

**IBM InfoSphere Master Data Management
Collaboration Server**



マイグレーション・ガイド

バージョン10 リリース1

**IBM InfoSphere Master Data Management
Collaboration Server**



マイグレーション・ガイド

バージョン10 リリース1

お願い

本書および本書で紹介する製品をご使用になる前に、89ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM InfoSphere Master Data Management Collaboration Server (プログラム番号 5724-V51) のバージョン 10 リリース 1 モディフィケーション 0、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC19-3498-01
IBM InfoSphere Master Data Management Collaboration
Server
Migration Guide
Version 10 Release 1

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.11

© Copyright IBM Corporation 2000, 2012.

目次

前書き	v	V6.0.0 と V10.1.0 でシグニチャーに違いのあるメソッド	53
バージョン 10.1.0 の新機能	v	V5.2.1 でのみ利用可能なメソッド	54
IBM ソフトウェア・サポートへのお問い合わせ	ix	V5.3.0.1 でのみ利用可能なメソッド	55
第 1 章 マイグレーション	1	V9.0.0 / V10.1.0 でのみ利用可能なメソッド	55
第 2 章 入門	3	V5.3.2 / V10.1.0 でのみ利用可能なメソッド	60
第 3 章 旧バージョンからのマイグレーション	5	V6.5.0 / V10.1.0 でのみ利用可能なメソッド	65
マスター・データを内部形式から XML 形式にマイグレーション	7	V5.2.1 および V10.1.0 からのスクリプト操作の動作に関する変更	65
V6.0.0 から V10.1.0 へのマイグレーション	9	V5.3.0.1 および V10.1.0 からの既存のスクリプト操作の動作に関する変更	68
V9.1.0 から V10.1.0 へのマイグレーション	13	V6.0.0 および V10.1.0 からのスクリプト操作の動作に関する変更	70
V10.0.0 から V10.1.0 へのマイグレーション	16	バージョン間での Java API の違い	70
Global Data Synchronization の V1.3.x から InfoSphere MDM Collaboration Server V10.1 へのマイグレーション	19	Java API メソッドおよび列挙型	71
GDS 1.3.x からのマイグレーションの準備	20	新しい Java API インターフェース	73
WebSphere Product Center V5.2.1.4 からのエクスポート	26	データベースのチューニング	74
InfoSphere MDM Collaboration Server V10.1 へのインポート	36	データ・モデル・メタデータの変換	74
マイグレーションした GDS 環境の検証	44	XSLT トランスフォーマーによる修正	75
スクリプトのコンパイルとデバッグ	44	スクリプトの手動マイグレーション	78
第 4 章 InfoSphere MDM Collaboration Server スクリプトのマイグレーション	49	第 5 章 トラブルシューティング	81
バージョン間でのスクリプトの違い	49	マイグレーション・スクリプトのトラブルシューティング	81
V5.2.1 と V10.1.0 でシグニチャーに違いのあるメソッド	50	コンパイル済みマイグレーション・スクリプトの使用	81
V5.3.0 と V5.3.0.1 でシグニチャーに違いのあるメソッド	52	一般的なスクリプト・コンパイル・エラー	82
V5.3.0.1 と V10.1.0 でシグニチャーに違いのあるメソッド	52	一般的な実行時のエラーと問題	83
		実行時のエラーと問題の解決	83
		checkForCompileError.sh の使用	84
		マイグレーション問題のトラブルシューティング	85
		特記事項	89
		商標	91
		索引	93

前書き

本書は、IBM InfoSphere Master Data Management Server for Product Information Management バージョン 6.0.0/9.1.0 や IBM® InfoSphere® Master Data Management Collaboration Server バージョン 10.0.0 から IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 に、いくつかのサブステップとマイグレーションを使用してマイグレーションする場合に活用できます。

本書について

以下の情報には、ステップバイステップの手順、ベスト・プラクティス、パフォーマンス・チューニング、およびトラブルシューティングに関連した一般的な情報が含まれています。

対象読者

このマイグレーション・ガイドは、製品を実際の環境にデプロイする作業に関係するインストール担当者、システム管理者、および IBM サービス担当員を対象としています。

前提条件および関連情報

このマイグレーション・ガイドを使用する前に、以下の情報について理解しておいてください。

- InfoSphere MDM Collaboration Server インフォメーション・センター
- アプリケーション・サーバーの資料
- 使用するデータベース・ソフトウェアの資料

ご意見の送付方法

IBM にお客様のご意見をお寄せください。本書や InfoSphere MDM Collaboration Server に関するその他のドキュメンテーションについてコメントがあるなら、以下の書式を使用してお送りください。

<http://www.ibm.com/software/data/rcf/>

バージョン 10.1.0 の新機能

IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 には、多くの新しいフィーチャーおよび機能があります。

InfoSphere MDM Collaboration Server は、InfoSphere MDM Enterprise Edition and InfoSphere MDM Collaborative Edition に組み込まれています。

InfoSphere MDM Collaboration Server を使用すれば、企業は、戦略的ビジネス・イニシアチブのために、その組織全体で使用可能な製品情報の単一かつ最新のリポジトリを作成することができます。

製品に対する主な変更点には、以下のものがあります。

Advanced Catalog Management (ACM)

Advanced Catalog Management (ACM) for Commerce は、InfoSphere MDM Collaboration Server のソリューション資産です。このソリューションは InfoSphere MDM Collaboration Server にあらかじめ組み込まれている構成であり、WebSphere Commerce ユーザーであれば直ちに使用できます。データ・モデルが直ちに使用できる状態になっていて、WebSphere Commerce のワークフロー、セキュリティー、およびエクスポート機能が取り込まれます。このソリューションは、InfoSphere MDM Collaboration Server および Commerce の実装を高速化することを目的としています。このソリューション・アクセラレーターは、ユーザーや実装チームが変更したり拡張したりできます。詳しくは、『WebSphere Commerce Server』を参照してください。

IBM WebSphere Operational Decision Management (WODM) 統合

IBM WebSphere Operational Decision Management (WODM) の統合ソリューション。高度なビジネス・ルールを作成したり製品やバンドルなどと関連付けたりするための機能が提供されます。この統合ソリューションはソリューション・アクセラレーターであり、ユーザーや実装チームが変更したり拡張したりできます。

インライン編集

単一/複数編集で現行ポップアップ・エディターの代わりに多くの属性タイプに対するインライン編集機能が提供されます。これにより、属性間を移動できます。必要に応じて (デフォルトではない) リッチ・テキスト・エディターを開くメカニズムが備わっています。詳しくは、『マスター・データを処理するためのヒント』を参照してください。

キャッシングの表示およびカタログ

キャッシング・パフォーマンスを最適化するために、キャッシュ値を使用してノード ID とノード・パスをキャッシングします。詳しくは、『カタログ・キャッシングを有効にするキャッシュ値の設定』を参照してください。

リンク表示属性

リンク属性の表示名を選択できるようになります。詳しくは、以下の資料を参照してください。

- ルックアップ・テーブル属性と関係属性
- ルックアップ・テーブル型の属性値の編集
- リンク属性の使用

InfoSphere MDM Collaboration Server ユーザーのためのマイグレーション・サポート

- V6.0.0 から V10.1.0 へのマイグレーション
- V9.1.0 から V10.1.0 へのマイグレーション
- V10.0.0 から V10.1.0 へのマイグレーション
- マスター・データを内部形式から XML 形式にマイグレーション

Global Data Synchronization ユーザーのためのマイグレーション・サポート

- Global Data Synchronization バージョン 1.3.x から InfoSphere MDM Collaboration Server バージョン 10.1.0 へのマイグレーション

エラー・メッセージおよび警告メッセージの強化

意味のあるエラー/警告メッセージ・テキストが提供されます。詳しくは、『InfoSphere MDM Collaboration Server メッセージ』を参照してください。

ソフトウェア・サポート・アップグレード

以下のアップグレードがサポートされるようになりました。詳しくは、「System requirements」を参照してください。

スタック・アップグレード

WebSphere Application Server バージョン 8.0

DB2 10

UI の強化

ユーザー・インターフェースで強化された部分を中心に以下のリストで説明します。

左ナビゲーション・ペインのツリーが最新表示のときに縮小表示されない

最新表示ボタンを押したときに、これまでは縮小表示されていた左ナビゲーション・ペインの表示が保持されます。

UI での数値属性に対する「次の値の間」検索

新規リッチ検索 UI で数値属性に対して「次の値の間」検索を実行できます。詳しくは、『UI での検索の使用』を参照してください。

UI での OR 検索機能

新規リッチ検索 UI で任意の 2 つの検索述部の間に 'OR' 論理演算を使用できます。詳しくは、『UI での検索の使用』を参照してください。

バックグラウンドでの検索の実行

バックグラウンド・モードで検索要求を実行するサポートが追加されました。これにより、検索がバックグラウンドでジョブとして実行されている間に製品の GUI を使用して他の機能を実行できます。

単一編集ポップアップ画面でのアイテム・ナビゲーション

複数編集画面から選択された複数のエントリーに対して単一編集画面を起動した後で、この機能を使用すれば、単一編集画面でこれらの任意のエントリーに素早くナビゲートできます。複数編集画面に戻る必要はありません。

使用不可になっているボタンを非表示

ユーザーがアクセスできないボタンは非表示になります。

ボタンの動的レイアウト

「オプション」メニュー・ボタンは除去されました。「オプション」の下で以前使用できていたメニュー項目はすべてトップレベルのボタンとして使用できます。右コンテンツ・ペインの現行幅の範囲内に一部のトップレベルのボタンをデフォルト・レイアウトでは表示できない場合、オーバーフローしたボタンは、右端にある「その他」メニュー・ボタンを使用することで使用できます。詳しくは、『オブジェクトを編集するための画面』を参照してください。

検索画面の強化

以下の検索画面のトピックが更新されました。

- フラグ属性による検索
- カタログ検索の作成
- 階層検索の作成
- コラボレーション・エリア検索の作成

単純検索で、索引が付けられていない属性を含むすべての属性が検索可能
索引が付けられていない属性を検索できる XML データ表現が追加
されました。また、より理解しやすい形式で InfoSphere MDM
Collaboration Server データにデータベースから直接アクセスできる
ようになりました。

Global Data Synchronization (GDS) の強化

Global Data Synchronization で強化された部分を中心に以下のリストで説明
します。

GDS データプールのアップグレード

最新 GDSN データプール・スキーマがサポートされています。

旧バージョンからの修正のマイグレーション

バージョン 9.1.0 IF7 およびバージョン 10.0 FP1 からの修正がバ
ージョン 10.1.0 の GDS コード行にマイグレーションされます。

最新 GDSN 規格に調整するための BMS V2.8 アップグレード

InfoSphere MDM Collaboration Server の GDS コンポーネントがア
ップグレードされて、最新の GDSN 仕様である BMS V2.8 仕様に
準拠するようになりました。

対象リテラーのアウトバウンド XML において ID としてグローバル・ レベルの GLN を持っているときに複数のリテラー GLN を使用するこ とでサブスクライブする機能

ユーザーはこの機能を使用して、GLN を自分の組織において複数所
有する場合があるシナリオを処理できます。リテラーのデータプ
ールにおいて、リテラー内でこのように下位組織化されたもの
は、単一の組織全体の GLN によって識別されます。

「メッセージの管理」 > 「探索画面 (Explore Screen)」 において GTIN で 検索する機能

この機能は、「メッセージの管理」 > 「探索画面 (Explore Screen)」
モジュールで提供される検索機能を拡張したものです。GTIN を検
索条件として使用すると、ロジスティック階層の最高レベルにその
GTIN を持つすべての CIN がフェッチされます。

「メッセージの管理」 > 「探索画面 (Explore Screen)」 において「トランザ クション処理」 ドロップダウンに対してオプション「すべて」を指定して 検索する機能:

ここで述べられているトランザクション処理には、すべてのタイプ
の CIN メッセージ、「同期の停止」メッセージ、および RFCIN
メッセージが含まれます。オプション「すべて」は、トランザク
ション処理のワイルドカード検索を意味します。この検索を GTIN と
ともに正しく使用すれば、特定の GTIN に関して受信されたさまざ
まなタイプの CIN メッセージをより良く表示できます。

「メッセージの管理」 > 「探索画面 (Explore Screen)」 で情報提供者の検索ポップアップを使用する機能

ユーザーは、値を入力する代わりに、GDS システムで作成した情報提供者を検索して選択できます。

GDS に取引アイテムがない場合でもタイプ ADD、CORRECT、および CHANGE の CIN メッセージを柔軟に処理できる

GDS に取引アイテムがない場合でも初期ロード CIN を処理する機能です。詳しくは、『Catalogue Item Notification Validation』を参照してください。

失敗ケースに関して、より有用な情報を提供する API - *HashMap MessageItemHandler::createTradeItems()* での機能

このスクリプト API に対するエラー処理や例外処理が向上しました。

失敗ケースに関して、より有用な情報を提供する CIC 生成 API - *sendConfirmation()* および *sendCICConfirmationReasonCodes()* での機能

このスクリプト API に対するエラー処理や例外処理が向上しました。

任意のタイプの CIC を GDS UI から送信する機能

CIC (カタログ・アイテム確認) を送信するスクリプト API が、GDS に用意されています。これは、プロセスを自動化するため、実装によって使用できます。多くのケースで、さまざまな理由によりベンダーに届かなかった CIC を再送できないことについてビジネス・ユーザーは不満に思っていました。

メッセージ・アイテムの取引アイテムを GDS UI で作成または保存する機能

GDS システムに着信した取引アイテムと CIN メッセージを、ユーザーが処理するための機能が向上しました。

IBM ソフトウェア・サポートへのお問い合わせ

IBM ソフトウェア・サポートは、製品の問題点、よくある質問への回答、および再ディスカバリーの実行で支援をします。

始める前に

技術情報などのその他の自助型の選択肢を使用して答えまたは解決策を見つけようとした後、IBM サポートに問い合わせることができます。IBM サポートに連絡を取る前に、お客様の会社に有効な IBM 保守契約があることが必要です。さらに IBM に問題を提出する担当者は、その権限がある担当者でなければなりません。使用可能なサポートのタイプについては、「*Software Support Handbook*」の『Support portfolio』のトピックを参照してください。

IBM ソフトウェア・サポートに連絡を取る場合には、IBM からどのようなサポートが受けられるのかを理解することが重要です。以下のサポートが受けられます。

- 製品の特定の質問: 製品の正規のトレーニングを受けた場合には、特定の質問または気がかりな点を送信できます。
- 欠陥のサポート: 製品の欠陥を解決するか次善策を提示するために必要な分析および必要な作業。

- スクリプト操作: スクリプト操作に関する質問の場合、またはスクリプト操作の欠陥の可能性を報告する場合、質問または問題をおおよそ 5 行以下のコードで簡潔に記述する必要があります。
 - IBM は、カスタム・スクリプトのデバッグおよび分析のサポートは行いません。
- 機能拡張の要求: 製品に追加される新機能の要求。

IBM ソフトウェア・サポートの担当範囲には含まれず、お客様の会社の販売または専門サービス担当員に連絡することが必要な要求には、次の要求があります。

- 各種作業: 複雑なインストールや複数のプラットフォームへのインストールなど、インストール中またはインプリメンテーション処理中に IBM が相当関わる必要がある作業。
- トレーニングを受けていないお客様へのサポート: IBM にステップごとの手引きを要求しているものの、必要な IBM 製品のトレーニングを受けていない場合。IBM 研修サービスでは、スタッフをさらにトレーニングすることが必要な場合のカスタマー・トレーニングを提供しています。受講できるコースに関する具体的な情報については、IBM Web サイトを確認してください。
- カスタム・スクリプトまたはスクリプト記述: お客様が作成したか、またはデバッグが必要なスクリプトについては、サポートは提供していません。プロフェッショナル・サービスにより作成されたカスタマイズ・スクリプトについては、サービスの作業完了後はすべてお客様が責任を持つことになり、お客様が保守する必要があります。カスタム・スクリプトの作成またはデバッグのためにテクニカル・サポートを利用することはできません。
- データ・リカバリー: 定期的にデータをバックアップするように強くお勧めします。
- カスタマー・データを必要とするパフォーマンスまたはチューニングの支援や推奨方法の提案。

手順

1. 問題に関する技術情報が既に存在しないかどうかを確認します。技術情報の表示に関する追加情報は、知識ベースの検索を参照してください。
2. 友人や社内のサポート・チームが支援を行えないかを確認します。
3. 問題を明記し、背景情報を収集し、問題の重大度を判断します。支援が必要な場合は、「ソフトウェア・サポート・ハンドブック」のソフトウェア・サポートにご連絡いただく前に参照してください。

必要な背景情報

- すべての構成設定および情報。
 - 期待される動作と現在の動作。
 - 最近インプリメントしたすべての変更または修正。
 - 問題を再現できるかどうかの判別。
 - 発生している問題に至るステップの明示。
4. 診断情報を収集します。
 - IBM Support Assistant の基本的なデータ収集機能を使用して重要なログおよび構成ファイルを収集することもできます。追加情報については、www.ibm.com/software/support/isa/ を参照してください。

- ワークベンチから、お客様の IBM Support Assistant Lite コレクター・ツールを作成します。
 - a. IBM Support Assistant ワークベンチで、製品のアドオンをダウンロードします。
 - b. 「データの収集と送信 (Collect and Send Data)」をクリックします。左側に 3 つのオプションが表示されます。
 - c. 「IBM Support Assistant Lite を使用する (...Using IBM Support Assistant Lite)」を選択します。
 - d. コレクターのそれぞれのオプションに入力して、生成されたコレクターを作成します。
 - コレクターを選択します (インストールされた製品のアドオンが表示されない場合、Must Gather 文書がないため、IBM Support Assistant Lite コレクターを作成できません)。
 - ターゲット・ディレクトリーを指定します。
 - ファイル名を指定します。
 - 「エクスポート (Export)」ボタンをクリックします。
 - e. エクスポートされた IBM Support Assistant Lite コレクターを、FTP またはその他の利用可能なファイル転送プロトコルを使用して、リモート・システムに転送します。
 - f. エクスポートされた IBM Support Assistant Lite コレクターを、ターゲットの場所に unzip します。
 - g. JAVA_HOME をセットアップするか、または Java™ が PATH ステートメントにあることを確認します。
 - h. リモート・システムのツールの IBM Support Assistant Lite ディレクトリーから、Windows の場合は **runISALite.bat** コマンド、UNIX および LINUX の場合は **runISALite.sh** スクリプトを実行します。
 - i. コレクション・プロンプトに、適切に応答します。
 - j. 収集されたファイルを確認し、IBM サポートに送信します。
- 5. 以下のいずれかの方法で問題を IBM ソフトウェア・サポートに送信します。
 - オンライン: (推奨)
 - a. IBM InfoSphere Master Data Management Collaboration Server サポート・サイトの左側ナビゲーション・バーにある「タスクの選択」セクションの「サービス・リクエスト (SR)」をクリックします。
 - b. 「ESR / SR」をクリックし、ESR Web サイトで提供される指示に従います。

ESR の使用と PMR の送信については、以下の指針とヒントに従います。

- 文書は、IBM サポート・サービスが使用できるように PMR に直接添付します。
- ESR は表示されたすべての PMR のリストを管理するため、それらの状況を確認するためにすぐにアクセスできます。
- ESR を使用すると、自動 PMR 通知をセットアップできます。
- 開いた PMR の更新情報は、ESR から直接 IBM サポート・サービスに送信します。

- PMR を送信する際、以下のように重大度レベルを示す必要があります。これによりサポート・サービスは、お客様が開いたその他の PMR との関係性を考慮した上で、サポートに使用するリソースに優先順位付けをすることができます。

重大度 1

業務へ重大な影響: これは、プログラムを使用できないために業務に重大な影響があることを示します。この状態は、直ちに解決する必要があります。

注: 業務へ重大な影響があるとは、通常、実動システムがダウンしていて、IBM ソフトウェア・サポートがお客様の営業時間外に作業を行うために、1日24時間週7日体制で電話連絡を取れるようにするための手配をお客様にさせていただく必要があるということを意味します。これにより、調査を行う技術者による問い合わせを受け付けたり、ご使用の実動インスタンスでテストを実行したりすることができます。

重大度 2

業務へ大きな影響: これは、プログラムは使用可能であるが、非常に制限されていることを示します。

重大度 3

業務へ多少の影響: これは、プログラムは利用可能であるが、重要度の低い機能 (業務への影響は重大ではない) が使用不可能になっていることを示します。

重大度 4

業務へ最小限の影響: これは、問題による業務への影響がほとんどないか、問題の妥当な回避策が実施されたことを示します。

- 機能拡張要求 (一般的な新規製品機能の要求で、特定のカスタマイズとは関連していないもの) を開くには、次のようにします。
 - 重大度を 2、3、または 4 (要求内容が業務に貢献する可能性の程度による) に設定して PMR を開くと、要求は IBM 製品管理に送信され、検討されます。追跡するための機能拡張要求番号を受け取ります。これは PMR 番号を置き換えるものです。通常、PMR を閉じるには、お客様の承認が必要です。
- 電話を使用: 国別の呼び出し用電話番号については、「ソフトウェア・サポート・ハンドブック」の電話によるサポート (ボイス・アクセス) のページを参照してください。

次のタスク

送信する問題がソフトウェアの欠陥である場合、または資料に欠落または不正確な部分がある場合には、IBM ソフトウェア・サポートは、Authorized Program Analysis Report (APAR) を作成します。APAR には、問題が詳細に記載されています。IBM ソフトウェア・サポートは、APAR が解決され、フィックスが配信されるまでに実施できる次善策を、可能な限り提供します。IBM は、解決された APAR を IBM Support Web サイトで毎日公表しているため、同じ問題が発生している他のユーザーは同じ解決方法を活用することができます。

第 1 章 マイグレーション

IBM InfoSphere Master Data Management Collaboration Server の旧バージョンを使用している場合、そのバージョンから最新のバージョンにマイグレーションできます。

このインストール情報は、「IBM InfoSphere Master Data Management Collaboration Server マイグレーション・ガイド」にも記載されています。これは、PDF ファイルのブック形式で入手できます。PDF ファイルをコンピューターにダウンロードするには、リンクを右クリックして、「**対象をファイルに保存**」を選択します。

第 2 章 入門

旧バージョンから IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1 にマイグレーションする場合、マイグレーション・スクリプトを使用して現在のインストール済み環境のデータベース・スキーマに変更を加える必要があります。

データベース内で表と索引を作成したり変更を加えたりするための適切な許可がなければなりません。また、\$TOP ディレクトリーのサブディレクトリー内でファイルを作成したり削除したりするための適切な許可もなければなりません。

マイグレーションには、基本的なステップとして以下のことが含まれます。

- 十分なデータベース表スペースが使用可能であることを確認します。
- データベース・スキーマを使用しているインスタンスを停止します。
- 既存のデータベース・スキーマをバックアップします。
- InfoSphere MDM Collaboration Server V10.1 をインストールして、旧インスタンスが使用していたものと同じデータベース・スキーマを指します。
- 該当するマイグレーション・スクリプトを実行して、データベース・スキーマに変更を加え、InfoSphere MDM Collaboration Server V10.1 レベルにします。

第 3 章 旧バージョンからのマイグレーション

特定の旧バージョンから IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 にマイグレーションできます。

バージョン 10.1.0 では、以下の 3 つのマイグレーション・パスがサポートされています。

- V6.0.0 フィックスパック 12 から V10.1.0 へのマイグレーション
- V9.1.0 フィックスパック 9 から V10.1.0 へのマイグレーション
- V10.0.0 フィックスパック 3 から V10.1.0 へのマイグレーション

上位のリリース (同じリリースのフィックスパックではない) にマイグレーションする場合は、JAR ファイルのバージョンが一致しない状態を避けるために、新規インストール・ディレクトリーを使用する必要があります。

旧バージョンのフィックスパックからマイグレーションする場合は、Excel スクリプトで削除された API を使用していないことを確認します。以下の API は最新の POI JAR ファイルでは使用できません。

- `public String getExcelCellEncoding();`
- `public void setEncoding(String encoding);`
- `public String getExcelCellEncoding(Object cellObj) throws AustinException;`

制約事項:

1. InfoSphere MDM Collaboration Server Excel Parser のサポートは、サード・パーティー JAR ファイル、POI-3.2-FINAL.jar を介して行われます。この JAR ファイルには、1 行に 255 個を超える列を使用することに対する制限についての修正がありません。
2. この JAR では Excel 2010 .xlsx がサポートされていません。

重要: マイグレーション・スクリプトを実行する前に、使用しているデータベースのバージョンが正しいことを確認するようにしてください。ご使用の製品バージョンに応じて、詳細についてシステム要件 (http://www-01.ibm.com/software/data/infosphere/mdm_server_pim/requirements.html および <http://www-01.ibm.com/support/docview.wss?uid=swg27035486>) を参照してください。InfoSphere MDM Collaboration Server データベース・ユーザーが、インストール・ガイドで説明されている適切な特権を持っていることを確認する必要があります。セキュリティ・ポリシーによるデータベース・ユーザー特権の変更に関しては、DBA チームに問い合わせてください。

システム要件で示されている DB2 データベース・バージョンまたは Oracle データベース・バージョンを使用する必要があります。ご使用の製品バージョンに応じて、詳細についてシステム要件 (http://www-01.ibm.com/software/data/infosphere/mdm_server_pim/requirements.html および <http://www-01.ibm.com/support/docview.wss?uid=swg27035486>) を参照してください。ご使用の DB2 または Oracle

データベースを、サポートされているバージョンにアップグレードするには、DB2 または Oracle の資料を参照してください。

DB2 および Oracle でカスタム表スペースを使用している場合の InfoSphere MDM Collaboration Server のマイグレーション・ステップ

デフォルトの USERS または INDX ではなく、カスタム表スペース名を使用している場合、InfoSphere MDM Collaboration Server へのマイグレーションを開始する前に、以下の SQL ファイル内の表スペース名を手動で変更する必要があります。

マイグレーションを開始する前に、データベースに十分なスペースがあることを確認する必要があります。

監査ログは V10.1.0 にマイグレーションされません。V9.1.0 で監査ログの新しい記述形式が導入されたため、V9.1.0 より前のリリースで表示されていた監査ログの記述は V10.1.0 では表示できません。したがって、マイグレーション中にデータベースから削除されます。マイグレーション前のリリースを使用すれば、古い監査ログを表示できます。

