

バージョン 6.1.0



WebSphere Process Server のマイグレーション

バージョン 6.1.0



WebSphere Process Server のマイグレーション

お願い

本書に記載されている情報をご使用になる前に、本書末尾の「特記事項」セクションに記載されている情報をお読みください。

本書は、WebSphere Process Server for Multiplatforms バージョン 6、リリース 1、モディフィケーション 0 (製品番号 5724-L01)、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。

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。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。

(URL は、変更になる場合があります)

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： WebSphere® Process Server for Multiplatforms
Version 6.1.0
Migrating WebSphere Process Server

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

目次

第 1 章 WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからのマイグレーション 1

マイグレーションの概要	1
事前マイグレーションの考慮事項	3
バージョン間マイグレーション用ツール	8
旧バージョンからマイグレーションするときのデータ処理方法	24
製品構成のマイグレーション時の構成マッピング	25
WebSphere アプリケーションのマイグレーション	31
マイグレーションのためのデータベースのアップグレード	32
手動による共通データベースのアップグレード	33
手動による Business Process Choreographer データベースのアップグレード	36
スタンドアロン・サーバーのマイグレーション	40
マイグレーション・ウィザードを使用したスタンドアロン・サーバーのマイグレーション	41
コマンド行ツールを使用したスタンドアロン・サーバーのマイグレーション	45
Network Deployment 環境のマイグレーション	49
デプロイメント・マネージャーのマイグレーション	50
非クラスター管理対象ノードのマイグレーション	60
クラスターのマイグレーション	75
リモート・システムへのマイグレーション	99
サポート対象外となったオペレーティング・システムからのマイグレーション	105
マイグレーションの検査	109
事後マイグレーション構成検査	111
環境のロールバック	111
デプロイメント・セルのロールバック	112
管理対象ノードのロールバック	115
Cloudscape データベースのマイグレーション	118
Cloudscape v10.1.x 自動マイグレーションの検査	120
手動による Cloudscape のアップグレード	125
UDDI レジストリーのマイグレーション	129
Business Process Choreographer に関するマイグレーションの考慮事項	132
バージョン間のマイグレーションのトラブルシューティング	134

第 2 章 以前の WebSphere 製品からのマイグレーション 145

WebSphere InterChange Server からのマイグレーション	146
--	-----

事前マイグレーションの考慮事項	146
reposMigrate コマンドを使用した WebSphere InterChange Server の成果物のマイグレーション	154
事後マイグレーションの考慮事項	158
WebSphere Business Integration データ・ハンドラーのサポート	175
サポートされる WebSphere InterChange Server API	176
WebSphereInterChange Server からマイグレーションする場合の制限事項	199
WebSphere InterChange Server からのマイグレーションのトラブルシューティング	200
WebSphere Studio Application Developer Integration Edition からのソース成果物のマイグレーション	203
WebSphere MQ Workflow からのマイグレーション	204

第 3 章 使用すべきでないフィーチャー 205 廃止リスト 205

WebSphere Process Server バージョン 6.1 で使用すべきでないフィーチャー	206
WebSphere Process Server バージョン 6.0.2 で使用すべきでないフィーチャー	210
WebSphere Process Server バージョン 6.0.1 で使用すべきでないフィーチャー	212
WebSphere Process Server バージョン 6.0 で使用すべきでないフィーチャー	212
WebSphere Business Integration Server Foundation バージョン 5.1.1 で使用すべきでないフィーチャー	216
WebSphere Business Integration Server Foundation バージョン 5.1 で使用すべきでないフィーチャー	216

第 4 章 マイグレーションのトラブルシューティング 219

バージョン間のマイグレーションのトラブルシューティング	219
WebSphere InterChange Server からのマイグレーションのトラブルシューティング	229
サポートされる WebSphere InterChange Server API のロギングおよびトレースの使用可能化	229
マイグレーションされた BPEL ファイルで直列化可能ではないオブジェクトを直列化しようとしたときの失敗	230

特記事項 233

第 1 章 WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからのマイグレーション

ご使用のインストール済みアプリケーションおよび構成は、WebSphere Process Server および WebSphere® Enterprise Service Bus の以前のバージョンから、WebSphere Process Server のバージョン 6.1 にマイグレーションすることができます。

マイグレーションの概要

WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからマイグレーションします。

WebSphere Process Server の 1 つのバージョンから WebSphere Process Server のより新しいリリースへ移行すること、場合によっては、あるバージョンの WebSphere Enterprise Service Bus から WebSphere Process Server のより高いリリース・レベルに移行することを、バージョン間マイグレーションと呼びます。バージョン間マイグレーションは、WebSphere Process Server などの製品の新しいバージョンをインストールし、古いインストールから新規インストールへ関連アプリケーションや構成データをコピーする場合に行われます。マイグレーションによって、新バージョンの製品が旧製品に加えてインストールされます。次に、旧バージョンの製品から新バージョンの製品にデータがコピーされます。マイグレーションは更新とは異なります。更新では、既存のインストール環境にある古いファイルまたはデータが現行の情報で置換されます。リフレッシュ・パック、暫定修正、フィックスパックなどが更新の例として挙げられます。更新の方法について詳しくは、「Update Installer を使用したフィックスパックおよびリフレッシュ・パックのインストール」を参照してください。

マイグレーションは、WebSphere Process Server の旧バージョンから、同じオペレーティング・システム上で稼働する新バージョンへのマイグレーションでなければなりません。異なるオペレーティング・システム間ではマイグレーションできません。例えば、ご使用の WebSphere Process Server バージョン 6.0.x が Microsoft Windows XP で稼働している場合、WebSphere Process Server のそのインスタンスから Windows XP で稼働している WebSphere Process Server バージョン 6.1 にデータをマイグレーションすることはできます。しかし、Windows XP で稼働する WebSphere Process Server バージョン 6.0.x から AIX で稼働する WebSphere Process Server バージョン 6.1 にデータをマイグレーションすることはできません。また、任意のオペレーティング・システム上の旧リリースから同じオペレーティング・システム上の新規にサポートされるリリースにマイグレーションできるのは、スタンドアロン・サーバーの場合のみです。例えば、スタンドアロン・サーバーでは、AIX 5.2 で稼働する WebSphere Process Server バージョン 6.0.x から、AIX 5.3 で稼働する WebSphere Process Server バージョン 6.1 にマイグレーション

することができます。(このマイグレーションについては、105 ページの『サポート対象外となったオペレーティング・システムからのマイグレーション』を参照してください。)

次の表に、このリリースの WebSphere Process Server でサポートされるバージョン間マイグレーションのシナリオを示します。「現在のインストール・バージョン」の下にリストされているすべての製品を WebSphere Process Server のバージョン 6.1 にマイグレーションすることができます。

現在のインストール・バージョン	新バージョン
WebSphere Process Server 6.0.1.x	WebSphere Process Server 6.1
WebSphere Process Server 6.0.2.x	WebSphere Process Server 6.1
WebSphere Enterprise Service Bus 6.0.1.x	WebSphere Process Server 6.1
WebSphere Enterprise Service Bus 6.0.2.x	WebSphere Process Server 6.1

バージョン間マイグレーションを行う理由

WebSphere Process Server では、以前のバージョンとのユーザー・アプリケーション・バイナリー互換性が維持されます。しかし、バージョン間マイグレーションを実行すれば、WebSphere Process Server の新バージョンに移行するときに、アプリケーションに加えて WebSphere Process Server 構成データも保持することができます。バージョン間マイグレーションを実行することによって、プロファイル、セル、クラスター、サーバー、およびノードの構成が保存されます。このマイグレーションを実行せずに新バージョンの WebSphere Process Server をインストールするだけの場合は、使用環境を最初から再構成する必要が生じます。

WebSphere Process Server の一部のリリースでは、「インプレース更新」すなわち保守パッケージの使用が可能です。この更新処理でも、構成データは保持されます。WebSphere Process Server のバージョン 6.0.x からバージョン 6.1 へ移行する場合のように、保守パッケージが使用できない状況では、構成データを保持するためにバージョン間マイグレーションが必要になります。

関連概念

開発およびデプロイメントのバージョン・レベル

ご使用の環境に必要な WebSphere Process Server のバージョン・レベルの決定は、アプリケーションが開発されたときのバージョン・レベルに依存します。一般に、前のバージョンの WebSphere Process Server にデプロイされたアプリケーションは、次に入手可能なバージョンの WebSphere Process Server 上で稼働します。

関連タスク

「Update Installer を使用したフィックスパックおよびリフレッシュ・パックのインストール」

IBM® Update Installer for WebSphere Software を使用して暫定修正、フィックスパック、およびリフレッシュ・パック (メンテナンス・パッケージと総称される) をインストールできます。Update Installer for WebSphere Software はまた、ア

アップデート・インストーラー・プログラム、UpdateInstaller プログラム、および
アップデート・インストール・ウィザードと呼ばれています。

事前マイグレーションの考慮事項

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

WebSphere Process Server のバージョン 6.1 がインストールされている場合は、マイグレーションおよび共存に関して以下の規則と制限事項が適用されます。

- バージョン 6.0.x のデプロイメント・マネージャーをバージョン 6.1.x のデプロイメント・マネージャーにマイグレーションできるのは、それらのデプロイメント・マネージャーが同じ拡張レベルにある場合に限られます。
- バージョン 6.0.x のスタンドアロン・サーバーからマイグレーションされたバージョン 6.1.x のスタンドアロン・サーバーは拡張できません。

バージョン 6.1.x で新規のスタンドアロン・プロファイルを作成し、それを拡張することができます。

- 6.0.x の管理対象サーバーからマイグレーションされたバージョン 6.1.x の管理対象サーバーは拡張できません。

バージョン 6.1.x で新規プロファイルを作成して拡張してから、拡張済みデプロイメント・マネージャーが組み込まれているバージョン 6.1.x セルに新規ノードを追加できます。

- セルのデプロイメント・マネージャーが、その管理対象ノードの中の最高拡張レベルと同じ拡張レベルまで拡張されている限り、拡張および拡張解除両方の管理対象ノードを含む混合セルを作成できます。例えば、デプロイメント・マネージャーが WebSphere Process Server 用に拡張されているならば、そのデプロイメント・マネージャーによって、WebSphere Process Server および WebSphere Application Server 用に拡張されているノードを正常に管理できます。ただし、WebSphere Application Server のみを対象に拡張されているデプロイメント・マネージャーが管理できるのは、WebSphere Application Server ノードに限られます。
- Business Process Choreographer をインストール済みの場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。
- WebSphere Process Server バージョン 6.1 をインストールした後は、完全なデプロイメント・セル構成を構築し、既存のセルまたはノードをマイグレーションする前に正常に動作することを確認します。

これにより、ご使用のシステムが必要な前提条件をすべて満たし、WebSphere Process Server の新規レベルをサポートすることが確認できます。

- バージョン 6.0 以降の WebSphere Process Server には、HA マネージャーとコア・グループ機能が組み込まれています。6.0.x からバージョン 6.1.x へのマイグレーションに影響する可能性があるコア・グループ構成およびトポロジーの考慮事項については、『コア・グループのマイグレーションに関する考慮事項』を参照してください。
- JDK 1.4 から JDK 5 (WebSphere Application Server バージョン 6.1 で導入されており、従って WebSphere Process Server バージョン 6.1 でも同様) にマイグレ

ーションする前に、Sun Microsystems の Java™ 仕様に基づいて、アプリケーションに必要な変更があるかどうかを検討してください。

『API および仕様マイグレーション』を参照してください。

- 複数のノードが存在するセルをマイグレーションする場合、アプリケーションは、すべてのノードがマイグレーションされるまで最低の JDK レベルに留まる必要があります。
- **Solaris** Solaris x64 で WebSphere Process Server バージョン 6.0.2 と連動する Java Native Interface (JNI) アプリケーションは、WebSphere Process Server バージョン 6.1 と連動させるために 64 ビット環境で再コンパイルする必要があります。これには、例えば Enterprise JavaBean (EJB) から呼び出される WebSphere Process Server プロセス・コードで実行されるすべての JNI アプリケーションが含まれます。

Solaris x64 上での WebSphere Process Server バージョン 6.0.2 は、基盤となるプラットフォームが 64 ビットであっても 32 ビット・アプリケーションとして実行されます。これは、基盤となる Java 仮想マシンが 32 ビットだからです。

WebSphere Process Server バージョン 6.1 は、基盤となる Java 仮想マシンが 64 ビットであるため、64 ビット・アプリケーションとして実行されます。バージョン 6.0.2 の 32 ビット環境でコンパイルされた JNI アプリケーションは、バージョン 6.1 の 64 ビット環境では実行できません。

- WebSphere Process Server のバージョン 6.1 は、前のレベルの WebSphere Process Server と共存する環境にインストールすることができます。しかし、いくつかの制約事項が存在します。

制約事項など、共存について詳しくは、他の WebSphere 製品のインストール済み環境との『共存』を参照してください。

- クラスター内で、バージョン 6.0.x メンバーとバージョン 6.1 メンバーを同時に稼働させることはできません。最初にバージョン 6.1 クラスター・メンバーを開始する前に、すべてのバージョン 6.0.x クラスター・メンバーを停止する必要があります。また、いったんバージョン 6.1 クラスター・メンバーを開始したら、そのクラスターでバージョン 6.0.x クラスター・メンバーを開始しないでください。
- マイグレーションにおいて、バージョン 6.1 のクラスター情報はセル全体に配布されます。WebSphere Process Server バージョン 6.0.1.3 以降のレベルではないバージョン 6.0.x ノードは、この情報の読み込みに失敗し、クラスター機能に障害が発生する可能性があります。バージョン 6.0.1.3 未満のノードで選択可能なオプションは 2 つです。1 つのオプションとしては、ご使用のデプロイメント・マネージャーをバージョン 6.1 にマイグレーションする前に、バージョン 6.1 セルに今後含まれる (またはそのセルと相互作用する) すべての 6.0.x ノードを、WebSphere Process Server バージョン 6.0.1.3 以降にアップグレードします。2 つ目のオプションとしては、WebSphere Process Server のバージョンを 6.0.1.3 以降に移行したくない場合に、基盤となる WebSphere Application Server をバージョン 6.0.2.11 以降に更新します。
- WebSphere Process Server バージョン 6.1 のマイグレーションでは、HTTP トランスポートがチャンネル・フレームワークによる Web コンテナー・トランスポート・チェーンに変換されます。

バージョン 6.1 のトランスポート・サポートについて詳しくは、以下のトピックを参照してください。

- トランスポート・チェーンの構成
- HTTP トランスポート・チャンネルの設定
- トランスポート・チェーン
- デプロイメント・マネージャーをマイグレーションする場合は、WebSphere Process Server バージョン 6.1 のセル名がバージョン 6.0.x のセル名と一致する必要があります。

新規のセル名でプロファイルを作成し、マイグレーション・ターゲットとしてこのプロファイルを使用すると、マイグレーションは失敗します。

- 統合ノードをマイグレーションする場合は、WebSphere Process Server バージョン 6.1 のセル名とノード名が、バージョン 6.0.x でそれぞれに対応する名前と一致する必要があります。
- マイグレーション要件 (命名要件など) を満たさないプロファイルを作成する場合は、WebSphere Process Server バージョン 6.1 製品をアンインストールして再インストールするのではなく、古いプロファイルを削除して新規プロファイルを作成することができます。
- ノードを WebSphere Process Server バージョン 6.1 にマイグレーションした後で、バージョン 6.0.x に戻る必要が生じた場合は、111 ページの『環境のロールバック』を参照してください。
- マイグレーション・ツールは、前のバージョンからの構成コピーのバックアップを格納したマイグレーション・バックアップ・ディレクトリーを作成します。バージョン 6.0.x からマイグレーションする場合は、このディレクトリーで使用可能なスペースのサイズを、少なくとも前のプロファイルの構成ディレクトリーおよびアプリケーションのサイズにしてください。
- バージョン 6.1 へのマイグレーション時にシステムが必要とするストレージの合計容量は、ユーザー環境および使用するマイグレーション・ツールに応じて異なります。

