソース制御システムに保存したりすることができます。

関連項目：

- [「変更リスト」 \(ページ 11\)](#)
- [「変更リストリファレンス」 \(ページ 69\)](#)

エクスポート可能なデザインオブジェクト

エクスポート可能なデザインオブジェクトの完全なリストについては、[「リポジトリマネージャでサポートされているデザインオブジェクト」 \(ページ 64\)](#)を参照してください。

関連項目：

- [「リポジトリマネージャでサポートされているデザインオブジェクト」](#) (ページ 64)
- [「変更リストリファレンス」](#) (ページ 69)

エクスポートした変更リスト XML ファイルの用途

作成した変更リスト XML ファイルを使用して以下の作業を行うことができます。

- 新しいデザインオブジェクトをリポジトリにインポートする。
- リポジトリ間で変更を昇格する。作成変更リストは昇格のソースリポジトリとして使用できます。
- ソース制御システムにアーカイブする。

関連項目：

- [「デザインオブジェクトのインポート」](#) (ページ 48)
- [「リポジトリ間での変更の昇格」](#) (ページ 28)
- [「視覚的昇格のソースリポジトリの選択」](#) (ページ 34)

エクスポートプロセスに関する考慮事項

リポジトリをエクスポートする前に、以下の事項を考慮します。



- エクスポートによってソースリポジトリが変更されることはありません。
- エクスポートの場合、リポジトリマネージャはエラーまたは重大（FATAL）な問題のない検証済みのソースリポジトリを必要とします。
- リポジトリマネージャのエクスポートおよびインポートプロセスでは、ROWID_OBJECT の値はデザインオブジェクトレコードに保持されません。デザインオブジェクトをエクスポートするとき、ソース ORS の ROWID_OBJECT の値はエクスポートファイルに保持されません。これらのデザインオブジェクトをターゲット ORS にインポートすると、インポートプロセスによって新しい ROWID_OBJECT の値が割り当てられます。この値は、ソース ORS の対応するレコードと異なる場合があります。

関連項目：

- [「問題の重要度レベル」](#) (ページ 20)
- [「オペレーショナルリファレンスストアのメタデータ検証」](#) (ページ 21)

[エクスポート] タブのコマンドボタン

[エクスポート] タブには以下のコマンドボタンがあります。

ボタン	説明
	変更リストを保存します。
	選択したリポジトリに対して、スキーマビューアを起動します。

リポジトリのエクスポート

リポジトリを作成変更リストとしてエクスポートする手順

1. リポジトリマネージャツールを起動します。
2. まだ選択されていない場合は、**【エクスポート】** タブをクリックします。
3. ドロップダウンリストから、エクスポートするリポジトリを選択します。

注: リポジトリマネージャでは、エラーおよび重大 (FATAL) な問題のないことが検証済みのリポジトリのみを使用できます。

4. **【保存】** ボタンをクリックします。
リポジトリマネージャに、**【リポジトリのエクスポート】** ダイアログが表示されます。
5. 変更リスト XML ファイルを保存するターゲットディレクトリに移動します。
6. リポジトリに関する以下の情報を指定します。

フィールド	説明
名前	変更リストファイルの論理名。この値は変更リスト XML ファイルの<name>タグに格納されます。
説明	この変更リストファイルの説明。この値は変更リスト XML ファイルの<description>タグに格納されます。
ファイル名	保存する変更リストファイルの名前。リポジトリマネージャにより、このファイル名に拡張子が付加されます (*.change.xml)。

7. **【エクスポート】** をクリックします。

リポジトリマネージャにより、指定した変更リストファイルが保存され、ターゲットの場所への書き込み中は進行状況バーが表示されます。

8. エディタで変更リスト XML ファイルを開いて内容を確認します。

特定の名前と説明が、タイムスタンプ、ファイル名、その他の情報とともに、<changelist>要素の属性として保存されます。

注: エクスポートファイルを作成後、インポートまたは昇格操作でファイルを使用する前に、ファイルの編集、デザインオブジェクト名の変更、無関係なセクションの削除、および必要に応じてその他の変更を行うことができます。ただし、Informatica は、このようにして変更した XML ファイルの結果に対して責任を負いません。

関連項目：

- [「問題の重要度レベル」 \(ページ 20\)](#)
- [「リポジトリマネージャの開始」 \(ページ 15\)](#)

デザインオブジェクトのサブセットのエクスポート

エクスポートを実行すると、ソースリポジトリ内のすべてのデザインオブジェクトが含まれる作成変更リスト XML ファイルが生成されます。

選択したデザインオブジェクトのサブセットのみを含む変更リストを作成することもできます。例えば、個別のクレンジング関数のみを昇格することができます。デザインオブジェクトのサブセットをエクスポートするには、[昇格] タブで以下のタスクを完了します。

1. エクスポートするデザインオブジェクトが含まれるソースリポジトリを選択します。
2. 空のターゲットリポジトリを選択します。
3. ソースリポジトリで、昇格するオブジェクトを選択します。

必要に応じて、一部のデザインオブジェクトを変更リスト内に別の名前で作成する場合は、オブジェクトを個別に選択して名前を変更できます。

4. 変更を変更リストに保存します。

保存したら、以下の手順を実行して、この比較変更リスト内の変更を任意のターゲットリポジトリに適用できます。

1. デザインオブジェクトの追加先のターゲットリポジトリを開きます。
2. 以前に作成した比較変更リスト XML ファイルを開きます。
3. 比較変更リストの変更を適用します。

第 7 章

共通ウェアハウスモデルのサポート

この章では、以下の項目について説明します。

- [概要, 59 ページ](#)
- [\[CWM ファイルからのインポート\] タブ, 60 ページ](#)
- [CWM ファイルからのデザインオブジェクトのインポート, 60 ページ](#)
- [\[CWM ファイルへのエクスポート\] タブ, 62 ページ](#)
- [CWM ファイルへのリポジトリのエクスポート, 63 ページ](#)