確実にマイグレーションを正常に実行するには、以下のステップを手動で実行しなければなりません。

1. 以下に示されるプロパティ・ファイルを手動でマイグレーションします。プロパティ・ファイルはローカル・ファイル・システムの一部なので、マイグレーション・メソッドはプロパティ・ファイルのエクスポートをサポートしません。旧リリースでカスタマイズした各プロパティは、対応する新規バージョンのファイルに合うように変更してください。V10.1.0 で新たに導入されたパラメーターがあるため、旧バージョンのプロパティ・ファイルを新規バージョンのプロパティ・ファイルに上書きすることはしないでください。各プロパティ・ファイルの詳細については、InfoSphere MDM Collaboration Server の資料を参照してください。
 - \$TOP/etc/default/common.properties
 - \$TOP/etc/default/logs.xml
 - \$TOP/etc/default/docstore_mount.xml
 - \$TOP/etc/default/user_jars.list
2. \$TOP/public_html/suppliers/<cmp code>/ctg_files ディレクトリーの内容を、各会社の新しいインスタンスに手動でコピーします。これらのファイルはローカル・ファイル・システムの一部なので、マイグレーション・メソッドは ctg_files ディレクトリーのエクスポートをサポートしません。
3. 必要なユーザー JAR ファイルを新しいインスタンスに手動で追加します。これらのファイルはローカル・ファイル・システムの一部であるため、マイグレーション・メソッドはユーザー JAR ファイルのエクスポートをサポートしません。ご使用のデプロイメントへの JAR ファイルの追加について詳しくは、『カスタム・ユーザー .jar ファイルのデプロイ』を参照してください。
4. デフォルトのリッチ検索結果レポート・スクリプトに対して変更を加えてカスタマイズを行った場合は、それを手動で再実行します。InfoSphere MDM Collaboration Server V10.1.0 へのマイグレーションによって、InfoSphere MDM Collaboration Server 文書ストア・ディレクトリー /scripts/report にある既存のスクリプトが上書きされます。このスクリプトは、InfoSphere MDM

Collaboration Server オブジェクト検索結果を Excel シートにエクスポートするために使用されます。このスクリプトの詳細については、検索結果の Excel へのエクスポートを参照してください。

マスター・データを内部形式から XML 形式にマイグレーション

旧バージョンから IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 へのマイグレーション時には、マスター・データを XML に変換する必要があります。

始める前に

1. インストール・ガイドの説明に従って XML 表スペースとバッファ・プールを作成します。
2. インストール・ガイドに記載されているとおりの適切なデータベース特権が InfoSphere MDM Collaboration Server アプリケーションのデータベース・ユーザーにあることを確認します。
3. ローカル・サーバー上の InfoSphere MDM Collaboration Server アプリケーションを停止します。
4. `delete_old_versions.sh` スクリプトを実行して旧バージョンをすべて削除します。 `code` パラメーターと `end_date` パラメーターには値として自分の会社コードと現在日付を使用します。
5. オプション - Java プロセスに使用できるメモリーを増やします。データ変換スクリプトを実行する処理では、多くのメモリーが必要となります。
`$TOP/bin/conf/env_settings.ini` ファイルにある `custom_java_options` パラメーターを更新して Java プロセスに 2 GB のメモリーを指定することをお勧めします。 `custom_java_options` パラメーターを変更したら `configureEnv.sh` を実行して、更新した新規メモリー設定を有効にしてください。

```
custom_java_options=-Xmx2048m -Xms256m
```

このタスクについて

データ・マイグレーション・スクリプトには、実行スレッド数の 2 倍の DB 接続数が必要です。例えば、コマンド行オプションで 10 個のスレッドを使用する場合は、20 個の DB 接続が必要となります。DB 接続数を構成するには、以下の 2 つの構成パラメーターを使用する必要があります。

- `db_maxConnection`
- `db_maxConnection_default`

どちらのパラメーターも `common.properties` ファイルにあります。これらのパラメーターの値がすでに、必要な DB 接続数と同じかそれ以上になっている場合は、その値を変更する必要はありません。これらのパラメーターの値が必要な DB 接続数より小さい場合は、速度を上げるために値を増やす必要があります。

カタログにあるマスター・データ・エントリーの数 が 100,000 個を超える場合は、スレッド数を増やしてください。例えば、50 個または 100 個のスレッドを使用すれば、データ・マイグレーションが完了するまでの時間を短縮できます。

InfoSphere MDM Collaboration Server で使用できるスレッドの最大数は、以下の変

動要素によって決まります。最適なスレッド数を評価するには、テスト環境でマイグレーション・スクリプトを実行してください。

- 使用可能なプロセッサ・コアの数
- 定義されている仕様の数
- 使用可能メモリー

InfoSphere MDM Collaboration Server 環境で定義されている仕様はすべてデータ変換処理で取得されて使用されます。仕様数が 100 個を超える場合は、仕様キャッシング・パラメーターを適切な値に更新し、データベース・トリップの頻度を下げ、スクリプト・パフォーマンス全体を向上させてください。

手順

1. 次のコマンドを使用して InfoSphere MDM Collaboration Server における仕様カウントを取得します。

```
$PERL5LIB/runSQL.pl --sql_command="select count(*) from spc;"
```

2. 以下のパラメーターを、ステップ 1 の照会で取得した値で更新します。これらのパラメーターは、`$TOP/etc/default` ディレクトリーの `mdm-cache-config.properties` ファイルにあります。

```
specCache_KEY_START_VERSION_TO_VALUE.maxElementsInMemory=<spec count>
```

```
specCache_KEY_TO_CURRENT_START_VERSION.maxElementsInMemory=<spec count>
```

```
specCache_KEY_VERSION_TO_START_VERSION.maxElementsInMemory=<spec count>
```

注: `mdm-cache-config.properties` ファイルを更新した後に `configureEnv.sh` スクリプトを実行すると、パラメーターがデフォルト値で上書きされます。

3. データ・マイグレーション・シェル・スクリプト `migrateDataToXml.sh` を実行します。このファイルは `$TOP/bin/migration` ディレクトリーにあります。

```
migrateDataToXml.sh --company=<company name> --threads=<number of threads>
```

ここで

--company

InfoSphere MDM Collaboration Server へのログオンに使用される会社コードを指定します。

--threads

スレッドの数を指定します。速度を上げるために多くのスレッドを使用してください。ただし、十分な数の DB 接続があることを確認してください。デフォルトは 1 です。推奨スレッド数として少なくとも 10 は指定してください。

例えば、以下のようにします。

```
migrateDataToXml.sh --company=ibm --threads=50
```

次のタスク

コンソールに表示されたエラーや、`$TOP/logs/default` ディレクトリーの `default.log` ファイルに記録されたエラーを参照すれば、スクリプトが意図したとおり動作したかどうかを確認できます。マイグレーションに失敗したエントリーのリストは実行終了時に表示されます。

データ・マイグレーション・スクリプトはデータベースをバッチで更新します。エラーが発生すると、バッチ全体がロールバックされます。そのため、マイグレーションに失敗したエントリーの数は実際の失敗数よりも多くなります。

common.properties ファイルにある **index_regeneration_batch_size** パラメーターの値を 1 に更新してデータ・マイグレーション・スクリプトを再実行し、実際の失敗数でエントリーを絞り込んでください。データ・マイグレーション時に発生したエラーについて考えられる理由と、推奨処置が表で示されます。

表 1. エラーについて考えられる理由と、推奨処置

データ・マイグレーション時のエラーの理由	推奨処置
データ・マイグレーションで次のエラーが表示される: java.sql.SQLException: ORA-00600: internal error code, arguments: [KGHALP1], [0x00000000], [], [], [], [], [], [], [], []	使用しているデータベースのパッチ・レベルが正しいことを確認してください。必要に応じて、データベースをアップグレードしてからデータ・マイグレーション・スクリプトを再実行してください。
データ・マイグレーションで次のエラーが表示される: java.sql.SQLException: ORA-31061: XDB error: XML event error ORA-19202: Error occurred in XML processing In line 46 of orastream: LPX-00241: entity reference is not well formed	エントリーに含まれるいずれかの属性値に、"&" 文字または無効な値が含まれています。InfoSphere MDM Collaboration Server のアイテム編集インターフェースでエントリーを更新して無効な値を除去し、データ・マイグレーションを再試行してください。

V6.0.0 から V10.1.0 へのマイグレーション

IBM® InfoSphere™ Master Data Management Server for Product Information Management バージョン 6.0.0 から IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 にマイグレーションするには、該当するマイグレーション・スクリプトを実行する必要があります。

始める前に

InfoSphere MDM Server for PIM V6.0.0 から InfoSphere MDM Collaboration Server V10.1.0 にマイグレーションする前に、V6.0.0 のフィックスパック 12 を使用していることを確認する必要があります。フィックスパック 1 から 11 までのいずれかのレベルで V6.0.0 を使用している場合には、「IBM InfoSphere Master Data Management Server for Product Information Management, Version 6.0.0 Release notes for fix pack 12」の資料に記載された説明に従って、V6.0.0 フィックスパック 12 にマイグレーションする必要があります。

以下のデータベース構成を確認する必要があります。

1. すべての XML 関連の表スペースがインストール・ガイドのとおり作成されていて、InfoSphere MDM Collaboration Server データベース・ユーザーに、新規表スペースを使用するための適切な特権がある。
2. 十分なフリー・スペースが表スペースにある。

デフォルトの `USERS` や `INDX` の代わりにカスタム表スペース名を使用している場合は、InfoSphere MDM Collaboration Server へのマイグレーションを開始する前に、以下の SQL ファイル内の表スペース名を手動で変更する必要があります。

表 2. DB2 または Oracle をバックエンド・データベースとして使用するときに変更する SQL ファイル

DB2 がバックエンド・データベースである場合に 変更する SQL ファイル	Oracle がバックエンド・データベースである 場合に 変更する SQL ファイル
<p>\$TOP/src/db/schema/dbscripts/db2 ディレクトリーにある以下の SQL ファイルで、表スペース名を <code>USERS</code>、<code>INDX</code>、および <code>BLOB_TBL_DATA</code> から、該当するカスタム表スペース名に変更します。</p> <pre>create_new_tables_of_650.sql create_new_tables_of_1000.sql create_tables_of_gds.sql recreate_coa_for_edit_flag.sql migrateItd.sql migrateDb1.sql migrate0bh.sql migrateSst.sql migrateCad.sql</pre>	<p>\$TOP/src/db/schema/dbscripts/oracle ディレクトリーにある以下の SQL ファイルで、表スペース名を <code>USERS</code>、<code>INDX</code>、および <code>BLOB_TBL_DATA</code> から、該当するカスタム表スペース名に変更します。</p> <pre>create_new_tables_of_650.sql create_new_tables_of_1000.sql create_tables_of_gds.sql recreate_coa_for_edit_flag.sql migrateItd.sql migrateDb1.sql migrate0bh.sql migrateSst.sql migrateCad.sql</pre>

DB2 データベースに関する注意事項

旧バージョンの DB2 から、サポートされているバージョンにアップグレードする場合、`CUR_COMMIT` データベース構成パラメーターが有効になっていることを確認する必要があります。このパラメーターは、旧バージョンからアップグレードされるデータベースに対してデフォルトでは無効になっているため、手動で設定する必要があります。

また、InfoSphere MDM Collaboration Server のマイグレーションを開始する前に、データベース管理者に以下の SQL コマンドの実行を依頼する必要があります。

```
alter tablespace blob_tbl_data convert to large;
alter tablespace blob_tbl_data file system caching
```

`migrateFrom600FP.sh` マイグレーション・スクリプト内の `_ERROR_LOGS` 変数に変更を加えて、エラー・ログ・ファイルの場所を構成できます。デフォルトでは、この変数は `$TOP/logs/errfile.log` に設定されます。

このタスクについて

以下は、InfoSphere MDM Server for PIM V6.0.0 からマイグレーションする唯一の方法です。環境エクスポート/インポート・ツールの使用など、他の手段をこのマイグレーションで使用することはできません。

手順

1. 既存のデータベース・スキーマをバックアップします。
2. InfoSphere MDM Collaboration Server V10.1.0 を別のディレクトリーにインストールします。
3. データベースをセットアップします。
 - a. DB2 データベースの場合は、インストール・ガイドの『DB2 データベースのセットアップ』に記載されているとおりに新規のバッファー・プールと表スペースを作成します。

注: XML_DATA 表スペースのサイズは、TCTG_ITD_ITEM_DETAIL 表スペースによって占有されているスペースと同じでなければなりません。この表のおおよそのサイズは、次の照会を使用して算出できます。

```
select (fpages * 16) / 1024 megabytes from syscat.tables
where tabschema = current schema
and tabname like 'TCTG_ITD_ITEM_DETAIL';
```

- b. Oracle データベースの場合は、インストール・ガイドの『Oracle データベースのセットアップ』に記載されているとおりに、新規表スペースを作成して Oracle XML DB コンポーネントをインストールします。
4. InfoSphere MDM Collaboration Server V10.1.0 インスタンスが V6.0.0 フィックスパック 12 データベース・スキーマを指すようにします。
 5. migrateFrom600FP.sh マイグレーション・スクリプトを実行します。このスクリプトは \$TOP/bin/migration ディレクトリーにあります。
 6. 『マスター・データを内部形式から XML 形式にマイグレーション』に記載されているとおりにデータをマイグレーションします。
 7. InfoSphere MDM Collaboration Server V10.1.0 インスタンスを開始します。

タスクの結果

個々のマイグレーション・コンポーネントの状況やマイグレーション全体のサマリーなどのすべてのメッセージがコンソール内に表示されます。

マイグレーション・スクリプトの実行後に、マイグレーションが正常に実行されたか失敗したかを示すメッセージがコンソール内に表示されます。

マイグレーションが正常に実行された場合は、以下のメッセージが表示されます。

```
-----
Summary of the migration
-----
```

```
Migration to IBM InfoSphere Master Data Management
Collaboration Server Version 10.1.0 is complete. Check the
messages for any errors.
```

```
Generating database verification report...
```

```
-----
Changed tables
-----
```

```
=====
There are no changed tables
-----
```

Missing Tables

=====
There are no missing tables

Changed Indexes

=====
There are no changed indexes

Missing Indexes

=====
There are no missing indexes

LOG FILE: \$TOP/logs/default/default.log

マイグレーションが失敗した場合は、以下のメッセージが表示されます。

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
Migration Failed : xxxx  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
Please see the log file for further details :  
/home/pimuser/mdmcs10/logs/errfile.log
```

特定のモジュールのマイグレーションが失敗した場合は、以下のメッセージが表示されます。

Summary of the migration

Migration of the following modules failed :
<module names>

次のメッセージは、データベースの表または索引に相違がある場合に表示されます。

Generating database verification report...

Changed tables

=====
<table names>

Missing Tables

=====
<table names>

Changed Indexes

=====
<index names>

Missing Indexes

=====
<index names>

LOG FILE: \$TOP/logs/default/default.log

errfile.log ファイル内の追加のメッセージを確認します。SQL エラーの場合は、SQL エラー・コードから詳細エラー・メッセージを見つけ、エラーを訂正して、マイグレーション・スクリプトを再実行してください。マイグレーション失敗時の推奨処置については、『マイグレーション問題のトラブルシューティング』を参照してください。エラーの訂正後に問題が持続する場合は、IBM ソフトウェア・サポートに連絡してください。

V9.1.0 から V10.1.0 へのマイグレーション

IBM InfoSphere Master Data Management Server for Product Information Management バージョン 9.1.0 から IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 にマイグレーションするには、該当するマイグレーション・スクリプトを実行する必要があります。

始める前に

InfoSphere MDM Server for PIM V9.1.0 から InfoSphere MDM Collaboration Server V10.1.0 にマイグレーションする前に、V9.1.0 のフィックスパック 9 を使用していることを確認する必要があります。フィックスパック 1 から 3 までのいずれかのレベルで V9.1.0 を使用している場合には、資料『IBM InfoSphere Master Data Management Server for Product Information Management, Version 9.1.0 Release notes for fix pack 9』に記載された説明に従って V9.1.0 フィックスパック 9 にマイグレーションする必要があります。

また、すべての XML 関連の表スペースがインストール・ガイドの説明どおりに作成されていること、および InfoSphere MDM Collaboration Server データベース・ユーザーに、新規表スペースを使用するための適切な特権があることを確認する必要があります。

migrateFrom910.sh マイグレーション・スクリプト内の `_ERROR_LOGS` 変数を変更することにより、エラー・ログ・ファイルの場所を構成できます。デフォルトでは、この変数は `$TOP/logs/errfile.log` に設定されます。

このタスクについて

以下は、InfoSphere MDM Server for PIM V9.1.0 から InfoSphere MDM Collaboration Server V10.1.0 にマイグレーションする唯一の方法です。環境エクスポート/インポート・ツールの使用など、他の手段をこのマイグレーションで使用することはできません。

手順

1. 既存のデータベース・スキーマをバックアップします。
2. InfoSphere MDM Collaboration Server V10.1.0 を別のディレクトリーにインストールします。
3. データベースをセットアップします。
 - a. DB2 データベースの場合は、インストール・ガイドの『DB2 データベースのセットアップ』に記載されているとおりに新規のバッファー・プールと表スペースを作成します。

注: XML_DATA 表スペースのサイズは、TCTG_ITD_ITEM_DETAIL 表スペースによって占有されているスペースと同じでなければなりません。この表のおおよそのサイズは、次の照会を使用して算出できます。

```
select (fpages * 16) / 1024 megabytes from syscat.tables
where tabschema = current schema
and tabname like 'TCTG_ITD_ITEM_DETAIL';
```

- b. Oracle データベースの場合は、インストール・ガイドの『Oracle データベースのセットアップ』に記載されているとおりに、新規表スペースを作成して Oracle XML DB コンポーネントをインストールします。
4. InfoSphere MDM Collaboration Server V10.1.0 インスタンスが V9.1.0 フィックスパック 9 データベース・スキーマを指すようにします。
 5. migrateFrom910FP.sh マイグレーション・スクリプトを実行します。このスクリプトは \$TOP/bin/migration ディレクトリーにあります。
 6. 『マスター・データを内部形式から XML 形式にマイグレーション』に記載されているとおりにデータをマイグレーションします。
 7. InfoSphere MDM Collaboration Server V10.1.0 インスタンスを開始します。

タスクの結果

個々のマイグレーション・コンポーネントの状況やマイグレーション全体のサマリーなどのすべてのメッセージがコンソール内に表示されます。

マイグレーション・スクリプトの実行後に、マイグレーションが正常に実行されたか失敗したかを示すメッセージがコンソール内に表示されます。

マイグレーションが正常に実行された場合は、以下のメッセージが表示されます。

```
-----
Summary of the migration
```

```
-----
Migration to IBM InfoSphere Master Data Management
Collaboration Server Version 10.1.0 is complete. Check the
messages for any errors.
```

```
Generating database verification report...
```

Changed tables

=====
There are no changed tables

Missing Tables

=====
There are no missing tables

Changed Indexes

=====
There are no changed indexes

Missing Indexes

=====
There are no missing indexes

LOG FILE: \$TOP/logs/default/default.log

マイグレーションが失敗した場合は、以下のメッセージが表示されます。

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
Migration Failed : xxxx  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
Please see the file for further details : /home/pimuser/mdmcs10/logs/errfile.log
```

特定のモジュールのマイグレーションが失敗した場合は、以下のメッセージが表示されます。

Summary of the migration

Migration of the following modules failed :
<module names>

次のメッセージは、データベースの表または索引に相違がある場合に表示されま
す。

Generating database verification report...

Changed tables

=====
<table names>

Missing Tables

=====
<table names>

Changed Indexes

=====
<index names>

Missing Indexes

=====
<index names>

LOG FILE: \$TOP/logs/default/default.log

errfile.log ファイル内の追加のメッセージを確認します。SQL エラーの場合は、SQL エラー・コードから詳細エラー・メッセージを見つけ、エラーを訂正して、マイグレーション・スクリプトを再実行してください。マイグレーション失敗時の推奨処置については、『マイグレーション問題のトラブルシューティング』を参照してください。エラーの訂正後に問題が持続する場合は、IBM ソフトウェア・サポートに連絡してください。

V10.0.0 から V10.1.0 へのマイグレーション

IBM InfoSphere Master Data Management Server for Product Information Management バージョン 10.0.0 から IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 にマイグレーションするには、該当するマイグレーション・スクリプトを実行する必要があります。

始める前に

InfoSphere MDM Collaboration Server を V10.0.0 から V10.1.0 にマイグレーションするには、明確に V10.0.0 のフィックスパック 3 を使用していることを確認しておく必要があります。V10.0.0 のフィックスパック 1 または 2 を使用している場合は、資料『IBM InfoSphere Master Data Management Collaboration Server, Version 10.0.0 Release notes for fix pack 3』に記載された説明に従って V10.0.0 のフィックスパック 3 にマイグレーションする必要があります。

migrateFrom910.sh マイグレーション・スクリプト内の `_ERROR_LOGS` 変数を変更することにより、エラー・ログ・ファイルの場所を構成できます。デフォルトでは、この変数は `$TOP/logs/errfile.log` に設定されます。

このタスクについて

以下は、InfoSphere MDM Collaboration Server を V10.0.0 から V10.1.0 にマイグレーションする唯一の方法です。環境エクスポート/インポート・ツールを使用する方法など、他の手段では、このマイグレーションは実行できません。

手順

1. 既存のデータベース・スキーマをバックアップします。
2. InfoSphere MDM Collaboration Server V10.1.0 を別のディレクトリーにインストールします。
3. InfoSphere MDM Collaboration Server V10.1.0 インスタンスが V10.0.0 フィックスパック 3 データベース・スキーマを指すようにします。
4. migrateFrom1000FP.sh マイグレーション・スクリプトを実行します。このスクリプトは \$TOP/bin/migration ディレクトリーにあります。
5. InfoSphere MDM Collaboration Server V10.1.0 インスタンスを開始します。

タスクの結果

個々のマイグレーション・コンポーネントの状況やマイグレーション全体のサマリーなどのすべてのメッセージがコンソール内に表示されます。

マイグレーション・スクリプトの実行後に、マイグレーションが正常に実行されたか失敗したかを示すメッセージがコンソール内に表示されます。

マイグレーションが正常に実行された場合は、以下のメッセージが表示されます。

```
-----  
Summary of the migration  
-----  
Migration to IBM InfoSphere Master Data Management  
Collaboration Server Version 10.1.0 is complete. Check the  
messages for any errors.  
  
Generating database verification report...  
  
-----  
Changed tables  
-----  
There are no changed tables  
-----  
  
-----  
Missing Tables  
-----  
There are no missing tables  
-----  
  
-----  
Changed Indexes  
-----  
There are no changed indexes
```