- **WBIPreUpgrade** ストレージの要件

- **ロケーション:** WBIPreUpgrade コマンドのパラメーターとして指定されるバックアップ・ディレクトリー
- **容量:** このコマンドを使用する場合のストレージ要件のおおまかな見積もりとして、次の容量を追加します。
 - 古い構成内のプロファイルすべてについて、以下の項目のサイズ:
 - *profile_root/installableApps* ディレクトリー
 - *profile_root/installedApps* ディレクトリー
 - *profile_root/config* ディレクトリー
 - *profile_root/properties* ディレクトリー
 - *libraries.xml* 構成ファイルで参照される共用ライブラリー
 - *resources.xml* 構成ファイルで参照されるリソース・アダプター・アーカイブ (RAR) ファイル
 - デフォルトどおりトレースを使用可能にする場合は、最大 200 MB (構成のサイズと複雑さによる)

このコマンドについて詳しくは、14 ページの『WBIPreUpgrade コマンド』を参照してください。

- **WBIPostUpgrade ストレージの要件**

- **ロケーション:** 新しい *profile_root* ディレクトリーに対する新規構成の相対位置
- **容量:** このコマンドを使用する場合のストレージ要件のおおまかな見積もりとして、次の容量を追加します。
 - マイグレーションする古いプロファイルについて以下の項目のサイズ:
 - *profile_root/installableApps* ディレクトリー
 - *profile_root/installedApps* ディレクトリー
 - *profile_root/config* ディレクトリー
 - *profile_root/properties* ディレクトリー
 - *libraries.xml* 構成ファイルで参照される共用ライブラリー
 - *resources.xml* 構成ファイルで参照されるリソース・アダプター・アーカイブ (RAR) ファイル
 - デフォルトどおりトレースを使用可能にする場合は、最大 200 MB (構成のサイズと複雑さによる)

このコマンドについて詳しくは、17 ページの『WBIPostUpgrade コマンド』を参照してください。

- Cloudscape™ データベースをマイグレーションする前に、Cloudscape データベースを使用するアプリケーションをホスティングしているサーバーがシャットダウンされているか確認します。シャットダウンされていないサーバーがあると、Cloudscape のマイグレーションに失敗します。
- マイグレーション・ツールを使用して WebSphere Process Server バージョン 6.1 にマイグレーションした後は、マイグレーション・ツールで自動的に実行されない操作を実行しなければならない場合があります。
 - バージョン 6.0.x で使用していた Lightweight Third Party Authentication (LTPA) セキュリティーの設定がある場合には、それらについて調べ、バージョン 6.1 のセキュリティが適切に設定されていることを確認してください。

『LTPA (Lightweight Third Party Authentication)』を参照してください。

- *logs* ディレクトリーの WBIPostUpgrade.log ファイルを調べ、マイグレーション・ツールによってマイグレーションされなかった JSP オブジェクトの詳細を確認します。

JSP オブジェクトが構成されているレベルをバージョン 6.1 がサポートしない場合は、マイグレーション・ツールが出力でオブジェクトを認識して、ログに記録します。

- ご使用の Java 仮想マシンの設定を確認し、『Java 仮想マシン設定』で説明されているデフォルトの推奨値を使用していることを確認してください。
- Cloudscape データベースの自動マイグレーションの結果を検証し、ツールによって自動的にマイグレーションされていない Cloudscape データベースがあれば手動でマイグレーションします。

118 ページの『Cloudscape データベースのマイグレーション』を参照してください。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要のある追加タスクに注意してください。

関連タスク

111 ページの『環境のロールバック』

WebSphere Process Server バージョン 6.1 環境へのマイグレーション後に、バージョン 6.0.x 環境にロールバックできます。これによって、構成はマイグレーション前の状態に戻ります。環境のロールバック後に、マイグレーション・プロセスを再開できます。

118 ページの『Cloudscape データベースのマイグレーション』

マイグレーション・ツールを使用して WebSphere Process Server バージョン 6.1 にマイグレーションした後、Cloudscape データベースの自動マイグレーションの結果を検査して、ツールによって自動的にマイグレーションされていない Cloudscape データベース・インスタンスがあれば、手動でマイグレーションします。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリに追加します。

関連情報



他の WebSphere 製品のインストール済み環境との『共存』

WebSphere Process Server バージョン 6.1 のインストール済み環境は、WebSphere Process Server または WebSphere Enterprise Service Bus のすべてのバージョンのインストール済み環境、および特定の WebSphere 製品の一部のバージョンと同じシステム上で共存することができます。

トランスポート・チェーンの構成

HTTP トランスポート・チャンネルの設定

トランスポート・チェーン

API および仕様マイグレーション

クラスターの作成

アプリケーション・サーバーの作成
Lightweight Third Party Authentication
コア・グループのマイグレーションに関する考慮事項
Java 仮想マシン設定

バージョン間マイグレーション用ツール

WebSphere Process Server または WebSphere Enterprise Service Bus の以前のバージョンからのマイグレーションは、マイグレーション「ウィザード」(グラフィカル・ユーザー・インターフェース) または一連のコマンドまたはスクリプトを使用して実行できます。

マイグレーションするために、マイグレーション・ウィザードまたは一連のコマンドを使用できます。

マイグレーション・ウィザード

マイグレーション・ウィザードは、マイグレーション・プロセスの手順を段階的に案内します。特定のフィールドに値を入力したり、ウィザードによって示される選択項目を選択したり、デフォルト値を使用したりするよう求められます。マイグレーション・ウィザードは、WebSphere Process Server の「ファースト・ステップ」パネルから起動できます。また、Windows® システムの場合は `install_root\bin\wbi_migration.bat` から、UNIX ベース・システムの場合は `install_root/bin/wbi_migration.sh` から直接起動することもできます。

マイグレーション・コマンド

マイグレーション・ウィザードを使用しない方法を選択する場合は、コマンド行から呼び出される一連のスクリプトを使用して、以前のバージョンの WebSphere Process Server からマイグレーションすることができます。次のコマンドは正しい順序で実行する必要があります。

1. `WBIPreUpgrade` - このコマンドを最初に実行します。既存の WebSphere Process Server 構成およびアプリケーションがマイグレーション固有のバックアップ・ディレクトリーに保存されます。
2. `WBIPostUpgrade` - このコマンドを 2 番目に実行します。`WBIPreUpgrade` コマンドによって作成されたマイグレーション固有バックアップ・ディレクトリーの内容を処理し、それを新規の WebSphere Process Server 環境にインポートします。
3. `WBIProfileUpgrade.ant` - このスクリプトは、クラスター化セルをマイグレーションする場合にのみ、`WBIPreUpgrade` コマンドと `WBIPostUpgrade` コマンドを実行した後に手動で実行する必要があります。また、マイグレーションが正常に行われなかったなどの特別な場合にも、このスクリプトを実行する必要があります。このスクリプトにより、プロファイルのエンタープライズ・アプリケーションと構成の設定が更新されます。`WBIPostUpgrade` コマンドはこのスクリプトを呼び出すので、非クラスター化シナリオにおいては、二度目は手動で再実行する必要はありません。

データベース・アップグレード・スクリプト

通常、WebSphere Process Server コンポーネントをサポートするデータベースのいずれかで更新が必要な場合には、マイグレーションされたサーバー・プロセスが開始するときに更新が自動的に完了します。ただし、サーバー・プロセスに十分な許可がない場合（つまり、該当するデータベースに対して十分な許可を持った正しいユーザー ID を使用して構成されていない場合）は、組み込まれているスクリプトを使用してデータベースを手動で更新する必要があります。

関連タスク

32 ページの『マイグレーションのためのデータベースのアップグレード』
マイグレーションと同時に、一部の WebSphere Process Server コンポーネントのデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

マイグレーション・ウィザード

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

バージョン間マイグレーション・ウィザードは、基本的な WebSphere Process Server マイグレーション・ツールである `WBIPreUpgrade` コマンドおよび `WBIPostUpgrade` コマンドとのグラフィカル・インターフェースです。

注: マイグレーション・ウィザードを非グラフィカル環境で実行することはできません。非グラフィカル環境の例としては `Telnet` セッションがあります。非グラフィカル環境でマイグレーションを実行する場合は、`WBIPreUpgrade` コマンドおよび `WBIPostUpgrade` コマンドを使用します。

マイグレーション・ウィザードの機能

マイグレーション・ウィザードは、`WBIPreUpgrade` コマンドおよび `WBIPostUpgrade` コマンドを使用して、データおよびアプリケーションを WebSphere Process Server の旧バージョンから新バージョンにマイグレーションします。

このステップでは、旧バージョンのサーバー・リソース、セキュリティー、変数、および仮想ホストに関するアプリケーションおよび構成情報が、新規バージョンのサーバーに転送されます。保存されている情報はすべて、各製品の `profile_dir/config/cells` ディレクトリー内の XML ファイルにあります。

`WBIPreUpgrade` ツールは、選択されたファイルを `install_root` および `profile_root` ディレクトリーから、ウィザード・パネルで指定したバックアップ・ディレクトリーに保存します。マイグレーションを行うと、ファイルはバックアップ・ディレクトリーの以下のサブディレクトリーに保存されます。

- `websphere_backup`
 - `bin`
 - `cloudscape`
 - `lib`
 - `ProcessChoreographer/Staff`

- properties
- *profile_root*
 - bin
 - config
 - etc
 - event
 - installedApps
 - installedConnectors
 - properties

後でマイグレーション・ウィザードは、WBIPostUpgrade ツールを使用して、バックアップ・ディレクトリー内の環境を WebSphere Process Server の新しいインストールに復元します。

マイグレーション・ウィザードへのアクセス

マイグレーション・ウィザードは、以下のいずれかの方法で起動します。

- WebSphere Process Server のファースト・ステップ・コンソールから「マイグレーション・ウィザード」を選択する。
- *install_dir/bin* ディレクトリーに保存されている以下のスクリプトの 1 つ (これはオペレーティング・システムによって異なります) を実行する。
 1. UNIX Linux UNIX® および Linux® システムの場合:
wbi_migration.sh
 2. Windows Windows システムの場合: wbi_migration.bat

始める前に

マイグレーション・ウィザードでは、プロンプトが表示されて情報を入力するよう要求されます。マイグレーション・ウィザードを起動する前に、以下の情報を収集しておいてください。

セル名 (デプロイメント・マネージャーのマイグレーションの場合のみ必須) マイグレーションしようとしているデプロイメント・マネージャーによって管理されているセルの名前。新バージョンのセル名は、旧バージョンの構成におけるセル名に一致している必要があります。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

マイグレーション・ウィザードの実行: プロシージャ

1. 「ようこそ」パネルを読んでマイグレーション・プロセスについて理解してから、「次へ」をクリックします。
2. マイグレーション元となる前のバージョンの WebSphere Process Server を選択または指定して、「次へ」をクリックします。

チェック・ボックスを選択して、以前のインストールの場所が選択リストに表示されない場合、この場所を入力します。

3. マイグレーションする以前のバージョンのプロファイルを選択して、「次へ」をクリックします。
4. バージョン 6.1 のインストールのための有効なプロファイルのリストからプロファイルを選択するか、「新規プロファイルを作成する」を選択して、「次へ」をクリックします。

古いプロファイルをマイグレーションする前に新しいプロファイルの構成のバックアップ・コピーを作成するよう、チェック・ボックスを選択します。チェック・ボックスを選択すると、ターゲット・プロファイルのバックアップ・コピーは、*profile_root/temp/MigrationBackup.time_stamp.zip* に書き込まれます。

5. 前述のパネルで「新規プロファイルを作成する」を選択した場合、プロファイルの名前とホスト名 (プロファイルがあるシステムの名前) を入力して、「次へ」をクリックします。可能であれば、古いプロファイルに使用されていたものと同じホスト名を使用してください。同じ名前を使用しない場合は、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。
6. 以前のバージョンからの構成のバックアップ・コピーを配置するマイグレーション・バックアップ・ディレクトリーを指定して、「次へ」をクリックします。

ディレクトリーが存在しない場合、作成されます。ディレクトリーが存在する場合、バックアップ操作によって既存のバックアップ・ファイルが上書きされるため、ディレクトリーは空である必要があります。

7. マイグレーションされるアプリケーションを配置する場所を指定して、「次へ」をクリックします。

以下のオプションのいずれか 1 つを選択できます。

- アプリケーションが現在配置されているディレクトリーと同じ場所にアプリケーションを保持します。

制約事項: このオプションを選択した場合、既存のインストールと新しいインストールによってこの場所は共有されます。マイグレーションされるアプリケーションを前のバージョンのときと同じロケーションに保持する場合は、次の制限が適用されます。

- ノードが混在した場合のサポートの制約に従う必要があります。つまり、wsadmin コマンドを呼び出したとき、以下のサポートが使用できません。
 - JSP のプリコンパイル
 - バイナリー構成の使用
 - EJB のデプロイ
 - 後でバージョン 6.0.x のインストール済み環境の管理 (アンインストールなど) を行うために、これらの場所からアプリケーションを削除する場合、マイグレーション済みアプリケーションを誤って失う危険性があります。
 - ターゲット・バージョンのデフォルト・ディレクトリーにアプリケーションをインストールするよう選択します。
 - マイグレーション済みアプリケーションをインストールするディレクトリーを指定します。
8. ポート値を割り当てるオプションを 1 つ選択して、「次へ」をクリックします。

ポート値については、以下のいずれか 1 つを実行するよう選択できます。

- 以前のインストール済み環境に割り当てたポート値を使用する。
- ターゲットのプロファイルに割り当てたポート値を使用する。
- ポート値をブロックとして定義する。

このオプションを選択した場合、割り当てる連続するポート番号のブロックの最初の値を指定します。

9. スクリプト互換性をサポートするようにマイグレーションする場合、チェック・ボックスを選択してから「次へ」をクリックします。

このオプションを選択した場合、マイグレーション・ウィザードでは、WebSphere Process Server のバージョン 6.0.x についての以下の構成定義が作成されます。

- トランスポート
- ProcessDef
- バージョン 6.0.x SSL

- バージョン 6.0.x ORB サービス・スレッド・プール

これらは以下の バージョン 6.1 構成定義の代替となります。

- チャンネル
- ProcessDefs
- バージョン 6.1 SSL
- バージョン 6.1 ORB サービス・スレッド・プール

既存の管理スクリプトへの影響を最小限にするには、このオプションを選択します。例えば、既存の構成定義を作成または変更するためにサード・パーティーの構成 API を使用する既存の **wsadmin** スクリプトまたはプログラムを持つ場合は、マイグレーション時にこのオプションを選択する必要があります。

注: これは、環境内のすべてのノードが新しいレベル (WebSphere Process Server バージョン 6.1 など) になるまでの、移行する上での一時的な手段です。これらがすべて新しいレベルになったら、以下の操作を実行する必要があります。

- a. 新しいバージョン (この場合は バージョン 6.1) のすべての設定を使用するよう管理スクリプトを変更する。
- b. `convertScriptCompatibility` コマンドを使用して、新しいバージョンに対応するすべての設定に一致するよう構成を変換する。

`convertScriptCompatibility` コマンドを参照してください。

注: このリンクの指示に従って、`convertScriptCompatibility` コマンドを使用するとき、`WASPostUpgrade` コマンドではなく `WBIPostUpgrade` コマンドを使用してください。