概要

リポジトリマネージャツールでは、共通ウェアハウスモデル（CWM）フォーマットのファイルがサポートされます。CWM ファイルは、Erwin などのサードパーティのソフトウェアを使用して生成できます。CWM を使用する利点は、Hub コンソールの CWM からのインポートオプションを使用して、シード ORS に CWM スキーマを適用できることです。

シード ORS に CWM スキーマを適用すると、ROWID_OBJECT や CREATE_DATE などのシステムカラムが自動的に追加されます。ROWID_OBJECT は、MDM によって強制されるテーブルのプライマリキーです。

EMI、EMO、XREF テーブルなどのコンテンツメタデータテーブルは自動的に作成されるため、シード ORS にインポートしないでください。

注: EMI テーブルはシステムテーブルで、その構造は定義した一致カラムに基づきます。

リポジトリマネージャツールには、CWM ファイルを扱うための、**[CWM ファイルからのインポート]** タブと **[CWM ファイルへのエクスポート]** タブがあります。




[CWM ファイルからのインポート] タブ

このツールは、CWM ファイルを MET 変更リストに変換するために使用します。MET 変更リストは、既存の ORS に適用したり、保存して後日使用したりすることができます。この操作の手順は以下のとおりです。

1. CWM ファイルをロードします。CWM ファイルがロードされ、インポート可能な要素（テーブル、ビュー、および外部キー）が選択ツリー形式で提示されます。ここでユーザーは、インポートするオブジェクト、およびテーブルの場合はインポートするテーブルタイプ（BO、ランディング）を選択できます。選択/タイプを変更するには、テーブルのチェックボックスを続けてクリックします。対応する外部キーとビューが自動的に選択/選択解除されます。
2. 変更リストを保存します。選択したオブジェクトを使用して、MET 変更リストが生成および保存されます。
3. 変更リストを適用します。選択したオブジェクトを使用して、MET 変更リストが生成され、選択した ORS に対して適用されます。

[CWM ファイルからのインポート] タブのコマンドボタン

[CWM ファイルからのインポート] タブには以下のコマンドボタンがあります。

ボタン	説明
	CWM ファイルをロードします。
	変更リストを保存します。
	CWM 変更リストをターゲットリポジトリに適用することによってオブジェクトをインポートします。

CWM ファイルからのデザインオブジェクトのインポート

この節では、リポジトリマネージャで CWM ファイルからデザインオブジェクトをインポートする方法について説明します。

注: CWM ファイルからインポートする前に、[「リポジトリのエクスポート」](#) (ページ 57) の手順に従ってターゲットリポジトリをバックアップします。

CWM ファイルからリポジトリにインポートするには、次の手順を実行します。

1. リポジトリマネージャで、**[CWM ファイルからのインポート]** タブをクリックします。
2. インポートする CWM ファイルをロードします。
 - a. **[CWM ファイルのロード]** ボタンをクリックします。
リポジトリマネージャに、**[開く]** ダイアログが表示されます。
 - b. インポートする必要がある .cwm ファイルに移動し、CWM ファイルを選択し、**[開く]** をクリックします。

リポジトリマネージャによって、選択した CWM ファイルがロードおよび検証されます。

- c. リポジトリマネージャにデザインオブジェクト階層として表示される、ロードされたソースリポジトリを確認します。

3. インポートするデザインオブジェクトを選択します。

- a. デザインオブジェクトを選択するには、デザインオブジェクトの横にあるチェックボックスをクリックします。

デザインオブジェクトを選択すると、関連付けられているすべてのデザインオブジェクトが自動的に選択されます。例えば、パッケージを選択すると、パッケージに基づくすべてのテーブルとクエリが自動的に選択（チェック）されます。同様に、クエリを削除（選択解除）すると、そのクエリに基づくパッケージも削除されます。インポートから除外するオブジェクトを個別に削除（選択解除）することができます。

- a. 特定のオブジェクトタイプのデザインオブジェクトをすべて選択するには、オブジェクトタイプの横にあるチェックボックスを選択します。

リポジトリマネージャによって、そのタイプのデザインオブジェクトがすべて自動的に選択されます。

- a. ソースリポジトリのデザインオブジェクトをすべて選択するには、階層上部のリポジトリの横にあるチェックボックスを選択します。

通常は、必要なデザインオブジェクトが誤って除外されないように、サブセットを個別にインポートするのではなく、デザインオブジェクト全体を一度にインポートすることを検討してください。例えば、ランディングテーブル、ステージングテーブル、マッピング、およびパッケージを個別にインポートするのではなく、依存デザインオブジェクトがすべてインポートされるよう、ベースオブジェクト全体をインポートすることを検討してください。

4. 選択したデザインオブジェクトをインポートします。

- a. **【変更リストの適用】** ボタンをクリックします。

リポジトリマネージャに、データベースリポジトリリストが含まれる **【リポジトリを開く】** ダイアログが表示されます。

5. インポートのターゲットリポジトリを選択します。

- a. **【データベースリポジトリ】** リストからターゲットリポジトリを選択し、**【OK】** をクリックします。

- b. インポートプロセスで障害が発生した場合は、リポジトリマネージャによって、ロールバックストラテジを選択するように求めるメッセージが表示されます。

次のいずれかのオプションを選択します。

オプション	説明
完全ロールバック	インポートプロセス中に適用された変更をすべてロールバックします。
最後の変更へのロールバック	インポートプロセスを中断する前に、最後に成功した変更までロールバックします。例えば、ベースオブジェクト A とベースオブジェクト B が正常にインポートされたが、カラム C がターゲットリポジトリに正常に作成される前にインポートプロセスが失敗した場合、変更はカラム B にロールバックされます。 このオプションは、多数のデザインオブジェクトをインポートしていて、障害が発生した場合に正常に適用された変更を保持する場合に使用します。