Missing Indexes

=====
There are no missing indexes

LOG FILE: \$TOP/logs/default/default.log

マイグレーションが失敗した場合は、以下のメッセージが表示されます。

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
Migration Failed : xxxx  
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
Please see the file for further details : /home/pimuser/mdmcs10/logs/errfile.log
```

特定のモジュールのマイグレーションが失敗した場合は、以下のメッセージが表示されます。

Summary of the migration

Migration of the following modules failed :
<module names>

次のメッセージは、データベースの表または索引に相違がある場合に表示されま
す。

Generating database verification report...

Changed tables

=====
<table names>

Missing Tables

=====
<table names>

Changed Indexes

=====
<index names>

Missing Indexes

<index names>

LOG FILE: \$TOP/logs/default/default.log

errfile.log ファイル内の追加のメッセージを確認します。SQL エラーの場合は、SQL エラー・コードから詳細エラー・メッセージを見つけ、エラーを訂正して、マイグレーション・スクリプトを再実行してください。マイグレーション失敗時の推奨処置については、『マイグレーション問題のトラブルシューティング』を参照してください。エラーの訂正後に問題が持続する場合は、IBM ソフトウェア・サポートに連絡してください。

Global Data Synchronization の V1.3.x から InfoSphere MDM Collaboration Server V10.1 へのマイグレーション

マイグレーション・プロセスのほとんどの手順を自動化するスクリプトを使用し、IBM InfoSphere Master Data Management Collaboration Server V10.1 へのマイグレーションを実行できます。正常なマイグレーションのために、InfoSphere MDM Collaboration Server ユーザー・インターフェースを使用して、いくつかの手動手順を実行することも必要です。

このタスクについて

GDS のマイグレーションを開始する前に、デマンド・サイドとサプライ・サイドの両方の実装環境に対してリリース GDS 1.3.1-IF011 の基本メンテナンスを使用していることを確認してください。

注: このトピックおよびそのサブトピックの説明では、「GDS 1.3.1-IF011」と表記する代わりに「GDS 1.3.x」と表記します。

手順

1. **エクスポート:** OOTB エクスポート・スクリプトを使用して、メタデータ (データ・モデル) とデータを GDS V1.3.x バックエンド WebSphere® Product Center V5.2.1.4 環境からエクスポートします。
 - a. **メタデータ:** 以下のファイルの内容を組み込んだ圧縮ファイルの形式でメタデータ定義をエクスポートします。
 - 1) **定義ファイル:** これらの XML ファイルは、エクスポートする各種オブジェクト (メタデータ) の実際の定義です。
 - 2) **制御ファイル:** インポートを制御する XML ファイル。圧縮ファイルの内容の参照が含まれています。このファイルの名前は ImportEnvControl.xml です。
 - 3) **データ・ファイル:** アイテムとカテゴリーに関する情報が含まれているコンマ区切り値 (CSV) 形式のファイル。

GDS 固有の文書ストアの内容 (GDS オブジェクトのメタデータを形成する内容) は、ファイル・システムのファイルとしてエクスポートします。

- b. **データ:** InfoSphere MDM Collaboration Server のデータは、アイテムまたはカテゴリーのいずれかになります。GDS で作成したアイテムとカテゴリーは、CSV 形式でエクスポートし、ファイル・システムに保存します。GDS

の通知とトランザクションは、メタデータの圧縮ファイルの形式に合わせて調整した別のファイルにエクスポートします。

2. **変換:** マイグレーション用のインポート・スクリプトには、エクスポートしたメタデータの圧縮ファイルを変換するための手順があります。

注: GDS に用意されていないエクスポート・ツールを使用して、WebSphere Product Center V5.2.1.4 の固有のメタデータをエクスポートする場合は、生成した圧縮ファイルを、InfoSphere MDM Collaboration Server V10.1 に用意されているマイグレーション・スクリプトを使用して変換する必要があります。つまり、InfoSphere MDM Collaboration Server V10.1 インスタンスにある変換ユーティリティ (<MDMPIM_Install_Dir>/bin/migrateCompanyZip.sh) で圧縮ファイルを処理し、InfoSphere MDM Collaboration Server V10.1 との互換性がある圧縮ファイルを生成します。

3. **インポート:** 組み込みのインポート・スクリプトを使用して、変換したメタデータの圧縮ファイルとエクスポートしたデータを InfoSphere MDM Collaboration Server V10.1 環境にインポートします。

GDS 1.3.x からのマイグレーションの準備

GDS 1.3.x から IBM InfoSphere Master Data Management Collaboration Server V10.1.x に正常にマイグレーションするには、次のような準備ステップを行う必要があります。

GDS データのクリーンアップ

ソース WebSphere Product Center V5.2.1.4 インスタンスをクリーンアップして、不要または未使用のデータ・エレメント、特に文書ストアにあるものを削除する必要があります。これらの要素を削除すると、マイグレーション・プロセスにかかる時間とリソース要件の削減に役立ちます。

始める前に

GDS データを削除する前に、以下の手順が必要です。

- 既存のデータベースをバックアップします。クリーンアップ・プロセス中に問題が発生した場合、データベースをロールバックする必要性が生じる可能性があります。
- GDS V1.3.x インスタンスが安定していることを確認します。インスタンスが安定している場合、インバウンド・キューにあるすべてのメッセージが GDS により処理され、メッセージをアウトバウンド・キューへ送信するアクションまたはスケジュールされたジョブが完了します。
- GDS が IBM WebSphere MQ に接続されておらず、GDS リスナー・サービスが実行されていないことを確認します。メッセージを送受信しないようにするには、すべてのキュー接続を停止します。
- GDS V1.3.x 環境変数が適切なディレクトリーを参照していることを確認します。
 - \$TOP 環境変数は、GDS インストール・ディレクトリーから参照されるときには GDS インストール・ディレクトリーを参照し、また、WebSphere Product Center インストール・ディレクトリーから参照されるときには、WebSphere Product Center インストール・ディレクトリーを参照します。

- WPC_INSTALL_DIR 環境変数は、WebSphere Product Center V5.2.1.4 インストール・ディレクトリーを参照します。
- データベースの表スペース、INDX、USERS、BLOB などのオブジェクトに対してスペースが使用可能かをデータベース管理者に確認してください。
- 膨大なトランザクションを収容するために十分なスペースがデータベース・トランザクション・ログ・ファイルに割り当てられているかどうかについてデータベース管理者に確認してください。
- <GDS1.3.x_Install_Dir>/etc/default ディレクトリーの gds.properties ファイルが正しく構成されていることを確認します。このファイルには、*company_code*、*gds_app_type*、および *ACTIVE_DATA_POOL_ID* 変数の有効な値が含まれている必要があります。
- <GDS1.3.x_Install_Dir>/etc/default ディレクトリーの common.properties ファイルが正しく構成されていることを確認します。このファイルには、*db_userName*、*db_password*、および *db_url* 変数の有効な値とメッセージングに関するパラメーターの有効な値が含まれている必要があります。
- <WPC5.2.1.4_Install_Dir >/etc/default ディレクトリーにある common.properties ファイルが正しく構成されていることを確認します。このファイルには、*db_userName*、*db_password*、および *db_url* 変数の有効な値とメッセージングに関するパラメーターの有効な値が含まれている必要があります。

手順

1. 文書ストアのうち、使用されていないファイルをすべて削除します。例えば、古いログ、フィード、および必要ないその他の生成されたファイルを削除します。
2. 必要ないジョブおよびそれに関連するスケジュールを削除します。
3. コラボレーション・エリアにチェックアウトされているすべてのアイテムおよびカテゴリーを移動します。すべてのコラボレーション・エリアを空にする必要があります。
4. すべての GDS ワークフローおよびそのワークフローに関するコラボレーション・エリアを削除します。
5. SQL 照会を実行して、GDS ワークフローにチェックアウトしたアイテム用に作成したすべての GDS 使用可能通知を削除します。これらの SQL 照会は、WebSphere Product Center V5.2.1.4 データベース管理コンソールまたは任意のデータベース・クライアントから実行できます。あるいは、データベース・サーバーにログインして、これらの SQL 照会を実行することもできます。これらの照会すべてにおいて、<company_name> を gds.properties ファイルの *company_code* パラメーターで定義されている実際の会社名に置き換える必要があります。このファイルは、<GDS_Install_Dir>/etc/default ディレクトリーにあります。

注: <company_name> を gds.properties ファイルの **company_code** パラメーターで定義されている実際の会社名に置き換える必要があります。

- a. 以下の SQL 照会を実行して、GDS ワークフロー・ステップでチェックアウトされたアイテムに関する通知を削除します。

デマンド・サイド SQL 照会

```

delete from rcp
where RCP_COMPANY_ID = (
  select CMP_COMPANY_ID from cmp
  where CMP_COMPANY_NAME='<company_name>')
and RCP_NTF_ID in (
  select NTF_ID from ntf
  where NTF_ACTION_CAUSE in (
    'SUBSCRIPTION_DELETION_APPROVAL',
    'SUBSCRIPTION_PENDING_APPROVAL',
    'REVIEW_INCORRECT_INITIAL_LOAD_PUBLICATION',
    'REVIEW_INCORRECT_ITEM_CHANGE_PUBLICATION',
    'REVIEW_INCORRECT_ITEM_CORRECT_PUBLICATION',
    'ITEM_DELETE_PUBLICATION_VALIDATION_FAILED',
    'REVIEW_INCORRECT_NEW_ITEM_PUBLICATION',
    'NEW_ITEM_CATEGORY_MAPPING_PENDING')
  )

```

```

delete from ntf
where NTF_COMPANY_ID = (
  select CMP_COMPANY_ID from cmp
  where CMP_COMPANY_NAME='<company_name>')
and NTF_ACTION_CAUSE in (
  'SUBSCRIPTION_DELETION_APPROVAL',
  'SUBSCRIPTION_PENDING_APPROVAL',
  'REVIEW_INCORRECT_INITIAL_LOAD_PUBLICATION',
  'REVIEW_INCORRECT_ITEM_CHANGE_PUBLICATION',
  'REVIEW_INCORRECT_ITEM_CORRECT_PUBLICATION',
  'ITEM_DELETE_PUBLICATION_VALIDATION_FAILED',
  'REVIEW_INCORRECT_NEW_ITEM_PUBLICATION',
  'NEW_ITEM_CATEGORY_MAPPING_PENDING'
  )

```

サブライ・サイド SQL 照会

```

delete from rcp
where RCP_COMPANY_ID = (
  select CMP_COMPANY_ID from cmp
  where CMP_COMPANY_NAME='<company_name>')
and RCP_NTF_ID in (
  select NTF_ID from ntf
  where NTF_ACTION_CAUSE in (
    'EDIT_ITEM_APPROVAL_PENDING',
    'ITEM_GLOBAL_ENRICHMENT_PENDING',
    'ITEM_LOCAL_ENRICHMENT_PENDING',
    'MODIFY_GLOBAL_ITEM_APPROVAL_PENDING',
    'MODIFY_ITEM_APPROVAL_PENDING',
    'MODIFY_ITEM_LINK_APPROVAL_PENDING',
    'MODIFY_TRADING_PARTNER_PENDING_APPROVAL',
    'NEW_ITEM_APPROVAL_PENDING',
    'NEW_ITEM_PENDING_PUBLICATIONS',
    'NEW_TRADING_PARTNER_PENDING_APPROVAL',
    'TRADING_PARTNER_DELETION_APPROVAL')
  )

```

```

delete from ntf
where NTF_COMPANY_ID = (
  select CMP_COMPANY_ID from cmp
  where CMP_COMPANY_NAME='<company_name>')
and NTF_ACTION_CAUSE in (
  'EDIT_ITEM_APPROVAL_PENDING',
  'ITEM_GLOBAL_ENRICHMENT_PENDING',
  'ITEM_LOCAL_ENRICHMENT_PENDING',
  'MODIFY_GLOBAL_ITEM_APPROVAL_PENDING',
  'MODIFY_ITEM_APPROVAL_PENDING',
  'MODIFY_ITEM_LINK_APPROVAL_PENDING',
  'MODIFY_TRADING_PARTNER_PENDING_APPROVAL',
  'NEW_ITEM_APPROVAL_PENDING',

```

```

        'NEW_ITEM_PENDING_PUBLICATIONS',
        'NEW_TRADING_PARTNER_PENDING_APPROVAL',
        'TRADING_PARTNER_DELETION_APPROVAL')
    )

```

- b. 以下の SQL 照会を実行して、ユーザーに表示するために作成されていない通知エントリーを削除します。

```

delete from rcp
where RCP_COMPANY_ID = (
    select CMP_COMPANY_ID from cmp
    where CMP_COMPANY_NAME='<company_name>')
and RCP_NTF_ID in (
    select NTF_ID from ntf where NTF_SOURCE_USER_ID = -1

```

```

delete from ntf where NTF_COMPANY_ID = (
    select CMP_COMPANY_ID from cmp
    where CMP_COMPANY_NAME='<company_name>')
and NTF_SOURCE_USER_ID = -1

```

- c. 以下の SQL 照会を実行して、宛先のない通知エントリーを削除します。

```

delete from rcp
where RCP_COMPANY_ID = (
    select CMP_COMPANY_ID from cmp
    where CMP_COMPANY_NAME='<company_name>')
and RCP_NTF_ID not in (select NTF_ID from ntf)

```

```

delete from ntf
where NTF_COMPANY_ID = (
    select CMP_COMPANY_ID from cmp
    where CMP_COMPANY_NAME='<company_name>')
and NTF_ID not in (select RCP_NTF_ID from rcp)

```

ここで、<company_name> を <GDS_Install_Dir>/etc/default/gds.properties ファイルの **company_code** パラメーターで定義された実際の会社名に置き換える必要があります。

階層およびカタログに付加された実装スクリプトの除去

データをエクスポートする前に、階層またはカタログに付加された実装スクリプトを手動ですべて切り離す必要があります。階層とカタログが正常にエクスポートされた後、スクリプトを再び付加することができます。

追加の検索可能属性の組み込み

ご使用の実装環境において、ターゲットの IBM InfoSphere Master Data Management Collaboration Server V9.1.x 環境で実行時に検索できる状態にしておかなければならない属性 (索引を生成しなければならない属性) がある場合は、Demand_Params.xml ファイル (デマンド・サイドの場合) または Supply_Params.xml ファイル (サプライ・サイドの場合) にそれらの属性を追加する必要があります。これらのファイルには、GDS データ・モデルで用意されているすべての検索可能属性が含まれています。このファイルは、\$TOP/GDSMigration ディレクトリーにあります。

このタスクについて

エクスポート・プロセスの FirstSteps ステップでは、\$TOP/GDSMigration/envInfo ディレクトリーに pkAndDspIndexedAttrs.xml ファイルが生成されます。このファイルには、デフォルトで索引が生成されるすべての主キー属性と表示属性が含まれています。\$TOP/GDSMigration ディレクトリーにある Demand_Params.xml ファイル

ル (デマンド・サイドの場合) または Supply_Params.xml ファイル (サプライ・サイドの場合) にそれらの属性を手動で追加する必要があります。

手順

1. テキスト・エディターを使用して、Demand_Params.xml ファイル (デマンド・サイドの場合) または Supply_Params.xml ファイル (サプライ・サイドの場合) を開きます。
2. 必要に応じて、Spec タグと Attribute タグを追加します。例えば、Master_Spec 仕様の sku 属性と productDetails 属性の索引を作成するには、以下のエレメントを Demand_Params.xml ファイルまたは Supply_Params.xml ファイルに追加します。

```
<Spec name="Master_Spec">
<Attribute>sku</Attribute>
<Attribute>productDetails/productName</Attribute>
</Spec>
```

その結果、Demand_Params.xml ファイルまたは Supply_Params.xml ファイルは、以下のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<IndexableAttributes>
...
...
...
<Spec name=" Master_Spec ">
  <Attribute>sku</Attribute>
  <Attribute>productDetails/displayAttribute</Attribute>
</Spec>
...
...
...
</IndexableAttributes>
```

3. ファイルを保存します。

gds.properties ファイルの更新

GDS トランザクション、通知、アイテム、およびカテゴリをエクスポートする前に、gds.properties ファイルを更新する必要があります。

このタスクについて

GDS V1.3.x からオブジェクトをエクスポートするには、以下のプロパティの値を指定する必要があります。

- txn_no_of_rows - 各 XML ファイルでエクスポートする必要のあるトランザクション・レコードの数を指定します。
- ntf_no_of_rows - 各 XML ファイルでエクスポートする必要のある通知レコードの数を指定します。
- item_no_of_rows - .csv ファイルにエクスポートされるアイテムまたはカテゴリの数を指定します。使用可能なハードウェア・リソースに応じて、より高い数値に変更することができます。

手順

1. \$TOP/etc/default ディレクトリーで、テキスト・エディターを使って gds.properties ファイルを編集モードで開きます。

2. エクスポート・プロパティを追加して、適切な値を設定します。例えば、以下のような推奨値にプロパティを設定できます。

- txn_no_of_rows=1000
- ntf_no_of_rows=1000
- item_no_of_rows=5000

3. ファイルを保存します。

GDS アプリケーションの停止

GDS V1.3.x インスタンスから GDS 環境をエクスポートする前に、指定された順序でスクリプトを実行することにより、実行中のすべての WebSphere Product Center サービス、GDS サービス、および RMI レジストリーを停止する必要があります。

手順

1. stop_scheduler.sh スクリプトを実行することにより、WebSphere Product Center のスケジューラーを手動で停止します。このスクリプトは <WPC_Install_Dir>/bin/go/stop ディレクトリーの中にあります。
重要: 保留中のクリティカル・ジョブがないこと、およびジョブ・キューが空であることを確認してください。
2. 以下に示す順序でスクリプトを実行することにより、GDS メッセージ・リスナーを停止します (これらのスクリプトは <GDS_Install_Dir//bin directory に入っています)。
 - a. stop パラメーターを指定して gdsmsg.sh スクリプトを実行します。このスクリプトは <GDS_Install_Dir>/bin ディレクトリーにあります。

```
$<GDS_Install_Dir>/bin/gdsmsg.sh stop
```
 - b. 数分が経過した後、abort パラメーターを指定して gdsmsg.sh スクリプトを実行します。このスクリプトは <GDS_Install_Dir>/bin ディレクトリーにあります。このように待機することにより、GDS メッセージ・リスナーは既に選び取ったメッセージの処理を完了することができます。

```
$<GDS_Install_Dir>/bin/gdsmsg.sh abort
```
3. stop_appsvr.sh スクリプトを実行することにより、GDS インスタンスを停止します。このスクリプトは <GDS_Install_Dir>/bin/go/stop ディレクトリーにあります。
4. abort_local.sh スクリプトを実行することにより、WebSphere Product Center インスタンスを停止します。このスクリプトは <WPC_Install_Dir>/bin/go ディレクトリーにあります。このスクリプトは、RMI レジストリー、およびサーバーで実行中のすべての WebSphere Product Center サービスを停止します。
5. インスタンス・シャットダウン後にまだ実行し続けている Java または RMI レジストリーのログ・プロセスがあれば、すべて強制終了します。すべての Java プロセスを強制終了するには、何度か試行しなければならない場合があります。実行中の不要なプロセスがなくなるまで、WebSphere Product Center または Global Data Synchronization によって開始されたすべての Java プロセスの強制終了を続ける必要があります。

データベースのバックアップ

マイグレーション・プロセスによりデータベース内のデータ・モデルが更新されるため、GDS V1.3.x 環境から IBM InfoSphere Master Data Management Collaboration Server V10.1.x 環境の GDS にマイグレーションする場合は、その前に GDS データベースのバックアップを取っておく必要があります。マイグレーションに関する問題が発生した場合は、旧バージョンの GDS データベースにロールバックできます。

このタスクについて

Oracle ユーザーの場合、Oracle 9i クライアント JDBC ドライバー (Oracle 9i または Oracle 10g のいずれかのサーバーに対するクライアント) を使用している際には、BLOB への直列化フォームの書き込みについて 4K の制限があります。InfoSphere MDM Collaboration Server V10.1.x へのマイグレーション中に、Oracle 10gR2 クライアントをインストールして、必要な 10gR2 jar ファイルを入手してください。ORACLE_HOME ディレクトリーの .jar ファイルを手動でアップデートするよりも、クライアントをインストールするほうが賢明です。

WebSphere Product Center V5.2.1.4 からのエクスポート

メタデータおよびデータをエクスポートするには、GDS に含まれているスクリプトを使用する必要があります。エクスポート・プロセスは、エラー処理およびリカバリーを管理できるようにするために、いくつかの部分に分割されています。

始める前に

GDS 環境をエクスポートする前に、次のことを行う必要があります。

- analyze_schema.sh スクリプトを実行して、データベース・スキーマを分析および改善します。このスクリプトは、\$TOP/src/db/schema/util ディレクトリーにあります。ここで、\$TOP は WebSphere Product Center のインストール・ディレクトリーです。このスクリプトを実行すると、エクスポート中の、特にアイテムやカテゴリーの量が多い場合に、エクスポートのパフォーマンスを改善します。
- ファイル・システムに十分なディスク・スペースがあることを確認します。
- 関連があるプロセスのみが実行されていることを確認します。パフォーマンスを改善するには、すべての GDS および WebSphere Product Center のサービスをシャットダウンします。
- docstore をクリーンアップした後も、docstore の内容の一部のみをエクスポートすることを検討してください。docstore 全体をエクスポートする代わりに、scripts や params などの関連があるディレクトリーから文書をエクスポートできます。例えば、次のスクリプトは、scripts ディレクトリーから文書をエクスポートします。

```
envObjList = new EnvObjectList ();
envObjList.addObjectByNameToExport ("/scripts","DOC_STORE");
sDocFilePath = "archives/docstoreExport.zip" +
out.writeln (exportEnv (envObjList, sDocFilePath));
```

このタスクについて

GDS システムが一貫性のある状態の場合、各オブジェクトを個別にエクスポートしたり、export_type=all オプションを使用してすべてのオブジェクトをまとめてエ

クサポートしたりできます。特定の順序でオブジェクトをエクスポートする必要はありません。ただし、すべてのオブジェクトをまとめてエクスポートすることは、プロセスの実行中に発生するエラーや例外を特定して解決することが難しくなるため、推奨されません。

次のスクリプトは、スクリプト内で `$TOP` として使用される GDS v1.3.x インストール・ディレクトリーから実行されるときに、すべてのオブジェクトをエクスポートします。これらのステップのログは、`$TOP/GDSMigration/Logs` ディレクトリーで取得できます。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh
--company_code=<cmp_code> --export_type=all
```

ただし、デバッグを容易にするために、次に詳細を説明する段階的なアプローチを使用することをお勧めします。エクスポートに指定されるオプションは、次のとおりです。

```
MigrateExport.sh --company_code=<code>
--export_type=<firstSteps/0/1/2/3/4/5/6/7/8/9>
```

次のように定義されるオプションを使用します。

company_code

会社コードを指定します。

以下のスクリプトで、`<cmp_code>` は、`$TOP/etc/default/gds.properties` ファイルの **company_code** プロパティーの値になります。これらのステップのログは、`$TOP/GDSMigration/Logs` ディレクトリーで取得できます。

export_type

エクスポート・オプションを指定します。次のリストはエクスポート・オプションを定義します。

- *firstSteps* - 階層とカタログに関連するスクリプトの検索、索引が付けられる属性の取り出し、データ・モデル情報の取得、およびカタログとアイテムの数の取り出しを実行します。
- 0 - エクスポートのインスタンスを準備します。
- 1 - 基本メタデータ [ルックアップ、仕様、属性コレクション、会社設定、階層、カタログなど] をエクスポートします。
- 2 - セキュリティー [ACG、役割、ユーザー、選択など] をエクスポートします。
- 3 - その他のメタデータ [ジョブ、フィード、レポートなど] をエクスポートします。
- 4 - GDS XML メッセージをエクスポートします。
- 5 - GDS メッセージング・オブジェクトをエクスポートします。

注: これはデマンド・サイドの場合にのみ必要です。

- 6 - トランザクションをエクスポートします。
- 7 - 通知をエクスポートします。
- 8 - 階層内容をエクスポートします。

- 9 - カタログの内容をエクスポートします。

手順

1. MigrateExport.sh スクリプトで、**export_type** パラメーターを firstSteps に設定して実行します。このスクリプトは \$TOP/src/db/schema/gds/migration/10.0_migration ディレクトリーにあります。このスクリプトは、階層とカタログに関連するスクリプトの検索、索引が付けられる属性の取り出し、データ・モデル情報の取得、およびカタログとアイテムの数の取り出しを実行します。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=firstSteps
```

このステップは、現行システムに関する情報、および実際のエクスポートを開始する前に行う必要があるいくつかのアクションに関する情報をユーザーに提供する準備のステップです。このステップでは、次のタスクを実行します。

- システムに存在するカタログに付加されるスクリプトの詳細を示すファイル \$TOP/GDSMigration/envInfo/CtgAttribInfo.txt を作成します。
- システムに存在する階層に付加されるスクリプトの詳細を示すファイル \$TOP/GDSMigration/envInfo/HierAttribInfo.txt を作成します。
- \$TOP/GDSMigration/Demand_Params.xml ファイル (デマンド・サイドの場合) または \$TOP/GDSMigration/Supply_Params.xml ファイル (サプライ・サイドの場合) に手動で追加される 1 次属性および表示属性を含むファイル \$TOP/GDSMigration/envInfo/pkAndDspIndexedAttrs.xml を作成します。これらのファイルは、InfoSphere MDM Collaboration Server V10.0 で索引を作成する必要があるデフォルトの検索可能属性をリストします (索引を作成することによって、これらの属性が検索可能になります)。
- 現在の GDS 1.3.x システムに存在するすべてのデータ・モデル・オブジェクトのリストを含むファイル \$TOP/GDSMigration/envInfo/metaDataInfoList.txt を作成します。
- 現在の GDS 1.3.x システムに存在するすべてのカタログおよびアイテムの数をを含むファイル \$TOP/GDSMigration/envInfo/itemAndCategoryCount.txt を作成します。

2. MigrateExport.sh スクリプトで、**export_type** パラメーターを 0 に設定して実行します。このスクリプトは、エクスポートのインスタンスを準備します。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=0
```

いくつかのオブジェクト・タイプおよびその定義は InfoSphere MDM Collaboration Server V10.0 でサポートされていないため、このオプションが必要です。既存のフォーマットでデータをエクスポートすると、インポート・プロセスが失敗します。例えば、GDS 1.3.x 環境では、タイプ STRING のノードを、他のノード (GROUPING) の親ノードとして持つことができます。これは、InfoSphere MDM Collaboration Server V10.0 環境では不可能です。このため、このメタデータをエクスポートする前に、修正する必要があります。

3. MigrateExport.sh スクリプトで、**export_type** パラメーターを 1 に設定して実行します。このスクリプトは、ルックアップ、仕様、属性コレクション、会社設定、階層、カタログなどの基本的なメタデータをエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=1
```

残りのステップでエクスポートされるメタデータは、これらのオブジェクトが正しくインポートされるかどうかには依存します。次のオブジェクトは、基本的なメタデータとして識別されます。

- COMPANY_ATTRIBUTES
- LOOKUP_TABLE_SPEC
- LOOKUP_TABLE
- LOOKUP_TABLE_CONTENT
- PRIMARY_SPEC
- SUB_SPEC
- DESTINATION_SPEC
- SECONDARY_SPEC
- SCRIPT_INPUT_SPEC
- FILE_SPEC
- ATTRIBUTE_COLS
- HIERARCHY
- CATALOG
- MAPS
- ACG

これらのオブジェクトは、COMPANYMETADATA.zip ファイルにエクスポートされてから、\$TOP/GDSMigration/ExportedZip ディレクトリーの下にダウンロードされます。

4. MigrateExport.sh スクリプトで、**export_type** パラメーターを 2 に設定して実行します。このスクリプトは、CONTAINER_ACCESSPRV、MY_SETTINGS、ROLES、USERS などのセキュリティー定義をエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=2
```

上記のオブジェクトは、SECURITY.zip ファイルにエクスポートされてから、\$TOP/GDSMigration/ExportedZip ディレクトリーの下にダウンロードされます。

5. MigrateExport.sh スクリプトで、**export_type** パラメーターを 3 に設定して実行します。このスクリプトは、ジョブ、フィールド、レポートなどのその他のメタデータをエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=3
```

このステップでは、エクスポート・スクリプトは次のようなその他のメタデータを処理します。

- DATASOURCE
- FEEDS
- JOBS
- DISTRIBUTION
- EXPORTS

- REPORTS
- HIERARCHY_VIEW
- CATALOG_VIEW
- DISTRIBUTION_GROUP
- ALERT
- QUEUE
- SEARCH_TEMPLATES
- UDL
- HIERARCHY_MAPS

上記のオブジェクトは、OTHERMETADATA.zip ファイルにエクスポートされてから、\$TOP/GDSMigration/ExportedZip ディレクトリーの下にダウンロードされます。

6. MigrateExport.sh スクリプトで、**export_type** パラメーターを 4 に設定して実行します。このスクリプトは、GDS XML メッセージをエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh
--company_code=<cmp_code> --export_type=4
```

GDS システムと接続されたデータプールの中で交換されたすべての XML メッセージは、WebSphere Product Center v5.2.1.4 docstore に保管されます。次のオブジェクトは GDS XML メッセージとして識別され、文書ストアの指示された部分をエクスポートする必要があります。

- docstore のパス /archive/messages/inbound/ の下に保管された XML メッセージ
- docstore のパス /archives/messages/outbound/ の下に保管された XML メッセージ

これらの XML メッセージは、エクスポートされてから \$TOP/GDSMigration/messages/inbound ディレクトリーおよび \$TOP/GDSMigration/messages/outbound ディレクトリーの下に制御ファイル uplInXMLMsg.xml および uplOutXMLMsg.xml と共にダウンロードされます。制御ファイルは両方とも \$TOP/GDSMigration/messages ディレクトリーに保管されます。

7. デマンド・サイドの場合のみ、**export_type** パラメーターを 5 に設定して MigrateExport.sh スクリプトを実行します。このスクリプトは、GDS メッセージング・オブジェクトをエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh
--company_code=<cmp_code> --export_type=5
```