10. 要約パネルの情報が正しいことを確認し、「次へ」をクリックしてマイグレーションを開始します。

新しいターゲット・プロファイルを作成するようオプションを選択した場合、作成の開始と結果がパネルに表示されます。

パネルにはマイグレーション・プロセスの進行状況が表示されます。

マイグレーションが正常でないとき、ウィザードに障害パネルが表示されます。マイグレーションが一部正常なとき、ウィザードに警告パネルが表示されます。問題を訂正して、マイグレーションを再試行してください。

事後マイグレーションが正常に行われたとき、ウィザードには正常であることが表示されます。

11. 「終了」をクリックしてマイグレーション・ウィザードを終了します。

結果

これで、マイグレーション済みのサーバーを WebSphere Process Server 環境において新しいリリース・レベルで始動できます。

関連概念

9 ページの『マイグレーション・ウィザード』
バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連タスク

 プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、`manageprofiles` コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

関連資料

『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された `backupDirectory` から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の `profile_root/installedApps` ディレクトリーに追加します。

関連情報

Java 仮想マシン設定

`convertScriptCompatibility` コマンド

WBIPreUpgrade コマンド

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

詳細

WBIPreUpgrade コマンドは、選択されたファイルを `install_root` および `profile_root` ディレクトリーから、指定したバックアップ・ディレクトリーに保存します。マイグレーションを行うと、ファイルは、バックアップ・ディレクトリーの以下のサブディレクトリーに保存されます。

- `websphere_backup`
 - `bin`
 - `cloudscape`
 - `lib`
 - `ProcessChoreographer/Staff`

- properties
- *profile_root*
 - bin
 - config
 - etc
 - event
 - installedApps
 - installedConnectors
 - properties

ロケーション

このコマンド・ファイルは、*install_dir/bin* ディレクトリーにあり、そこで実行される必要があります。

構文

構文は以下のとおりです。

UNIX

Linux

```
WBIPreUpgrade.sh backupDirectory
                  currentWebSphereDirectory
                  [-traceString trace_spec [-traceFile file_name ]]
```

Windows

```
WBIPreUpgrade.bat backupDirectory
                  currentWebSphereDirectory
                  [-traceString trace_spec [-traceFile file_name ]]
```

パラメーター

パラメーターは以下のとおりです。

backupDirectory

必須パラメーターです。第 1 パラメーターとして指定する必要があります。値 *backupDirectory* には、コマンド・スクリプトにより保存される構成の格納先のディレクトリー名を指定します。

また、このディレクトリーから WBIPostUpgrade コマンドにより構成が読み取られます。

ディレクトリーが存在しない場合、WBIPreUpgrade コマンド・スクリプトによりディレクトリーが作成されます。

currentWebSphereDirectory

必須パラメーターです。第 2 パラメーターとして指定する必要があります。マイグレーションがサポートされる任意の版の WebSphere Process Server を指定できます。

値 *currentWebSphereDirectory* には、既存の WebSphere Process Server インストール済み環境のインストール・ルート・ディレクトリーの名前を指定します。

-traceString

これはオプション・パラメーターです。値 *trace_spec* には、収集するトレース情報を指定します。

すべてのトレース情報を収集するには、"**=all=enabled*" (引用符付き) を指定します。

-traceString パラメーターも **-traceFile** パラメーターも指定しない場合、デフォルトでコマンドによりトレース・ファイルが作成され、*backupDirectory/logs* ディレクトリーに配置されます。

このパラメーターを指定する場合は、**-traceFile** パラメーターも指定する必要があります。

-traceFile

これはオプション・パラメーターです。値 *file_name* には、トレース情報の出力ファイル名を指定します。

-traceString パラメーターも **-traceFile** パラメーターも指定しない場合、デフォルトでコマンドによりトレース・ファイルが作成され、*backupDirectory/logs* ディレクトリーに配置されます。

-traceString パラメーターを指定して **-traceFile** パラメーターを指定しない場合は、スクリプトによってトレース・ファイルは生成されません。

ロギング

WBIPreUpgrade ツールでは、実行中に、画面に状況が表示されます。また、このツールによって、さらに詳細なログ情報が、*backupDirectory/logs* ディレクトリーに作成される *WBIPreUpgrade.time_stamp.log* ファイルに保存されます。ここで、*backupDirectory* は *backupDirectory* パラメーターで指定した値です。*WBIPreUpgrade.time_stamp.log* ファイルはテキスト・エディターで表示できます。

関連概念

9 ページの『マイグレーション・ウィザード』

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連資料

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

22 ページの『WBIProfileUpgrade スクリプト』

クラスターをマイグレーションする場合とその他の特殊な状況において、WBIProfileUpgrade スクリプトを使用して WebSphere Process Server プロファイルのアプリケーションと構成の設定を更新します。

関連情報

clientUpgrade コマンド

WBIPostUpgrade コマンド

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

ロケーション

コマンド・ファイルは、*install_root/bin* ディレクトリーに格納されており、そのディレクトリーで実行されるようにします。

構文

構文は以下のとおりです。

UNIX

Linux

```
WBIPostUpgrade.sh backupDirectory
    [-oldProfile profile_name]
    [-profileName profile_name]
    [-scriptCompatibility true | false]
    [-portBlock port_starting_number]
    [-backupConfig true | false]
    [-replacePorts true | false]
    [-keepAppDirectory true | false]
    [-keepDmgrEnabled true | false]
    [-appInstallDirectory user_specified_directory]
    [-traceString trace_spec [-traceFile file_name]]
```

Windows

```
WBIPostUpgrade.bat backupDirectory
    [-oldProfile profile_name]
    [-profileName profile_name]
    [-scriptCompatibility true | false]
    [-portBlock port_starting_number]
    [-backupConfig true | false]
    [-replacePorts true | false]
    [-keepAppDirectory true | false]
    [-keepDmgrEnabled true | false]
    [-appInstallDirectory user_specified_directory]
    [-traceString trace_spec [-traceFile file_name]]
```

パラメーター

パラメーターは以下のとおりです。

backupDirectory

これは必須パラメーターです。値 *backupDirectory* には、WBIPreUpgrade ツールが保存済み構成およびファイルを保管し、WBIPostUpgrade ツールがその構成およびファイルを読み取るディレクトリーの名前を指定します。

-oldProfile

これは、前のバージョンからインスタンスまたはプロファイルをマイグレーションする場合のオプション・パラメーターです。インスタンスまたはプロファイル

は、このコマンドが実行される前に、マイグレーション・バックアップ・ディレクトリーにすでに存在している必要があります。

-oldProfile パラメーターを指定しない場合には、デフォルト・プロファイルが使用されます。デフォルト・プロファイルが見つからない場合、システムはエラーを報告します。

-profileName

これは、特定のプロファイルにマイグレーションする場合のオプション・パラメーターです。値 *profile_name* には、最新バージョンの WebSphere Process Server によってすでに作成されており、スクリプトが構成をマイグレーションする先のプロファイルの名前を指定します。WBIPostUpgrade コマンドを呼び出す前に、このプロファイルを作成しておく必要があります。

-profileName パラメーターを指定しない場合には、デフォルト・プロファイルが使用されます。デフォルト・プロファイルが見つからない場合、システムはエラーを報告します。

-scriptCompatibility

これは、マイグレーションで以下のバージョン 6.0.x 構成定義を作成するかどうかを指定するためのオプション・パラメーターです。

- トランスポート
- ProcessDef
- バージョン 6.0.x の SSL

これらは以下のバージョン 6.1 構成定義の代替となります。

- チャネル
- ProcessDefs
- バージョン 6.1 の SSL

デフォルトは true です。

既存の管理スクリプトへの影響を最小限に抑えるには、このパラメーターを true に指定します。例えば、サード・パーティーの構成 API を使用してバージョン 6.0.x 構成定義を作成または変更する既存の wsadmin スクリプトまたはプログラムが存在する場合は、マイグレーション時にこのオプションを true に指定できます。

注: これは、環境内のノードすべてがバージョン 6.1 レベルになるまで一時的な遷移を提供することを意味します。これらがすべて新しいレベルになったら、以下の操作を実行する必要があります。

1. バージョン 6.1 の設定すべてを使用するように管理スクリプトを変更します。
2. `convertScriptCompatibility` コマンドを使用して構成を変換し、バージョン 6.1 の設定すべてに一致させます。

詳しくは、『`convertScriptCompatibility`コマンド』を参照してください。

-portBlock

これはオプション・パラメーターです。*port_starting_number* 値には、コマンド・スクリプトの実行時に割り当てる連続ポート番号ブロックの最初の番号を指定します。

-backupConfig

これは、WBIPostUpgrade ツールによって何らかの変更が行われる前に、既存の構成を保存するかどうかを指定するためのオプション・パラメーターです。デフォルトでは true に設定されているので、WBIPostUpgrade は、現在の構成のコピーを *profile_name/temp* ディレクトリーに保存します。

必要に応じて、restoreConfig コマンドを使用してその構成を復元します。

『restoreConfigコマンド』を参照してください。

-replacePorts

このオプション・パラメーターは、仮想ホストおよび Web コンテナのトランスポート・ポートのポート値のマップ方法を指定するために使用します。

• False

マイグレーション時には、バージョン 6.0.x のポート定義を置き換えないでください。

- 前のバージョンの構成は別個に残され、チャンネルはどれも削除されません。
- 次の 4 つの名前付きチャンネルは、前のリリースで設定されていた値と等しい値に設定されます。
 - WC_adminhost
 - WC_defaulthost
 - WC_adminhost_secure
 - WC_defaulthost_secure
- マイグレーション・プロセスでは、-scriptCompatibility 設定に基づいて、前のリリースのすべてのポートに対するトランスポートまたはチャンネルが作成されます。
- マイグレーション・プロセスでは、すべての非 Web コンテナ・ポートが、前のリリースで設定されていた値に設定されます。
- マイグレーション・プロセスで、Web コンテナで定義されるものと同じトランスポートまたはチャンネルが作成されると、ポート競合が発生する可能性があります。

これはデフォルトです。

• True

マイグレーション時に、すべての仮想ホスト別名ポートの設定をバージョン 6.1 のポート定義に置き換えます。

マイグレーション・プロセスではデフォルトで、前の環境の構成データを新規 WebSphere Process Server 環境のデータに追加します。ただし場合によっては、これは、これらのポート値に対して望ましい振る舞いではないことがあります。例えば、前のリリースからの既存のポート定義は、他の製品とのポート競合を回避するために慎重に設定されていたとします。そのような場合には、それらの設定を新規バージョンの構成にマイグレーションすることを望むものです。新規値が追加される前に一致する仮想ホストのポートすべてが新規バージョンの構成から削除されるようにするには、このパラメーターに true を指定します。

- 前のリリースで設定されていた値と等しい値に設定される次の 4 つの名前付きチャンネルを除いて、Web コンテナーに関連付けられているすべてのトランスポート・チャンネルが削除されます。
 - WC_adminhost
 - WC_defaulthost
 - WC_adminhost_secure
 - WC_defaulthost_secure
- マイグレーション・プロセスでは、-scriptCompatibility 設定に基づいて、前のリリースのすべてのポートに対するトランスポートまたはチャンネルが作成されます。
- マイグレーション・プロセスでは、すべての非 Web コンテナー・ポートが、前のリリースで設定されていた値に設定されます。

-keepAppDirectory

これは、すべてのアプリケーションを現在配置されているのと同じディレクトリにインストールするかどうか指定するためのオプション・パラメーターです。デフォルトは false です。

このパラメーターを true に指定すると、個々のアプリケーションがそのロケーションを維持します。

制約事項: このパラメーターを true に指定すると、そのロケーションは既存の WebSphere Process Server と新規インストールによって共有されます。マイグレーションされるアプリケーションを前のバージョンのときと同じロケーションに保持する場合は、次の制限が適用されます。

- バージョン 6.1 の混合ノードのサポート制限に従う必要があります。つまり、wsadmin コマンドを呼び出したとき、以下のサポートが使用できません。
 - JSP のプリコンパイル
 - バイナリー構成の使用
 - EJB のデプロイ
- 後になって、以前に存在していたインストールを管理する (例えば、アンインストール) ときにこれらのロケーションからアプリケーションを削除すると、マイグレーションされたアプリケーションを意図せずに失ってしまう危険性があります。

-keepDmgrEnabled

これは、既存の WebSphere Process Server デプロイメント・マネージャーを使用不可に設定するかどうかを指定するためのオプション・パラメーターです。デフォルトは false です。

このパラメーターを true に指定すると、マイグレーションが完了する間、既存のデプロイメント・マネージャーを使用できます。この設定は、デプロイメント・マネージャーをマイグレーションする場合にのみ有効であり、他のすべてのマイグレーションでは無視されます。

注意: このパラメーターは慎重に使用してください。

- 既存の WebSphere Process Server デプロイメント・マネージャーの構成が通常は停止および使用不可にされる理由は、複数のデプロイメント・マネー

ャーが同じノードを管理しないようにするためです。新規バージョンのデプロイメント・マネージャーを使用し始める前に、既存のデプロイメント・マネージャーを停止する必要があります。この操作を行わない場合に最も生じやすいエラー状態は、デプロイメント・マネージャーの 2 番目のインスタンスが開始するときに発生するポート競合です。

- このパラメーターに `true` を指定すると、マイグレーション中に古い構成で行われた構成変更は、どれもマイグレーションされない可能性があります。

-appInstallDirectory

これは、マイグレーション時にすべてのアプリケーションをインストールする場合に使用するディレクトリー名を渡すためのオプション・パラメーターです。このパラメーターが指定されない場合は、デフォルト値 `profile_name¥installedApps` が使用されます。

名前に 1 つ以上のブランクが含まれている場合は、ディレクトリー名を引用符で囲む必要があります。

-traceString

これはオプション・パラメーターです。値 `trace_spec` には、収集するトレース情報を指定します。

すべてのトレース情報を収集するには、"`*=all=enabled`" (引用符付き) を指定します。

`-traceString` パラメーターも `-traceFile` パラメーターも指定しない場合、デフォルトでコマンドによりトレース・ファイルが作成され、`backupDirectory/logs` ディレクトリーに配置されます。

このパラメーターを指定する場合は、`-traceFile` パラメーターも指定する必要があります。

-traceFile

これはオプション・パラメーターです。値 `file_name` には、トレース情報の出力ファイル名を指定します。

`-traceString` パラメーターも `-traceFile` パラメーターも指定しない場合、デフォルトでコマンドによりトレース・ファイルが作成され、`backupDirectory/logs` ディレクトリーに配置されます。

`-traceString` パラメーターを指定して `-traceFile` パラメーターを指定しない場合は、スクリプトによってトレース・ファイルは生成されません。

ロギング

WBIPostUpgrade ツールでは、実行中に、画面に状況が表示されます。また、このツールによって、さらに詳細なログ情報が、`backupDirectory/logs` ディレクトリーにある `WBIPostUpgrade.time_stamp.log` ファイルに保存されます。

`WBIPostUpgrade.time_stamp.log` ファイルはテキスト・エディターで表示できます。

関連概念

9 ページの『マイグレーション・ウィザード』

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

『WBIProfileUpgrade スクリプト』

クラスターをマイグレーションする場合とその他の特殊な状況において、WBIProfileUpgrade スクリプトを使用して WebSphere Process Server プロファイルのアプリケーションと構成の設定を更新します。

関連情報

convertScriptCompatibility コマンド

restoreConfig コマンド

WBIProfileUpgrade スクリプト

クラスターをマイグレーションする場合とその他の特殊な状況において、WBIProfileUpgrade スクリプトを使用して WebSphere Process Server プロファイルのアプリケーションと構成の設定を更新します。