- c. **【OK】** をクリックします。

ターゲットリポジトリに現在プロキシユーザーが登録されている場合、リポジトリマネージャによって、リポジトリの所有者パスワードを入力するように求めるメッセージが表示されます。

- d. パスワードを入力するように求めるメッセージが表示されたら、ORS スキーマの所有者のパスワードを入力します。

リポジトリマネージャに、進行状況バーが表示されます。インポートプロセスが完了すると、リポジトリマネージャにインポートプロセスが成功したかどうかを示すメッセージが表示されます。

6. 比較変更リスト XML ファイルの変更を保存します。

- a. **【保存】** ボタンをクリックします。

リポジトリマネージャに、[変更リストの保存] ダイアログが表示されます。

- b. 変更リスト XML ファイルを保存する必要があるターゲットディレクトリに移動します。

- c. [ファイル名] フィールドに、保存する変更リストファイルの名前を入力します。

リポジトリマネージャによって、ファイルに拡張子が追加されます (*.change.xml)。

- d. **【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 change.xml ファイルが保存され、ターゲットの場所への書き込み中は進行状況バーが表示されます。

注: エディタで変更リスト XML ファイルを開いて内容を確認します。



[CWM ファイルへのエクスポート] タブ

このツールは、ORS メタデータを CWM 形式にエクスポートするために使用します。この操作には以下の 2 つの手順があります。

1. リポジトリをロードする - ソース ORS リポジトリをツールにロードします。エクスポート可能な ORS オブジェクトの選択ツリーが表示されます。ツリーの要素を選択することで、CWM 形式にエクスポートする ORS オブジェクトを指定できます。
2. CWM ファイルを保存する - 選択したオブジェクトが CWM 形式にエクスポートされ、ファイルが（拡張子 XML で）保存されます。タブをクリックし、フォルダアイコンをクリックすると、CWM ファイルがインポートまたはエクスポートされます。

[CWM ファイルへのエクスポート] タブのコマンドボタン

[CWM ファイルへのエクスポート] タブには以下のコマンドボタンがあります。

ボタン	説明
	リポジトリをロードします。
	変更リストを保存します。

CWM ファイルへのリポジトリのエクスポート

CWM XML ファイルからリポジトリをエクスポートするには、次の手順を実行します。

1. リポジトリマネージャで、**[CWM ファイルへのエクスポート]** タブをクリックします。
2. エクスポートのデータベースリポジトリをロードします。
 - a. **[リポジトリのロード]** ボタンをクリックします。

リポジトリマネージャに、**[リポジトリを開く]** ダイアログが表示されます。
 - b. **[データベースリポジトリ]** リストからリポジトリを選択し、**[OK]** をクリックします。

注: リポジトリマネージャでは、エラー、重大なエラー、または致命的なエラーのないことが検証済みのリポジトリのみを使用できます。
 - c. CWM ファイルにエクスポートするリポジトリオブジェクトを選択します。
3. CWM ファイルを保存します。
 - a. **[保存]** ボタンをクリックします。

リポジトリマネージャに、**[保存]** ダイアログが表示されます。
 - b. 変更リスト XML ファイルを保存する必要があるターゲットディレクトリに移動します。
 - c. **[ファイル名]** フィールドに、保存する変更リストファイルの名前を入力します。

リポジトリマネージャによって、ファイルに拡張子が付けられます (*.CWM.xml)。
 - d. **[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 CWM.xml ファイルが保存され、ターゲットの場所への書き込み中は進行状況バーが表示されます。

注: エディタで CWM XML ファイルを開いて内容を確認します。

付録 A

デザインオブジェクトリファレンス

この付録では、以下の項目について説明します。

- [概要, 64 ページ](#)
- [リポジトリマネージャでサポートされているデザインオブジェクト, 64 ページ](#)
- [デザインオブジェクトの依存関係, 66 ページ](#)
- [名前の変更が可能なデザインオブジェクト, 67 ページ](#)

概要

この付録では、リポジトリマネージャで管理できるデザインオブジェクトのリファレンスを提供します。

リポジトリマネージャでサポートされているデザインオブジェクト

リポジトリマネージャを使用すると、オペレーショナル参照ストア内のメタデータを管理できます。以下の表に、リポジトリマネージャを使用して管理できる具体的なデザインオブジェクトの詳細リストを示します。この表では、リポジトリマネージャで以下の操作がサポートされるかどうかも示します。

- デザインオブジェクトのタイプの**検証**。
- リポジトリ間でのデザインオブジェクトの**コピー**。

さらに、『*Multidomain MDM の設定ガイド*』（またはその他のドキュメント）内の特定の要素について詳述したトピックも記載します。

以下の表は、リポジトリマネージャによって管理できるデザインオブジェクトの一覧です。

Informatica MDM Hub コンポーネント	検証	昇格、インポート、エクスポート
マッピング	○	○
クレンジング		
- プロセスサーバー	○	×
- クレンジング関数	○	○
クエリ		
- クエリ	○	○
- カスタムテーブルを使用したカスタムクエリ	○	×
パッケージ	○	○
- パッケージ	○	○
- カスタムクエリを使用したパッケージ	○	×
スキーマ		
- ベースオブジェクトテーブルおよびカラム	○	○
- 一致ルール、ルールセット、パスコンポーネントなどの一致	○	○
- 外部一致	○	○
- 検証	○	○
- メッセージトリガ	○	○*
- リレーション	○	○
- ステージングテーブル	○	○
- ベースオブジェクトのカスタムインデックス	○	○
信頼		
- ソースシステムの信頼	○	○
- カラムの信頼	○	○
システムの状態	○	○
バッチグループ（カスタムオブジェクトを除く）	○	○
メッセージキュー	一部（メッセージトリガのみ）	一部（メッセージトリガのみ）
セキュリティアクセスマネージャ	○	○