GDS と接続されたデータプールの中の XML メッセージの交換プロセスで作成されたすべての GDS メッセージング・オブジェクトは、WebSphere Product Center v5.2.1.4 docstore に保管され、エクスポートする必要があります。

次のオブジェクトは、GDS メッセージング・オブジェクトとして識別されます。

- /object/messaging/ の下の docstore に保管されるすべてのオブジェクト

これらのオブジェクトは、エクスポートされてから \$TOP/GDSMigration/object/messaging ディレクトリーの下に \$TOP/GDSMigration/object 内の制御ファイル uplMessObj.xml とともにダウンロードされます。

8. MigrateExport.sh スクリプトで、**export_type** パラメーターを 6 に設定して実行します。このスクリプトはトランザクションをエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=6
```

トランザクションは中核となる GDS オブジェクトで、TRANSACTIONS として識別されます。これらのオブジェクトは、TRANSACTIONS.zip ファイルにエクスポートされてから \$TOP/GDSMigration/ExportedZip ディレクトリーの下にダウンロードされます。

9. MigrateExport.sh スクリプトで、**export_type** パラメーターを 7 に設定して実行します。このスクリプトは、通知をエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=7
```

通知は中核となる GDS オブジェクトで、NOTIFICATIONS として識別されます。これらのオブジェクトは、NOTIFICATIONS.zip ファイルにエクスポートされてから、\$TOP/GDSMigration/ExportedZip ディレクトリーの下にダウンロードされます。

10. MigrateExport.sh スクリプトで、**export_type** パラメーターを 8 に設定して実行します。このスクリプトは、階層内容をエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=8
```

このフェーズは、階層内容がエクスポートの大半を占めるため、他の会社データ・オブジェクトから分離されています。データ (カテゴリー) の量が非常に多い階層がある場合、このフェーズはさらに階層レベルに分割され、各エクスポートには 1 つの階層内容が含まれます。別の可能性として、データ量が小さい階層のすべてを .zip ファイルにグループ化して、大きい階層をそれぞれ個別の zip ファイルにエクスポートする場合があります。GDS の実装では、階層内容のほとんどが大量であり、WebSphere Product Center のエクスポート環境を使用してそのままエクスポートすることはできません。このために、カスタマイズされたエクスポート・ユーティリティーが、階層内容をエクスポートするために提供されています。HIERARCHY_CONTENT(Categories) オブジェクトは、階層内容として識別されます。

これらのオブジェクトは、エクスポートされてから \$TOP/GDSMigration/feed_files/hierarchies ディレクトリーの下に \$TOP/GDSMigration/feed_files 内の制御ファイル uplHieCSV.xml とともにダウンロードされます。

注: データ・ファイル (.csv) は、中に少なくとも 1 つのカテゴリーを持つ階層に対してのみ生成されます。

11. MigrateExport.sh スクリプトで、**export_type** パラメーターを 9 に設定して実行します。このスクリプトは、カタログの内容をエクスポートします。

```
$TOP/src/db/schema/gds/migration/9.1.0_migration/MigrateExport.sh  
--company_code=<cmp_code> --export_type=9
```

このステップでは、GDS のカタログの内容がエクスポートされます。このフェーズは、カタログの内容がエクスポートの大半を占めるため、他の会社データ・オブジェクトから分離されています。データ (アイテム) の量が非常に多いカタログがある場合、このフェーズはさらにカタログ・レベルに分割され、各エクスポートには 1 つのカタログの内容が含まれます。別の可能性として、データ量が小さいカタログのすべてを .zip ファイルにグループ化して、大きいカタログをそれぞれ個別の zip ファイルにエクスポートする場合があります。GDS の実装では、カタログの内容のほとんどが大量であり、WebSphere Product Center のエクスポート環境を使用してそのままエクスポートすることはできません。このために、カスタマイズされたエクスポート・ユーティリティーが、カタログの内容をエクスポートするために提供されています。CATALOG_CONTENT オブジェクトは、カタログの内容として識別されます。

これらのオブジェクトは、エクスポートされてから、\$TOP/GDSMigration/feed_files 内の制御ファイル uplCtgCSV.xml とともに \$TOP/GDSMigration/feed_files/catalogs ディレクトリの下にダウンロードされます。

注: データ・ファイル (.csv) は、少なくとも 1 つのアイテムが含まれるカタログに対してのみ生成されます。

csv ファイルに含まれているアイテムの数は、GDS V1.3.x 環境の gds.properties ファイルにある item_no_of_rows に対して設定されているパラメーター値によって決まります。

次のタスク

エクスポートがどのように完了したかを調べるには、ログをチェックします。正常にエクスポートしなかったオブジェクトがいくつかある場合は、そのオブジェクトのみについて再度エクスポートを実行することができます。作成された圧縮ファイルは、解凍し、エラー・ログに基づいて、情報欠落がないかどうかをチェックすることができます。エクスポート・スクリプトは、2 回以上実行できます。

エクスポート・スクリプトの制限事項

エクスポート操作の完了後、すべてのオブジェクトがエクスポートされたこと、およびすべての依存関係が解決されていることを確認しなければなりません。エクスポート・スクリプトでは、依存関係にあるすべてがエクスポートされたことの検証はしません。また、作成された圧縮ファイルが正常にインポート可能であるかどうかの判断も行いません。

WebSphere Product Center レベルの制限事項

- 旧バージョンのエクスポート: マイグレーション・ツールでは WebSphere Product Center でタグ付けされたメタデータおよびデータのすべてのバージョンのエクスポートは行いません。エクスポートできるのは、そのようなオブジェクトの最新バージョンのみです。インポートされたデータの現在のエクスポートから過去のレベルへのカタログ・ロールバックはサポートされていません。カタログおよびカテゴリのタグのすべてが存在しているわけではないからです。バージョン間の相違をエクスポートすることはできません。ターゲット・システムでは最新バージョンのみが使用可能だからです。例えば、WebSphere Product Center にカタログ・オブジェクトの複数のバージョンがある場合、カタログの最新バージョンのみがエクスポートされます。

- デフォルト・オブジェクトのエクスポート: マイグレーション・ツールでは、会社を新規作成した際にデフォルトで作成されたオブジェクトはエクスポートしません。変更されたオブジェクト (デフォルト階層 1 次仕様、ルックアップ・テーブル・インポート仕様、デフォルト・ルックアップ・テーブル階層など) が現在の実装に含まれている場合でも、ツールはそれらがターゲット会社で見つかることを想定するので、変更内容の持ち越しは行いません。これらのオブジェクトをインポートする前に、それらのオブジェクトの WebSphere Product Center V5.2.1.4 での構造を反映するように IBM InfoSphere Master Data Management Collaboration Server V10.1.x で当該オブジェクトを手動で変更する必要があります。
- 組織階層: WebSphere Product Center V5.2.1.4 では、マイグレーション・ツールを使用して組織階層のエクスポートを行うことはできません。そのような階層はターゲット会社で手動で作成する必要があります。
- アイテム/カテゴリーの順序付け: アイテム/カテゴリーの順序付けのエクスポートはサポートされていません。
- 役割からロケールへのマッピング: 役割からロケールへのマッピングのエクスポートはサポートされていません。
- 属性値サイズ: integer MAX_VALUE より大きな整数値を持つ属性はインポートされません。
- 役割のないユーザー: WebSphere Product Center の役割が関連付けられていないユーザーはエクスポートされません。
- 拡張選択照会言語: 拡張選択照会言語 (ルール) は InfoSphere MDM Collaboration Server v10.1.x で変更されました。 WebSphere Product Center V5.3.0 より前のバージョンで作成された照会はすべて手動でマイグレーションしなければなりません。

InfoSphere MDM Collaboration Server V10.1.x では、拡張選択は自動的にマイグレーションされません。ルールは InfoSphere MDM Collaboration Server V10.1.x で書き直す必要があります。 InfoSphere MDM Collaboration Server v10.1.x での選択の作成には以下のスクリプト操作を使用できます。

- new DynamicSelection(String name, String WQL)
- DynamicSelection::saveSelection()

WebSphere Product Center V5.3.2 では WebSphere Query Language (WQL) が無効である場合に saveSelection 操作は使用できないので、修正された有効な WQL を使用してこのスクリプトを実行しなければなりません。動的選択は以下の手順でマイグレーションできます。

1. すべての選択を WebSphere Product Center V5.2.1 およびそれ以降のフィックスパックから .csv ファイル形式でエクスポートします。ここで、最初の値は名前、2 番目の値はルール・ストリングになります。
2. ルールを有効な WQL に変換します。WQL の詳細については、『照会言語』を参照してください。
3. ルールと選択を、スクリプト操作を使用してインポートします。 .csv ファイルを文書ストアにアップロードするか、またはサンドボックス内のスクリプト入力ペインから値を直接読み取ることができます。

簡単に参照できるように、以下にサンプル・スクリプトを示します。

スクリプト入力ペイン

```
selection1, select item from catalog('cata')
  where item['spec1/primary key'] = '11'
selection2, select item from catalog('cata')
  where item['spec1/primary key'] = '11'
selection3, select item from catalog('cata')
  where item['spec1/primary key'] = '11'
```

スクリプト・ペイン

```
var parser = newCSVParser(in);
var bDone = false;
var e = "";
catchError(e)
{
  while(!bDone)
  {
    attrs = parser.splitLine();
    if(attrs!= null)
    {
      a = new DynamicSelection(attrs[0], attrs[1]);
      a.saveSelection();
      out.writeln(attrs[0] + " created");
    }
    bDone = (null == attrs);
  }
}
out.writeln(e);
```

- 関係情報: 関係属性用に指定されたデータを持つカテゴリーを含む階層については、階層コンテンツのインポートは正常に行われません。この問題を回避するには、以下の手順に従います。
 - 会社のエクスポートで生成された圧縮ファイルを解凍します。
 - *HIERARCHY_CONTENTS* ディレクトリーを参照します。
 - 各 *.csv* ファイルを編集して、関係データを指定している列を削除します。編集後に必ずファイルを保存します。
 - 圧縮ファイルを新規作成し、更新したファイルをそこに追加します。 *zip* ファイルのディレクトリー構造は、解凍した圧縮ファイルの構造と同じでなければなりません。 *ImportEnvControl.xml* ファイルと *archives* ディレクトリーはルート・レベルに置く必要があります。
 - エクスポートされた *zip* ファイルではなく、新規の *zip* ファイルをインポートします。
- プロパティ・ファイルのマイグレーション: *WebSphere Product Center* プロパティ・ファイルはすべて手動でマイグレーションする必要があります。各プロパティ・ファイルの詳細については、*InfoSphere MDM Collaboration Server V10.1.x* の製品資料を参照してください。
- *ctg_files* ディレクトリーのマイグレーション: *\$TOP/public_html/suppliers/<cmp code>/ctg_files* ディレクトリーの内容は、各会社の新しいインスタンスに手動でコピーする必要があります。
- ユーザー *.jar* ファイルのマイグレーション: 必要なユーザー *.jar* ファイルは新しいインスタンスに追加する必要があります。 *InfoSphere MDM Collaboration Server V10.1.x* インフォメーション・センターで、必要なユーザー *.jar* ファイルの新しいインスタンスへの追加に関する情報を参照してください。

- 継承されたデータのエクスポートとインポート: 継承されたデータのエクスポートとインポートはできません。継承をサポートするカタログと階層のコンテンツは、エクスポートもインポートもできません。

システム・レベルの制限事項

マイグレーション・ツールを使用したエクスポート操作は、システム・レベルの制限に達した場合は正常に行われません。一般的な制限事項には以下のものがあります。

- ファイル・システムの制限: 多くのオペレーティング・システムではファイル・システムに制限があります。特定のファイル・システムを使用する際に、メタデータとデータのサイズに基づいてエクスポートを分割する方法を決定しなければなりません。
- 圧縮ファイル形式の制限: マイグレーション・ツールでは、エクスポートしたファイルを圧縮し、それを文書ストアにアップロードします。圧縮ユーティリティーには、単一の圧縮ファイルのサイズの上限、および 1 つの圧縮ファイルに対するファイルの最大数が設定されている場合があります。この問題に対処するため、エクスポートを論理的に分割することにより、生成されるファイル数が上限より少なくなるようにしてください。
- エクスポート操作を使用して生成される圧縮ファイルが空のように見える場合: エクスポート操作を使用して生成される圧縮ファイルを Windows XP の圧縮 (zip 形式) フォルダ・ユーティリティーで開くと空であるように見えます。これは既知の問題です。この問題を回避するには、WinZip アプリケーションを使用して圧縮ファイルを開いてください。

文書ストアのエクスポート

文書ストアは別個にエクスポートする必要があります。

文書ストアには多くのコンテンツが格納され、他のオブジェクトと比較してエクスポートに多くの時間がかかるため、文書ストアは別個にエクスポートしてください。文書ストア全体のエクスポート、または文書ストアの一部のエクスポートを行います。文書ストアは 1 つの圧縮ファイルにするには大きすぎるが多いため、部分に分けてエクスポートするのがよい場合があります。

部分的/選択的な文書ストアのエクスポート

ここでは、文書ストアを部分に分けてエクスポートします。以下のサンプル・スクリプトで、/object/messaging/ という名前を指定した文書ストア部分をエクスポートする例を示します。

```
//Sample script

var docStorePath = "GDSMigration/ExportedZip/";
var sDocFilePath = docStorePath+"MESSAGINGOBJECTS.zip";
var envObjList = new EnvObjectList();
envObjList.setTypeToExport("DOC_STORE");
envObjList.addObjectByNameToExport("/object/messaging/");
//export the selected objects to the docstore
var result = exportEnv(envObjList, sDocFilePath);
//check whether export was successful
if( result.contains("Uploaded the archived file to docstore") ) {
    out.println("result " + result);
}
```

```

else {
  //if the export fails, the text of the result
  //will appear as an error message
  throwError(result);
}

```

このサンプル・スクリプトでは、文書ストアの /object/messaging/ セクションをエクスポートし、これを /GDSMigration/ExportedZip/MESSAGINGOBJECTS.zip として文書ストアに保存します。

完全文書ストア・エクスポート

このアプローチでは、文書ストア全体をエクスポートします。文書ストア全体のエクスポートは、文書ストアのコンテンツが単一の圧縮ファイルに収まる場合にのみ行ってください。以下のサンプル・スクリプトでは文書ストア全体をエクスポートします。

```

//Sample script

var docStorePath = "GDSMigration/ExportedZip/";
var sDocFilePath = docStorePath+"COMPANYDOCSTORE.zip";
var envObjList = new EnvObjectList();
envObjList.addAllObjectsToExport("DOC_STORE");
  //export all the objects to the docstore
out.println("Saving environment as " + sDocFilePath);
var result = exportEnv(envObjList, sDocFilePath);
//check whether export was successful
if( result.contains("Uploaded the archived file to docstore") ) {
  out.println("result " + result);
}
else {
  //if the export fails, the text of the result will
  //appear as an error message
  throwError(result);
}

```

このスクリプトでは、文書ストア全体をエクスポートし、文書ストアに生成された圧縮ファイルを /GDSMigration/ExportedZip/COMPANYDOCSTORE.zip として保管します。

注: このサンプル・スクリプトは MigrateExport.sh スクリプトには含まれていません。オブジェクト DOC_STORE は文書ストアとも呼ばれます。

InfoSphere MDM Collaboration Server V10.1 へのインポート

MigrateImport.sh スクリプトを使用して、GDS オブジェクトを IBM InfoSphere Master Data Management Collaboration Server V10.1 環境にインポートします。

始める前に

GDS 環境をインポートする前に、以下の点を確認する必要があります。

- InfoSphere MDM Collaboration Server V10.1 の GDS で必要なすべての構成およびセットアップが完了していることを確認します。動作中の GDS 1.3.x インスタンスで GDS プロパティの構成を参照することができます。詳細については、『GDS フィーチャーの構成』を参照してください。

- スキーマを作成するスクリプト (`create_schema.sh`) と会社を作成するスクリプト (`create_cmp.sh`) が正常に実行されていることを確認します。どちらのスクリプトも、`$(install_dir)/bin/db` ディレクトリーにあります。
- GDS 1.3.x のインストール・ディレクトリーにある `GDSMigration` ディレクトリーとそのすべての内容を `InfoSphere MDM Collaboration Server V10.1` のインストール・ディレクトリーにコピーします。

このタスクについて

`InfoSphere MDM Collaboration Server V10.1` にデータを正常にインポートするには、段階的な方式に従って、特定の順序でインポート・プロセスの手順を実行する必要があります。それぞれの手順では、`import_type` パラメーターを別々の値に設定してインポート・スクリプトを実行しなければなりません。一部の手順でインポートするオブジェクトは、その前の手順でインポートするオブジェクトに依存しているため、各手順が正常に完了したことを確認してから、次の手順を開始するようにしてください。指定の順序を守れば、オブジェクトの依存関係に起因する失敗を回避できます。

ベスト・プラクティス: インポート時にどのアイテムが失敗したのかを確認する方法はありません。ツールから生成されるデバッグ・レポートのログには、そのような詳細は記録されません。アイテムやカテゴリーなどの内容をインポートするスクリプトは、以下のとおりです。

- `/scripts/import/ctg/environment_catalog_content_import.script`
- `/scripts/import/ctr/environment_category_tree_content_import.script`

失敗したアイテムやカテゴリーのロギングを可能にするために、これらのスクリプトを変更できます。

`InfoSphere MDM Collaboration Server V10.1` では、`common.properties` ファイルの `debug_profile_db_connections` プロパティ、およびデータベース・マネージャーとして `LiteDBManager` を作成する関連コードが削除されました。そのため、`ThinPoolDBManager` は `InfoSphere MDM Collaboration Server V10.1` での唯一のデータベース・マネージャーです。

次のスクリプトは、スクリプト内で `$TOP` として使用される `InfoSphere MDM Collaboration Server V10.1` インストール・ディレクトリーから実行されるときに、オブジェクトをインポートします。これらの手順に関するログは、`$TOP/GDSMigration/Logs` ディレクトリーにあります。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=<gdsprev/0/1/2/3/4/5/6/7/8/9/10>
```

次のように定義されるオプションを使用します。

company_code

会社コードを指定します。

以下のスクリプトで、`<cmp_code>` は、`$TOP/etc/default/gds.properties` ファイルの `company_code` プロパティの値になります。

import_type

インポート・オプションを指定します。インポート・オプションの定義を以下のリストにまとめます。

- *gdsprev* - GDS の特権をロードします。
- 0 - エクスポートした圧縮ファイルを変換します。
- 1 - ルックアップ、仕様、属性コレクション、会社設定、階層、カタログなどのメタデータをインポートします。
- 2 - 階層の内容のカテゴリをインポートします。
- 3 - 役割、ユーザー、選択などのセキュリティー・オブジェクトをインポートします。
- 4 - GDS のワークフローを作成し、スクリプトをロードします。
- 5 - その他のメタデータ (ジョブ、フィールド、レポートなど) をインポートします。
- 6 - GDS の XML メッセージをインポートします。
- 7 - GDS のメッセージング・オブジェクトをインポートします。

注: これはデマンド・サイドの場合にのみ必要です。

- 8 - カタログの内容のアイテムをインポートします。
- 9 - GDS のトランザクションをインポートします。
- 10 - GDS の通知をインポートします。

手順

1. **import_type** パラメーターを *gdsprev* に設定して *MigrateImport.sh* スクリプトを実行します。このスクリプトは `$TOP/src/db/schema/gds/migration/10.0_migration` ディレクトリーにあります。このスクリプトでは、GDS の特権をロードします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh  
--company_code=<cmp_code> --import_type=gdsprev
```

この手順では、スクリプトによって GDS の特権を InfoSphere MDM Collaboration Server V10.1 環境にロードします。デフォルトでは、GDS のデータ・モデルはロードされないので、最初に特権をロードすることが必須になります。

注: この手順は、必須の手順です。

2. **import_type** パラメーターを 0 に設定して *MigrateImport.sh* スクリプトを実行します。このスクリプトでは、エクスポートした圧縮ファイルを変換します。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh  
--company_code=<cmp_code> --import_type=0
```

この手順では、GDS v1.3.x インスタンスからエクスポートした圧縮ファイルを InfoSphere MDM Collaboration Server V10.1 の形式に変換します。このスクリプトでは、`$TOP/GDSMigration/ExportedZip` ディレクトリーに配置されているすべてのエクスポート圧縮ファイルを読み取り、変換してから、変換後の圧縮ファイルを `$TOP/GDSMigration/TransformedZip` ディレクトリーに配置します。

注: COMPANYMETADATA.zip ファイルの変換時に SAXParseException が表示されます。MAPS.xml ファイルは、.xml 拡張子が付いているものの、実際には XML ファイルではないからです。この例外は無視しても差し支えありません。

3. **import_type** パラメーターを 1 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、ルックアップ、仕様、属性コレクション、会社設定、階層、カタログなどのメタデータをインポートします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=1
```

この手順は、エクスポートしたメタデータとデータを実際にインポートする最初の手順です。最初に基本的なメタデータの圧縮ファイルをインポートする必要があります。その後のインポートは、そのメタデータに依存するからです。

このスクリプトでは、\$TOP/GDSMigration/TransformedZip ディレクトリーにある変換後の COMPANYMETADATA.zip ファイルをインポートします。

4. **import_type** パラメーターを 2 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、階層と内容のカテゴリーをインポートします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=2
```

この手順では、会社のメタデータまたは基本メタデータが InfoSphere MDM Collaboration Server V10.1 インスタンスに正常にインポートされた後、階層内容がインポートされます。この手順では、階層のすべての内容 (.csv ファイルの内容) とインポート・スクリプトを文書ストアにロードし、このスクリプトの正常終了後に手動で開始しなければならない階層インポート・ジョブを作成します。このスクリプトを正常に実行してから、インポート・ジョブを手動で開始してください。階層内容を InfoSphere MDM Collaboration Server V10.1 インスタンスにインポートするときには、『階層とカタログの内容のインポート・ジョブを即時実行する』を参照してください。

5. **import_type** パラメーターを 3 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、ACG、役割、ユーザー、選択などのセキュリティー・コンポーネントをインポートします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=3
```

セキュリティー・コンポーネントには、ACG、役割、ユーザー、選択、コンテナ・アクセス権、マイ・セッティングなどがあります。このスクリプトでは、\$TOP/GDSMigration/TransformedZip ディレクトリーにある変換後の SECURITY.zip ファイルの内容をインポートします。

6. **import_type** パラメーターを 4 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、GDS のワークフローを作成し、スクリプトをロードします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=4
```

GDS 固有のスクリプト、ワークフロー、およびコラボレーション・エリアは、GDS v1.3.x 環境からエクスポートされませんが、このマイグレーション手順で作成またはロードされます。

7. **import_type** パラメーターを 5 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、その他のメタデータ (ジョブ、フィード、レポートなど) をインポートします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=5
```

このスクリプトでは、前の手順でインポートしたオブジェクトに依存する残りのメタデータをすべてインポートします。このインポート・スクリプトでは、\$TOP/GDSMigration/TransformedZip ディレクトリーにある変換後の OTHERMETADATA.zip ファイルの内容をインポートします。

8. **import_type** パラメーターを 6 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、GDS の XML メッセージを InfoSphere MDM Collaboration Server V10.1 ターゲット・インスタンスの文書ストアにインポートします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=6
```

9. デマンド・サイドの場合のみ、**import_type** パラメーターを 7 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、GDS のメッセージング・オブジェクトを InfoSphere MDM Collaboration Server V10.1 ターゲット・インスタンスの文書ストアにインポートします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=7
```

10. **import_type** パラメーターを 8 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、カタログの内容のアイテムをインポートします。詳細については、『階層とカタログの内容のインポート・ジョブをただちに実行する操作』を参照してください。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=8
```

トランザクションと通知以外のすべてのオブジェクト、および文書ストアの一部が InfoSphere MDM Collaboration Server V10.1 インスタンスに正常にインポートされた後に、カタログの内容をインポートします。

このスクリプトでは、カタログのすべての内容 (.csv ファイルの内容) とカタログ・インポート・スクリプトを文書ストアにロードし、カタログ・インポート・ジョブを作成します。csv ファイルに含まれているアイテムの数は、エクスポートした GDS 1.3.x 環境の gds.properties ファイルにある

item_no_of_rows パラメーターに設定されているパラメーター値によって決まります。例えば、カタログの内容を組み込む各 .csv ファイルに 1000 個のアイテムを入れる、といった設定が可能です。その場合は、1000 個のアイテムごとに別の .csv ファイルが作成されます。

.csv ファイルの名前は、<Catalog_Name>[1,2,3..].csv という形式になります。例えば、Global_Catalog に 2000 個のアイテムが含まれていれば、このカタログのアイテム・データのために 2 つの .csv ファイルが作成されます。ファイル名はそれぞれ、Global_Catalog1.csv、Global_Catalog2.csv になります。

注: TradeCatalog の内容をインポートできるのは、Global_Catalog と Global_Local_Catalog の内容を正常にインポートした後に限られます。

11. **import_type** パラメーターを 9 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、GDS のトランザクションをインポートします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=9
```

GDS のトランザクションは、ユーザー、アイテム、カテゴリーに依存しています。トランザクションをインポートできるのは、それらのオブジェクトをすべて正常にインポートした後に限られます。

このスクリプトでは、\$TOP/GDSMigration/TransformedZip ディレクトリーにある変換後の TRANSACTIONS.zip ファイルの内容をインポートします。

12. **import_type** パラメーターを 10 に設定して MigrateImport.sh スクリプトを実行します。このスクリプトでは、GDS の通知をインポートします。