目的

WBIProfileUpgrade は、 WebSphere Process Server に付属するサポート・アプリケーションと、WebSphere Process Server プロファイル内の構成設定を更新する Apache™ Ant スクリプトです。通常は、マイグレーション・ウィザードまたは WBIPostUpgrade コマンドによって自動的に呼び出されます。ただし、クラスターのマイグレーション時に、マイグレーション・ウィザードまたは WBIPostUpgrade コマンドがクラスター・メンバー上で実行されたら、WBIProfileUpgrade を手動で呼び出す必要があります。また、内部サポート・アプリケーションを手動でマイグレーションする場合や、マイグレーション中にエラーが発生する特殊な状況では、WBIProfileUpgrade を呼び出すことが必要になります。

構文

構文は以下のとおりです。

UNIX

Linux

```
profile_root/bin/ws_ant.sh -f install_root/util/WBIProfileUpgrade.ant  
-DmigrationDir=backupDirectory [-Dcluster=clustername ]
```

Windows

```
profile_root\bin ws_ant.bat -f install_root/util/WBIProfileUpgrade.ant  
-DmigrationDir=backupDirectory [-Dcluster=clusterName ]
```

注: Windows システムでは、スラッシュを 1 つ使用する (c:/migration/backup など) か二重の円記号 (¥) を使用して (c:¥migration¥backup など)、パラメーターへのパスを指定します。

パラメーター

profile_root

マイグレーションされるプロファイルがあるディレクトリー。ws_ant コマンドは、このプロファイル・ディレクトリーの bin サブディレクトリーにあります。

install_root

WebSphere Process Server をインストールするディレクトリ。

backupDirectory

これは必須パラメーターです。*backupDirectory* は、WBIPreUpgrade コマンドによって作成されるマイグレーション固有のバックアップ・ディレクトリーがあるディレクトリーの名前を指定します。

clusterName

これにより、マイグレーションする必要があるクラスターの名前を指定します。

ロギング

WBIPreUpgrade スクリプトは、実行されるときにログを作成し、そのログは、*backupDirectory/logs/WBIPreUpgrade.ant-time_stamp.log* ディレクトリーに書き込まれます。ここで、*backupDirectory* は *backupDirectory* パラメーターに指定された値です。*WBIPreUpgrade.ant-time_stamp.log* ファイルはテキスト・エディターで表示できます。

例

注: 以下の例はそれぞれ 1 行のコマンド行ですが、読みやすくするために複数行に分けて表示しています。

UNIX

Linux

```
/opt/IBM/WPS61/profiles/DMgr/bin>ws_ant.sh -f
../../../../util/WBIPreUpgrade.ant -DmigrationDir=/tmp/migrationBackup
-Dcluster=clusterA
```

Windows

```
C:\IBM\WPS61\profiles\DMgr\bin>ws_ant.bat -f
../../../../util/WBIPreUpgrade.ant -DmigrationDir=c:\\temp\\migrationBackup
-Dcluster=clusterA
```

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

旧バージョンからマイグレーションするときのデータ処理方法

WebSphere Process Server のバージョン間マイグレーション・ツールは、さまざまなデータ・セット (エンタープライズ・アプリケーション・データ、構成データ、およびシステム・アプリケーション・データ) をそれぞれ異なる方法で処理します。

構成データのマイグレーション

バージョン間マイグレーション・ツール (ウィザードまたはスクリプト) は、以前のプロファイルの構成設定を、マイグレーション・プロセスで作成される新規プロファイルに自動的に適用します。新規プロファイルが既に構成されており、古いプロファイルと新規プロファイルの値が一致しない場合には、それらの値が次のように処理されます。

- 新規プロファイルで既に構成済みのインストール・ディレクトリー名は、新規プロファイルに保持されます。
- インストール・ディレクトリー名以外で、新規プロファイルにある一致しない値は、古いプロファイルの値に置き換えられます。

アプリケーションのマイグレーション

ご使用のアプリケーション (WebSphere Process Server 製品に付属していないアプリケーションすべて) は、サポートされるマイグレーション・シナリオではバイナリー互換です。(サポートされるマイグレーションのシナリオについては、1 ページの『マイグレーションの概要』を参照してください)。アプリケーションは、WebSphere Process Server の新バージョンで実行するためにその一部に変更を加える必要はありません。

注: バージョン 6.0.1 WebSphere Adapters の場合、互換性を保つにはいくつかの追加ステップが必要です。このことや、その他の例外については詳しくは、WebSphere Process Server の技術情報 (WebSphere Process Server 技術情報の Web サイト) を参照してください。

サンプル・アプリケーションを除いて、WebSphere Process Server 製品の一部として提供されるアプリケーションはそれらのアプリケーションの最新バージョンにマイグレーションされます。これらは以下のように処理されます。

- すべてのシステム・アプリケーション (*install_root/systemApps* ディレクトリーに存在するアプリケーション) には、新バージョンがインストールされます。
- すべてのサポート・アプリケーション (Business Rules Manager や Business Process Choreographer アプリケーションなどの WebSphere Process Server に付属するアプリケーション) では、古いバージョンが最新バージョンに更新されます。

サンプル・アプリケーションは別の方法で処理されます。スタンドアロン・プロファイルの場合、マイグレーション・プロセスではサンプル・アプリケーションが何もインストールされません。スタンドアロン・プロファイルでサンプル・アプリケーションを使用可能にする場合は、新しいバージョンの WebSphere Process Server 用インストール・ウィザードを使用してインストールすることができます。

Network Deployment プロファイルの場合、旧バージョンの WebSphere Process Server を使用してインストールされたサンプルは、新バージョンへのマイグレーション時にインストールされます。

データベースのマイグレーション

Cloudscape データベースを使用している場合は、一部の例外を除いて、データベース構成がマイグレーション・ツールによって自動的にマイグレーションされます。詳しくは、118 ページの『Cloudscape データベースのマイグレーション』を参照してください。また、Cloudscape データベースは Derby データベースに変換されます。Derby は Cloudscape の後継データベースで、WebSphere Process Server のバージョン 6.1 でサポートされています。

Cloudscape 以外のデータベースを使用している場合は、マイグレーション・ツールにより、既存のデータ・ソースおよびプロバイダーごとにプロバイダー定義とデータ・ソース定義が自動的にマイグレーションされます。しかし、データベース・スキーマのアップグレードも必要な場合があるため、特別な注意が必要です。サーバー・プロセスに必須のデータベース権限がある場合 (データベースによってはその他の要件も満たしている場合)、スキーマのアップグレードは、サーバーを最初に始動した際に自動的行われます。

必須の権限がサーバー・プロセスにないか、その他の要件を満たしていない場合、またはデータベース・スキーマを手動で更新する場合、提供されたスクリプトを使用する必要があります。詳しくは、32 ページの『マイグレーションのためのデータベースのアップグレード』を参照してください。

関連概念

1 ページの『マイグレーションの概要』

WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからマイグレーションします。

関連タスク

118 ページの『Cloudscape データベースのマイグレーション』

マイグレーション・ツールを使用して WebSphere Process Server バージョン 6.1 にマイグレーションした後、Cloudscape データベースの自動マイグレーションの結果を検査して、ツールによって自動的にマイグレーションされていない Cloudscape データベース・インスタンスがあれば、手動でマイグレーションします。

製品構成のマイグレーション時の構成マッピング

製品構成のマイグレーション時には、さまざまな構成がマッピングされます。

マイグレーション操作は必ず、1 つのプロファイルから、同じマシンまたは別のマシンにある別のプロファイルへのマイグレーションを伴います。例えば、バージョン 6.1 のデプロイメント・マネージャー・プロファイルに WebSphere Process Server バージョン 6.0.x デプロイメント・マネージャーをマイグレーションする場合や、バージョン 6.1 のスタンドアロン・サーバー・プロファイルにバージョン 6.0.x のスタンドアロン・サーバーをマイグレーションする場合などがあります。

マイグレーションには多くのシナリオがあります。マイグレーション・ツールは、マイグレーション元となるバージョンに存在するオブジェクトおよび属性を、新規バージョン環境の対応するオブジェクトおよび属性にマッピングします。

ブートストラップ・ポート

マイグレーション・ツールは、デフォルト以外の値をバージョン 6.1 の環境に直接マップします。

ただし、WBIPostUpgrade への呼び出し中に `-portBlock` パラメーターを指定すると、バージョン 6.1 にマイグレーションした各サーバーに新規のポート値が指定されます。

コマンド行パラメーター

マイグレーション・ツールは、適切なコマンド行パラメーターを、サーバー・プロセス定義の Java 仮想マシン (JVM) 設定に変換します。ほとんどの設定は直接にマッピングされます。一部の設定は、WebSphere Process Server のバージョン 6.1 構成内にそのロールが存在しない、構成内での意味が異なる、または構成内でのスコープが異なる、などの理由によりマイグレーションされません。

プロセス定義設定の変更方法について詳しくは、WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『プロセス定義設定』を参照してください。Java 仮想マシンの設定の変更方法について詳しくは、WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『Java 仮想マシン設定』を参照してください。

EAR ファイルをマイグレーションする場合の Java ヒープ・サイズ

wsadmin ツールを使用してすべての WebSphere Process Server EAR ファイルをバージョン 6.1 にマイグレーションする場合、WBIPostUpgrade ツールはデフォルトの最大 Java ヒープ・サイズ値である 64 MB を使用して EAR ファイルをインストールします。

マイグレーション時に、Java ヒープ・サイズの大きさが不十分であるためにバージョン 6.1 の EAR ファイルのインストールに失敗すると、次のようなメッセージが表示されます。

```
java.lang.OutOfMemoryError JVMXE006:OutOfMemoryError
```

最大 Java ヒープ・サイズを大きくし、以下の例に従ってアプリケーションをインストールしてください。

バージョン 6.1 の WebSphere Process Server でのアプリケーションのインストール例

以下のように想定します。

インストール・ルート

```
C:¥WebSphere¥DeploymentManager
```

番号記号 (###)

```
最大ヒープ・サイズ値
```

EAR_file_name

```
EAR ファイルの名前
```

app_name
アプリケーションの名前

cluster_name
EAR ファイルをインストールするクラスターの名前

コマンドは、分かりやすくするために複数の行で表示しています。

```
wsadmin -conntype NONE
        -javaoption
        -Xmx###m
        -c "$AdminApp install
            C:¥¥WebSphere¥¥DeploymentManager¥¥installableApps¥¥
            EAR_file_name>
        {-nodeployejb
        -appname app_name
        -cluster cluster_name}"
```

バージョン 6.0.x のノードからバージョン 6.1 のノードへのマイグレーション

セルに属する WebSphere Process Server のバージョン 6.0.x のノードは、セルからそのノードを除去せずに WebSphere Process Server のバージョン 6.1 にマイグレーションできます。

セル内のベース・ノードをマイグレーションする前に、まずデプロイメント・マネージャーをマイグレーションしてください。

バージョン 6.0.x からバージョン 6.1 にマイグレーションするときには、同じセル名を使用します。異なるセル名を使用すると、WebSphere Process Server のバージョン 6.1 のセルに統合ノードを正常にマイグレーションできなくなります。

セル内のベース WebSphere Process Server ノードをバージョン 6.1 にマイグレーションすると、ノード・エージェントもバージョン 6.1 にマイグレーションされます。セルには、いくつかのバージョン 6.1 のノードと、バージョン 6.0.x レベルにある他のノードを含めることができます。リリースが混在したセルを使用する上での制限については、他の WebSphere 製品のインストール済み環境との『共存』を参照してください。詳しくは、75 ページの『クラスターのマイグレーション』および 60 ページの『非クラスター管理対象ノードのマイグレーション』を参照してください。

ポリシー・ファイル

WebSphere Process Server のバージョン 6.1 は、バージョン 6.0.x のポリシー・ファイルと共にインストールされた、次の特性を持つポリシー・ファイルすべてをマイグレーションします。

- バージョン 6.1 のポリシー・ファイル内のコメントはすべて保持されます。バージョン 6.0.x のポリシーに含まれるコメントは、どれもバージョン 6.1 に組み込まれません。
- マイグレーションでは、アクセス権限または認可のマージは試みられません。これは、完全に追加型のマイグレーションです。アクセス権限または認可がバージョン 6.1 のファイル内に存在しない場合は、マイグレーションによって持ち込まれます。
- セキュリティーは重要なコンポーネントであるため、マイグレーションによって追加される場合は、元の .policy ファイルの最後にある「MIGR0372I: Migrated grant permissions follow」コメントの直後に追

加されます。これにより管理者は、マイグレーションによって生じたポリシー・ファイルの変更を検証できます。

プロパティと lib/app ディレクトリー

マイグレーションでは、以前のバージョン・ディレクトリーからバージョン 6.1 の WebSphere Process Server 構成にファイルをコピーします。

プロパティ・ファイル

WebSphere は、設定をバージョン 6.1 のプロパティ・ファイルにマージすることにより、バージョン 6.0.x でインストールされたバージョン 6.1 の WebSphere Process Server プロパティ・ファイルすべてをマイグレーションします。

マイグレーションによってプロパティ・ファイルがオーバーレイされることはありません。

J2C リソースが参照するリソース・アダプター・アーカイブ (RAR)

J2C リソースが参照する RAR は、WebSphere Process Server の古いインストールに存在していればマイグレーションされます。この場合に RAR は、WebSphere Process Server の新規インストールでの対応するロケーションにコピーされます。Relational Resource Adapter の RAR はマイグレーションされません。

クラスター・レベルのリソースのマイグレーション:

クラスター・レベルのリソースは、クラスターのディレクトリーの下にある resourcexxx.xml ファイルで構成されます。以下に例を示します。

```
<resources.j2c:J2CResourceAdapter xmi:id="J2CResourceAdapter_1112808424172"
  name="ims" archivePath="{WAS_INSTALL_ROOT}¥installedConnectors¥x2.rar">
  ...
</resources.j2c:J2CResourceAdapter>
```

クラスター・レベルのリソースを使用している場合は、各クラスター・メンバー (ノード) の同じロケーションにこのリソースが存在します。したがって、上の例で説明すると、各クラスター・メンバーでは、ロケーション `{WAS_INSTALL_ROOT}¥installedConnectors¥x2.rar` に RAR ファイルがインストールされています。`{WAS_INSTALL_ROOT}` は、各クラスター・メンバーで解決され、正確なロケーションに到達します。

デプロイメント・マネージャーのマイグレーションでは、resourcexxx.xml ファイルを含むデプロイメント・マネージャーのクラスター・ファイルがツールによってマイグレーションされます。

管理対象ノードのマイグレーションでは、ツールが各 J2C アダプターを処理します。RAR ファイルなどのファイルは、次のようにして、バージョン 6.0.x からバージョン 6.1 にマイグレーションされます。

バージョン 6.0.x からバージョン 6.1 へのマイグレーションでは、RAR ファイルなどのファイルが、`WAS_INSTALL_ROOT` から `WAS_INSTALL_ROOT` へ、また `USER_INSTALL_ROOT` から `USER_INSTALL_ROOT` へコピーされます。

例えば、バージョン 6.0.x の WAS_INSTALL_ROOT に RAR ファイルがある場合、マイグレーション・ツールは WAS_INSTALL_ROOT から USER_INSTALL_ROOT へのファイルのコピーを自動的には行いません。これにより、クラスター・レベルの J2C リソースの整合性が維持されます。ただし、バージョン 6.0.x で RAR ファイルへのパスをハードコーディングしている場合 (例えば archivePath="C:/WAS/installedConnectors/x2.rar")、バージョン 6.1 のマイグレーション・ツールでは、そのことを反映するように archivePath 属性を変更することはできません。これは、その属性を変更すると、マイグレーションされていない他のクラスター・メンバーのすべてを切断することになるからです。

サンプル

デプロイメント・マネージャーのマイグレーション時には、WebSphere Process Server の統合ノード用サンプルは何もマイグレーションされません。すべてのバージョン 6.1 サンプルに対して、同等のバージョン 6.1 サンプルが使用可能です。

セキュリティ

WebSphere Process Serverバージョン 6.1 でセキュリティを有効にすると、Java 2 セキュリティがデフォルトで有効になります。Java 2 セキュリティでは、セキュリティ・アクセス権限を明示的に与える必要があります。