Informatica MDM Hub コンポーネント	検証	昇格、インポート、エクスポート
ユーザーオブジェクト	○	一部
階層マネージャ	○	○
ビジネスエンティティ/ビジネスエンティティ設定		
- ビジネスエンティティ	○	○
- ビジネスエンティティサービス	○	○
- REST ビジネスエンティティサービス設定	○	○
- SearchableCO ビジネスエンティティサービス設定	○	○
- WriteCO ビジネスエンティティサービス設定	○	○

関連項目：

- [「メタデータの検証」 \(ページ 18\)](#)
- [「リポジトリ間での変更の昇格」 \(ページ 28\)](#)
- [「デザインオブジェクトのインポート」 \(ページ 48\)](#)
- [「リポジトリのエクスポート」 \(ページ 55\)](#)
- [「変更リスト XML ファイル内のデザインオブジェクトと変更」 \(ページ 70\)](#)

デザインオブジェクトの依存関係

以下の表に、デザインオブジェクト間の依存関係を示します。

オブジェクトタイプ	依存関係
システム	なし
ベースオブジェクト	<ul style="list-style-type: none"> - カラム - 一致設定 - 検証ルール <p>注:</p> <ul style="list-style-type: none"> - ステージングテーブルのルックアップ、一致パス、および検証ルールが他のデザインオブジェクトを参照している場合、そのデザインオブジェクトは必要な依存関係です。 - 外部キーは、必要な依存関係を関連するデザインオブジェクトに示しません。外部キーは、両方のデザインオブジェクトがインポートされる場合にのみ、インポートされます。インポート前に、インポートされない外部キーがユーザーに通知されます。
ステージングテーブル	ベースオブジェクト、カラム、システム

オブジェクトタイプ	依存関係
ランディングテーブル	カラム
クエリ	クエリで使用されているデザインオブジェクト（ベースオブジェクトなど）
パッケージ	クエリ
マッピング	<ul style="list-style-type: none"> - ランディングテーブル - ステージングテーブル
クレンジング関数	使用されている他のクレンジング関数。 注: クレンジング関数によってカスタム Java ライブラリのある関数がインポートされると、そのライブラリのすべての関数がインポートされます。
階層マネージャ	設定で使用されているベースオブジェクトおよびパッケージ
ビジネスエンティティ	ベースオブジェクト
ビジネスエンティティフィールド	<ul style="list-style-type: none"> - ベースオブジェクトのカラム - 親ビジネスエンティティノード
ビジネスエンティティの子ベースオブジェクト	親ベースオブジェクトと子ベースオブジェクトのリレーション
ビジネスエンティティの referenceOne および referenceMany 要素	参照されるベースオブジェクト
ビジネスエンティティサービス	ビジネスエンティティから生成された入力
writeCO ビジネスエンティティサービス	ビジネスエンティティの定義
searchCO ビジネスエンティティサービス	ビジネスエンティティの定義
階層マネージャ対応のベースオブジェクトに基づくビジネスエンティティ	階層やリレーションタイプなどの階層マネージャ項目

関連項目：

- [「依存関係」 \(ページ 10\)](#)
- [「依存関係の競合」 \(ページ 30\)](#)

名前の変更が可能なデザインオブジェクト

昇格またはインポート時に名前を変更できるのは、次のデザインオブジェクトのみです。

- ベースオブジェクト

- クレンジングライブラリ

注: 名前を変更できるのは、クレンジング関数ツールで作成した内部クレンジングライブラリ（ユーザーライブラリや Java ライブラリ）のみです。サードパーティのクレンジングエンジンの外部クレンジング関数の名前変更はサポートされていません。

付録 B

変更リストリファレンス

この付録では、以下の項目について説明します。

- [概要, 69 ページ](#)
- [変更リスト XSD ファイル, 69 ページ](#)
- [変更リスト XML ファイルのルートタグと属性, 70 ページ](#)
- [変更リスト XML ファイル内の変更のタイプ, 70 ページ](#)
- [変更リスト XML ファイル内のデザインオブジェクトと変更, 70 ページ](#)

概要

この付録では、Informatica MDM Hub の変更リスト XML ファイルのリファレンスを提供します。

注: 変更リスト XML ファイルは、その変更リスト XML ファイルに含まれていないシステムオブジェクト（管理システム、システムクレンジング関数、および階層マネージャの RBO オブジェクトなど）への参照を含むことができます。リポジトリマネージャでは、これらの参照が内部的に解決されます。

関連項目：

- [「変更リスト」 \(ページ 11\)](#)

変更リスト XSD ファイル

siperian-changelist.xsd ファイルは、変更リスト XML ファイルの構造を定義します。このファイルは、MDM Hub Resource Kit に含まれています。詳細については、『*Multidomain MDM のリソースキットガイド*』を参照してください。

変更リスト XML ファイルのルートタグと属性

タグ	説明
<changeList>	変更リスト XML ファイルのルートタグ 属性には以下のものがあります。 <ul style="list-style-type: none">- xmlns:java="http://java.sun.com"- xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"- xsi:noNamespaceSchemaLocation="siperian-changelist.xsd"- xsi:schemaLocation="http://java.sun.com java.xsd"- creationDate=ファイルが作成されたときのタイムスタンプ- description=ユーザーによって指定された説明。比較変更リストでは、デフォルトの説明にはソースリポジトリとターゲットリポジトリが記載されます（例: "Compare docs--ORS2 (source) to docs--ORS1 (target)"）- listType="creation"または"comparison"のいずれかの値になります。- name=ユーザーによって指定されたわかりやすい名前。- version=変更リスト XML スキーマのバージョン番号（バージョン=2 など）
<changes>	ターゲットリポジトリに適用される変更のリスト。