```
$TOP/src/db/schema/gds/migration/v9/MigrateImport.sh
--company_code=<cmp_code> --import_type=10
```

GDS の通知は、ユーザー、アイテム、カテゴリー、トランザクションに依存しています。通知をインポートできるのは、それらのオブジェクトをすべて正常にインポートした後に限られます。このスクリプトでは、

\$TOP/GDSMigration/TransformedZip ディレクトリーにある変換後の NOTIFICATIONS.zip ファイルの内容をインポートします。

タスクの結果

インポートが正常に完了した後に、GDS 1.3.1 と IBM InfoSphere Master Data Management Collaboration Server V10.1 環境の GDS の間でアイテム、トランザクション、および通知の数に違いがあることが分かります。この違いは、インポート・プロセス中のこれら相互の依存関係が原因となって発生します。

例えば、アイテムが GDS 1.3.1.x 環境から削除された場合に、そのトランザクションと通知は削除されません。エクスポート時に、このようなアイテムはフェッチされないため、インポートにおいてもそのような環境を作成できません。トランザクションと通知をインポートしているときに、インポート・プロセスは、これらのトランザクションと通知が記録されているアイテムがあるかどうかを検査します。アイテムがない場合、これらのトランザクションと通知はインポートされません。これは、このようなトランザクションや通知があると、作業対象となる関連アイテムがない通知をユーザーがクリックした場合に、ユーザー・インターフェースに例外が発生することがあるためです。そのため、2 つの環境間に違いが生じることになります。

次のタスク

- インポート操作で生成されたレポートをチェックして、操作が正常に終了したことを確認する必要があります。
- さらに、InfoSphere MDM Collaboration Server のログで、エラーや例外があるかどうか確認できます。
- インポートしたデータに対して InfoSphere MDM Collaboration Server V10.1 のサービスを開始する前に、以下の手順を実行する必要があります。

1. GDS V1.3.x で実装チームが作成したルックアップ・テーブルに対応する項目を InfoSphere MDM Collaboration Server V10.1 インスタンスの GDSCacheRegistry.xml ファイルに追加します。このファイルは、\$TOP/etc/default ディレクトリーにあります。
2. \$TOP/etc/default ディレクトリーにある GDSCacheRegistry.xml ファイルで以下のリストを更新します。
 - 体積計測単位
 - 重量計測単位
 - 線形計測単位
 - 言語ルックアップ

階層とカタログの内容のインポート・ジョブを即時実行する

会社のメタデータまたは基本メタデータが IBM InfoSphere Master Data Management Collaboration Server V10.1.x インスタンスに正常にインポートされた後、階層内容をインポートする必要があります。インポート・スクリプトが正常に実行された後、インポート・ジョブを手動で開始して、カタログの内容を IBM InfoSphere Master Data Management Collaboration Server V10.1.x インスタンスにインポートする必要があります。

始める前に

記述フィールドに CURRENT_DATE スtringが含まれる追加のジョブがいくつか作成されている場合があります。これらのジョブは、実行せずに削除してください。

このタスクについて

MigrateImport.sh インポート・スクリプトを実行した後、**import_type** パラメーターを次のように設定します。

- 階層には 2
- カタログには 8

階層およびカタログのインポート・ジョブを手動で開始する必要があります。

手順

1. \$TOP/logs ディレクトリーを削除します。
2. インポートの処理には時間がかかるため、ユーザー・セッションがタイムアウトになることがあります。\$TOP/etc/default/common.properties ファイルの **max_inactive_interval** プロパティーに正しい値が設定されていることを確認してください。詳しくは、トピック『max_inactive_interval』を参照してください。
3. InfoSphere MDM Collaboration Server appserver およびスケジューラー・サービスのみを開始します。これらのサービスを開始する方法については、『サービスの開始』を参照してください。
4. 管理者として InfoSphere MDM Collaboration Server V10.1.x にログインします。

5. 「データ・モデル・マネージャー」->「スケジューラー」->「ジョブ・コンソール」に移動します。
6. インポート対象となる階層またはカタログの名前 (例えば HierarchyName.csv) が記述に含まれるジョブに関して、「追加」をクリックします。
7. 「スケジュール」には「即時実行」を選択します。

注: アプリケーション・サーバーおよびデータベース・サーバーの間で日付や時刻が異なる場合、「同時実行」オプションを使ってジョブの日付を遡らせることにより、ジョブを開始できます。

8. デフォルト値「指定なし」を使って「実行依頼」をクリックします。ジョブ・コンソール画面にジョブ状況が表示されます。
9. 完了状況を表示するには「アクション」アイコンをクリックします。
10. ジョブが完了した後、エラーや障害があるかどうか確認します。ジョブが完了するまで、十分な時間を与える必要があります。
11. \$TOP ディレクトリーの下にあるスケジューラーおよび appserver のログに障害や例外が含まれていないことを確認します。
12. インポートする必要のある残りの階層とカタログに関して、手順 5 から 11 までを繰り返します。
13. データ・ソース・フィード・ファイル (.csv 形式) および正常に完了したインポート・ジョブを、クリーンアップ・パースペクティブから削除します。
14. InfoSphere MDM Collaboration Server サービスを停止して、次のインポート・ステップに進みます。

インポート・スクリプトの制限事項

インポート・スクリプトの制限事項を回避するため、このガイドラインを使用してください。

- 圧縮ファイルが IBM InfoSphere Master Data Management Collaboration Server V10.1.x と互換性があることを確認します。インポート・スクリプトでは、IBM InfoSphere Master Data Management Collaboration Server V10.1.x との互換性のない圧縮ファイルは正常にインポートできません。
- インポート操作の停止がユーザーによるものではなく、システム・レベルの障害でもないことを確認します。インポート操作が停止した場合、アプリケーションの状態は不明となり、データベースは破損している可能性があります。
- 圧縮ファイルに従属オブジェクトがすべて含まれていることを確認します。圧縮ファイルに従属オブジェクトのすべてが含まれていない場合、アプリケーションで、または以前にインポートした圧縮ファイルを使用して、それらを手動で作成する必要があります。

インポートの障害に関する情報をデバッグ・レポートで確認できます。ただし、デバッグ・レポートには障害の的確な理由が記録されていない可能性があります。インポートの障害について詳しくは、InfoSphere MDM Collaboration Server ログを参照してください。

GDS V1.3.x インスタンスでのスクリプトへの変更内容はマイグレーションで失われます。変更は InfoSphere MDM Collaboration Server V10.1.x に手動でマイグレーションする必要があります。f

GDS マイグレーションによって処理されない Collaboration Server オブジェクト

いくつかの Collaboration Server オブジェクトは、ほとんどがカスタマイズされたオブジェクトであり、GDS マイグレーションによって処理されません。

例えば、レポートは GDS マイグレーションによって処理されない Collaboration Server オブジェクトです。

実装スクリプトに付加されたレポートは、IBM InfoSphere Master Data Management Collaboration Server V10.1 で使用する際にいくつかの変更が必要になります。import_type=5 のマイグレーション・ステップを実行中に、このようなレポートに関連したエラーがコンソールにスローされます。これらのエラーは、インポートに CTGTODB を使用してレポートに REPORTEXE を使用するジョブ・タイプに関連しています。インポート中に、これらのレポートは作成されますが、当該レポートに付加されているジョブは作成されません。したがって、これらのレポートをスケジュールに入れるためには、InfoSphere MDM Collaboration Server V10.1 でジョブを手動で作成する必要があります。GDS マイグレーションによって処理されないその他のオブジェクトとしては、Web サービス、拡張選択、実装スクリプト、および Collaboration Server 実装に関する文書ストアのコンテンツがあります。

マイグレーションした GDS 環境の検証

マイグレーション操作を正常に実行した後で、マイグレーションした GDS 環境を検証する必要があります。

インポートの整合性を手短かにテストするには、以下の 3 つの方法があります。

- データベース全体で、アイテム、カタログ、および仕様の数を比較します。例えば、CTG、CTR、および SPC テーブルに対して何回か照会を実行すると、インポートされた仕様とカタログの数を確かめるのに役立ちます。
- さらに総合的なテストを実施するには、InfoSphere MDM Collaboration Server V10.1.x から再度エクスポートして、WebSphere Product Center V5.2.1.4 での XSL 変換された .zip ファイルとの差分検出 (diff) を実行します。
- ビジネス・シナリオがエンドツーエンドでテストされたことを確認します。ワークフローが使用されている場合には、アイテムをチェックアウトし、ワークフロー内のすべてのステップでこれらのアイテムが期待どおり機能していることを確認し、正常終了して、スクリプトやアイテム属性に関する潜在的な問題を見つけ出します。

スクリプトのコンパイルとデバッグ

ここで示す簡単な方法でスクリプトのコンパイルとデバッグができます。

- コンパイル済みスクリプトを使用するには、common.properties ファイルに必ず以下の設定を含めてください。

```
script_execution_mode=compiled_only
```

- 個々のスクリプトについて、スクリプトの先頭に以下のディレクティブを含めることにより、スクリプトのコンパイルをオフに設定できます。

```
script_execution_mode=not_compiled
```

ただし、この方法は性能を大きく低下させるため推奨しません。非コンパイル・スクリプトを使用する場合、スクリプト・レベルでコンパイルをオフに設定するのではなく、このパラメーターを `not_compiled` に設定できます。

注: `common.properties` ファイルのサーバー設定が `not_compiled` に設定されている場合、スクリプト・レベルでディレクティブを使用しても、個々のスクリプトのスクリプト・コンパイルをオンにすることはできません。

コンパイル済みスクリプトと非コンパイル・スクリプトを組み合わせることは、パフォーマンスに悪影響を与えるため推奨しません。しかし、どうしてもそのような組み合わせを使用しなければならない場合には、非コンパイル・スクリプトでコンパイル済みスクリプトの関数を呼び出すことは可能ですが、逆の呼び出しは不可能であることに注意してください。

一般的なスクリプト・コンパイル・エラー

コンパイル済みスクリプトを使用する場合、スクリプトをスクリプト・コンソールに保存できるのは、それが正しくコンパイルされる場合だけです。エラーがある場合は、`appsvr` ログ・ディレクトリーにある `svc.out` ファイルを開いて、Java コンパイラー (`javac`) 出力の全体とエラー・メッセージを確認してください。一般的なコンパイル・エラーのいくつかを以下に示します。

- `forEach*Element()` ブロック内の `break` または `return` ステートメントは、「到達不能コード」エラーとなり、コンパイルできません。この問題を修正するには、

```
forEachItemSetElement(itemSet, item)
{
    return item;
}
```

これを次のように変更します。

```
forEachItemSetElement(itemSet, item)
{
    if (item != null) {
        return item;
    }
}
```

これは、機能として同等ですがコンパイラーでエラーにならずに済みます。

- 関数から値を返す場合、あらゆる場合に値を返す必要があります。次のサンプル・コードはコンパイルできません。

```
function sample() {
    var e = null;
    catchError (e) {
        // do something...
        return "a string";
    } if (e != null) {
        reportError(...);
    }
}
```

この関数は、`catchError` ブロックで例外が発生した場合に値が返されないため、変更しなければなりません。

```
function sample() {
    var e = null;
    catchError (e) {
```

```
// do something...

return "a string";
}
if (e != null) {
reportError(...);
}
return null;
}
```

- コンパイル時に発生する問題のほとんどについては、生成される .java ファイルを見ることができます。生成されるそれらのファイルは、common.properties ファイル内のパラメーター **tmp_dir** で指定されるディレクトリーにあります。最近、Java ファイルの命名規則は、スクリプト名と生成順序番号を含めるように変更されました。例えば、MyScript12345.java のようになります。
- 文書ストアのスクリプトの絶対パスが、生成される各 .java ファイルの先頭にコメントとして含まれるようになりました。文書ストアをファイル・システムにマッピングする場合、**grep** コマンドを再帰的に実行することにより、どの .java ファイルがスクリプトに一致するかを検出することができます。

一般的な実行時エラー

- 引数タイプが無効: 関数に誤ったタイプの引数を渡した可能性があります。例えば、String の代わりに HashMap を渡したような場合です。また、このエラーは WPC が型を正しく推論できない場合にも発生します。その場合、明示的に checkString() などのスクリプト操作を使用する必要があるかもしれません。
- 比較対象の引数のタイプが不一致: 条件演算子 (「==」, 「>」, 「<」, 「<=」など) の両辺のデータ型が同じではない場合、式の評価結果は false になります。この場合エラー・メッセージを生じさせる結果にはなりません、対応するコードは実行されません。例えば、以下のコードは動作しません。

```
var id = "12345" ;
var my_id = item.getEntryAttrib
(path to some attribute that is a sequence) ;
if ( id == my_id) {
// statements that need to be executed but won't be
}
```

これを解決するには、以下のように明示的にコーディングします。

```
var id = "12345" ;
var my_id = checkString(item.getEntryAttrib
(//some attribute that is a sequence),"") ;
if ( id == my_id) {
// statements to be executed
}
```

- XML 構文解析: 以下のコードは、非コンパイル・モードでも、さらにはコンパイル・モードでもサンドボックスから実行した場合には動作します。

```
new XmlDocument(xmlDoc) ;
forEachXmlNode("item") {
// do the needful
}
```

しかし、コンパイル・モードの場合に、このコードが複数ユーザーから呼び出されるスクリプト・ライブラリー関数で使用されるのであれば、forEachXmlNode ブロック内のステートメントは実行されません。エラー・メッセージは表示されません。回避策は以下のとおりです。

```

var doc = new new XmlDocument(xmlDoc) ;
var xmlNode ; foreachXmlNode(doc, "item", xmlNode) {
//do the needful
}

```

実行時エラーの解決

appserver での実行時エラーを解決する方法については、appsvr log ディレクトリー内のファイル svc.out を参照してください。exception.log および default.log を調べることが役立つ場合がしばしばあります。生成される .java ファイルの新しい命名規則により、どのスクリプトがエラーになったのかを容易に識別できるようになりました。また、エラー・メッセージには、生成された .java ファイルの中の行番号も示されます。問題を解決するには、生成された .java ファイルを表示して、実行時エラーの発生した行までスクロールしてください。現在では、生成された Java コードにおいて、数行ごとに実際のスクリプト・コードがコメントの形で挿入されるようになりました。例えば、生成された .java ファイルのサンプル内の、以下のコード部分を検討してみてください。

```

// function checkIfPartyPartyTypeExist(party, partyType)
public static Object ScriptFunction__checkIfPartyPartyTypeExist
  (HashMap hmContext, Object party, Object
  partyType) throws Exception
{
// var bRet = false;
Object bRet = (java.lang.Boolean) Boolean.FALSE; //
var rootEntry = party.getRootEntryNode();
Object rootEntry = GenGetRootEntryNodeOperation.execute(hmContext ,
  (IEntry) party);
// var entryNodes = rootEntry.getEntryNodes(getCatalogSpecName() +
  "/Party Types/Party Type Code");
Object entryNodes = GenGetEntryNodesOperation.execute(hmContext ,
  (EntryNode) rootEntry, (String)
  BinaryOperation.execute(BinaryOperation.PLUS,
  ScriptFunction__getCatalogSpecName(hmContext), "/Party
  Types/Party Type Code"));
// var entryNodesSize = entryNodes.size();
Object entryNodesSize = (java.lang.Integer)
  GenSizeOperation.execute(hmContext , (HashMap) entryNodes);

```

第 4 章 InfoSphere MDM Collaboration Server スクリプトのマイグレーション

IBM InfoSphere Master Data Management Collaboration Server の使用時に、スクリプトを使用して複数の関数を実行できます。スクリプトは、操作の効率や開発しやすさを損なうことなく、高度な柔軟性を実現します。

バージョン間でのスクリプトの違い

IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 で使用される基本バージョンは WebSphere Product Center バージョン 5.0.1 です。そのため、V5.2.1 と V10.1.0 の間には機能および動作の違いがあります。

データ型 *Flag* の属性によって返される値が V10.1.0 では異なっています。V5.2.1 において *Flag* データ型が返すのは TRUE/FALSE というストリングでした。V10.1.0 では、*Flag* データ型によって Java の Boolean 値に相当するものが返され、その値は true/false/null です。したがって、フラグ属性を使用する比較演算を実行する場合、比較は以下のようにしなければなりません。