バージョン 6.1 では、数種類の技術を使用して異なるレベルの Java 2 セキュリティを定義できます。その 1 つでは、アプリケーションの一部として was.policy ファイルを作成し、すべてのセキュリティ・アクセス権限を使用可能に設定します。マイグレーション・ツールは、wsadmin コマンドを呼び出して、バージョン 6.1の properties ディレクトリーにある was.policy ファイルを、マイグレーション中にエンタープライズ・アプリケーションに追加します。

バージョン 6.1 の WebSphere Process Server にマイグレーションする場合は、スクリプト互換性をサポートするようにマイグレーションするかどうかの選択によって、結果が 2 つに分かれます。

- スクリプト互換性をサポートするマイグレーションを選択すると、セキュリティ構成は変更なしでバージョン 6.1 に引き渡されます。

これはデフォルトです。

- スクリプト互換性をサポートするマイグレーションを選択しない場合、セキュリティ構成は WebSphere Process Serverバージョン 6.1 のデフォルト構成に変換されます。バージョン 6.1 デフォルト・セキュリティ構成は旧バージョンとほとんど同様に動作しますが、いくつかの変更点があります。

例えば、既存の鍵ファイルとトラスト・ファイルは SSLConfig レポートリー外に移され、新しい鍵ストア・オブジェクトとトラストストア・オブジェクトが作成されました。

同じセキュリティ設定を維持するには、バージョン 6.0.x で設定されている WebSphere Application Server のセキュリティ設定をマイグレーションする必要があります。バージョン 6.1 へのセキュリティ構成のマイグレ

ーションについて詳しくは、WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『マイグレーション、共存、および相互運用 - セキュリティに関する考慮事項』を参照してください。

stdin、stdout、stderr、passivation、および作業ディレクトリー

これらのディレクトリーのロケーションは、通常、前のバージョンのインストール・ディレクトリー内にあります。stdin、stdout、および stderr のデフォルト・ロケーションは、バージョン 6.1 の WebSphere Process Server インストール・ルートの logs ディレクトリーです。

マイグレーション・ツールは、既存の非活性化ディレクトリーと作業ディレクトリーのマイグレーションを試みます。それ以外のディレクトリーについては、バージョン 6.1 の適切なデフォルト設定が使用されます。

非活性化ディレクトリーについて詳しくは、『EJBコンテナ設定』を参照してください。作業ディレクトリーについて詳しくは、『プロセス定義設定』を参照してください。

共存シナリオにおいて、異なるバージョンどうしで共通ディレクトリーを使用すると問題が発生します。

ポートのトランスポート

マイグレーション・ツールにより、すべてのポートがマイグレーションされます。このツールは、構成内でポートがすでに定義されている場合、ポート競合警告をログに記録します。サーバーを同時に実行できるようにするには、すべてのポート競合を解決しておく必要があります。

WBIPostUpgrade コマンドで `-portBlock` パラメーターを指定すると、マイグレーションされた各トランスポートに新規の値が割り当てられます。

WBIPostUpgrade コマンドについて詳しくは、17 ページの『WBIPostUpgrade コマンド』を参照してください。

トランスポート・チェーンとトランスポート・チャンネルについて詳しくは、『トランスポート・チェーン』を参照してください。

ポートごとに仮想ホスト別名エントリーを手動で追加する必要があります。詳しくは、『仮想ホストの構成』を参照してください。

Web モジュール

バージョン 6.0.x の WebSphere Process Server に実装された Java 2 Platform, Enterprise Edition (J2EE) の仕様レベルでは、コンテンツ・タイプを設定する場合に Web コンテナでの振る舞いを変更する必要がありました。デフォルトのサブレット書き込みプロセスによってコンテンツ・タイプの設定が行われない場合は、Web コンテナがそのデフォルトに設定されなくなるだけでなく、Web コンテナがその呼び出しを「null」として戻します。この状況になると、一部のブラウザで結果の Web コンテナ・タグが正しく表示されなくなる場合があります。この問題の発生を防止するために、エンタープライズ・アプリケーションのマイグレーションでは、Web モジュールの `autoResponseEncoding` IBM 拡張子が「true」に設定されます。

関連概念

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

『WebSphere アプリケーションのマイグレーション』

マイグレーションのために既存の WebSphere Process Server アプリケーションを変更する必要はありません。さまざまなタイプの WebSphere アプリケーションのマイグレーションについて詳しくは、WebSphere Application Server Network Deployment インフォメーション・センターの情報を参照してください。

75 ページの『クラスターのマイグレーション』

クラスターのマイグレーションは、特別な手順に従って、クラスター・メンバーを含む各プロファイルを順番にマイグレーションすることによって行います。クラスター・サービスのダウン時間を最小限に抑えるには、追加のステップを実行してください。

関連タスク

60 ページの『非クラスター管理対象ノードのマイグレーション』

WebSphere Process Server 管理対象ノードを、ニーズに応じていくつかの方法から選択してマイグレーションします。

関連資料

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

関連情報

プロセス定義設定

Java 仮想マシン設定

マイグレーション、共存、および相互運用 - セキュリティーに関する考慮事項

EJB コンテナ設定

トランスポート・チェーン

仮想ホストの構成

タスクの概説: アプリケーションでのエンタープライズ Bean の使用

WebSphere アプリケーションのマイグレーション

マイグレーションのために既存の WebSphere Process Server アプリケーションを変更する必要はありません。さまざまなタイプの WebSphere アプリケーションのマイグレーションについて詳しくは、WebSphere Application Server Network Deployment インフォメーション・センターの情報を参照してください。

ご使用のアプリケーション (WebSphere Process Server 製品に付属していないアプリケーションすべて) は、サポートされるマイグレーション・シナリオではバイナリー互換です。(サポートされるマイグレーションのシナリオについては、

1 ページの『マイグレーションの概要』を参照してください)。アプリケーションは、WebSphere Process Server の新バージョンで実行するためにその一部に変更を加える必要はありません。

注: バージョン 6.0.1 WebSphere Adapters の場合、互換性を保つにはいくつかの追加ステップが必要です。このことや、その他の例外については、WebSphere Process Server の技術情報 (WebSphere Process Server 技術情報の Web サイト) を参照してください。

特定タイプの WebSphere アプリケーションのマイグレーションについては、WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターで、『WebSphere アプリケーションのマイグレーション』とその下位トピックを参照してください。WebSphere Process Server は WebSphere Application Server Network Deployment を基にしているので、同じ情報が適用されます。

WebSphere Adapters のマイグレーションについては、IBM WebSphere Business Process Management バージョン 6.1 インフォメーション・センターの WebSphere Integration Developer の資料の中から、ご使用のアダプターに関する資料を参照してください。

WebSphere Process Server の最新バージョンで開発されたアプリケーションは、旧バージョンでは実行できません。ランタイムの互換性についての詳細は、開発およびデプロイメントのバージョン・レベルを参照してください。

関連概念

1 ページの『マイグレーションの概要』

WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからマイグレーションします。



開発およびデプロイメントのバージョン・レベル

ご使用の環境に必要な WebSphere Process Server のバージョン・レベルの決定は、アプリケーションが開発されたときのバージョン・レベルに依存します。一般に、前のバージョンの WebSphere Process Server にデプロイされたアプリケーションは、次に入手可能なバージョンの WebSphere Process Server 上で稼働します。

マイグレーションのためのデータベースのアップグレード

マイグレーションと同時に、一部の WebSphere Process Server コンポーネントのデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

このタスクについて

通常では、新しいバージョンの WebSphere Process Server で必要なデータベースの変更は、自動的に適用されます。サーバーが最初に始動されるときに、データベース・テーブルが新規スキーマ・バージョンにマイグレーションされます。ただし、サーバーがデータベース・スキーマにアクセスするための十分な権限を持たない場

合、あるいはその他のデータベース固有の要件が満たされていない場合は、データベースを手動で更新する必要があります。

WebSphere Process Server バージョン 6.1 では、スキーマのアップグレードが必要なデータベースは以下のとおりです。

- 共通データベース (デフォルト名 WPRCSDB)
- Business Process Choreographer (デフォルト名 BPEDB)

マイグレーション後の手動によるスキーマのアップグレードは、データベースを使用するサーバーを始動する前に実行してください。

Business Process Choreographer の自動アップグレードに必要な追加条件については、36 ページの『手動による Business Process Choreographer データベースのアップグレード』を参照してください。

これらのデータベースを手動でアップグレードする手順については、以下のサブトピックを参照してください。

関連タスク

118 ページの『Cloudscape データベースのマイグレーション』

マイグレーション・ツールを使用して WebSphere Process Server バージョン 6.1 にマイグレーションした後、Cloudscape データベースの自動マイグレーションの結果を検査して、ツールによって自動的にマイグレーションされていない Cloudscape データベース・インスタンスがあれば、手動でマイグレーションします。

手動による共通データベースのアップグレード

サーバーを前のバージョンからマイグレーションした後、サーバーを始動する前に「共通」データベースの新規データベース・スキーマにアップグレードする必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければなりません。

始める前に

- 既にマイグレーション・ウィザードまたはマイグレーション・スクリプトを実行して、サーバー (クラスターの場合はクラスター内のサーバー) をマイグレーション済みである必要があります。
- サーバー (またはクラスター内のサーバー) が停止したままであることを確認してください (マイグレーション・ウィザードまたはスクリプトの実行後、データベース・アップグレードが完了する前に、サーバーを始動しないでください)。

このタスクについて

マイグレーション済みサーバーがアクセスするどのデータベースでも、サーバーを始動する前にスキーマを更新する必要があります。クラスターの場合、どのマイグレーション済みクラスター・メンバーがアクセスするどのデータベースでも、クラスター・メンバーのいずれかを開始する前にスキーマを更新する必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければ

ばなりません。共通データベースの場合、データ・ソース用に構成されたデータベース・ユーザーが、テーブルの作成と変更、および索引とビューの作成と除去の操作のすべてを実行することを許可されている必要があります。

プロシージャ

1. 必ずデータベース・スキーマを更新可能な権限を持つユーザー ID を使用してください。
2. データベースをバックアップします。
3. データベース・スクリプトが格納されているディレクトリーを見つけます。

- **Windows** Windows プラットフォームの場合:
`install_root\dbscripts\component_name\database_type`

- **Linux** **UNIX** Linux および UNIX プラットフォームの場合:
`install_root/dbscripts/component_name/database_type`

各部の意味は、次のとおりです。

install_root

WebSphere Process Server バージョン 6.1 がインストールされているルート・ディレクトリーです。

component_name

CommonDB です。

database_type

は、使用中のデータベース製品に対応する名前です。利用可能なデータベース・タイプおよびそのディレクトリー名は、以下のとおりです。

データベース・タイプ	ディレクトリー名
DB2 Universal Database™ (z/OS® および i5/OS® 以外のすべてのオペレーティング・システム)	DB2
DB2® for z/OS および OS/390® v. 7.x	DB2zOSV7
DB2 for z/OS、および OS/390 v.8.x とバージョン 9.x	DB2zOSV8
Derby	Derby 注: 以前の WebSphere Process Server (6.0.x) インストール済み環境用の既存の Cloudscape データベースがある場合、データベースは、まず Cloudscape データベースから Derby データベースにアップグレードする必要があります。118 ページの『Cloudscape データベースのマイグレーション』を参照してください。次に、その同じバージョン (6.0.x) の WebSphere Process Server に対応する Derby データベースが生成されたら、そのデータベースを、WebSphere Process Server バージョン 6.1 に対応するように、このディレクトリー内のスクリプトを使用してアップグレードする必要があります。

データベース・タイプ	ディレクトリー名
Informix®	Informix
Oracle	Oracle
Microsoft® SQL Server	SQLServer

特定のデータベース製品名およびバージョンについては、データベース仕様書を参照してください。

- データベースおよび現在のスキーマ・バージョンのマイグレーション・スクリプトを探します。ここで、*schema_version* の値は、バージョン 6.0.1 の場合は 601、バージョン 6.0.2 の場合は 602 です。例えば、共通データベース・スキーマを WebSphere Process Server 6.0.1 からバージョン 6.1 にアップグレードするには、*install_root/dbscripts/CommonDB/upgradeSchema601.sql* が必要です。共通データベース・スキーマを WebSphere Process Server 6.0.2 からバージョン 6.1 にアップグレードするには、*install_root/dbscripts/CommonDB/upgradeSchema602.sql* が必要です。

注: 他のスクリプトも同じ CommonDB ディレクトリーにあります。これには、データベースのアップグレード・プロセスの一部で使用されるものも含まれます。*upgradeSchema601.sql* または *upgradeSchema602.sql* という名前のもので使用します。これらは必要なすべてのスクリプトを一度に実行します。

- 適切なスクリプトを、そのスクリプトがあるディレクトリーからデータベースを実行中のシステムにコピーします。
- SQL スクリプトを確認し、必要に応じて自分の要件を満たすように変更します。これには、*upgradeSchemannn* という構文 (ここで *nnn* は製品バージョン番号に対応) に従って名前が付けられた、ディレクトリー内のすべてのスクリプトが含まれます。例えば、ユーザー名、パスワード、またはファイル・パスの変更が必要な場合があります。
- データベース・クライアントを使用して、データベースに接続します。これは、接続できることを確認するために実行します。
- ご使用のバージョンのアップグレード SQL スクリプトを実行します。ご使用のデータベースで .sql スクリプトを実行する方法については、そのデータベース製品の資料を参照してください。
- エラーが発生した場合、またはデータベース・クライアントの出力で障害があることが示された場合、報告されたエラーを修正して、ステップ 8 を再試行します。

結果

データベース・スキーマが更新されました。アップグレードした後に最初にサーバーを開始するときに、新しいスキーマに応じてデータがマイグレーションされます。データのマイグレーション後は、データベースに対してバージョン 6.0.x サーバーを実行することはできません。

関連タスク

36 ページの『手動による Business Process Choreographer データベースのアップグレード』

Business Process Choreographer が構成されているサーバーをマイグレーションした後は、サーバーを始動する前に、関連付けられた Business Process

Choreographer データベースのスキーマをアップグレードする必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければなりません。

手動による Business Process Choreographer データベースのアップグレード

Business Process Choreographer が構成されているサーバーをマイグレーションした後は、サーバーを始動する前に、関連付けられた Business Process Choreographer データベースのスキーマをアップグレードする必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければなりません。

始める前に

- 既にマイグレーション・ウィザードまたはマイグレーション・スクリプトを実行して、サーバー (クラスターの場合はクラスター内のサーバー) をマイグレーション済みである必要があります。
- サーバー (またはクラスター内のサーバー) が停止したままであることを確認してください (マイグレーション・ウィザードまたはスクリプトの実行後、データベース・アップグレードが完了する前に、サーバーを始動しないでください)。

このタスクについて

マイグレーション済みサーバーがアクセスする、Business Process Choreographer に関連付けられたデータベースでは、サーバーを始動する前にスキーマを更新する必要があります。クラスターの場合、どのマイグレーション済みクラスター・メンバーがアクセスするどのデータベースでも、クラスター・メンバーのいずれかを開始する前にスキーマを更新する必要があります。以下の条件のいずれかに該当する場合は、手動でアップグレードする必要があります。

- Business Process Choreographer データベースにデフォルトのテーブル・スペースを使用しなかった。Business Process Choreographer のサンプル構成を使用したか、サンプル SQL スクリプトで指定されたデフォルト・テーブル・スペース内のすべてのデータベース・オブジェクトを作成済みの場合、データベースではデフォルト・テーブル・スペースを使用します。通常、テスト環境がこれに該当します。
- BPEDB データ・ソース用に構成されたデータベース・ユーザーが、テーブルの作成と変更、および索引とビューの作成と除去の操作のすべてを実行することを許可されているわけではなく、テーブル SCHEMA_VERSION の場合は、照会、更新、削除、および挿入のすべての実行を許可されているわけではない。