変更リスト XML ファイル内の変更のタイプ

リポジトリマネージャでは、**変更**とは、ターゲットリポジトリに対して実行される変更リスト内の操作を指します。例えば、変更によって、テーブルの追加、一致ルールでの設定の更新、パッケージの削除、別のクエリグループへのクエリの移動などが実行される可能性があります。以下の表に、変更リスト XML ファイルで定義される変更のタイプを示します。

変更	説明
addObjectType	プロパティが関連付けられている、指定されたデザインオブジェクトを追加します。
modifyObjectType	プロパティが関連付けられている、指定されたデザインオブジェクトを追加します。
deleteObjectType	プロパティが関連付けられている、指定されたデザインオブジェクトを追加します。
revertObjectType	階層マネージャのデザインオブジェクトに適用され、階層マネージャエンティティオブジェクトまたはリレーションオブジェクトを Informatica MDM Hub のベースオブジェクトに戻すプロセスを指します。

変更リスト XML ファイル内のデザインオブジェクトと変更

以下の表に、変更リスト XML ファイルに出現する可能性のある、デザインオブジェクトのタイプ、および関連する変更を示します。各デザインオブジェクトに関連付けられているプロパティの詳細な説明については、*Multidomain MDM の設定ガイド*のデザインオブジェクトのドキュメント、または必要に応じて他のドキュメ

ントを参照してください。ColumnDataType や ChangeError など、変更リスト XML ファイル内で指定される他のオブジェクトについては、「[変更リスト XSD ファイル](#)」 (ページ 69)を参照してください。

コンポーネント	デザインオブジェクト	変更
バッチ	BatchGroup	addBatchGroup
		deleteBatchGroup
		modifyBatchGroup
クレンジング	CleanseFunction	addCleanseFunction
		deleteCleanseFunction
		modifyCleanseFunction
	CleanseLibrary	addCleanseLibrary
		deleteCleanseLibrary
		modifyCleanseLibrary
階層マネージャ	HmBlob	addHmBlob
		deleteHmBlob
		modifyHmBlob
	HmColumnPackage	addHmColumnPackage
		deleteHmColumnPackage
		modifyHmColumnPackage
	HmEntityObject	addHmEntityObject
		revertHmEntityObject
	HmEntityType	addHmEntityType
		deleteHmEntityType
		modifyHmEntityType
	HmHierarchy	addHmHierarchy
		deleteHmHierarchy
		modifyHmHierarchy
	HMPackage	addHmPackage
		deleteHmPackage
	HmProfile	addHmProfile

コンポーネント	デザインオブジェクト	変更
		deleteHmProfile
		modifyHmProfile
	HmRelationshipObject	addHmRelationshipObject
		revertHmRelationshipObject
	HmRelationshipType	addHmRelationshipType
		deleteHmRelationshipType
		modifyHmRelationshipType
	HmSandbox	addHmSandbox
		deleteHmSandbox
		modifyHmSandbox
一致	MatchColumn	addMatchColumn
		deleteMatchColumn
		modifyMatchColumn
	MatchPathComponent	addMatchPathComponent
		deleteMatchPathComponent
		modifyMatchPathComponent
	MatchPathComponentFilter	addMatchPathComponentFilter
		deleteMatchPathComponentFilter
		modifyMatchPathComponentFilter
	MatchPopulation	addMatchPopulation
		modifyMatchPopulation
	MatchRuleSet	addMatchRuleSet
		deleteMatchRuleSet
		modifyMatchRuleSet
	PrimaryKeymatchRule	addPrimaryKeyMatchRule
		deletePrimaryKeyMatchRule
		modifyPrimaryKeyMatchRule

コンポーネント	デザインオブジェクト	変更
メッセージトリガ	メッセージトリガ	addMessageTrigger
		deleteMessageTrigger
		modifyMessageTrigger
パッケージ	パッケージ（カスタムクエリに基づくパッケージを含む）	addPackage
		deletePackage
		modifyPackage
	PackageColumn	modifyPackageColumn
クエリ	クエリ	addQuery
		deleteQuery
		modifyQuery
	QueryGroup	addQueryGroup
		deleteQueryGroup
		modifyQueryGroup
スキーマ	BaseObject	addBaseObject
		deleteBaseObject
		modifyBaseObject
		modifyCascadeUnmerge
	BaseObjectColumn	addBaseObjectColumn
		deleteBaseObjectColumn
		modifyBaseObjectColumn
	ForeignKey	addForeignKey
		deleteForeignKey
	Index	addIndex
		deleteIndex
		modifyIndex
	LandingTable	addLandingTable
		deleteLandingTable

コンポーネント	デザインオブジェクト	変更
	LandingTableColumn	modifyLandingTable
		addLandingTableColumn
		deleteLandingTableColumn
	Mapping	modifyLandingTableColumn
		addMapping
		deleteMapping
	StagingTable	modifyMapping
		addStagingTable
		deleteStagingTable
	StagingTableColumn	modifyStagingTable
		addStagingTableColumn
		deleteStagingTableColumn
	SystemColumnTrust	modifyStagingTableColumn
		modifySystemColumnTrust
		addSystemTable
セキュリティアクセス マネージャ	SystemTableColumn	systemTableColumn
	SystemTableOrderColumn	orderColumn
		addResourceGroup
	Role	deleteResourceGroup
		modifyResourceGroup
		addRole
	SecureResource	deleteRole
		modifyRole
		addSecureResource
検索	SearchableField	deleteSecureResource
		addSearchableField
		modifySearchableField