```
if (flagValue==null)
```

また、例えば `out.write(flagValue);` のようにして *Flag* の値を出力する場合、出力される値は null、false、または true のいずれかになります。

compiled_only_mode を使用している場合にはコードが Java としてコンパイルされるため、IBM InfoSphere Master Data Management Collaboration Server スクリプトで Java のキーワードを変数として使用しているなら、例外がスローされます。また、スクリプトで、生成されるコードの中に InfoSphere MDM Collaboration Server で既に使用されている変数 (*logger* や *hmContext* など) が含まれている場合、コンパイルは失敗します。

別の重要な違いは、*ItemSet* または *CategorySet* のいずれかを返すメソッドに関するものです。V5.2.1 では、そのようなメソッドで結果を取り出すことができなかった場合、*null* を返していました。V5.3.2 の場合、および、結果として V10.1.0 の場合、それらのメソッドは空集合を返します。

- *ItemSet* を返すメソッドは以下のとおりです。
 - `getCtgItemByAttributeValue(String sNodePath, String sValue)`
 - `getItemSetForCategory(Catalog ctg [, Boolean ordered])`
 - `getItemSetForCatalog()`
 - `getItemSetForUnassigned(CategoryTree ctr, boolean readOnly)`
 - `getItemSetForPrimaryKeys(Array pkeys, Boolean bOptimize)`
 - `getItemSetForSelection()`
 - `getSubset(IItemSet items, Integer start_point [, Integer end_point])`
- *CategorySet* を返すメソッドは以下のとおりです。
 - `getCategorySet([Boolean bReadOnly])`

- getDescendentCategorySetForCategory([Boolean bReadOnly])
- getCategorySetByAttributeValue(String attribPath, Object attribValue, [Boolean bReadOnly])
- getCategorySetByPrimaryKey(String primaryKey, [Boolean bReadOnly])
- getCategorySetByLevel(Integer level, [Boolean bReadOnly])
- getCategorySetByFullNamePath(String[] fullNamePaths, String delimiter)
- getCategorySetByItemSecondarySpec(String specName)
- getCategorySetByStandAloneSpec(String specName)
- getLocationsAddedAvailability(CategoryTree locationHierarchy)
- getLocationsChangedToHaveData(CategoryTree locationHierarchy)
- getLocationsChangedToHaveNoData(CategoryTree locationHierarchy)
- getLocationsHavingChangedData(CategoryTree locationHierarchy)
- getAvailableLocations(Object locationOrCategoryTree)

V5.2.1 と V10.1.0 でシグニチャーに違いのあるメソッド

次の表は、IBM InfoSphere Master Data Management Collaboration Server V10.1.0 のスクリプト・ライブラリーに追加されたスクリプト操作を比較したものです。

表3. V5.2.1 と V10.1.0 でシグニチャーに違いのあるメソッドの比較

V5.2.1 の操作	V10.1.0 の操作
checkString (String str, String defaultValue)	checkString (String str, String defaultValue [, Boolean trim])
getEntriesInfoXMLInStep (String stepName,ItemSet itemSet, HashMap hmAttrPaths[, int entriesCount])	getEntriesInfoXMLInStep (String stepName,HashMap itemSet, HashMap hmAttrPaths[, int entriesCount,String dateFormat])
userDefinedLogDeleteEntriesFor (IEntry entry)	userDefinedLogDeleteEntriesFor (Entry entry [, Entry category])
userDefinedLogAddEntry (Entry entry, [String log_message])	userDefinedLogAddEntry (Entry entry, [String log_message], [Entry category])
put (String sLkpTableName, String sStartKey, [String sEndKey,] String[] asValues)	put (String sLkpTableName, String sStartKey, String[] asValues)
createWebService (String name, String desc, String wsdlDocPath, String protocol, String implScriptPath, Boolean storeIncoming, Boolean storeOutgoing, Boolean deployed [,String style])	createWebService (String name, String implclass, String desc, String wsdlDocPath, String wsddDocPath, String protocol, String style, String implScriptPath, Boolean storeIncoming, Boolean storeOutgoing, Boolean deployed [, Boolean authRequired, Boolean skipRequestValidation, Boolean skipResponseValidation])
new\$CtgItem ([String sCtgName/Catalog ctg], [Boolean bRunEntryBuildScript], [Boolean bBuildNonPersisted])	new\$CtgItem (String sCtgName/Catalog ctg], [Boolean bRunEntryBuildScript], [Boolean bBuildNonPersisted], [Boolean bBuildEmptyEntryPicture])

表 3. V5.2.1 と V10.1.0 でシグニチャーに違いのあるメソッドの比較 (続き)

V5.2.1 の操作	V10.1.0 の操作
printXML (String sTagName, String sValue, String sAttributes)	printXML (String sTagName, String sValue [, String sAttributes] [, boolean escape])
forEachXMLNode (String xpath [, XMLNode node])	forEachXMLNode ([XMLNode rootNode], String xpath [, XMLNode node])
setHierarchyMapToExport (String sourceHierarchy, String destHierarchy)	setHierarchyMapToExport (String sourceHierarchy, String destHierarchy [,String sActionMode])
newUserDefinedLogEntry (Date date, Container container, Entry entry, String log)	newUserDefinedLogEntry (Date date, Container container, Entry entry, String log [, Entry category])
addObjectByNameToExport (String sEntityName[, String sObjectType])	addObjectByNameToExport (String sEntityName[, String sObjectType, [String sActionMode]])
userDefinedLogGetEntriesFor (IEntry entry)	userDefinedLogGetEntriesFor (Entry entry[, Entry category])
buildSpec (String specName, String specType)	buildSpec (String specName, String specType, [Spec specFileType])
addAllObjectsToExport ([String sObjectType])	addAllObjectsToExport ([String sObjectType, [String sActionMode]])
new Spec (String specName, String specType)	new Spec (String specName, String specType, [String specFileType])
getItemsInStepByAttribute (String stepName,String searchAttributePath,String searchAttrValue,String sSortColumn[,boolean isAscending,int startIndex, int endIndex])	getItemsInStepByAttribute (String stepName,String searchAttributePath,String searchAttrValue,String sSortColumn[,boolean isAscending,int startIndex, int endIndex,String categoryPKList])
setItemCategoryMapToExport (String sCatalog, String sHierarchy)	setItemCategoryMapToExport (String sCatalog, String sHierarchy [,String sActionMode])
getItemXMLRepresentation (Spec spec, boolean includePrimaryKeyValue[, boolean addNameSpace, String dateFormat])	getItemXMLRepresentation (Spec spec, boolean includePrimaryKeyValue[, String dateFormat])
forEachUserDefinedLogEntry (UserDefinedLog UDL, String[] entries [, Boolean bReturnMultipleEntries = true])	forEachUserDefinedLogEntry (UserDefinedLog UDL, [Entry e,] String[] logEntries [, Boolean bReturnMultipleLogEntries = true])
setEntryAttrib (String sAttribPath, Object sValue, [Boolean bDoChecks])	setEntryAttrib (String sAttribPath, Object sValue)
invokeSoapServer (String sURL, String sMethodName, Object[] aParamValues [,String[] aParamNames])	invokeSoapServer (String sURL, String sMethodName, Object[] aParamValues [,String[] aParamNames[, String userName, String password]])
importEnv (String sDocFilePath)	importEnv (String sDocFilePath, [bFromFileSystem])
getDocByPath(String sPath)	getDocByPath(String sPath , Boolean forceSync)

表 3. V5.2.1 と V10.1.0 でシグニチャーに違いのあるメソッドの比較 (続き)

V5.2.1 の操作	V10.1.0 の操作
jmsGetQueueConnection()	jmsGetQueueConnection([String username], [String password])
sendEmail(String emailTos, String emailSubject, String emailBody, [Doc emailAttachment Doc[] emailAttachments])	sendEmail(String emailTos, String emailSubject, String emailBody, [Doc emailAttachment Doc[] emailAttachments, String ccList, String bccList])
sendHTMLEmail(String emailTos, String emailSubject, String emailBody, [Doc emailAttachment Doc[] emailAttachments])	sendHTMLEmail(String emailTos, String emailSubject, String emailBody, [Doc emailAttachment Doc[] emailAttachments, String ccList, String bccList])
addEntryIntoColArea(Entry entry, String stepPath)	addEntryIntoColArea(Entry entry, String stepPath, [boolean waitForStatus])
getColAreaByName(String colAreaName)	getColAreaByName(String colAreaName,[boolean useCache])

V5.3.0 と V5.3.0.1 でシグニチャーに違いのあるメソッド

以下の表では、WebSphere Product Center V5.3.0 と V5.3.0.1 でシグニチャーに違いのあるメソッドを比較しています。

表 4. V5.3.0 と V5.3.0.1 でシグニチャーに違いのあるメソッドの比較

V5.3.0 の操作	V5.3.0.1 の操作
getSpecByName (String name)	getSpecByName (String name, [Boolean bImmutable])

WebSphere Product Center V5.3.0.1 と InfoSphere MDM Collaboration Server V10.0.0 の間でシグニチャーに違いのあるメソッドに関する情報については、『V5.3.0.1 と V10.1.0 でシグニチャーに違いのあるメソッド』を参照してください。

V5.3.0.1 と V10.1.0 でシグニチャーに違いのあるメソッド

次の表は、InfoSphere MDM Collaboration Server V10.1.0 のスクリプト・ライブラリに追加されたスクリプト操作を比較したものです。

WebSphere Product Center V5.3.0.1 と IBMInfoSphere MDM Collaboration Server V10.1.0 でシグニチャーに違いのあるメソッド。

表 5. V5.3.0.1 と V10.1.0 でシグニチャーに違いのあるメソッドの比較

V5.3.0.1 の操作	V10.1.0 の操作
jmsGetQueueConnection()	jmsGetQueueConnection([String username], [String password])

表 5. V5.3.0.1 と V10.1.0 でシグニチャーに違いのあるメソッドの比較 (続き)

V5.3.0.1 の操作	V10.1.0 の操作
createWebService(String name, String implclass, String desc, String wsdlDocPath, String wsddDocPath, String protocol, String style, String implScriptPath, Boolean storeIncoming, Boolean storeOutgoing, Boolean deployed [, Boolean authRequired])	createWebService (String name, String implclass, String desc, String wsdlDocPath, String wsddDocPath, String protocol, String style, String implScriptPath, Boolean storeIncoming, Boolean storeOutgoing, Boolean deployed [, Boolean authRequired, Boolean skipRequestValidation, Boolean skipResponseValidation])
getDocByPath(String sPath)	getDocByPath(String sPath , Boolean forceSync)
sendEmail(String emailTos, String emailSubject, String emailBody, [Doc emailAttachment Doc[] emailAttachments])	sendEmail(String emailTos, String emailSubject, String emailBody, [Doc emailAttachment Doc[] emailAttachments, String ccList, String bccList])
sendHTMLEmail(String emailTos, String emailSubject, String emailBody, [Doc emailAttachment Doc[] emailAttachments])	sendHTMLEmail(String emailTos, String emailSubject, String emailBody, [Doc emailAttachment Doc[] emailAttachments, String ccList, String bccList])
addObjectByNameToExport (String sEntityName[, String sObjectType])	addObjectByNameToExport (String sEntityName[, String sObjectType, [String sActionMode]])
addAllObjectsToExport ([String sObjectType])	addAllObjectsToExport ([String sObjectType, [String sActionMode]])
setHierarchyMapToExport (String sourceHierarchy, String destHierarchy)	setHierarchyMapToExport (String sourceHierarchy, String destHierarchy [,String sActionMode])
setItemCategoryMapToExport (String sCatalog, String sHierarchy)	setItemCategoryMapToExport (String sCatalog, String sHierarchy [,String sActionMode])
addEntryIntoColArea(Entry entry, String stepPath)	addEntryIntoColArea(Entry entry, String stepPath, [boolean waitForStatus])
getColAreaByName(String colAreaName)	getColAreaByName(String colAreaName,[boolean useCache])

V6.0.0 と V10.1.0 でシグニチャーに違いのあるメソッド

次の表は、IBM InfoSphere Master Data Management Collaboration Server V10.1.0 のスクリプト・ライブラリーに追加されたスクリプト操作を比較したものです。

IBM InfoSphere MDM Server for PIM V6.0.0 と InfoSphere MDM Collaboration Server V10.1.0 でシグニチャーに違いのあるメソッドです。

表 6. V6.0.0 と V10.1.0 でシグニチャーに違いのあるメソッドの比較

V6.0.0 の操作	V10.1.0 の操作
String exportEnv(EnvObjectList envObjList, String sDocFilePath)	String exportEnv(EnvObjectList envObjList, String sDocFilePath, [String sMapFilePath])

V5.2.1 でのみ利用可能なメソッド

ここでは、WebSphere® Product Center V5.2.1 でのみ利用可能なメソッドのリストを示します。

- populateFromPathValueMap (HashMap pathValueHashMap)
- getItemNode (String sPath)
- getCategoryPathsForInheritedValue (String sPathDelimiter, [Boolean bWithRoot])
- setTabular (String str)
- getRootItemNode ()
- setInheritanceTargets (String[][] containers)
- getItemNodes (String sPath)
- setWflStepAttributeGroups (String[]/AttrGroup[] attrGroups)
- getMappedAttributeGroups ()
- dummy ()
- deleteInheritanceRule ()
- getInheritanceTargets ()
- getItemNodePath ()
- getCtgItemAttribs ()
- getItemSetAndCountForSelection ()
- getItemNodeValue ()
- getEntryAttribModifier (String sAttribPath)
- getChangedAttributes (Entry secondEntry)
- massUpdate (Selection sel, String scriptPath, String parametersPath)
- isEntryNodeInheritedDataFromItem ()
- getChangedAttributesForMultiOccurrence (Entry oldEntry)
- new\$AdvancedSelection (Catalog catalog, String name, String expression)
- getInheritanceRuleByName (String sRuleName)
- getCtgItemRelatedItemInfo (Integer iItemId)
- disableInheritance ()
- setItemSetFetchCategorySpecificAttributes (Boolean b)
- setItemNode (String sPath)
- debug (String s)
- getEntryNodeInheritedValueSourceEntryUniqueID ()
- getSelectionRule ()
- getItemsInheritingDataForPath (String sAttribPath)
- setItemNodeRelationshipValueUsingItem (Item relatedItem)
- new\$InheritanceRule (Container container, String ruleName)
- getItemNodeChildren ()
- getCtgItemAttributesStatus ()
- getEntryNodeInheritedDataContainerName ()
- addSubNode (Node node)

- setItemNodeRelationshipValue (Catalog relatedItemCtg, StringsRelatedItemPrimaryKey)
- reflattenAllInheritanceRules ()
- getEntryAttribModificationTime (String sAttribPath)
- removeAttributeGroup (String attributeGroupName)
- saveRule ()
- setItemNodeValue (Object value)
- addAttributeGroup (String attributeGroupName)
- setEntryAttributesFromXMLRepresentation (String XPath [,String dateFormat)
- resetAttributeGroupsToProcess ()

V5.3.0.1 でのみ利用可能なメソッド

このトピックでは、WebSphere Product Center V5.3.0.1 でのみ利用可能なメソッドのリストを示します。

- loadCatalog(String docStorePathForFileToLoad, Spec fileSpec, SpecMap specMap, String feedType [itmlicmlctr])
- getEntryRelatedItemInfo(int itemID)

V9.0.0 / V10.1.0 でのみ利用可能なメソッド

このトピックでは、IBM InfoSphere Master Data Management Collaboration Server V9.0.0 / V10.1.0 でのみ利用可能なメソッドのリストを示します。

- commitDB()
- new LinkHandler(String wflName)
- updateLinks(IEntrySet entrySet)
- New PathSelectionHandler(ICollaborationArea colArea)
- selectAndSetPath(IEntrySet entrySet,String propertyName)
- MultiMoveHandler(String wflName)
- moveEntries(IEntrySet entrySet,String stepPath)
- loadWF(String filePath)
- checkOutItem(String wfName, String itemID)
- checkOutCategory(String wfName, String categoryId,String categoryTreeName)
- new ChildCheckinHandler(String wflName)
- checkInChilds(IEntrySet entrySet, String stepPath)
- TIAttrVerifier(String wflName)
- verifyAttrGroupsAndSetExitValues(IEntrySet entrySet,String [] attrCollectionNames)
- new GDSXMLDocument(Doc doc/String str)
- getXMLNodeValue(Boolean bRequired)
- setXMLNodeValue (String sPath, String value)
- getGDSXMLNodeName()
- getXMLNodePath()
- getGDSXMLNode()

- addXMLNode(GDSXMLNode xmlNode [,String path])
- removeXMLNode(Object obj)
- getXMLNode(String path)
- cloneXMLNode()
- setValue(String path, String value)
- getGDSXMLNodes(String path)
- removeAustinNodes(String path)
- getXMLString()
- getGDSXMLNodeFromXMLNode(XMLNode xmlnode)
- getNewPublicationMessageIds()
- getNewPublicationMessageIdsByType(String[] messageTypes)
- getPublicationMessageIdsByDate(String fromDate, String toDate [,String recipientGLN])
- MessageItemHandler(String messageId)
- getTradeItems()
- sendConfirmation(String confirmationType,String[] messageIds)
- getLinks()
- changeState()
- getHierarchyTree()
- getAssociatedLinks(String gtin, Boolean fetchChildLinks)
- getInternalCode()
- setNodeType(String nodeType)
- sendCICConfirmationReasonCodes(String confirmationType,String MsgId,HashMap hmCICReasonCodeData)
- new CICStatus(String confirmationStatusCode, String confirmationStatusCodeDescription, String languageISOCODE [,String additionalConfirmationStatusDescription] [,String correctiveAction] [,String expectedCorrectiveInformation])
- getParent(int index)
- getChild(int index)
- getNumberOfChildren()
- getNumberOfParents()
- getRootTradeItem()
- isLeaf()
- getLinkQuantity()
- createNotification(IBusinessObject iObj, String notifType, String userName)
- new TradeItemSS(String sGtin, String sGTINName, String sProductType, String sInternalCatCode, [String sDataPoolCatCode])
- getTradeItemSS(String sGTIN,String sTM,String sIP)
- setTradeItemAttributesSS(HashMap hmAttributePathsToValues)
- setTradeItemAttributeSS(String attributePath,String attributeValue)

- `getTradeItemAttributeValueSS(String sAttrPath)`
- `toBoolean(String str)`
- `validateGTIN(String sGTIN)`
- `validateGLN(String sGLN)`
- `getGTINCategory(String sGTIN)`
- `getGTINTree(String sGTIN, String sProductType, Integer iLinkQuantity, String sLinkStatus)`
- `addChild(GTINTree tree)`
- `removeChild(String sGTIN)`
- `saveTradeItemSS()`
- `searchTradeItems(HashMap hmSearchParamsToValues)`
- `getTradeItemParentLinks()`
- `getTradeItemChildLinks()`
- `getModifiedTradeItems(String transactionType, Date startDate, Date endDate)`
- `getItems()`
- `setUDEXCategoryCode(String code)`
- `setInternalCategoryCode(String code)`
- `validateTradeItem()`
- `validateTradeItemAttribute(String sAttrPath)`
- `validateTradeItemAttributeGroup(String sAttributeGroup)`
- `new TradeItemDS(String sEANUCCCode, String sEANUCCCodeType, String sExtensionName, String sProductType, String sGPCCCode[,String sInternalCatCode ,String sGTIN ,String sGTINName])`
- `getExchangeSpecificAttribute()`
- `getTradeItemDS(String sEANUCCCode,String sTM,String sIP,[String sGTIN],[String recipientGLN])`
- `setTradeItemAttributesDS(HashMap hmAttributePathsToValues)`
- `setTradeItemAttributeDS(String attributePath,String attributeValue)`
- `getTradeItemAttributeValueDS(String sAttrPath)`
- `saveTradeItemDS()`
- `getAllTradeItemAttributes()`
- `searchTradeItemsDS(HashMap hmSearchParamsToValues)`
- `deleteTMIPVariant()`
- `deleteGTIN()`
- `isTradeItemCheckedOut()`
- `getWWREExchangeSpecificAttribute()`
- `isWWREInteroperablee()`
- `addItemSS(HashMap sEntryMap, String strDate)`
- `synchronizeItemChangesSS(HashMap sEntryMap, String strDate, String operationType)`

- publishNewItemSS(HashMap sEntryMap, HashMap tradingPartners,String strDate, String operationType)
- sendInitialLoadSS(HashMap sEntryMap, HashMap tradingPartners, String strDate, String operationType)
- addItemLinksForMassTransactionsSS(HashMap sEntryMap, String strDate)
- getCtgItemAttribValues()
- readGDSPropertiesSS(String sPropkey)
- searchTradeItems(HashMap hmSearchParamsToValues,String start, String end)
- processMassTransaction(String massJobID,HashMap hmSearchParamsToValues)
- HashMap setMassImportSS(boolean bImport)
- publishPublicationList(HashMap hmSearchParamsToValues)
- new TradeItemLink(String parentGTIN, String childGTIN [,Integer linkQuantity])
- setLinkAttributes(HashMap hmAttributeNameToValues [,String informationProvider])
- setLinkAttribute(String attributeName,String attributeValue [,String informationProvider])
- getTradeItemLink(String parentGTIN, String childGTIN)
- saveTradeItemLink()
- deleteTradeItemLink()
- getLinkAttributeValue(String sAttrName)
- getParentGTIN()
- getChildGTIN()
- new TradingPartner(String sGLN, String sPartnerName, String sCountry, String sGLNID, String itemType)
- new TradingPartner(String sGLN, String sPartnerName, String sCountry, String itemType)
- getTradingPartner(String sGLN)
- setPartnerAttributes(HashMap hmAttributePathsToValues)
- setPartnerAttribute(String attributePath,String attributeValue)
- saveTradingPartner()
- searchTradingPartners(HashMap hmSearchParamsToValues)
- getPartnerAttributeValue(String sAttrPath)
- createTransaction(IBusinessObject iObj, String txnType, String userName)
- isString()
- iterator()
- hasNext()
- next()
- isLangSpecificAttr(String sAttrPath)
- getGDSLanguages()
- flushGDSCache()
- new WWREMessageHandler(String wflName)
- sendMessage(IEntrySet entrySet,String messageType[, List transactionVOList])

- new MessageValidationResult(String sMsgID, String sMessage)
- new TradeItemValidationResult(String sKey)
- new AttributeValidationResult(String sKey)
- new ValidationResultObject(String sRuleName)
- addValidationResult(ValidationResultObject refValidationResultObject, String sMessage, String sStatus, String sType, Object objHolder)
- addValidationResultObject(AbstractValidationResult refAbstractValidationResult, ValidationResultObject refValidationResultObject)
- addTradeItemValidationResult(MessageValidationResult refMessageValidationResult, TradeItemValidationResult refTradeItemValidationResult)
- addAttributeValidationResult(TradeItemValidationResult refTradeItemValidationResult, AttributeValidationResult refAttributeValidationResult)
- getValidationResultObjectList(AbstractValidationResult refAbstractValidationResult)
- getTradeItemValidationResultList(MessageValidationResult msgValidationResult)
- getAttributeValidationResultList(TradeItemValidationResult tradeItemValidationResult)
- setValidationResultStatus(boolean bStatus)
- getValidationResultStatus()
- setValidationResultMessage(String message)
- setLocalizedValidationResultMessage(String messageID [,String[] arguments])
- addLocalizedValidationResultMessage(String messageID [,String[] arguments])
- getValidationResultMessage()
- setValidationResultHolder(Object holder)
- getValidationResultHolder()
- setValidationResultType(String sType)
- getValidationResultType()
- getTradeItemsInMsgItemDS(Item msgItem)
- getValidationResult(AbstractValidationResult refAbstractValidationResult)
- getMessageTypeName(Item msgItem)
- getDataPoolNameForMsgItem(Item msgItem)
- new TransactionHandlerDS()
- insertTransactionDS(ICollaborationArea colArea, IEntrySet entrySet, String stepPathString, String transactionType, String status, String stepTransition)
- updateTransactionDS(ICollaborationArea colArea, IEntrySet entrySet, String stepPathString, String transactionType, String status, String stepTransition)
- new StatusUpdaterDS()
- updateTradeItemGlobalStatusDS(IEntrySet entrySet, String status)
- updateTradeItemLocalStatusDS(IEntrySet entrySet, String status)
- updateStatusDS(IEntrySet entrySet, String status)
- new NotificationHandlerDS(String wflName)
- notifyUsersForActionDS(IEntrySet entrySet, String action)
- notifyUsersForActionWithStatusDS(IEntrySet entrySet, String action)

- new InternalSyncHandler()
- synchronize(IEntrySet entrySet)
- new EnrichmentHandler()
- canStartEnrichment(IEntrySet entrySet)
- startEnrichment(IEntrySet entrySet)
- new InitialLoadSaver()
- saveInitLoadEntries(IEntrySet entrySet)
- new MessageValidator()
- checkInitLoadAndSetExitVal(IEntrySet entrySet)
- new CategoryMappingValidator()
- verifyAndSetExitValues(IEntrySet entrySet)
- getTransactionDetails(ICollaborationArea colArea,IEntrySet entrySet,String stepPathString,String transactionType,String status,String stepTrasition)
- runWFTTest(String ar1,String arg2, String arg3)
- createTestData(String ar1,String arg2)
- GDSCreateMassPublicationItems(IEntrySet entrySet)
- GDSDeleteMassPublicationItems(IEntrySet entrySet)
- GDSCheckFilters(IEntrySet entrySet)
- LogInboundTransaction(IItem msgItem,String transactionType, String senderGLN, String msgFilePath)
- TransactionHandlerSS()
- insertTransactionSS(ICollaborationArea colArea,IEntrySet entrySet,String stepPathString,String transactionType,String status,String stepTrasition)
- updateTransactionSS(ICollaborationArea colArea,IEntrySet entrySet,String stepPathString,String transactionType,String status,String stepTrasition)
- new StatusUpdaterSS()
- updateStatusSS(IEntrySet entrySet,String status)
- updateTradeItemGlobalStatusSS(IEntrySet entrySet,String status)
- updateTradeItemLocalStatusSS(IEntrySet entrySet,String status)
- new NotificationHandlerSS(String wflName)
- notifyUsersForActionSS(IEntrySet entrySet,String action)
- notifyUsersForActionWithStatusSS(IEntrySet entrySet,String action,String status)
- invokeWorkflowSS(HashMap entryMap, String sWFName)

V5.3.2 / V10.1.0 でのみ利用可能なメソッド

ここでは、WebSphere Product Center V5.3.2 / IBM InfoSphere Master Data Management Collaboration Server V10.1.0 でのみ利用可能なメソッドのリストを示します。

- publishEntriesToSrcContainer (EntrySet entrySet)
- setLdapOperation (LdapOperation operation)

- `setVisibleAttributeGroups` (String subViewType, String[]/AttrGroup[] attrGroups, [String locationHierarchyName])
- `isItemAvailableInLocation` (Category location)
- `isBinary` ()
- `reset` ()
- `getCurrentLocation` ()
- `getEntryInStep` (String stepPath)
- `deleteLookupTable` (LookupTable lkp)
- `assertFalse` (Boolean condition, [String message])
- `getEntryAttribValues` (String sAttribPath)
- `getSearchTemplateName` ()
- `getLdapObjectclasses` ()
- `getCategoryParentsUsingCache` (CategoryCache cat_cache)
- `getCtgItemLocationAttribsForKeys` (Category location, Object[] aAttribPath [, String sDelimiter])
- `getAddedAttributePathsNewEntry` ([Category location])
- `getItemRootEntryNodesHavingLocationData` ()
- `getEntryChangedDataSinceLastSave` ()
- `getDate` (int columnIndex)
- `addLdapAttribute` (LdapAttribute attribute)
- `getLocationPathForInheritedValue` (String delimiter)
- `getModifyLocationHierarchyAvailability` (String locationHierarchyName)
- `makeItemAvailableInLocations` (Category[] locations, [Boolean bRecursive])
- `encodeUsingCharset` (String charset)
- `getLocationsAddedAvailability` (CategoryTree locationHierarchy)
- `getEntryRelationshipAttrib` (String sAttribPath)
- `new$LdapObjectclass` (String objclass)
- `getWflStepView` (String subViewType, [String locationHierarchyName])
- `getLocationsChangedToHaveData` (CategoryTree locationHierarchy)
- `setLdapDistinguishedName` (LdapAttribute dn)
- `getLocation` ()
- `xmlDocToString` ()
- `getAvailableLocations` (Object locationOrCategoryTree)
- `writeFileUsingReader` (Reader r)
- `setEditableAttributeGroups` (String subViewType, String[]/AttrGroup[] attrGroups, [String locationHierarchyName])
- `getDynamicSelectionQueryString` ()
- `setActionModeToExport` (String sActionMode)
- `addLdapEntry` (LdapEntry entry)
- `getLdapOperation` ()

- getCategory (int columnIndex)
- getHierarchy (int columnIndex)
- removeLocationSpecificData (CategoryTree ctr)
- parseDate (String value, String format[,Locale locale])
- new\$SearchQuery (String queryString)
- getStepsForEntry (Entry entry)
- getWsddDocPath ()
- setRequiredAttributeGroups (String subViewType, String[]/AttrGroup[] attrGroups, [String locationHierarchyName])
- getWflStepViews ()
- startExportByName (String sName)
- getErrorsForLocation ()
- getInt (int columnIndex)
- parseXMLNodeWithNameSpace (String sXMLSubPath)
- initializeKeyValueMapping (Object[] aKeyset)
- assertTrue (Boolean condition, [String message])
- getCtgItemAttribsForKeys (Object[] aAttribs [, String sDelimiter])
- getNodeDisplayName ([Locale locale])
- getLocationErrors ()
- getLdapEntries ()
- getRequiredAttributeGroups ([String subViewType], [String locationHierarchyName])
- getEntryChangedData (Entry oldEntry, Entry newEntry)
- getLocaleNode (Locale locale)
- locationHasData (Category location)
- execute ()
- makeItemUnavailableInLocations (Category[] locations, [Boolean bRecursive])
- getLdapAttributes ()
- assertNotNull (Object obj, [String message])
- getLocationsHavingChangedData (CategoryTree locationHierarchy)
- getLocationsHavingData (Object locationOrCategoryTree)
- getSpec (int columnIndex)
- getItemLocationAttrib (Category location, String sAttribPath)
- runQuery (String qryString)
- getLdapAttributeValue ()
- defineLocationSpecificData (CategoryTree ctr, Spec spc, AttrGroup[] inhAttrGrps)
- new\$SearchTemplate (String name, Array attrGroupNames, Container container, String desc, [String colAreaName, String stepPath])
- getBoolean (int columnIndex)
- setItemLocationData (Item item, CategoryTree locationTree, String[] locFullPaths, String delimiter, Boolean rootIncluded[, Boolean append])

- setImplclass (String implclass)
- isInheriting (Category location, String sAttribPath)
- new\$LdapEntry ()
- getLdapAttributeType ()
- setXMLNodeValue (String sPath, String value)
- parseLDIFFile (String filename)
- dropEntry (Entry entry)
- new\$DynamicSelection (String selectionName, String queryString)
- makeItemAvailableInLocation (Category location, [Boolean bRecursive])
- getCategoryChildrenUsingCache (CategoryCache cat_cache)
- isDefined (int columnIndex)
- getEntryNodeType ()
- setAuthRequired (Boolean authRequired)
- flushScriptCache ()
- getLocationsRemovedAvailability (CategoryTree locationHierarchy)
- getScriptExecutionMode ()
- getModifiedAttributePathsOldEntry ([Category location])
- copySearchItemData (Item searchItem[, Boolean append])
- copySearchItemLocationTreeData (Item searchItem, CategoryTree locationTree[, Boolean append])
- createLDIFFile (String docstore_filename, LdapEntrySet entrySet)
- getLong (int columnIndex)
- decodeUsingCharset (String charset)
- moveEntryToNextStep (Entry entry, String stepPath, String exitValue)
- isAuthRequired ()
- getSearchTemplateByName (String name)
- getItemRootEntryNodeForLocation (Category location)
- getViewableAttributeGroups ([String subViewType], [String locationHierarchyName])
- moveEntryToColArea (Entry entry, String destColAreaName)
- getCtgViewType ()
- getPossibleEntryNodeValues ()
- getItemUsingEntryRelationshipAttrib (String sAttribPath)
- javaArrayFromScriptArray (Array scriptArray, String type)
- getXMLNode (String nodePath)
- assertEquals (Object expectedValue, Object actualValue, [String message])
- addLdapObjectclass (LdapOperation objectClass)
- new\$LdapEntrySet ()
- setNodeIndexed (Boolean)
- isNodeIndexed ()

- getProductCenterURL - common.properties ファイルで定義されているプロパティ `trigo_web_url` を返します。これには、ユーザーがこの InfoSphere MDM Collaboration Server のインスタンスにアクセスするためにブラウザーにアドレスを入力する Web サイトの完全修飾 URL (ポート番号も含む) が保持されます。
- assertNull (Object obj, [String message])
- setXMLNodeValues (String sPath, String[] values)
- setModifyLocationHierarchyAvailability (String locationHierarchyName, Boolean canModifyAvailability)
- getFlatEntryNodesOf (EntryNode en)
- checkoutEntry (Entry entry, [String stepPath], [boolean waitForStatus])
- getLocationHierarchyNames ([Boolean canModifyAvailability])
- getGlobalErrors ()
- getItem (int columnIndex)
- getModifiedAttributePathsNewEntry ([Category location])
- deleteCtgView (CtgView ctgView)
- getLocationsChangedToHaveNoData (CategoryTree locationHierarchy)
- new\$searchSelection (Catalog catalog, String name)
- getDouble (int columnIndex)
- moveCursor (int position)
- getXMLNodes (String sPath)
- getLdapObjectclass ()
- new\$LdapAttribute (String attributeType, Object attributeValue [, Boolean isBinary, Boolean isExternal])
- getImplclass ()
- getAllUsers ()
- getEditableAttributeGroups ([String subViewType], [String locationHierarchyName])
- getXMLNodeValues (String nodePath [, Boolean bRequired])
- getFloat (int columnIndex)
- makeItemUnavailableInLocation (Category location, [Boolean bRecursive])
- deleteSearchTemplate ()
- getDeletedAttributePathsOldEntry ([Category location])
- isExternal ()
- setInheriting (Category location, String sAttribPath, [Boolean flag])
- setWsddDocPath (String wsddDocPath)
- getString (int columnIndex)
- setItemLocationAttrib (Category location, String sAttribPath, Object sValue)
- scriptArrayFromJavaArray (OneDimensionalJavaArray)
- setEntryAttribValues (String sAttribPath, Object[] sValues)
- getLdapDistinguishedName ()
- executeInBackground (String selectionName)
- getLocationForErrors ()

- disableEntryValidation(Entry collabAreaEntry, String workflowStepAtPath, Boolean disableValidation)
- isEntryNew()
- item.getCtgItemRelatedItemInfo()

注:

- これは、WebSphere Product Center V5.2.1 での使用すべきでないスクリプト操作です。
- このスクリプト操作は、InfoSphere MDM Collaboration Server V9.0.0 において **entry::getEntryRelationshipAttrib(string attribpath)** に置き換えられました。

- moveUserToOrganization(Category srcOrg,Category dstOrg)
- copyUserToOrganizations(Category[] organizations)
- removeUserFromOrganization(Category organization)
- invocationCacheGet(String key)
- invocationCachePut(String key, Object value)
- invocationCacheRemove(String key)
- invocationCacheClear()
- setDynamicSelectionQueryString(String queryString)
- getExactAttributePath()

V6.5.0 / V10.1.0 でのみ利用可能なメソッド

このトピックでは、IBM InfoSphere Master Data Management Collaboration Server V6.5.0 / V10.1.0 でのみ利用可能なメソッドのリストを示します。

- String ExcelCell::getExcelCellEncoding()

V5.2.1 および V10.1.0 からのスクリプト操作の動作に関する変更

このトピックでは、WebSphere Product Center V5.2.1 と IBM InfoSphere Master Data Management Collaboration Server V10.1.0 の両方のスクリプト・ライブラリーにおけるスクリプト操作の動作に関する変更をリストします。

setEntryNodeValue()

V5.2.1 での動作

1. ノードが存在しない場合、新規作成されます。
2. 設定された値を返します。

V10.1.0 での動作

1. ノードがまだ存在しない場合は例外をスローします。その場合、`entry.setEntryAttrib()` を使用して新しいノードを作成する必要があります。
2. 実行されたアクションに応じて 1/0/-1 (成功/変更なし/エラー) を返します。

getEntryNode()

V5.2.1 での動作

1. 新しい項目/カテゴリが作成されると、デフォルトで **MultiAttribute** ノードの 1 つのオカレンスが存在します。