プロシージャ

1. 必ずデータベース・スキーマを更新可能な権限を持つユーザー ID を使用してください。
2. データベースをバックアップします。
3. データベース・スクリプトが格納されているディレクトリーを見つけます。

- Windows Windows プラットフォームの場合:
`install_root¥dbscripts¥component_name¥database_type`
- Linux UNIX Linux および UNIX プラットフォームの場合:
`install_root/dbscripts/component_name/database_type`

各部の意味は、次のとおりです。

install_root

WebSphere Process Server バージョン 6.1 がインストールされているルート・ディレクトリーです。

component_name

ProcessChoreographer です。

database_type

は、使用中のデータベース製品に対応する名前です。利用可能なデータベース・タイプおよびそのディレクトリー名は、以下のとおりです。

データベース・タイプ	ディレクトリー名
DB2 Universal Database (z/OS および i5/OS 以外のすべてのオペレーティング・システム)	DB2
DB2 for z/OS および OS/390 v. 7.x	DB2zOSV7
DB2 for z/OS、および OS/390 v.8.x とバージョン 9.x	DB2zOSV8
Derby	Derby 注: 以前の WebSphere Process Server (6.0.x) インストール済み環境用の既存の Cloudscape データベースがある場合、データベースは、まず Cloudscape データベースから Derby データベースにアップグレードする必要があります。118 ページの『Cloudscape データベースのマイグレーション』を参照してください。次に、その同じバージョン (6.0.x) の WebSphere Process Server に対応する Derby データベースが生成されたら、そのデータベースを、WebSphere Process Server バージョン 6.1 に対応するように、このディレクトリー内のスクリプトを使用してアップグレードする必要があります。
Informix	Informix
Oracle	Oracle
Microsoft SQL Server	SQLServer

特定のデータベース製品名およびバージョンについては、データベース仕様書を参照してください。

- データベースおよび現在のスキーマ・バージョンのマイグレーション・スクリプトを探します。ここで、*schema_version* の値は、バージョン 6.0.1 の場合は 601、バージョン 6.0.2 の場合は 602 です。

Linux、UNIX、および Windows 上の DB2 の場合:

以下のアップグレード・スクリプトのいずれかを使用します。

- `upgradeSchemaschema_version.sql` は、スキーマの作成時に `createTablespace.sql` スクリプトを使用して作成されたテーブル・スペース内に新規データベース・オブジェクトを作成します。
- `upgradeSchemaschema_versionnonp.sql` は、デフォルトのテーブル・スペースに新しいオブジェクトを作成します。

z/OS および OS/390 上の DB2 の場合:

以下の順序で、2 つのスクリプトを実行します。

- a. `upgradeTablespaceschema_version.sql` は、データベース・オブジェクトをアップグレードする前にテーブル・スペースをアップグレードします。
- b. `upgradeSchemaschema_version.sql` は、テーブル・スペースをアップグレードした後にデータベース・オブジェクトをアップグレードします。

Derby の場合:

以下のアップグレード・スクリプトのいずれかを使用します。

- `upgradeSchemaschema_version.sql` は、スキーマ修飾子を使用してスキーマをアップグレードします。
- `upgradeSchemaschema_versionnonp.sql` は、カスタム・スキーマ修飾子を使用せずにスキーマをアップグレードします。

Informix Dynamic Server の場合:

以下のアップグレード・スクリプトのいずれかを使用します。

- `upgradeSchemaschema_version.sql` は、スキーマの作成時に `createDbpace.sh` または `createDbpace.bat` シェル・スクリプトを使用して作成されたデータベース・スペース内に新規データベース・オブジェクトを作成します。
- `upgradeSchemaschema_versionnonp.sql` は、デフォルトのデータベース・スペースに新しいオブジェクトを作成します。

Oracle の場合

スクリプト `upgradeSchemaschema_version.sql` を使用します。

Microsoft SQL Server の場合

以下のアップグレード・スクリプトのいずれかを使用します。

- `upgradeSchemaschema_version.sql` は、カスタム・スキーマ修飾子を使用してスキーマをアップグレードします。
- `upgradeSchemaschema_versionnonp.sql` は、カスタム・スキーマ修飾子を使用せずにスキーマをアップグレードします。
- `upgradeSchema602Unicode.sql` は、WebSphere Process Server バージョン 6.0.2 からアップグレードし、`createSchemaUnicode.sql` スクリプトまたは `createDatabaseUnicode.sql` スクリプトを使用して UNICODE をサポートする 6.0.2 スキーマを作成済みである場合、およびカスタム・スキーマ修飾子を使用したい場合に使用します。
- `upgradeSchema602UnicodeNonp.sql` は、WebSphere Process Server バージョン 6.0.2 からアップグレードし、`createSchemaUnicode.sql` スクリプトまたは `createDatabaseUnicode.sql` スクリプトを使用して

UNICODE をサポートする 6.0.2 スキーマを作成済みである場合、およびカスタム・スキーマ修飾子を使用しない場合に使用します。

注: upgradeSchema...Nonp.sql バージョンの Microsoft SQL Server アップグレード・スクリプトは、スキーマ修飾子を利用しません。それらはユーザー・スキーマ内のデータベース・オブジェクトをアップグレードします。

5. 適切なスクリプト (複数可) を、そのスクリプトがあるディレクトリーからデータベースを実行中のシステムにコピーします。
6. コピーした SQL スクリプトを確認し、必要に応じて自分の要件を満たすように変更します。例えば、ユーザー名、パスワード、スキーマ修飾子、またはファイル・パスの変更が必要な場合があります。
7. データベース・クライアントを使用して、データベースに接続します。これは、接続できることを確認するために実行します。
8. ご使用のバージョンのアップグレード SQL スクリプトを実行します。ご使用のデータベースで .sql スクリプトを実行する方法については、そのデータベース製品の資料を参照してください。
9. エラーが発生した場合、またはデータベース・クライアントの出力で障害があることが示された場合、報告されたエラーを修正して、ステップ 8 を再試行します。

結果

データベース・スキーマが更新されました。アップグレードした後に最初にサーバーを開始するときに、新しいスキーマに応じてデータがマイグレーションされます。データのマイグレーション後は、データベースに対してバージョン 6.0.x サーバーを実行することはできません。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要がある追加タスクに注意してください。

関連タスク

33 ページの『手動による共通データベースのアップグレード』

サーバーを前のバージョンからマイグレーションした後、サーバーを始動する前に「共通」データベースの新規データベース・スキーマにアップグレードする必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければなりません。

61 ページの『マイグレーション・ウィザードを使用した非クラスター管理対象ノードのマイグレーション』

マイグレーション・ウィザードを使用して、非クラスター管理対象ノードを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

 コマンド行ツールを使用した管理対象ノードのマイグレーション

コマンド行ツールを使用して、非クラスター管理対象ノードを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

75 ページの『クラスターのマイグレーション』

クラスターをマイグレーションするには、そのクラスターのメンバーを含む各プロファイルを一度に 1 つずつマイグレーションします。マイグレーションには、非クラスター環境には不要な追加のステップが必要になります。

87 ページの『最小限のダウン時間でのクラスターのマイグレーション』

ダウン時間を最小限にしてクラスターをマイグレーションするには、クラスターに寄与するプロファイルの約半分を最初にマイグレーションしてから、残りの半分をマイグレーションします。最初のプロファイルのセットをマイグレーションした後に、クラスター・マイグレーションに必要な追加のステップを実行します。

スタンドアロン・サーバーのマイグレーション

スタンドアロンの WebSphere Process Server サーバーを、ニーズに応じていくつかの方法から選択してマイグレーションします。

始める前に

1 ページの『マイグレーションの概要』および 3 ページの『事前マイグレーションの考慮事項』を参照してください。

WebSphere Process Server スタンドアロン・サーバーを、WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションする方法についての情報を参照するには、適切なマイグレーション・シナリオを選択してください。(例えば、WebSphere Process Server バージョン 6.0.2 スタンドアロン・サーバーから WebSphere Process Server バージョン 6.1 スタンドアロン・サーバーへのマイグレーション。)

- 41 ページの『マイグレーション・ウィザードを使用したスタンドアロン・サーバーのマイグレーション』

このトピックには、古いバージョンの WebSphere Process Server スタンドアロン・サーバーから新しいバージョンのスタンドアロン・サーバーに、マイグレーション・ウィザード (グラフィカル・ユーザー・インターフェース) を使用してマイグレーションするための説明が記載されています。

- 45 ページの『コマンド行ツールを使用したスタンドアロン・サーバーのマイグレーション』

このトピックには、古いバージョンの WebSphere Process Server スタンドアロン・サーバーから新しいバージョンのスタンドアロン・サーバーに、マイグレーション・ウィザードでなくマイグレーション・コマンド行ツールを使用してマイグレーションするための説明が記載されています。

- 99 ページの『リモート・システムへのマイグレーション』

このトピックには、リモート・システム上で古いバージョンの WebSphere Process Server サーバーから新しいバージョンのサーバーにマイグレーションするための説明が記載されています。

- 105 ページの『サポート対象外となったオペレーティング・システムからのマイグレーション』

このトピックには、WebSphere Process Server がサポートしなくなったオペレーティング・システム上で実行中の WebSphere Process Server スタンドアロン・サーバーをマイグレーションするための説明が記載されています。

ヒント: マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

マイグレーション・ウィザードを使用したスタンドアロン・サーバーのマイグレーション

マイグレーション・ウィザードを使用して、スタンドアロン・サーバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。

注: 新しい物理コンピューター・システムにマイグレーションする場合、99 ページの『リモート・システムへのマイグレーション』で説明されている代替の手順を使用します。WebSphere Process Server を実行しているコンピューター上のオペレーティング・システムのバージョンをアップグレードする場合は、105 ページの『サポート対象外となったオペレーティング・システムからのマイグレーション』で説明されている代替の手順を使用します。

- 古い WebSphere Process Server バージョンで作成された統合されていないプロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、`backupConfig` コマンドまたは望ましいバックアップ・ユーティリティーを使用して、既存の構成をバックアップします。『`backupConfig` コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

プロシージャ

1. Linux または UNIX システムでは root として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション対象のノード上でバージョン 6.0.x サーバーが実行中の場合、そのサーバーを停止します。影響を受けるサーバーのプロファイルの `profile_dir/bin` ディレクトリーで `stopServer` コマンドを使用するか、プロファイルのファースト・ステップ・コンソールでサーバーを停止します。

`stopServer` コマンドについて詳しくは、『`stopServer` コマンド』を参照してください。

Linux 例えば、Linux システムで以下のコマンドを実行して、`server1` を停止します。

```
./stopServer.sh server1
```

セキュリティーが有効になっている場合、`stopServer` コマンドで `-username` および `-password` パラメーターを指定します。入力するユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーである必要があります。

Windows Windows システムで以下のコマンドを実行して、`server1` を停止します。

```
stopServer.bat server1
```

Windows オペレーティング・システム上では、セキュリティーが有効になっていても、サーバーが Windows サービスとして実行されていれば、`-username` および `-password` パラメーターを指定する必要はありません。この場合、パラメーターは、Windows サービスがシステムのシャットダウンに使用するスクリプトに自動的に渡されます。

注: サーバーはマイグレーション中に停止していても実行中でも構いません。サーバーを停止せずにマイグレーションできます。ただし、新しいサーバー (新しいバージョンの WebSphere Process Server を実行するサーバー) を始動するには、古いサーバーを停止する必要があります。また、サーバー構成をマイグレーションするのに、サーバーを稼働させておく必要はありません。マイグレーション・ツールはサーバーが停止していても、すべての構成データを取り出すことができます。

3. オプション: マイグレーション・ウィザード以外のツールを使用して、新しいバージョン 6.1 プロファイルを作成します。

マイグレーション・ウィザードでは、既存のバージョン 6.0.x プロファイルをベースとして新規プロファイルを作成したり、ターゲットのプロファイル用に既存のバージョン 6.1 プロファイルを使用したりすることができます。このような既存のバージョン 6.1 プロファイルを作成するには、マイグレーション・ウィザードを実行する前に、プロファイル管理ツールまたは `manageprofiles` コマンドを使用して、新しい (バージョン 6.1) プロファイルを作成します。ノード名、セル名、および拡張レベルがバージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用し

て、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。マイグレーション・ウィザードでは新規プロファイルを検出し、ターゲット・プロファイルの可能な選択項目としてプロファイルの名前を表示します。プロファイルの作成については、『プロファイルの作成』を参照してください。

4. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

5. マイグレーション・ウィザードを起動します。

マイグレーション・ウィザードは、以下のいずれかの方法で起動します。

- WebSphere Process Server のファースト・ステップ・コンソールから「マイグレーション・ウィザード」を選択する。
- `install_dir/bin` ディレクトリーに保存されている以下のスクリプトの 1 つ (これはオペレーティング・システムによって異なります) を実行する。

- a. **UNIX** **Linux** UNIX および Linux システムの場合:

`wbi_migration.sh`

- b. **Windows** Windows システムの場合: `wbi_migration.bat`

マイグレーション・ウィザードが実際に実行する処理については、9 ページの『マイグレーション・ウィザードの機能』を参照してください。

6. 11 ページの『マイグレーション・ウィザードの実行』で説明するように、マイグレーション・ウィザードのプロンプトに従います。
7. 必要に応じて、WebSphere Process Server で使用されているデータベースを手動で更新します。通常では、新しいバージョンの WebSphere Process Server で必要なデータベースの変更は、自動的に適用されます。サーバーが最初に始動されるときに、データベース・テーブルが新規スキーマ・バージョンにマイグレーションされます。ただし、サーバーがデータベース・スキーマにアクセスするための十分な権限を持たないか、その他のデータベース固有の要件が満たされてい

ない場合、データベースを手動で更新する必要があります。詳しくは、32 ページの『マイグレーションのためのデータベースのアップグレード』を参照してください。

結果

これでスタンドアロン・サーバーをマイグレーションできました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要がある追加タスクに注意してください。

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

9 ページの『マイグレーション・ウィザード』

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連タスク

11 ページの『マイグレーション・ウィザードの実行』

 コマンド行ツールを使用したスタンドアロン・サーバーのマイグレーション

コマンド行ツールを使用して、スタンドアロン・サーバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

 プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、manageprofiles コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』
WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

111 ページの『事後マイグレーション構成検査』

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

32 ページの『マイグレーションのためのデータベースのアップグレード』

マイグレーションと同時に、一部の WebSphere Process Server コンポーネントのデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

99 ページの『リモート・システムへのマイグレーション』

マイグレーション・ツールを使用して、あるシステムの古いバージョンから、別のリモート・システム上の新しいバージョンの WebSphere Process Server にマイグレーションします。(スタンドアロン・サーバーのみ。)

105 ページの『サポート対象外となったオペレーティング・システムからのマイグレーション』

マイグレーション・ツールを使用して、新しいバージョンでサポートされないオペレーティング・システム上で実行中の以前の WebSphere Process Server リリースをマイグレーションします。(スタンドアロン・サーバーのみ。)

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

関連情報

backupConfig コマンド

stopServer コマンド

コマンド行ツールを使用したスタンドアロン・サーバーのマイグレーション

コマンド行ツールを使用して、スタンドアロン・サーバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。

注: 新しい物理コンピューター・システムにマイグレーションする場合、99 ページの『リモート・システムへのマイグレーション』で説明されている代替の手順を使用します。 WebSphere Process Server を実行しているコンピューター上のオペレーティング・システムのバージョンをアップグレードする場合は、105 ページの『サポート対象外となったオペレーティング・システムからのマイグレーション』で説明されている代替の手順を使用します。