コンポーネント	デザインオブジェクト	変更
		deleteSearchableField
ソースシステム	DistinctSystem	addDistinctSystem
		deleteDistinctSystem
		modifyDistinctSystem
	ImmutableSystem	modifyImmutableSystem
	システム	addSystem
		deleteSystem
		modifySystem
検証	ValidationRule	addValidationRule
		deleteValidationRule
		modifyValidationRule

関連項目：

- [「リポジトリマネージャでサポートされているデザインオブジェクト」](#) (ページ 64)
- [「変更リスト XSD ファイル」](#) (ページ 69)

付録 C

MetCommand リファレンス

この付録では、以下の項目について説明します。

- [概要, 76 ページ](#)
- [MetCommand について, 76 ページ](#)
- [作業を開始する前に, 77 ページ](#)
- [使用方法, 77 ページ](#)
- [例, 79 ページ](#)
- [戻りコード, 80 ページ](#)
- [スクリプトの実行, 81 ページ](#)
- [MetCommand の拡張, 81 ページ](#)

概要

この付録では、MetCommand ユーティリティのリファレンスを提供します。

MetCommand について

MetCommand は、リポジトリマネージャ API のコマンドラインラッパーです。Metadata Manager API は、MDM Hub の実装内でのメタデータの管理に使用されます。

API 呼び出し	説明
<code>applyChangeList</code>	変更リストを適用します。
<code>createChangeList</code>	変更リストを作成します。
<code>getOrsMetadata</code>	指定されたファイルにメタデータをエクスポートします。
<code>validateChangeList</code>	変更リストを検証します。
<code>validateMetadata</code>	リポジトリを検証します。

これらの API の詳細については、*Multidomain MDM サービスの統合フレームワークガイド*および *Multidomain MDM Javadoc* を参照してください。

作業を開始する前に

MetCommand の使用を開始する前に、この節の説明をお読みください。

要件

MetCommand は Java プログラムです。したがって、JRE または JDK がインストールされている必要があり、OS パスに java コマンドが必要です。

接続の設定

Informatica MDM Hub に接続するためのプロパティは以下のファイルにあります。

MetCommand\source\resources\properties\SiperianConnection.properties

このファイルを編集し、Informatica MDM Hub のインストールのユーザー名、パスワード、および orsId を設定する必要があります。

- 使用しているアプリケーションサーバーのプロトコルをコメント解除します。
- 使用していないアプリケーションサーバーをコメントアウトします。
- アプリケーションサーバーが localhost がない場合は、localhost をアプリケーションサーバーのホスト名で置き換えます。
- orsId は、デフォルトで sourceOrsId として使用されます。これはコマンドライン引数でオーバーライドできます。orsId は、Hub コンソール内のデータベースツールに登録されている ORS の名前です。

使用方法

この節では、MetCommand の使用方法について説明します。

ヘルプの出力

コマンドラインからパラメータを指定せずにこのユーティリティを実行すると、metcommand.cmd ファイル (Windows) に使用方法に関する情報が表示されます。

```
>metcommand
usage: MetCommand
-applyChangeList          apply a changelist (-sourceXmlFilename, -targetOrsId)
-createChangeList         create a changelist (-sourceOrsId|-sourceXmlFilename,
                        -targetOrsId, -outputFilename)
-getOrsMetadata           export metadata to the specified file (-sourceOrsId,
                        -outputFilename)
-outputFilename <arg>    output file name
-password                Owner password
-propertiesFilename <arg> hub client properties file name
-rollbackToLast          rollback to last change
-sourceOrsId <arg>       source ors id
-sourceXmlFilename <arg> source ors file
-targetOrsId <arg>       target ors id
-validateChangeList       validate a changelist (-sourceXmlFilename, -targetOrsId)
-validateMetadata         validate a ORS (-targetOrsId)
```

コマンドラインの引数

以下の表に、コマンドライン引数を示します。

引数	説明	関連する引数
-applyChangeList	変更リストを適用します。	-sourceXmlFilename -targetOrsId -rollbackToLast -password
-createChangeList	変更リストを作成します。	-sourceOrsId または -sourceXmlFilename -targetOrsId -outputFilename
-getOrsMetadata	指定されたファイルにメタデータをエクスポートします。	-sourceOrsId -outputFilename
-outputFilename <arg>	出力ファイル名。	<arg>=valid filename
-password	指定した場合、MetCommand はコマンドウィンドウまたはシェルでユーザーに ORS スキーマ所有者のパスワードを入力するように求めます。	
-propertiesFilename <arg>	クライアントのプロパティファイル名。	<arg>=valid filename
-rollbackToLast	最後の変更にロールバックします。指定しなかった場合、applyChangeList のデフォルトは完全ロールバックになります。	
-sourceOrsId <arg>	ソースリポジトリの ORS ID。	<arg>=valid ORS ID
-sourceXmlFilename<arg>	ソースリポジトリ (changelist) のファイル名。	<arg>=valid filename
-targetOrsId <arg>	ターゲットリポジトリの ORS ID。	<arg>=valid ORS ID
-validateChangeList	変更リストを検証します。	-sourceXmlFilename -targetOrsId -password
-validateMetadata	ORS を検証します。	-targetOrsId

XML Over HTTP

注: プロパティは XML over HTTP を介して Hub と通信するように設定されています。metcommand.cmd ファイルは、このプロトコルをサポートする JAR ファイルのみを参照し、EJB や SOAP に必要な JAR ファイルは参照しません。これは、ここで他のプロトコルを使用することの利点が少なく、JAR ファイルユーザーは使用されているアプリケーションサーバーに依存していないためです。

プロキシユーザーのアクセス

現在プロキシユーザーが登録されている ORS の場合、リポジトリを検証したり変更リストを適用したりするために、スキーマ所有者のパスワードを求めるように MetCommand を設定する必要があります（-password 引数）。ターゲット ORS で MetCommand が正常に実行されるようにするためには、ユーザーが有効なパスワードを入力する必要があります。