```
itemNode.getEntryNode("/SpecForMigTest/Multi#0/Sub-Multi#0")
```

は、ノードを返します。

2. 指定されたパスが見つからない場合、V5.2.1 ではそれを作成して、ノードを返します。パスが無効の場合はヌル (null) を返します。

V10.1.0 での動作

1. パスがまだ存在していない場合、V10.1.0 では例外がスローされます。
2. V10.1.0 ではノードが作成されません。

setEntryNode()

V5.2.1 での動作

1. V5.2.1 では、ノードを作成してそれを返します。パスが無効の場合は、ヌル (null) を返します。

V10.1.0 での動作

1. パスがまだ存在していない場合、V10.1.0 では例外がスローされます。

getEntryNodeChildren()

V5.2.1 での動作

1. V5.2.1 では、グループ化ノードの実際の子を返します。

```
itemNode = item.getRootEntryNode(); node = itemNode.setEntryNode("/SpecForMigTest/Multi");
```

出力:

```
SpecForMigTest/Multi/Sub-Multi
```

V10.1.0 での動作

1. V10.1.0 では、実際の子ではなく、Multi のオカレンスを返します。例えば、次のようになります。

```
itemNode = item.getRootEntryNode(); node =  
itemNode.setEntryNode("/SpecForMigTest/Multi");
```

出力: /SpecForMigTest/Multi#0 /SpecForMigTest/Multi#1

getEntryNodes()

V5.2.1 での動作

1. V5.2.1 では、スクリプト操作に指定されているパスが子の子に含まれている場合に、子も返されます。

```
itemNode = item.getRootEntryNode(); nodes =  
itemNode.getEntryNodes("/SpecForMigTest/Multi"); out.println(nodes);
```

V10.1.0 での動作

1. V10.1.0 では、子は返されません。

getEntryNodeParent()

V5.2.1 での動作

1. V5.2.1 では、実際の親が返されます。

```

itemNode = item.getRootEntryNode(); node =
itemNode.getEntryNodes("/SpecForMigTest/Multi#0/Sub-Multi#0");
out.writeln("Node:: " + node[0].getEntryNodePath());
out.writeln("Node Parents:: "
+ node[0].getEntryNodeParent().getEntryNodePath());

```

出力:

```
Node:: SpecForMigTest/Multi/Sub-Multi Node Parents:: SpecForMigTest/Multi
```

V10.1.0 での動作

1. V10.1.0 では、実際の親ではなくグループ・ノードが返されます。

```

itemNode = item.getRootEntryNode(); node =
itemNode.getEntryNodes("/SpecForMigTest/Multi#0/Sub-Multi#0");
out.writeln("Node:: " + node[0].getEntryNodePath());
out.writeln("Node Parents:: "
+ node[0].getEntryNodeParent().getEntryNodePath());

```

出力:

```
Node:: SpecForMigTest/Multi/Sub-Multi Node Parents::
SpecForMigTest/Multi/Sub-Multi
```

getFlatEntryNodes()

V5.2.1 での動作

1. このメソッドは、このエントリーの `EntryNodes` のフラットな配列を返します。

V10.1.0 での動作

旧バージョンでは、このメソッドは例外をスローします。その例外の原因は、スクリプトにおいて、`getFlatEntryNodes` から返される `entryNode` に対して `getEntryNodeValue` が呼び出されることによります。これは、現在では直列化形式で、`getFlatEntryNodes` から新しくディレクトリー・ノードと呼ばれる `entrynode` が返されるからです。複数回出現する単一属性については、余分の `entrynode` が返され、その `node::type` は、`STRING` か、または属性のタイプになります。スクリプトは、値を取得してそれを報告するよう試みます。そのため、バージョン 5.3.2 では修正が用意されており、`getFlatEntryNodes` がノードを返した後、新しいメソッド `getEntryNodeType()` によってそのタイプがチェックされます。タイプが `V` であれば、その場合のみエントリー・ノードが処理されます。メソッドのプロトタイプは次のとおりです。

```

<prototype>
  <![CDATA[String EntryNode::getEntryNodeType()]&gt;
</prototype>
<description>
  <![CDATA[return "V" = value, "G" = Grouping or top level of spec directory,
  "M" = Multi-directory (contains multiple occurrences of values or groupings)]]&gt;
</description>

```

実装例を以下に示します。

```

ctg1 = getCtgByName( "<catalog name>" );
itm1 = ctg1.getCtgItemByPrimarykey( "<primary key of item>" );
ents1 = itm1.getFlatEntryNodes();

for( i=0; i<ents1.size(); i++ )
{
  var val1 = "----";
  var ent1 = ents1[i];

  var ePath1 = ent1.getEntryNodeExactPath();
  out.write( ePath1 + " " );

  var type1 = ent1.getEntryNodeType();
  out.write( type1 + " " );

  if( type1 == "V" )
  {
    val1 = "" + ent1.getEntryNodeValue();
  }
}

```

```

    }
    else
    {
        val1 = "" + ent1.getEntryNodeChildren().size();
    }
    out.writeIn( val1 );
}

```

setTypeToExport(String sObjectType)

V5.2.1 での動作

- sObjectType に対して COLLABORATION_AREA_CONTENT は許容値です。

V10.1.0 での動作

- sObjectType に対して、COLLABORATION_AREA_CONTENT は許容値のリストから削除されています。

addObjectByNameToExport(String sEntityName[, String sObjectType, [String sActionMode]])

V5.2.1 での動作

- sObjectType に対して COLLABORATION_AREA_CONTENT は許容値です。

V10.1.0 での動作

- sObjectType に対して、COLLABORATION_AREA_CONTENT は許容値のリストから削除されています。

addAllObjectsToExport([String sObjectType, [String sActionMode]])

V5.2.1 での動作

- sObjectType に対して COLLABORATION_AREA_CONTENT は許容値です。

V10.1.0 での動作

- sObjectType に対して、COLLABORATION_AREA_CONTENT は許容値のリストから削除されています。

V5.3.0.1 および V10.1.0 からの既存のスクリプト操作の動作に関する変更

このトピックでは、WebSphere Product Center V5.3.0.1 と IBM InfoSphere Master Data Management Collaboration Server V10.1.0 の両方のスクリプト・ライブラリーにおけるスクリプト操作の動作に関する変更をリストします。

setContainerProperties()

V5.3.0.1 での動作

- PROPERTIES ハッシュ・マップで指定されているプロパティーが、このコンテナに対して設定されます。

V10.1.0 での動作

- PROPERTIES ハッシュ・マップで指定されているプロパティーが、このコンテナに対して設定されます。ハッシュ・マップ・キー「SCRIPT_NAME」は現在は使用すべきでなく、代わりに「POST_SCRIPT_NAME」を使用する必要があります。

getCategoryAttrib()

V5.3.0.1 での動作

- このカテゴリの属性 sAttribPath (spec_name/attribute_name) の値を返します。

V10.1.0 での動作

- 「表示」または「属性コレクション」に「属性」が含まれる場合のみ、このカテゴリの属性 sAttribPath (spec_name/attribute_name) の値を返します。含まれない場合はヌルを返します。

getEntryAttrib()

V5.3.0.1 での動作

- このエントリーの属性 sAttribPath (spec_name/attribute_name) の値を返します。

V10.1.0 での動作

- 「表示」または「属性コレクション」に「属性」が含まれる場合のみ、このエントリーの属性 sAttribPath (spec_name/attribute_name) の値を返します。含まれない場合はヌルを返します。

getCtgItemAttrib()

V5.3.0.1 での動作

- このアイテムの属性 sAttribPath (spec_name/attribute_name) の値を返します。

V10.1.0 での動作

- 「表示」または「属性コレクション」に「属性」が含まれる場合のみ、このアイテムの属性 sAttribPath (spec_name/attribute_name) の値を返します。含まれない場合はヌルを返します。

removeNode()

V5.3.0.1 での動作

- Boolean を返します。

V10.1.0 での動作

- void を返します。

setTypeToExport()

V5.3.0.1 での動作

- エクスポートするオブジェクト・タイプを設定します。

V10.1.0 での動作

- エクスポートするオブジェクト・タイプを設定します。 sObjectType に対して、COLLABORATION_AREA_CONTENT、INHERITANCE_RULES は許容値のリストから削除されています。

addObjectByNameToExport()

V5.3.0.1 での動作

- sObjectType に対して COLLABORATION_AREA_CONTENT は許容値です。

V10.1.0 での動作

- sObjectType に対して、COLLABORATION_AREA_CONTENT は許容値のリストから削除されています。

addAllObjectsToExport()

V5.3.0.1 での動作

- sObjectType に対して COLLABORATION_AREA_CONTENT は許容値です。

V10.1.0 での動作

- sObjectType に対して、COLLABORATION_AREA_CONTENT は許容値のリストから削除されています。

disableContainerProcessingOptions(String[] options)

V5.3.0.1 での動作

- 指定された処理オプションを使用不可にします。

V10.1.0 での動作

- 指定された処理オプションを使用不可にします。他に考えられるオプション・パラメーター値は、ENTRY_BUILD_SCRIPT および UNIQUE_VALIDATION です。

V6.0.0 および V10.1.0 からのスクリプト操作の動作に関する変更

このトピックでは、IBM InfoSphere Master Data Management Collaboration Server V6.0.0 および V10.1.0 のスクリプト・ライブラリーにおけるスクリプト操作の動作に関する変更をリストします。

exportEnv ()

V6.0.0 での動作

- 指定された文書ストア・パスにある、envObjList の中で指定された InfoSphere MDM Collaboration Server オブジェクトをエクスポートします。 *sDocFilePath* 変数には、文書ストアにエクスポートされる .zip ファイルのパスが入ります。このメソッドは、ログをストリングとして返します。

V10.1.0 での動作

- 指定された文書ストア・パスにある、envObjList の中で指定された InfoSphere MDM Collaboration Server オブジェクトをエクスポートします。 *sDocFilePath* 変数には、文書ストアにエクスポートされる .zip ファイルのパスが入ります。 *sDocFilePath* が「file:///」およびそれに続くファイル・システムのパスとして指定される場合、.zip ファイルは、文書ストアではなくファイル・システムに直接エクスポートされます。オプション・パラメーター *sMapFilePath* は、ファイル・システム上の名前マッピング・ファイルのパスです。このファイルは、エクスポートされるオブジェクトのための、オブジェクトからファイルへの名前マッピングを示すものです。このメソッドは、ログをストリングとして返します。

バージョン間での Java API の違い

IBM InfoSphere Master Data Management Collaboration Server バージョン 10.1.0 の Java API で使用されている基本バージョンは、WebSphere Product Center バージョン 5.3.2 です。

Java API メソッドおよび列挙型

このトピックでは、IBM InfoSphere MDM Server for PIM V6.0.0 フィックスパック 5、V9.1.0、および IBM InfoSphere Master Data Management Collaboration Server V10.0 に存在するインターフェースに新しく追加されたメソッドおよび列挙型をリストします。

AttributeDefinition.java

列挙型: AttributeDefinition.Type: RICH_TEXT および EXTERNAL_CONTENT_REFERENCE を追加

メソッド: public int getId()

AttributeDefinitionProperty.java

列挙型: AttributeDefinitionProperty.Name: GROUP および LOCALE_NODE が非推奨

列挙型: AttributeDefinitionProperty.Name: RICH_TEXT を追加

AttributeInstance.java

メソッド: public void setValueUsingJSON(Object value)

AttributeOwner.java

メソッド: public AttributeInstance getAttributeInstance(int[] nodeIdArray, int[] indexArray, boolean bFollowInheritance, boolean createOccurrencesIfNeeded)

Item.java

メソッド: public OrderedJSONObject toJSON(boolean includeAttributeType)

メソッド: public void addStatusAndAvailability(OrderedJSONObject itemJSONObj)

Catalog.java

メソッド: public PIMCollection<Item> getItems(int[] itemIds)

メソッド: public PIMCollection<Item> getItems(Object[] itemIds)

メソッド: public void setIgnoreCache(boolean ignoreCache)

メソッド: public boolean getIgnoreCache()

メソッド: public boolean isCached()

メソッド: public void setCached(boolean setCached)

メソッド: public OrderedJSONObject toJSON()

CatalogManager.java

メソッド: public Catalog getCatalog(String catalogName, boolean returnItem)

メソッド: public Catalog getCatalog(int catalogId)

メソッド: public Catalog getCatalog(int catalogId, boolean returnItem)

CategoryCollaborationArea.java

メソッド: public CollaborationCategory createCollaborationCategory()

CollaborationCategory.java

列挙型: CollaborationCategory.SaveResult

メソッド: public SaveResult getSaveResult()

メソッド: public Collection<SecondarySpec> getItemSecondarySpecs()

メソッド: public OrderedJSONObject toJSON(boolean includeAttributeType)

メソッド: public void addStatusAndAvailability(OrderedJSONObject

collaborationCategoryJSONObj)

メソッド: public void addItemSecondarySpec(SecondarySpec spec, Collection<Catalog> catalogs, boolean addToChildCategories, boolean addAcrossMapping)

メソッド: public void addSecondarySpec(SecondarySpec spec)

メソッド: public void removeItemSecondarySpec(SecondarySpec spec)

CollaborationItem.java

メソッド: public OrderedJSONObject toJSON(boolean includeAttributeType)

メソッド: public void addStatusAndAvailability(OrderedJSONObject collaborationItemJSONObj)

CollaborationStep.java

メソッド: public CollaborationObject getCollaborationObject(int id)

列挙型: ValidationError.Type

TOTAL_DIGITS、MIN_EXCLUSIVE、MIN_INCLUSIVE、MAX_EXCLUSIVE、MAX_INCLUSIVE を追加

Context.java

メソッド: SearchQuery createSearchQuery(String queryString, Collection<Item> items, Catalog catalog)

メソッド: public String setSavepoint()

メソッド: void rollbackToSavepoint(String name)

メソッド: public SearchHelper getSearchHelper()

メソッド: public HistoryManager getHistoryManager()

メソッド: public MessageBundle getMessageBundle()

メソッド: public MessageBundle getMessageBundle(Locale locale)

Category.java

列挙型: Category.SaveResult

メソッド: public Collection<String[]> getFullDisplayPaths()

メソッド: public Collection<Integer> getSecondarySpecIdsForChildren()

メソッド: public OrderedJSONObject toJSON(boolean includeAttributeType)

メソッド: public void addStatusAndAvailability(OrderedJSONObject categoryJSONObj)

Hierarchy.java

メソッド: public Category createCategory()

メソッド: public PIMCollection<Category> getCategories(int[] categoryIds)

メソッド: public PIMCollection<Category> getCategories(Object[] categoryIds)

メソッド: public OrderedJSONObject toJSON()

HierarchyManager.java

メソッド: public Hierarchy getHierarchy(int hierarchyId)

OrganizationManager.java

メソッド: public OrganizationHierarchy getOrganizationHierarchy(int id)

SearchQuery.java

列挙型: SearchQuery.ColumnType: LOGENTRY を追加 列挙型:
SearchQuery.SearchQueryOp

SearchResultSet.java

メソッド: public UserDefinedLogEntry getUserDefinedLogEntry(int
columnIndex) throws PIMSearchException

SelectionManager.java

メソッド: public Selection getSelection(int selectionId)

Spec.java

メソッド: public AttributeDefinition getAttributeDefinition(int
attributeDefinitionId)

SpecManager.java

メソッド: public Spec getSpec(int specId)

EnvironmentExporter.java

メソッド: public String export(ExportList exportList, String documentPath,
String mappingPath)

メソッド: public String export(ExportList exportList, String documentPath,
boolean

checkForRequisites, String mappingPath)

メソッド: public String exportAll(String documentPath, String mappingPath)

新しい Java API インターフェース

このトピックでは、IBM InfoSphere MDM Server for PIM V9.1.0 および IBM InfoSphere Master Data Management Collaboration Server V10.0 で導入された新しいインターフェースをリストします。

- Filter
- PIMObject
- HistoryManager
- CMSContentURN
- CMSInstance
- CMSManager
- AuthorizationHelper
- SavedSearch
- SearchHelper
- LeftPaneDataObject
- LeftPaneDataObjectFactory
- RichTextEditor
- MessageBundle

データベースのチューニング

データベース・サーバーとして IBM DB2® を使用している場合、ヒープ・サイズ設定値を増やすことによりデータベースをチューニングする必要があります。

始める前に

DB2 データベースのチューニングを行う前に、DB2 管理者としてログインする必要があります。

手順

以下のコマンドを実行してヒープ・サイズ設定値を増やします。

```
db2 update database configuration for database <dbname> using applheapsz 8192
db2 update database configuration for database <dbname> using app_ctl_heap_sz 8192
db2 update database configuration for database <dbname> using LOGFILSIZ 4000
```

ここで、*dbname* は DB2 データベース・インスタンスの実際の名前です。これらのパラメーターの現行値を確認するには、以下のコマンドを使用できます。

```
db2 get dbm cfg
db2 get db cfg for <dbname>
```

注: 上記の DB2 コマンドでの数値は、説明のための例にすぎません。実際の環境における正しいパラメーター値を DBA に問い合わせてください。

データ・モデル・メタデータの変換

IBM InfoSphere Master Data Management Collaboration Server V9.1.0 で使用するために WebSphere Product Center V5.2.1.4 のメタデータを変換する必要があります。

このタスクについて

WebSphere Product Center V5.2.1.4 と InfoSphere MDM Collaboration Server V9.1.0 は、それぞれ異なる方法でデータを保管し、データ・モデルを処理します。メタデータを InfoSphere MDM Collaboration Server V9.1.0 規格に準拠させるには、メタデータを変換する必要があります。

データを変換するには `migrateCompanyZip.sh` スクリプトを使用できます。このスクリプトはエクスポート処理で生成された圧縮ファイルを読み取り、圧縮ファイルの内容を調べてデータ・モデル非互換性の問題を検査し、同じ名前の新しい圧縮ファイルを作成します (それを InfoSphere MDM Collaboration Server V9.1.0 インスタンスにインポートできます)。このシェル・スクリプトは、圧縮ファイルの内容に基づいて開始される XSL 変換を内部的に使用します。

また、このスクリプトを実行するときには、ターゲットの会社に関して実行時に検索可能となる必要のある属性を定義する XML ファイルを指定することもできます。この機能はパフォーマンスを改善するために導入されました。

データのマイグレーション前に、実行時に検索可能となる必要のあるすべての属性が RTS として定義されていることを確認してください。「仕様」コンソールを介

してマイグレーション後に属性が RTS に変更された場合、この変更前にマイグレーションまたは作成されたすべてのアイテムを RTS にするために、それらを再び保存する必要があります。そのようなアイテムが多数ある場合、この作業に多くの労力が必要になる可能性があります。マイグレーション前にこのような属性を検査することをお勧めします。

手順

migrateCompanyZip.sh スクリプトを実行します。このスクリプトは \$TOP/bin ディレクトリにあります。

```
migrateCompanyZip.sh [--debug=yes] [--verbose=yes]
[--param=<filename>] --out=<output directory> <input zip>
```

この例で、migrateCompanyZip.sh は companySpecs.zip ファイルの内容を変換します。さらに、ターゲット会社で実行時検索可能 (RTS) になる必要のあるすべての属性を識別するために params.xml ファイルを使用します。

```
$TOP/bin/migrateCompanyZip.sh --debug=yes --verbose=yes
--param=$TOP/params.xml --out=$TOP/out $TOP/input/companySpecs.zip
```

ここで、\$TOP は InfoSphere MDM Collaboration Server のインストール・ディレクトリを指定します。

XSLT トランスフォーマーによる修正

XSLT トランスフォーマーは、選択エラーを修正するように設計されています。

仕様の変更

IBM InfoSphere Master Data Management Collaboration Server V9.1.0 は、以下のものをサポートしていません。

- PERIOD SpecNode タイプ - WebSphere Product Center V5.2.1.x からマイグレーションしていて PERIOD SpecNode タイプを使用している場合、マイグレーション・ユーティリティーは PERIOD SpecNode タイプを GROUP に変換します。
- 属性値が NULL の SpecNode。例えば、SpecNode 属性 MAX_INCLUSIVE、SCALE、MAX_EXCLUSIVE など。MIN_EXCLUSIVE は NULL 値にすることはできません。変換ユーティリティーは、このような属性プロパティを削除します。

ワークフローの変更

InfoSphere MDM Collaboration Server V9.1.0 には StepType PARTIAL_UNDO がありません。これと置き換わる同等の値は INTERIM_CHECKOUT です。したがって、変換ユーティリティーは PARTIAL_UNDO ステップ・タイプのすべての出現箇所を INTERIM_CHECKOUT に変換します。

ステップに関連付ける属性

WPC V5.2.1.x では、ステップ PARTIAL_UNDO および MODIFY だけが、それらに関連付けられた属性コレクションを持つという制約がありました。

InfoSphere MDM Collaboration Server V9.1.0 では、ステップ AUTOMATED、INTERIM_CHECKOUT、AND_APPROVAL、MODIFY、INTERIM_CHECKIN、GENERAL、OR_APPROVAL、および DISPATCH が、それらに関連付けられた属性コレクションを持つという制約があります。

ワークフローのインポートの前に、上記のステップ・タイプについて、関係する属性コレクションを含めることにより、zip ファイル内の XML を変更する必要があります。XML は次のようになります。

```
<Step>
....
  <StepType>AUTOMATED</StepType>
....
  <EditableAttributesCollections>
    <ItemEdit>
      <AttributeCollection>Attribute Collection 1</AttributeCollection>
    </ItemEdit>
  </EditableAttributesCollections>
....
</Step>
```

コラボレーション・エリアのある左ペイン

InfoSphere MDM Collaboration Server V9.1.0 では、左ペインのコラボレーション・エリアを、カタログや階層と同じように使用できます。これは、コラボレーション・エリア・コンソールに移動しなくても、異なるステップに素早くナビゲートする手段となります。

始動時のワークフロー・エンジンのリカバリー

WPC V5.3 以降、ワークフロー・エンジンが始動するときに、依然として PROCESSING とマークされているワークフロー・イベントがあれば、それらすべては ERROR に設定されて、問題があったことを示します。これがいっそう重要になったのは、ほとんどのワークフロー・イベント処理がアプリケーション・サーバーではなくワークフロー・エンジンで行われるためです。

ワークフローにおける属性の検証

InfoSphere MDM Collaboration Server V9.1.0 では、ワークフロー内の各ステップで属性コレクションを割り当てることができます。旧バージョンの WPC では、ユーザーはワークフロー全体で属性コレクションを選択していました。

検証する属性

WPC V5.3 以降、各ステップが 3 つまで、割り当てられた属性コレクションを持ちます。

- 表示 - これらの属性の値が表示されるのを許可します。
- 編集 - これらの属性の値が表示および編集されるのを許可します。保管の際に、これらの属性は仕様内で定義されたルールに照らして検証されます。
- 必須 - これらの属性は、表示と編集が可能です。すべての必須属性に、値が提供されなければなりません。保管の際に、これらの属性は仕様内で定義されたルールに照らして検証されます。

InfoSphere MDM Collaboration Server V9.1.0 でのワークフロー 検証ルール

- 終了しようとしているステップのルールによって有効であると見なされない限り、アイテムは次のステップに移動することが許可されません。これが最も重要なルールです。言い換えると、ソース・コンテナの検証ルール、特に必須属性は、ワークフローの検証設定によってオーバーライドされることになります。例えば、MinOccurs=1 などの必須属性は、ワークフローの検証設定によってオーバーライドされます。
- ワークフロー内で必須であり変更可能な属性は、たとえ仕様にルールが指定されている場合でも、それらの属性を変更できるステップ上の必須属性コレクションに追加されなければなりません。
- 各アイテムが固有の基本キーを持っていないければなりません。
- 各オブジェクトは、チェックイン (その時点でソース・コンテナ・キーの固有性がチェックされる) まで、コラボレーション・エリア内で固有でなければなりません。
- ソース・コンテナ内のエントリーと競合するエントリーは FIXIT にルーティングされます。
- ステップ内の編集可能属性または必須属性だけが検証されます。

ドラフトとしての保存

WPC V5.3 では、ワークフロー・ステップ内の個々の無効データを保存しておき、あとでそのステップに戻ることができます。旧バージョンでは、他の画面にナビゲートしたときに、何らかの無効な属性があった場合にはデータが失われていました。このことが生じるのは、「保存」または「完了」をクリックして、画面上に検証エラーが表示されることになる場合だけです。ドラフトとしての保存は、型変換を通過する値だけに機能します。ドラフトとしての保存のオプションが機能するためには、コラボレーション・エリア内で基本キーが有効で固有でなければなりません。無効データを持つエントリーは、それを修正するまでは次のステップに移動することができません。正常に終了するには、ステップのルールによってエントリーが有効とされなければならないからです。

ドラフトとしての保存の落とし穴

タイムアウトのトリガーを経てエントリーがステップから出る場合、またはエントリーが FIXIT (調整) に送られている場合、検証は実行されません。FIXIT ステップで検証するために、ワークフローがかかわりを持つすべての属性を編集可能属性として割り当てていることを確認してください。そうしないと、無効属性を調べたり訂正したりすることはできません。その場合に可能なオプションは、以下のものだけとなります。

- 無効データをチェックインする。
- コラボレーション・エリアからアイテムを除去する。

あらゆるタイムアウトのターゲット・ステップに、タイムアウトが起動される前に変更された可能性のあるすべての属性が組み込まれていることを確認してください。この点で、タイムアウト・チェーンや、ネストしているワークフロー・タイムアウト・エスカレーションでは細心の注意を払う必要があります。疑わしい場合には、すべての属性を組み込んでください。

ゴースト・アイテムの処理

WPC V5.2.1.1 では、ゴースト・アイテムが発生する原因のほとんどは除去されました。ただし、ゴースト・カテゴリについては、対応がなされませんでした。その後、以下のように、いくつかの変更が行われました。

- ステップ遷移中のトランザクション・ロジックの強化により、ワークフロー・テーブル更新への「全部かゼロか」のアプローチが 1 つのトランザクションで行われることが保証されます。
- 競合を減らすために、ワークフロー・エンジンを 1 つにする制限が実施されるようになりました。
- ステップ・スクリプトの接続分離。
- ステップ・スクリプトは、それらの失敗がワークフロー・テーブル更新に影響しないように、それぞれ別個のデータベース接続で実行されるようになりました。
- ワークフロー処理は、アプリケーション・サーバー上で同期的に行われるのではなく、ワークフロー・エンジン上ですべて非同期で行われるようになりました。これによってワークフロー遷移が実行される前にデータベースに確実に記録されることにもなり、より良いリカバリーとエラー処理が可能になりました。

文書ストアの変更

WPC V5.2.1 でのエクスポート zip ファイル DOCSTORE.xml に対して変換ユーティリティー migrateCompanyZip.sh を実行している間に、このファイルから以下のファイルが除去されます。したがって、このファイルは InfoSphere MDM Collaboration Server V9.1.0 にマイグレーションされません。このファイルを InfoSphere MDM Collaboration Server V9.1.0 に変更する必要がある場合は、手動でマイグレーションしなければなりません。