- 古い WebSphere Process Server バージョンで作成された統合されていないプロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、backupConfig コマンドまたは望ましいバックアップ・ユーティリティーを使用して、既存の構成をバックアップします。『backupConfig コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

プロシージャー

1. Linux または UNIX システムでは root として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション対象のノード上でバージョン 6.0.x サーバーが実行中の場合、そのサーバーを停止します。影響を受けるサーバーのプロファイルの `profile_dir/bin` ディレクトリーで `stopServer` コマンドを使用するか、プロファイルのファースト・ステップ・コンソールでサーバーを停止します。

`stopServer` コマンドについて詳しくは、『`stopServer` コマンド』を参照してください。

Linux 例えば、Linux システムで以下のコマンドを実行して、`server1` を停止します。

```
./stopServer.sh server1
```

セキュリティーが有効になっている場合、`stopServer` コマンドで `-username` および `-password` パラメーターを指定します。入力するユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーである必要があります。

Windows システムで以下のコマンドを実行して、server1 を停止します。

```
stopServer.bat server1
```

Windows オペレーティング・システム上では、セキュリティーが有効になっていても、サーバーが Windows サービスとして実行されていれば、-username および -password パラメーターを指定する必要はありません。この場合、パラメーターは、Windows サービスがシステムのシャットダウンに使用するスクリプトに自動的に渡されます。

注: サーバーはマイグレーション中に停止していても実行中でも構いません。サーバーを停止せずにマイグレーションできます。ただし、新しいサーバー (新しいバージョンの WebSphere Process Server を実行するサーバー) を始動するには、古いサーバーを停止する必要があります。また、サーバー構成をマイグレーションするのに、サーバーを稼働させておく必要はありません。マイグレーション・ツールはサーバーが停止していても、すべての構成データを取り出すことができます。

3. プロファイル管理ツールまたは **manageprofiles** コマンドを使用して、新しいバージョンの WebSphere Process Server で新規プロファイルを作成します。 ノード名、セル名、および拡張レベルが バージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。プロファイルの作成については、『プロファイルの作成』を参照してください。
4. マイグレーション・バックアップ・ディレクトリー名と既存の WebSphere Process Server ディレクトリー名を指定して、WBIPreUpgrade コマンドを実行します。 WBIPreUpgrade ツールは、*install_root* および *profile_root* ディレクトリーから、指定したバックアップ・ディレクトリーに、選択されたファイルを保存します。詳しくは、14 ページの『WBIPreUpgrade コマンド』を参照してください。
5. マイグレーション・バックアップ・ディレクトリーを指定して、WBIPostUpgrade コマンドを実行します。 WBIPostUpgrade ツールによって、バックアップ・ディレクトリー内の環境が、新しい WebSphere Process Server スタンドアロン・サーバー・インストール済み環境にリストアされます。詳しくは、17 ページの『WBIPostUpgrade コマンド』を参照してください。
6. 必要に応じて、WebSphere Process Server で使用されているデータベースを手動で更新します。通常では、新しいバージョンの WebSphere Process Server で必要なデータベースの変更は、自動的に適用されます。サーバーが最初に始動されるときに、データベース・テーブルが新規スキーマ・バージョンにマイグレーションされます。ただし、サーバーがデータベース・スキーマにアクセスするための十分な権限を持たないか、その他のデータベース固有の要件が満たされていない場合、データベースを手動で更新する必要があります。詳しくは、32 ページの『マイグレーションのためのデータベースのアップグレード』を参照してください。

結果

これでスタンドアロン・サーバーをマイグレーションできました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要がある追加タスクに注意してください。

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

9 ページの『マイグレーション・ウィザード』

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連タスク

41 ページの『マイグレーション・ウィザードを使用したスタンドアロン・サーバーのマイグレーション』

マイグレーション・ウィザードを使用して、スタンドアロン・サーバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、manageprofiles コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』

WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

111 ページの『事後マイグレーション構成検査』

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

32 ページの『マイグレーションのためのデータベースのアップグレード』
マイグレーションと同時に、一部の WebSphere Process Server コンポーネント
のデータベース・スキーマをアップグレードする必要があります。これは自動的に
実行させることができますが、場合によっては手動でスキーマをアップグレード
しなければなりません。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインス
トールされたバージョンの WebSphere Process Server の構成をマイグレーショ
ン固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマ
ンドによって作成された保存済み構成を、指定された *backupDirectory* から取り
出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンド
は、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere
Process Server にマイグレーションし、マイグレーションされたすべてのアプリ
ケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに
追加します。

関連情報

backupConfig コマンド

stopServer コマンド

Network Deployment 環境のマイグレーション

WebSphere Process Server Network Deployment 環境をマイグレーションします。

始める前に

1 ページの『マイグレーションの概要』および 3 ページの『事前マイグレーション
の考慮事項』を参照してください。

このタスクについて

Network Deployment 環境をマイグレーションするには、デプロイメント・マネー
ジャーを先にマイグレーションしてから、その管理対象ノードをマイグレーションす
る必要があります。

WebSphere Process Server バージョン 6.1 デプロイメント・セルにマイグレーショ
ンする方法についての情報を参照するには、適切なマイグレーション・シナリオを
選択してください。

注: クラスター環境内にはない管理対象ノードについては、60 ページの『非クラスタ
ー管理対象ノードのマイグレーション』を参照してください。クラスター環境内の
管理対象ノードについては、75 ページの『クラスターのマイグレーション』を参照
してください。

ヒント: マイグレーションするときの問題のトラブルシューティングのヘルプにつ
いては、134 ページの『バージョン間のマイグレーションのトラブルシューティン
グ』を参照してください。

デプロイメント・マネージャーのマイグレーション

WebSphere Process Server デプロイメント・マネージャーを、ニーズに応じていくつかの方法から選択してマイグレーションします。

始める前に

1 ページの『マイグレーションの概要』および3 ページの『事前マイグレーションの考慮事項』を参照してください。

WebSphere Process Server デプロイメント・マネージャーを、WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションする方法についての情報を参照するには、適切なマイグレーション・シナリオを選択してください。(例えば、WebSphere Process Server バージョン 6.0.2 デプロイメント・マネージャーから WebSphere Process Server バージョン 6.1 デプロイメント・マネージャーへのマイグレーション。)

- 『マイグレーション・ウィザードを使用したデプロイメント・マネージャーのマイグレーション』

このトピックには、古いバージョンの WebSphere Process Server デプロイメント・マネージャーから新しいバージョンのデプロイメント・マネージャーに、マイグレーション・ウィザード (グラフィカル・ユーザー・インターフェース) を使用してマイグレーションするための説明が記載されています。

- 56 ページの『コマンド行ツールを使用したデプロイメント・マネージャーのマイグレーション』

このトピックには、古いバージョンの WebSphere Process Server デプロイメント・マネージャーから新しいバージョンのデプロイメント・マネージャーに、マイグレーション・ウィザードでなくマイグレーション・コマンド行ツールを使用してマイグレーションするための説明が記載されています。

ヒント: マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

関連概念

75 ページの『クラスターのマイグレーション』

クラスターのマイグレーションは、特別な手順に従って、クラスター・メンバーを含む各プロファイルを順番にマイグレーションすることによって行います。クラスター・サービスのダウン時間を最小限に抑えるには、追加のステップを実行してください。

マイグレーション・ウィザードを使用したデプロイメント・マネージャーのマイグレーション

マイグレーション・ウィザードを使用して、デプロイメント・マネージャーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

注: セルを構成する管理対象ノードをマイグレーションする前に、WebSphere Process Server バージョン 6.0.x デプロイメント・マネージャーをバージョン 6.1 にマイグレーションします。デプロイメント・マネージャーは、セル内のすべてのノードを管理するために、常にセル内で最新のリリース・レベルおよびフィックス・レベルである必要があります。バージョン 6.1 デプロイメント・マネージャーは、バージョン 6.0.1、バージョン 6.0.2、およびバージョン 6.1 の管理対象ノードを管理できます。これによって、セル内で実行中のアプリケーションに及ぼす影響を最小限に抑えながら、セルでは一度に 1 ノードずつ新規リリースにアップグレードできます。リリースが混在したセルを使用する上での制限については、『他の WebSphere 製品のインストール済み環境との共存』を参照してください。

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。
- 古い WebSphere Process Server バージョンで作成されたデプロイメント・マネージャー・プロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件について詳しくは、3 ページの『事前マイグレーションの考慮事項』を参照してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、backupConfig コマンドまたは望ましいバックアップ・ユーティリティーを使用して、既存の構成をバックアップします。『backupConfig コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

プロシージャ

1. Linux または UNIX システムでは root として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

セル名 マイグレーションしようとしているデプロイメント・マネージャーによって管理されているセルの名前。新バージョンのセル名は、旧バージョンの構成におけるセル名に一致している必要があります。

ノード名

マイグレーションするノードの名前。新バージョンのノード名は、旧バージョンの構成における名前一致している必要があります。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

3. オプション: マイグレーション・ウィザード以外のツールを使用して、新しいバージョン 6.1 プロファイルを作成します。

マイグレーション・ウィザードでは、既存のバージョン 6.0.x プロファイルをベースとして新規プロファイルを作成したり、ターゲットのプロファイル用に既存のバージョン 6.1 プロファイルを使用したりすることができます。このような既存のバージョン 6.1 プロファイルを作成するには、マイグレーション・ウィザードを実行する前に、プロファイル管理ツールまたは `manageprofiles` コマンドを使用して、新しい (バージョン 6.1) プロファイルを作成します。ノード名、セル名、および拡張レベルがバージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。マイグレーション・ウィザードでは新規プロファイルを検出し、ターゲット・プロファイルの可能な選択項目としてプロファイルの名前を表示します。プロファイルの作成については、『プロファイルの作成』を参照してください。

4. マイグレーションしようとするデプロイメント・マネージャーを停止します。デプロイメント・マネージャーの `profile_dir/bin` ディレクトリーか、またはデプロイメント・マネージャーのファースト・ステップ・コンソールから、**stopManager** コマンドを使用します。

`stopManager` コマンドについて詳しくは、`stopManager` を参照してください。

Linux 例えば、Linux プラットフォームでは以下のコマンドを使用します。

```
./stopManager.sh
```

セキュリティーを有効に設定した場合、コマンドの `-username` および `-password` パラメーターを指定します。

デプロイメント・マネージャーが実行中でも停止中でも、マイグレーションできます。いずれの状態でも、マイグレーション・ツールはすべての構成データを取

り出すことができます。ただし、インストールする新しいバージョンのデプロイメント・マネージャーを開始できるようにするには、古いバージョンのデプロイメント・マネージャーを先に停止する必要があるので、ここで停止するのが妥当です。

5. マイグレーション先の WebSphere Process Server からマイグレーション・ウィザードを起動します。

マイグレーション・ウィザードは、以下のいずれかの方法で起動します。

- WebSphere Process Server のファースト・ステップ・コンソールから「**マイグレーション・ウィザード**」を選択する。
- `install_dir/bin` ディレクトリに保存されている以下のスクリプトの 1 つ (これはオペレーティング・システムによって異なります) を実行する。
 - a.   UNIX および Linux システムの場合:
`wbi_migration.sh`
 - b.  Windows システムの場合: `wbi_migration.bat`

マイグレーション・ウィザードが実際に実行する処理については、9 ページの『マイグレーション・ウィザードの機能』を参照してください。

6. 11 ページの『マイグレーション・ウィザードの実行』で説明するように、マイグレーション・ウィザードのプロンプトに従います。
7. 共通データベースを手動で更新する必要がある場合は、ここで実行します。詳しくは、33 ページの『手動による共通データベースのアップグレード』を参照してください。通常では、新しいバージョンの WebSphere Process Server で必要なデータベースの変更は、自動的に適用されます。サーバーが最初に始動されるときに、データベース・テーブルが新規スキーマ・バージョンにマイグレーションされます。ただし、サーバーがデータベース・スキーマにアクセスするための十分な権限を持たないか、その他のデータベース固有の要件が満たされていない場合、データベースを手動で更新する必要があります。

注: サーバーまたはクラスターで Business Process Choreographer が構成されている場合、デプロイメント・マネージャーのマイグレーション時ではなく、サーバーまたはクラスターをマイグレーションした後に、データベースを更新してください。

8. バージョン 6.1 デプロイメント・マネージャーを始動します。 `profile_dir/bin` ディレクトリまたはファースト・ステップ・コンソールから `startManager` コマンドを使用します。`startManager` コマンドについて詳しくは、`startManager` コマンドを参照してください。
9. オプション: バージョン 6.0.x デプロイメント・マネージャーをアンインストールします。

このステップを実行するのは、削除予定のデプロイメント・マネージャーの構成を正しくマイグレーションしたことを確認できた後にしてください。アンインストールについて詳しくは、『ソフトウェアのアンインストール』を参照してください。

結果

デプロイメント・マネージャーはマイグレーションされました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要がある追加タスクに注意してください。

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

9 ページの『マイグレーション・ウィザード』

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連タスク

11 ページの『マイグレーション・ウィザードの実行』

 コマンド行ツールを使用したデプロイメント・マネージャーのマイグレーション

コマンド行ツールを使用して、デプロイメント・マネージャーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

60 ページの『非クラスター管理対象ノードのマイグレーション』

WebSphere Process Server 管理対象ノードを、ニーズに応じていくつかの方法から選択してマイグレーションします。

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』

WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

111 ページの『事後マイグレーション構成検査』

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

32 ページの『マイグレーションのためのデータベースのアップグレード』

マイグレーションと同時に、一部の WebSphere Process Server コンポーネント

のデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、`manageprofiles` コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

112 ページの『デプロイメント・セルのロールバック』

restoreConfig および **wsadmin** コマンドを使用して、マイグレーション済みの WebSphere Process Server バージョン 6.1 デプロイメント・セルを、バージョン 6.0.x にロールバックすることができます。これによって、構成はマイグレーション前の状態に戻ります。デプロイメント・セルをロールバックした後、マイグレーション・プロセスを再開できます。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

関連情報

`backupConfig` コマンド

`stopServer` コマンド

ソフトウェアのアンインストール

IBM WebSphere Process Server をアンインストールするためのさまざまな方法について説明します。

他の WebSphere 製品のインストール済み環境との『共存』

WebSphere Process Server バージョン 6.1 のインストール済み環境は、WebSphere Process Server または WebSphere Enterprise Service Bus のすべてのバージョンのインストール済み環境、および特定の WebSphere 製品の一部のバージョンと同じシステム上で共存することができます。

コマンド行ツールを使用したデプロイメント・マネージャーのマイグレーション

コマンド行ツールを使用して、デプロイメント・マネージャーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

注: セルを構成する管理対象ノードをマイグレーションする前に、WebSphere Process Server バージョン 6.0.x デプロイメント・マネージャーを バージョン 6.1 にマイグレーションします。デプロイメント・マネージャーは、セル内のすべてのノードを管理するために、常にセル内で最新のリリース・レベルおよびフィックス・レベルである必要があります。バージョン 6.1 デプロイメント・マネージャーは、バージョン 6.0.1、バージョン 6.0.2、およびバージョン 6.1 の管理対象ノードを管理できます。これによって、セル内で実行中のアプリケーションに及ぼす影響を最小限に抑えながら、セルでは一度に 1 ノードずつ新規リリースにアップグレードできます。リリースが混在したセルを使用する上での制限については、『他の WebSphere 製品のインストール済み環境との共存』を参照してください。

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。
- 古い WebSphere Process Server バージョンで作成されたデプロイメント・マネージャー・プロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件について詳しくは、3 ページの『事前マイグレーションの考慮事項』を参照してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、backupConfig コマンドまたは望ましいバックアップ・ユーティリティーを使用して、既存の構成をバックアップします。『backupConfig コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