以下に例を示します。

```
cmd /c metcommand -validateChangeList -targetOrsId %target_ors% -sourceXmlFilename  
%changelist_file% -password  
cmd /c metcommand -applyChangeList -targetOrsId %target_ors% -sourceXmlFilename  
%changelist_file% -password
```

パスワードのプロンプトは、他の MetCommand 操作では使用されません。プロキシユーザー設定の詳細については、『*Multidomain MDM のインストールガイド*』および『*Multidomain MDM の設定ガイド*』を参照してください。

変更リスト適用時の変更のロールバック

MetCommand で applyChangeList 引数を使用する場合、部分ロールバックを実行するには -rollbackToLast 引数を使用します。以下に例を示します。

```
cmd /c metcommand -applyChangeList -targetOrsId %target_ors% -sourceXmlFilename  
%changelist_file% -rollbackToLast -password
```

指定しなかった場合、applyChangeList のデフォルトは完全ロールバックになります。

例

この節では、MetCommand の使用例を示します。

メタデータの取得

```
metcommand -getOrsMetadata -sourceOrsId localhost-orcl-cmx_ors2  
ORS Metadata has been written to: localhost-orcl-cmx_ors2.change.xml
```

変更リストの作成

```
metcommand -createChangeList -sourceOrsId localhost-orcl-cmx_ors1 -targetOrsId  
localhost-orcl-cmx_ors2  
Change list has been written to: localhost-orcl-cmx_ors1.change.xml  
または  
metcommand -createChangeList -sourceXmlFilename localhost-orcl-cmx_ors1.change.xml  
-targetOrsId localhost-orcl-cmx_ors2 -outputFilename changelist.change.xml  
Change list has been written to: changelist.change.xml
```

変更リストの検証

```
metcommand -validateChangeList -sourceXmlFilename changelist.change.xml  
-targetOrsId localhost-orcl-cmx_ors2  
The change list is valid.
```

現在プロキシユーザーが登録されている ORS の場合（-password 引数が必要）

```
metcommand -validateChangeList -sourceXmlFilename changelist.change.xml -targetOrsId
localhost-orcl-cmx_ors2 -password
The change list is valid.
```

変更リストの適用

```
metcommand -applyChangeList -sourceXmlFilename changelist.change.xml -targetOrsId
localhost-orcl-cmx_ors2
The change list has been applied.
```

現在プロキシユーザーが登録されている ORS の場合（-password 引数が必要）

```
metcommand -applyChangeList -sourceXmlFilename changelist.change.xml -targetOrsId
localhost-orcl-cmx_ors2 -password
The change list has been applied.
```

RollbackToLast

```
metcommand -applyChangeList -sourceXmlFilename changelist.change.xml -targetOrsId
localhost-orcl-cmx_ors2 -rollbackToLast
A partial rollback will happen.
```

現在プロキシユーザーが登録されている ORS の場合（-password 引数が必要）

```
metcommand -applyChangeList -sourceXmlFilename changelist.change.xml -targetOrsId
localhost-orcl-cmx_ors2 -rollbackToLast -password
A partial rollback will happen.
```

メタデータの検証

```
metcommand -validateMetadata -targetOrsId localhost-orcl-cmx_ors2
The ORS is valid.
```

戻りコード

各 METCommand API 呼び出しオプションは整数値を返します（成功した場合は 0、失敗した場合は-1）。返された整数値は、スクリプト内でトラップまたは処理できます。

以下の表に、API 呼び出しオプションの失敗（呼び出しコード=-1）に関連付けられている意味を示します。

API 呼び出し	意味
validateMetadata	MET 検証が正常に開始されなかったか、重要度レベルがエラーおよび重大（FATAL）の検証メッセージが返されました。
validateChangeList	変更リストの検証が成功しませんでした。
createChangeList	変更リストが作成されませんでした。
applyChangeList	変更リストが適用されませんでした。
rollbackToLast	最後の変更へのロールバックが行われませんでした。
getOrsMetadata	ORS のメタデータがファイルに書き込まれませんでした。

スクリプトの実行

MetCommand は、ORS 間でのデザインオブジェクトの昇格を自動化するスクリプト内で実行できます。?

カスタムスクリプトの実行

カスタムスクリプトを実行する場合

1. 新たに作成したスクリプトを以下のディレクトリから実行します。
Windows:
resourcekit\samples\MetCommand
UNIX:
resourcekit/samples/metcommand
2. コマンドラインウィンドウにエラーメッセージが表示されていないか確認するか、出力をリダイレクトして Met API 呼び出しが実行されていることを確認します。

スクリプトの例

以下のコードは、さまざまな MetCommand 呼び出しを行う Windows バッチスクリプトを示しています。使用時には、実際のスキーマ名を使用してください。

```
REM Sample Windows batch script using METCommand
echo off
set target_ors=localhost-orcl-target_ors
set source_ors=localhost-orcl-newtest1
set changelist_file=changelist1.change.xml
cmd /c metcommand -validateMetadata -targetOrsId %source_ors%
IF NOT %ERRORLEVEL% == 0 GOTO METCOMMAND_ERRORRED
cmd /c metcommand -validateMetadata -targetOrsId %target_ors%
IF NOT %ERRORLEVEL% == 0 GOTO METCOMMAND_ERRORRED
cmd /c metcommand -createChangelist -targetOrsId %target_ors% -sourceOrsId %source_ors%
    -outputFilename %changelist_file%
IF NOT %ERRORLEVEL% == 0 GOTO METCOMMAND_ERRORRED
cmd /c metcommand -validateChangelist -targetOrsId %target_ors% -sourceXmlFilename
    %changelist_file%
IF NOT %ERRORLEVEL% == 0 GOTO METCOMMAND_ERRORRED

cmd /c metcommand -applyChangelist -targetOrsId %target_ors% -sourceXmlFilename
    %changelist_file%
IF NOT %ERRORLEVEL% == 0 GOTO METCOMMAND_ERRORRED
cmd /c metcommand -validateMetadata -targetOrsId %target_ors%
IF NOT %ERRORLEVEL% == 0 GOTO METCOMMAND_ERRORRED
GOTO DONE
:METCOMMAND_ERRORRED
ECHO MetCommand Failed
:DONE

ECHO Done
```