```
/scripts/import/ctg/environment_catalog_content_import.script  
/scripts/import/ctr/environment_category_tree_content_import.script  
/scripts/logout/Logout.wpcs  
/scripts/login/Login.wpcs  
/scripts/triggers/LDAPLibrary.wpcs  
/scripts/triggers/LdapLibraryLogger.wpcs
```

ワークフロー内のデフォルト・ステップに関連付けられたスクリプトは、それに相当する大文字の名前に変更されます。例えば、Fixit スクリプトは FIXIT と名前変更されます。

スクリプトの手動マイグレーション

ターゲットの会社のデータをインポートする前に、マイグレーションしたスクリプトが正常に機能することを確認する必要があります。

このタスクについて

スクリプトをコンパイルすることによってシグニチャーの変更や戻り値の欠落のような構文レベルのエラーのほとんどを識別し修正することができます。

checkForCompileError.sh スクリプトを使用して、スクリプトを自動でコンパイルできます。このスクリプトを実行すると、各スクリプトのエラーをリストしたレポートが生成されます。シグニチャーと動作に関するスクリプト・レベルでの変更の完全なリスト、および一般的な問題とその解決方法については、『InfoSphere

MDM Collaboration Server スクリプトのコンパイルとデバッグ』を参照してください。

手順

1. checkForCompileError.sh スクリプトを実行します。このスクリプトは、\$TOP/bin ディレクトリーにあります。

```
$TOP/bin/checkForCompileError.sh --company_code=<code>
--script_dir=<path/to/wpc/script/directory> --logfile=<logfile path>
```

このスクリプトは、すべてのスクリプトが格納されているディレクトリーを引数として取り、そのディレクトリーとそのサブディレクトリーの各スクリプトのコンパイルを試行します。この例では、checkForCompileError.sh スクリプトを実行して、/scripts ディレクトリーのスクリプトをコンパイルします。compiled_only スクリプト・モードを使用して /scripts ディレクトリーにあるすべてのスクリプトを確認するには、common.properties ファイルを修正して、**script_execution_mode=compiled_only** を設定し、それから checkForCompileError.sh スクリプトを実行する必要があります。

```
$TOP/bin/checkForCompileError.sh --company_code=test
--script_dir=/scripts --logfile=logfile.txt
```

logfile.txt ファイルが生成されます。このファイルの末尾に、以下の形式のレポートを確認できます。

```
Fri Jan 07 15:25:06 IST 2011 - *****
Fri Jan 07 15:25:06 IST 2011 - Script Mode : 'compiled_only'
Fri Jan 07 15:25:06 IST 2011 - Total Scripts : '10'
Fri Jan 07 15:25:06 IST 2011 - Successful : '8'
Fri Jan 07 15:25:06 IST 2011 - Failed      : '2'
Fri Jan 07 15:25:06 IST 2011 - /scripts/import/ctr/abhitest -
The script could not be compiled.
Fri Jan 07 15:25:06 IST 2011 - /scripts/import/ctr/AbhiFailureWrongScript -
The script object could not be built.
[[Undefined operation: getEntryAttribs at line 3]]
Fri Jan 07 15:25:06 IST 2011 - *****
```

2. checkChangedScriptOps.sh を実行して、WebSphere Product Center v5.2 およびそのフィックスパックに対するシグニチャーや動作に関するスクリプトのリスト、または WebSphere Product Center V5.3.x から削除されたスクリプトのリストを取得します。このスクリプトは、\$TOP/bin ディレクトリーにあります。スクリプトの作動方法の変更について詳しくは、InfoSphere MDM Collaboration Server のインフォメーション・センターを参照してください。このスクリプトは、初期の見積もりとその後のインプリメンテーション・フェーズにおいて役立ちます。ここで、さらに注意を向ける必要のあるスクリプトのリストと、変更がなされたと思われるスクリプト内の正確な位置を取得できます。

```
$TOP/bin/checkChangedScriptOps.sh --company_code=<company code>
--use_docstore=<yes/no> --src_path=<source path>
[--dst_path=<destination path>] --logfile=<log file path>
```

dst_path

宛先パスを指定します。これはオプション・パラメーターです。引数としてこのパラメーターを指定しない場合、スクリプトは **src_path** パラメーターにより指定されたソース・フォルダーのすべてのスクリプトをスキャンして上書きします。

logfile 以下の形式でログ・ファイルの詳細を指定します。

<スクリプト・ファイルの完全パス > <行番号> <推奨されないスクリプト操作の名前>

Comment

特定の推奨されないスクリプト操作に対するコメントを指定します。

dst_path 引数をシェル・スクリプトに指定すると、推奨されないスクリプト操作の出現箇所の上にコメントを追加することによってソース・スクリプトを変更し、その変更されたスクリプトを宛先ロケーションに書き込みます。すべてのスクリプト修正には、識別しやすいように `//modified_by wpc_automated_tool` というタグが付けられます。

use_docstore パラメーターに `no` が設定されているときには、**company_code** パラメーターは必要ありません。値を指定しても、その値は無視されます。

InfoSphere MDM Collaboration Server V9.1.0 では、関連付けられていない型の値を変数に保管できない型付き言語でのスクリプト記述がさらに強制されたため、実行時エラーには型の不一致によるものが含まれます。いくつかのスクリプト操作で動作に変更が生じるため、戻り値により予期しないエラーが発生する可能性があります。これらのエラーにはケース・バイ・ケースで対処する必要があります。スクリプトの変更について詳しくは、『InfoSphere MDM Collaboration Server スクリプトのコンパイルとデバッグ』を参照してください。

注: このユーティリティでは、WPC v5.3.2 後のバージョンで生じたスクリプト API の変更を処理しません。WPC V5.2.1.4 と V5.3.2 とのスクリプト操作の違いについて確認してください。

第 5 章 トラブルシューティング

一般的なマイグレーション問題を解決するには、推奨ソリューションを使用できません。

マイグレーション・スクリプトのトラブルシューティング

マイグレーション・スクリプトを IBM InfoSphere Master Data Management Collaboration Server のスクリプト・コンソール内で表示したり実行したりするには、その前にマイグレーション・スクリプトをコンパイルする必要があります。InfoSphere MDM Collaboration Server 機能を使用して、スクリプトをトラブルシューティングすることもできます。

コンパイル済みマイグレーション・スクリプトの使用

コンパイル済みスクリプトを使用するには、その前に `common.properties` ファイルに変更を加える必要があります。

コンパイル済みスクリプトを使用するには、`$TOP/etc/default` ディレクトリー内の `common.properties` ファイルに以下の設定が含まれていることを確認してください。

```
script_execution_mode = compiled_only
```

スクリプトの先頭に以下のディレクティブを含めるなら、個々のスクリプトについてスクリプトのコンパイルをオフにすることができます。

```
script_execution_mode=not_compiled
```

しかし、スクリプト・レベルでスクリプトのコンパイルをオフにするとパフォーマンスが大幅に低下するため、それは勧められていません。非コンパイル・スクリプトを使用する際のパフォーマンス低下を防ぐには、スクリプト・レベルでコンパイルをオフにするオプションを使用するのではなく、サーバーの設定を `not_compiled` に変更してください。

`common.properties` ファイルのサーバー設定が `not_compiled` に設定されている場合、スクリプト・レベルでディレクティブを使用しても、個々のスクリプトのスクリプト・コンパイルをオンにすることはできません。

注: コンパイル済みスクリプトと非コンパイル・スクリプトを組み合わせで使用することは、パフォーマンス低下を招くため、勧められていません。しかし、どうしてもそのような組み合わせを使用しなければならない場合には、1 つの制限があります。非コンパイル・スクリプトでコンパイル済みスクリプトの関数を呼び出すことは可能ですが、コンパイル済みスクリプトから非コンパイル・スクリプト中の関数を呼び出すことはできません。

一般的なスクリプト・コンパイル・エラー

コンパイル済みスクリプトを使用する場合、スクリプトをスクリプト・コンソールに保存できるのは、それが正しくコンパイルされる場合だけです。エラーがある場合は、`appsvr ログ・ディレクトリー`にある `svc.out` を見て、`javac` 出力の全体とエラー・メッセージを確認してください。

一般的なコンパイル・エラーのいくつかを以下に示します。

1. `forEach*Element()` ブロック内の `break` または `return` ステートメントは、「到達不能コード」エラーとなり、コンパイルできません。この問題を修正するには、

```
forEachItemSetElement(itemSet, item)
{
    return item;
}
```

これを次のように変更します。

```
forEachItemSetElement(itemSet, item)
{
    if (item != null)
    {
        return item;
    }
}
```

これは、機能として同等ですがコンパイラーでエラーにならずに済みます。

2. 関数から値を返す場合、あらゆる場合に値を返す必要があります。つまり、以下のコードはコンパイルを通りません。

```
function sample() {
    var e = null;
    catchError (e) {
        // do something...
        return "a string";
    } if (e != null) {
        reportError(...);
    }
}
```

`catchError` ブロックで例外が発生すると、値が返されないこととなります。次のように変更する必要があります。

```
function sample() {
    var e = null;
    catchError (e) {
        // do something...

        return "a string";
    }
    if (e != null) {
        reportError(...);
    }
    return null;
}
```

3. コンパイル時に発生する問題のほとんどについては、生成される Java ファイルを見ることができます。生成されるそれらの Java ファイルは、`common.properties` 内のパラメーター `tmp_dir` で指定されるディレクトリーにあります。最近、Java ファイルの命名規則は、スクリプト名と生成順序番号を含めるように変更されました (`MyScript12345.java` など)。

- さらに、文書ストアのスキプットの絶対パスが、生成される各 Java ファイルの先頭にコメントとして含まれるようになりました。文書ストアをファイル・システムにマッピングする場合、`grep` コマンドを再帰的に実行することにより、どの Java ファイルがスキプットに一致するかを検出することができます。

一般的な実行時のエラーと問題

スキプット実行時に発生する可能性がある既知の問題がいくつかあります。以下の例は、それら既知の問題を例示するものであり、またそれを回避または解決するためのヒントとなるものです。

無効な引数型

これは、関数に渡す引数の型が間違っている場合に発生します (例えば `String` が必要な場合に `HashMap` を渡すなど)。また、IBM InfoSphere Master Data Management Collaboration Server が型を正しく推論できない場合にも発生します。これを解決するには、`checkString()` などのスキプット操作を使用することにより、明示的に型を指定することが必要かもしれません。

比較演算で引数の型の不一致

条件演算子 (`==`、`>`、`<`、`<=` など) の両辺のデータ型が違う場合、式の評価結果は `false` になります。これはエラー・メッセージにはなりませんが、対応するコードは実行されません。

例えば、以下のコードは動作しません。

```
var id = "12345" ;
var my_id = item.getEntryAttrib(path to some attribute that is a sequence) ;
if ( id == my_id ) {
  // statements that need to be executed but won't be
}
```

これを解決するには、以下のように明示的にコーディングします。

```
var id = "12345" ;
var my_id = checkString(item.getEntryAttrib("//some attribute that is a sequence"),"");
if ( id == my_id ) {
  // statements to be executed
}
```

XML 構文解析

以下のコードは、非コンパイル・モードでも、さらにはコンパイル・モードでもスキプット・サンドボックスから実行した場合には動作します。

```
new XmlDocument(xmlDoc) ;
forEachXmlNode("item") {
  // do the needful
}
```

しかし、コンパイル・モードの場合に、このコードが複数ユーザーから呼び出されるスキプット・ライブラリー関数で使用されるのであれば、`forEachXmlNode` ブロック内のステートメントは実行されません。エラー・メッセージは表示されません。回避策として、以下を使用してください。

```
var doc = new XmlDocument(xmlDoc) ;
var xmlNode ;
forEachXmlNode(doc, "item", XmlNode) {
  //do the needful
}
```

実行時のエラーと問題の解決

`appserver` での実行時エラーを解決する方法については、`appsvr log` ディレクトリ内のファイル `svc.out` を参照してください。 `exception.log` および `default.log` を調べるのが役立つ場合がしばしばあります。

新たに生成される Java ファイルの命名規則により、どのスクリプトがエラーになったのかを容易に識別できます。また、エラー・メッセージには、生成された Java ファイルの中の行番号も示されます。問題を解決するには、生成された Java ファイルを表示して、実行時エラーの発生した行までスクロールしてください。現在では、生成された Java コードにおいて、数行ごとに実際のスクリプト・コードがコメントの形で挿入されるようになりました。

例えば、生成された Java ファイルのサンプルのうち、以下のコード部分を考慮してみましょう。

```
// function checkIfPartyPartyTypeExist(party, partyType)
public static Object ScriptFunction__checkIfPartyPartyTypeExist(HashMap hmContext, Object party, Object
partyType) throws Exception
{
// var bRet = false;
Object bRet = (java.lang.Boolean) Boolean.FALSE; //
var rootEntry = party.getRootEntryNode();
Object rootEntry = GenGetRootEntryNodeOperation.execute(hmContext , (IEntry) party);
// var entryNodes = rootEntry.getEntryNodes(getCatalogSpecName() + "/Party Types/Party Type Code");
Object entryNodes = GenGetEntryNodesOperation.execute(hmContext , (EntryNode) rootEntry, (String)
BinaryOperation.execute(BinaryOperation.PLUS, ScriptFunction__getCatalogSpecName(hmContext), "/Party
Types/Party Type Code"));
// var entryNodesSize = entryNodes.size();
Object entryNodesSize = (java.lang.Integer) GenSizeOperation.execute(hmContext , (HashMap) entryNodes);
```

上記のうち // で始まるコメント行のそれぞれは、IBM InfoSphere Master Data Management Collaboration Server スクリプトの中で対応する実際のコードです。これにより、スクリプトのどこで障害が発生したかを突き止めるのが容易になります。

checkForCompileError.sh の使用

スクリプトにコンパイル・エラーがないか、チェックする必要があります。

ターゲットの会社へのインポートを開始する前に、コンパイル中に捕らえたすべてのエラーを確実に訂正してください。

自動化されたスクリプトのコンパイルを可能にするために、**checkForCompileError.sh** が用意されています。

その使用法は次のとおりです。

```
checkForCompileError.sh --company_code=<code> --script_dir=<path/to/wpc/script/directory> --logfile=<logfile path>
```

このコマンドは、すべてのスクリプトが存在する場所である `script_dir` (docstore パス) を引数として取ります。このスクリプトは、指定されたスクリプト・ディレクトリーとそのサブディレクトリーに存在するそれぞれのスクリプトのコンパイルを試行して、スクリプト名とエラーを含むレポートを生成します。

`compiled_only` スクリプト・モードを使用して `/scripts/import/ctr/docstore` ディレクトリー内で使用可能なすべてのスクリプトをチェックしたい場合には、まず、`script_execution_mode=compiled_only` 値を持つよう `common.properties` ファイルを変更して、`$TOP` ディレクトリーから `checkForCompileError.sh` を実行します。

```
./bin/checkForCompileError.sh --company_code=test --script_dir=/scripts --logfile=logfile.txt
```

それによって `logfile.txt` ファイルが生成され、そのファイルの末尾には次の形式のレポートが見られます。

```
Fri Jul 21 15:25:06 IST 2006 - *****
Fri Jul 21 15:25:06 IST 2006 - Script Mode : 'compiled_only'
Fri Jul 21 15:25:06 IST 2006 - Total Scripts : '10'
Fri Jul 21 15:25:06 IST 2006 - Successful : '8'
```

```
Fri Jul 21 15:25:06 IST 2006 - Failed : '2'  
Fri Jul 21 15:25:06 IST 2006 - /scripts/import/ctr/abhtest - The script could not be  
compiled.  
Fri Jul 21 15:25:06 IST 2006 - /scripts/import/ctr/AbhiFailureWrongScript - The script  
object could not be built. [[Undefined operation: getEntryAttribs at line 3]]  
Fri Jul 21 15:25:06 IST 2006 - *****
```

マイグレーション問題のトラブルシューティング

IBM InfoSphere Master Data Management Collaboration Server でマイグレーション時に発生する可能性のあるトラブルシューティング対象の一般的な問題について説明します。

問題: 索引が足りない

InfoSphere MDM Collaboration Server バージョン 10.0.0 へのマイグレーション時に、マイグレーション・コンソール・レポートで、索引が足りないと報告されることがあります。

索引が足りないことに関する以下のエラーがマイグレーション時に発生することがあります。

SCA 表で索引が足りない

```
Missing Indexes  
|=====|  
| SCA_0_PK SCA_CATEGORY_IDSCA_SELECTION_IDSCA_COMPANY_IDSCA_ |  
| CATALOG_IDSCA_CAT_TREE_ID |
```

次の PERL スクリプトを実行すれば、この問題を解決できます。ディレクトリーには db2 または oracle を適宜選択してください。

```
perl $PERL5LIB/runSQL.pl --sql_file=$TOP/src/db/schema  
/dbscripts/<db2 or oracle>/sca_pk.sql
```

DBV 表で索引が足りない

```
Missing Indexes  
|=====|  
| DBV_0_UK DBV_VERSION |
```

次の PERL スクリプトを実行すれば、この問題を解決できます。ディレクトリーには db2 または oracle を適宜選択してください。

```
perl $PERL5LIB/runSQL.pl --sql_file=$TOP/src/db/schema  
/gen/<db2 or oracle>/ddl_ver_synchronize.sql  
. $TOP/bin/compat.sh  
$JAVA_RT com.ibm.ccd.synchronize.DBSchemaVersion --autoupd
```

DOA 表と CTG 表で索引が足りない (DB2 のみ)

```
Missing Indexes  
|=====|  
| CTG_1_UK CTG_COMPANY_IDCTG_NAME |  
| DOA_0_UK DOA_DOC_IDDOA_CMP_IDDOA_NAME |
```

次の PERL スクリプトを実行すれば、この問題を解決できます。

```
perl $PERL5LIB/runSQL.pl  
--sql_command="alter table tctg_ctg_catalog drop constraint ctg_1_uk ;"  
perl $PERL5LIB/runSQL.pl  
--sql_command="drop index ictg_ctg_2;"
```

```
perl $PERL5LIB/runSQL.pl  
--sql_command="alter table tctg_ctg_catalog  
add constraint ctg_1_uk unique (ctg_company_id, ctg_name);"
```

```

perl $PERL5LIB/runSQL.pl
--sql_command="create index ictg_ctg_2 on
tctg_ctg_catalog ( ctg_name, ctg_company_id)
ALLOW REVERSE SCANS;"

perl $PERL5LIB/runSQL.pl
--sql_command="alter table TDOC_DOA_DOC_ATTRIBUTES
drop constraint doa_0_uk ;"

perl $PERL5LIB/runSQL.pl
--sql_command="drop index idoc_doa_0;"

perl $PERL5LIB/runSQL.pl
--sql_command="alter table tdoc_doa_doc_attributes
add constraint doa_0_uk unique (doa_doc_id, doa_cmp_id, doa_name);"

perl $PERL5LIB/runSQL.pl
--sql_command="create index idoc_doa_0
on tdoc_doa_doc_attributes ( doa_name, doa_doc_id, doa_cmp_id)
ALLOW REVERSE SCANS;"

```

問題: 表 ITX および CAX がない

InfoSphere MDM Collaboration Server バージョン 10.0.0 へのマイグレーション時に DB 検査レポートで次のエラーが報告されることがあります。

```

Missing Tables
=====
TCTG_CAX_CATEGORY_CONTENT
TCTG_ITX_ITEM_CONTENT

Missing Indexes|
CAX_0_PK CAX_ENTRY_IDCAX_NEXT_VERSION_ID
ITX_0_PK ITX_ENTRY_IDITX_NEXT_VERSION_ID

```

このエラーは、表スペースの作成、XDB のインストール、または DB パラメーターに関する問題を示しています。 \$TOP/logs ディレクトリーにある errfile.log ファイルで正確なメッセージを確認してください。データベースが正しくセットアップされていることを確認して、マイグレーション・スクリプトを再実行してください。

問題: 特権不足エラー

InfoSphere MDM Collaboration Server バージョン 10.0.0 にマイグレーションするためのマイグレーション・スクリプトを実行するときに、コンソールに特権不足エラーが表示されることがあります。代表的なエラーの例を以下に示します。

Oracle データベース

Oracle データベースを使用している場合のエラーは次のエラーのように表示されます。

```

create index icnt_eem_2 on tcnt_eem_entry_entry_map (
*
ERROR at line 1:
ORA-01031: insufficient privileges

```

DB2 データベース

DB2 データベースを使用している場合のエラーは次のエラーのように表示されます。

```
SQL0551N "USERNAME" does not have the privilege to perform operation
"CREATE INDEX" on object "USERNAME.TWFL_WFE_WORKFLOW_EVENT".
SQLSTATE=42501
```

データベース・ユーザーに、データベース内で索引を作成するのに十分な特権がありません。データベース・ユーザーに「索引の作成」特権を付与してから、マイグレーション・スクリプトを再実行しなければなりません。また、表の作成や変更を実行できるアクセス権をユーザーに付与しなければなりません。

問題: マイグレーション・スクリプト失敗

InfoSphere MDM Collaboration Server バージョン 10.0.0 にマイグレーションするためのマイグレーション・スクリプトを実行すると、そのスクリプトが失敗することがあります。スクリプト失敗の代表的なシナリオを以下に示します。

モジュール `data_maintenance_reports` での失敗

```
-----
Summary of the migration
-----
```

```
Migration of the following modules failed:
data_maintenance_reports
```

\$TOP/logs/errfile.log ファイルには以下の内容が含まれています。

```
net.sf.ehcache.distribution.
MulticastKeepaliveHeartbeatReceiver$MulticastReceiverThreadrun
SEVERE: Multicast receiver thread caught throwable. Cause was null. Continuing...
```

これは、キャッシュ構成パラメーターに関する問題が発生したことが原因です。ただし、マイグレーション・スクリプトは続行されて正常に完了しました。キャッシュ構成パラメーターを調べてください。マイグレーション・スクリプトを再実行する必要はありません。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
J46A/G4
San Jose, CA 95141-1003
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのもと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願ひします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。当該サンプル・プログラムは、いかなる保証も提供されることなく、特定物として

現存するままの状態を提供されます。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. _年を入れる_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corp. の商標です。

他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Intel、Intel Centrino、Celeron、Intel Xeon、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft および、Windows は、Microsoft Corporation の米国およびその他の国における商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

インポート 36, 42
インポート・スクリプト
 gds 43
エクスポート 26, 35
エクスポート・スクリプト
 GDS 32

[カ行]

階層内容
 GDS 42
確認 78
カタログの内容
 GDS 42
環境
 GDS 26, 36
関連情報 v
組み込み 23
クリーンアップ 20
検索可能属性
 gds 23
検証 44
更新 24
コメント
 送信 v
コメントの送信 v
コンパイル 44

[サ行]

実装スクリプト
 GDS 23
準備 20
商標 91
除去 23
スクリプト
 一般的なコンパイル・エラー 82
 一般的な実行時エラー 83
 コンパイルとデバッグ 81
 実行時エラーの解決 84
 違い 52, 53
 5.3.0 52
 V10.1.0 のメソッド 55, 60

スクリプト (続き)
 V5.2.1 および V10.1.0 からの動作に関する変更 65
 V5.2.1 と V10.1.0 の違い 50
 V5.2.1 のメソッド 54
 V5.3.0.1 および V10.1.0 からの動作に関する変更 68
 V5.3.0.1 のメソッド 55
 V6.0.0 および V10.1.0 からの動作に関する変更 70
スクリプトの違い 49
制限 32, 43
前提条件の情報 v

[タ行]

チューニング 74
追加 23
データのインポート 84
データベース
 GDS 26, 74
データ・モデル
 GDS 74
停止 25
デバッグ 44
 gds 44
トラブルシューティング
 マイグレーション問題 85
 IBM ソフトウェア・サポートへのお問い合わせ ix
トランスフォーマー
 GDS 75

[ハ行]

バックアップ 26
フィーチャー
 バージョン 10.1 での新機能 v
文書ストア 35
変換 74

[マ行]

マイグレーション 19
 概要 1
 旧バージョンからのマイグレーション 5
 コンパイル・エラーのチェック 84
 スクリプト 49, 81
 入門 3

マイグレーション (続き)
 マイグレーション・スクリプトのトラブルシューティング 81
 問題 81
 GDS 1.3.x 20
 v10.0.0 から v10.1.0 へのマイグレーション 16
 v6.0.0 から v10.0.0 へのマイグレーション 9
 v9.1.0 から v10.1.0 へのマイグレーション 13
マイグレーションした環境
 GDS 44
マイグレーション・スクリプト
 GDS 78
メソッド
 V6.5.0 / V10.1.0 でのみ利用可能 65
 V9.0.0 65
メタデータ 74

[数字]

10.0 71
10.1.0 52, 53
5.3.0.1 52
6.0.0 53
9.1.0 71, 73

G

GDS 19, 25
 データ 20
GDS V1.3.x 19
gds.properties
 GDS 24

J

Java API
 インターフェース 73
 メソッド 71
 列挙型 71
 10.0 73
 6.0.0 71
Java API の違い 71

M

MigrateExport.sh スクリプト 26

X

XSLT 75



Printed in Japan

SA88-4603-01



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21