MetCommand の拡張

MetCommand のソースコードとビルドファイルは、リポジトリマネージャ API の使用例として Resource Kit に含まれます。これには必要に応じて変更や拡張を加えることができます。

索引

J

Java クレンジングアダプタ [13](#)

M

MetCommand [76](#)

S

Services Integration Framework (SIF) の要件 [12](#)
siperian-changelist.xsd ファイル [69](#)

い

依存関係 [10](#)
依存関係の競合 [30](#)
インポート
 Java クレンジングアダプタ [13](#)
 インポートについて [48](#)
 オブジェクトのサポート [64](#)
 階層マネージャの要件 [13](#)
 考慮事項 [49](#)
 コマンドボタン [50](#)
 ソースリポジトリ [50](#)
 ターゲットリポジトリ [51](#)
 タスクのサマリ [50](#)
 デザインオブジェクトの選択 [52](#)
 名前の変更 [52](#)
 変更の適用 [53](#)
 ユーザーイグジット [13](#)
 ロールバックオプション [39, 45, 53](#)

え

エクスポート
 エクスポートについて [55](#)
 考慮事項 [56](#)
 コマンドボタン [56](#)
 サポートされているオブジェクト [64](#)

か

階層マネージャのライセンス要件 [13](#)
格上げ
 サポートされているオブジェクト [64](#)
 視覚的
 選択されたデザインオブジェクトの昇格 [36](#)
 ユーザーイグジット [13](#)
 考慮事項 [31](#)
 昇格について [28](#)

き

競合
 依存関係の競合 [30](#)
 解決するためのアクション [30](#)
 競合インジケータ [35](#)
 検索 [38](#)
 プロパティの競合 [29](#)
共通ウェアハウスモデル (CWM)
 [インポート] タブ [59](#)
 [エクスポート] タブ [59](#)
 概要 [59](#)
 ファイルのインポート [60](#)
 ファイルのエクスポート [63](#)

け

検査
 重要度レベル [20](#)
 [情報] ペイン [23](#)
 スコープ [19](#)
 物理モデル [18](#)
 プロセス概要 [19](#)
 プロパティペイン [24](#)
 履歴 [25](#)
 論理モデル [18](#)
検証
 検証インジケータ [20](#)
 検証について [18](#)
 コマンドボタン [20](#)
 サポートされているオブジェクト [64](#)
検証結果
 保存 [25](#)

こ

コマンドラインユーティリティ [76](#)

し

視覚的 昇格
 デザインオブジェクトの階層 [34](#)
視覚的昇格
 [視覚] タブへの移動 [33](#)
システムオブジェクト [11](#)
重要度レベル [20](#)
昇格
 Java クレンジングアダプタ [13](#)
 依存関係の競合 [30](#)
 階層マネージャの要件 [13](#)
 競合インジケータ [35](#)
 視覚的
 関連するデザインオブジェクト [36](#)

昇格 (続く)

視覚的 (続く)

競合の検索 [38](#)

コマンドボタン [33](#)

コンテキストメニュー [36](#)

自動競合解決 [38](#)

手動競合解決 [37](#)

ソースリポジトリ、選択 [34](#)

ターゲットリポジトリ、選択 [34](#)

タスクのサマリ [32](#)

複数選択 [36](#)

プロパティパネル [36](#)

変更の適用 [39](#)

変更リストへの変更の保存 [39](#)

マークアップモード [38](#)

シナリオ [28](#)

選択的昇格 [29](#)

同期化昇格 [29](#)

プロパティの競合 [29](#)

変更のロールバック [39, 45, 53](#)

変更リスト

シミュレーションの実行 [45](#)

ターゲットリポジトリ、選択 [42](#)

タスクのサマリ [40](#)

変更の保存 [44](#)

そ

ソースリポジトリ [11](#)

た

ターゲットリポジトリ [11](#)

て

デザインオブジェクト

依存関係 [10](#)

デザインオブジェクトについて [10](#)

デザインオブジェクトの名前変更 [52](#)

は

排他ロック [17](#)

ひ

比較変更リスト [12](#)

ふ

プロパティの競合 [29](#)

へ

変更リスト

siperian-changelist.xsd ファイル [69](#)

XML ファイル [12](#)

作成変更リスト [12](#)

タイプ [12](#)

比較変更リスト [12](#)

変更の詳細 [44](#)

変更の説明 [43](#)

変更リストについて [11](#)

変更リストの適用 [45](#)

リファレンス [69](#)

変更リストの昇格

コマンドボタン [41](#)

変更内の移動 [43](#)

変更リスト XML ファイルを開く [43](#)

[変更リスト] タブへの移動 [41](#)

リポジトリの比較 [42](#)

め

メタデータの検証

検証チェック [23](#)

メタデータ

格納場所 [11](#)

検証 [23](#)

マスターデータベースのメタデータ [11](#)

メタデータについて [10](#)

リポジトリメタデータ [11](#)

ゆ

ユーザーイグジット [13](#)

り

リポジトリ

ソースリポジトリ [11](#)

ターゲットリポジトリ [11](#)

リスト [16](#)

リポジトリについて [11](#)

リポジトリマネージャ

移動 [15](#)

開始 [15](#)

コマンドボタン [16](#)

タブ [16](#)

リポジトリマネージャについて [9](#)

リポジトリリスト [16](#)