プロシージャ

1. Linux または UNIX システムでは root として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

セル名 マイグレーションしようとしているデプロイメント・マネージャーによ

って管理されているセルの名前。新バージョンのセル名は、旧バージョンの構成におけるセル名に一致している必要があります。

ノード名

マイグレーションするノードの名前。新バージョンのノード名は、旧バージョンの構成における名前に一致している必要があります。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

3. プロファイル管理ツールまたは **manageprofiles** コマンドを使用して、新しいバージョンの WebSphere Process Server で新規プロファイルを作成します。ノード名、セル名、および拡張レベルがバージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。プロファイルの作成については、『プロファイルの作成』を参照してください。
4. マイグレーションしようとするデプロイメント・マネージャーを停止します。デプロイメント・マネージャーの *profile_dir/bin* ディレクトリーか、またはデプロイメント・マネージャーのファースト・ステップ・コンソールから、**stopManager** コマンドを使用します。

stopManager コマンドについて詳しくは、**stopManager** を参照してください。

Linux

例えば、Linux プラットフォームでは以下のコマンドを使用します。

```
./stopManager.sh
```

セキュリティーを有効に設定した場合、コマンドの **-username** および **-password** パラメーターを指定します。

デプロイメント・マネージャーが実行中でも停止中でも、マイグレーションできます。いずれの状態でも、マイグレーション・ツールはすべての構成データを取り出すことができます。ただし、インストールする新しいバージョンのデプロイ

メント・マネージャーを開始できるようにするには、古いバージョンのデプロイメント・マネージャーを先に停止する必要があるので、ここで停止するのが妥当です。

5. マイグレーション・バックアップ・ディレクトリー名と既存の WebSphere Process Server ディレクトリー名を指定して、WBIPreUpgrade コマンドを実行します。WBIPreUpgrade ツールは、*install_root* および *profile_root* ディレクトリーから、指定したバックアップ・ディレクトリーに、選択されたファイルを保存します。詳しくは、14 ページの『WBIPreUpgrade コマンド』を参照してください。
6. マイグレーション・バックアップ・ディレクトリーを指定して、WBIPostUpgrade コマンドを実行します。WBIPostUpgrade ツールによって、バックアップ・ディレクトリー内の環境が、新しい WebSphere Process Server スタンドアロン・サーバー・インストール済み環境にリストアされます。詳しくは、17 ページの『WBIPostUpgrade コマンド』を参照してください。
7. 共通データベースを手動で更新する必要がある場合は、ここで実行します。詳しくは、33 ページの『手動による共通データベースのアップグレード』を参照してください。通常では、新しいバージョンの WebSphere Process Server で必要なデータベースの変更は、自動的に適用されます。サーバーが最初に始動されるときに、データベース・テーブルが新規スキーマ・バージョンにマイグレーションされます。ただし、サーバーがデータベース・スキーマにアクセスするための十分な権限を持たないか、その他のデータベース固有の要件が満たされていない場合、データベースを手動で更新する必要があります。

注: サーバーまたはクラスターで Business Process Choreographer が構成されている場合、デプロイメント・マネージャーのマイグレーション時ではなく、サーバーまたはクラスターをマイグレーションした後に、データベースを更新してください。

8. バージョン 6.1 デプロイメント・マネージャーを始動します。*profile_dir/bin* ディレクトリーまたはファースト・ステップ・コンソールから *startManager* コマンドを使用します。*startManager* コマンドについて詳しくは、*startManager* コマンドを参照してください。
9. オプション: バージョン 6.0.x デプロイメント・マネージャーをアンインストールします。

このステップを実行するのは、削除予定のデプロイメント・マネージャーの構成を正しくマイグレーションしたことを確認できた後にしてください。アンインストールについて詳しくは、『ソフトウェアのアンインストール』を参照してください。

結果

デプロイメント・マネージャーはマイグレーションされました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process

Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要がある追加タスクに注意してください。

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

9 ページの『マイグレーション・ウィザード』

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連タスク

50 ページの『マイグレーション・ウィザードを使用したデプロイメント・マネージャーのマイグレーション』

マイグレーション・ウィザードを使用して、デプロイメント・マネージャーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

60 ページの『非クラスター管理対象ノードのマイグレーション』

WebSphere Process Server 管理対象ノードを、ニーズに応じていくつかの方法から選択してマイグレーションします。

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』

WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

111 ページの『事後マイグレーション構成検査』

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

32 ページの『マイグレーションのためのデータベースのアップグレード』

マイグレーションと同時に、一部の WebSphere Process Server コンポーネントのデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、

manageprofiles コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

112 ページの『デプロイメント・セルのロールバック』

restoreConfig および **wsadmin** コマンドを使用して、マイグレーション済みの WebSphere Process Server バージョン 6.1 デプロイメント・セルを、バージョン 6.0.x にロールバックすることができます。これによって、構成はマイグレーション前の状態に戻ります。デプロイメント・セルをロールバックした後、マイグレーション・プロセスを再開できます。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

関連情報

backupConfig コマンド

stopServer コマンド



ソフトウェアのアンインストール

IBM WebSphere Process Server をアンインストールするためのさまざまな方法について説明します。



他の WebSphere 製品のインストール済み環境との『共存』

WebSphere Process Server バージョン 6.1 のインストール済み環境は、WebSphere Process Server または WebSphere Enterprise Service Bus のすべてのバージョンのインストール済み環境、および特定の WebSphere 製品の一部のバージョンと同じシステム上で共存することができます。

非クラスター管理対象ノードのマイグレーション

WebSphere Process Server 管理対象ノードを、ニーズに応じていくつかの方法から選択してマイグレーションします。

始める前に

1 ページの『マイグレーションの概要』および 3 ページの『事前マイグレーションの考慮事項』を参照してください。

WebSphere Process Server 非クラスター管理対象ノードを、 WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションする方法については、該当するサブトピックを選択してください。

注: 管理対象ノードがクラスターに属する場合は、 75 ページの『クラスターのマイグレーション』の説明に従ってください。

ヒント: マイグレーションするときの問題のトラブルシューティングのヘルプについては、 134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

関連タスク

75 ページの『クラスターのマイグレーション』

クラスターをマイグレーションするには、そのクラスターのメンバーを含む各プロファイルを一度に 1 つずつマイグレーションします。マイグレーションには、非クラスター環境には不要な追加のステップが必要になります。

87 ページの『最小限のダウン時間でクラスターのマイグレーション』

ダウン時間を最小限にしてクラスターをマイグレーションするには、クラスターに寄与するプロファイルの約半分を最初にマイグレーションしてから、残りの半分をマイグレーションします。最初のプロファイルのセットをマイグレーションした後に、クラスター・マイグレーションに必要な追加のステップを実行します。

50 ページの『マイグレーション・ウィザードを使用したデプロイメント・マネージャーのマイグレーション』

マイグレーション・ウィザードを使用して、デプロイメント・マネージャーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

 コマンド行ツールを使用したデプロイメント・マネージャーのマイグレーション

コマンド行ツールを使用して、デプロイメント・マネージャーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

マイグレーション・ウィザードを使用した非クラスター管理対象ノードのマイグレーション

マイグレーション・ウィザードを使用して、非クラスター管理対象ノードを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。

- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。
- 古いバージョンの WebSphere Process Server で作成された統合プロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。
- マイグレーションしようとする管理対象ノードを管理するデプロイメント・マネージャーが、既に新しいバージョンの WebSphere Process Server にマイグレーションされていて、実行中です。

注: WebSphere Process Server バージョン 6.0.x 管理対象ノードをバージョン 6.1 管理対象ノードにマイグレーションするには、まずバージョン 6.0.x デプロイメント・マネージャーをバージョン 6.1 デプロイメント・マネージャーにマイグレーションする必要があります。詳しくは、50 ページの『デプロイメント・マネージャーのマイグレーション』を参照してください。このトピックの指示に従って続行する前に、デプロイメント・マネージャーのマイグレーションを完了してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、`backupConfig` コマンドまたは望ましいバックアップ・ユーティリティーを使用して、管理対象ノードの構成をバックアップします。『`backupConfig` コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

このタスクについて

古いバージョンのデプロイメント・マネージャーを新しいバージョンの WebSphere Process Server にマイグレーションすれば、新しいバージョンのデプロイメント・マネージャーはデフォルトでは互換モードで実行され、このとき新旧両方のバージョンの WebSphere Process Server を管理できます。例えばマイグレーションの後で、バージョン 6.1 デプロイメント・マネージャーはバージョン 6.0.x とバージョン 6.1 のリリースのノードの両方を管理できます。バージョン 6.0.x デプロイメント・マネージャーの管理対象ノードは、バージョン 6.0.x 管理対象ノードとしてバージョン 6.1 デプロイメント・マネージャー内で実行されています。リリースが混在したセルを使用する上での制限については、他の WebSphere 製品のインストール済み環境との『共存』を参照してください。

バージョン 6.0.x WebSphere Process Server 管理対象ノード (バージョン 6.1 デプロイメント・マネージャーによって管理されるサーバー) を、バージョン 6.1 管理対象ノードにそれぞれ順次マイグレーションしていきます。すべてのバージョン 6.0.x 管理対象ノードのマイグレーションが終わったら、`convertScriptCompatibility` スクリプトを使用して、バージョン 6.0.x 管理スクリプトの後方互換性をサポート

するノードから、バージョン 6.1 のみをサポートするノードに、デプロイメント・マネージャーを変更します。 `convertScriptCompatibility` コマンドを参照してください。

注: このリンクの指示に従って、`convertScriptCompatibility` コマンドを使用するとき、`WASPostUpgrade` コマンドではなく `WBIPostUpgrade` コマンドを使用してください。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

プロシージャ

1. Linux または UNIX システムでは `root` として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション対象のノード上でバージョン 6.0.x サーバーが実行中の場合、そのサーバーを停止します。影響を受けるサーバーのプロファイルの `profile_dir/bin` ディレクトリで `stopServer` コマンドを使用するか、プロファイルのファースト・ステップ・コンソールでサーバーを停止します。

`stopServer` コマンドについて詳しくは、『`stopServer` コマンド』を参照してください。

Linux 例えば、Linux システムで以下のコマンドを実行して、`server1` を停止します。

```
./stopServer.sh server1
```

セキュリティーが有効になっている場合、`stopServer` コマンドで `-username` および `-password` パラメーターを指定します。入力するユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーである必要があります。

Windows Windows システムで以下のコマンドを実行して、`server1` を停止します。

```
stopServer.bat server1
```

Windows オペレーティング・システム上では、セキュリティーが有効になっている場合、サーバーが Windows サービスとして実行されていれば、`-username` および `-password` パラメーターを指定する必要はありません。この場合、パラメーターは、Windows サービスがシステムのシャットダウンに使用するスクリプトに自動的に渡されます。

注: サーバーはマイグレーション中に停止していても実行中でも構いません。サーバーを停止せずにマイグレーションできます。ただし、新しいサーバー (新しいバージョンの WebSphere Process Server を実行するサーバー) を始動するには、古いサーバーを停止する必要があります。また、サーバー構成をマイグレーションするのに、サーバーを稼働させておく必要はありません。マイグレーション・ツールはサーバーが停止していても、すべての構成データを取り出すことができます。

3. マイグレーション対象のノードのノード・エージェントを停止します。プラットフォームに応じて、以下のいずれかのコマンドを実行し、ノード・エージェント・プロセスを停止します (ここで、*profile_root* は統合ノードのインストール・ディレクトリーを表します)。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:

```
profile_root/bin/stopNode.sh
```

- **Windows** **Windows** プラットフォームの場合:

```
profile_root%bin%stopNode.bat
```

セキュリティーが使用可能な場合は、代わりに以下のコマンドのいずれかを使用します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:

```
profile_root/bin/stopNode.sh -username user_ID -password password
```

- **Windows** **Windows** プラットフォームの場合:

```
profile_root%bin%stopNode.bat -username user_ID -password password
```

注: ノードはマイグレーション中に停止していても実行中でも構いません。管理対象ノードは停止せずに (ノード・エージェントを停止せずに) マイグレーションできます。ただし、新しいノード (新しいバージョンの WebSphere Process Server を実行するノード) を始動するには、古いノードを停止する必要があります。また、ノード構成をマイグレーションするのに、ノードを稼働させておく必要はありません。マイグレーション・ツールはノードが停止していても、すべての構成データを取り出すことができます。

4. オプション: マイグレーション・ウィザード以外のツールを使用して、新しいバージョン 6.1 プロファイルを作成します。

バージョン 6.1 スタンドアロン・サーバー・プロファイルまたはカスタム・プロファイルのいずれかをターゲットとして作成できます。カスタム・プロファイルを作成する場合、マイグレーション前にノードを統合しないでください。マイグレーション・ツールがマイグレーション中にノードをフェデレートします。

注: マイグレーションが正常に行われるためには、バージョン 6.0.x からバージョン 6.1 にマイグレーションする各ノードについて、同じノード名とセル名を使用する必要があります。

ヒント: マイグレーションする前に、仮想ホスト情報の変更などのセル・レベルの変更を新しいバージョン 6.1 ノードに行った場合、これらの変更はマイグレーション中に失われます。したがって、ノードがマイグレーションされるまで待つから、このような変更を行う必要があります。そうしない場合、仮想ホストおよびホスト別名情報への変更などの新しいセルへのすべての変更を、デプロイメント・マネージャー上で実行される管理コンソールを使用して、マイグレーション後に手動で再び行う必要があります。

5. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

- バージョン 6.1 デプロイメント・マネージャーが稼働中であるか確認してください。

注: ノードが実行中でも停止中でも、バージョン 6.0.x ノードはマイグレーションできます。いずれの状態でも、マイグレーション・ツールはすべての構成データを取り出すことができます。ただし、インストールするバージョン 6.1 ノードを開始するには、バージョン 6.0.x ノードを先に停止する必要があるため、ここで停止するのが妥当です。

- マイグレーション・ウィザードを起動します。

マイグレーション・ウィザードは、以下のいずれかの方法で起動します。

- WebSphere Process Server のファースト・ステップ・コンソールから「マイグレーション・ウィザード」を選択する。
- `install_dir/bin` ディレクトリーに保存されている以下のスクリプトの 1 つ (これはオペレーティング・システムによって異なります) を実行する。

-   UNIX および Linux システムの場合:

`wbi_migration.sh`

-  Windows システムの場合: `wbi_migration.bat`

マイグレーション・ウィザードは、構成およびアプリケーションをバージョン 6.0.x 管理対象ノードからバージョン 6.1 管理対象ノードにコピーします。すべてのデータのマイグレーションが終わったら、ウィザードはバージョン 6.1 管理対象ノードをデプロイメント・マネージャー・セルに統合します。

- サーバーおよびノード・エージェントを停止します (まだ停止していない場合)。サーバーが停止していない場合、ステップ 2 で説明するようにサーバーを停止します。ノード・エージェントが停止していない場合、ステップ 3 で説明するようにノード・エージェントを停止します。
- Business Process Choreographer が構成されているサーバーをマイグレーションしており、その関連データベースを手動でアップグレードする必要がある場合

は、このステップをここで実行してください。36 ページの『手動による Business Process Choreographer データベースのアップグレード』を参照してください。

10. ノード・エージェントを再始動します。ノード・エージェントを始動するには、`profile_root%bin%startNode` コマンドを実行します (ここで、`profile_root` は管理対象ノードのインストール・ディレクトリーを表します)。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:

```
profile_root/bin/startNode.sh
```

- **Windows** **Windows** プラットフォームの場合:

```
profile_root%bin%startNode.bat
```

11. このノード上で実行中のサーバーを始動します。startServer コマンド、管理コンソール、またはプロファイルのファースト・ステップ・コンソールを使用して、各サーバーを始動します。詳しくは、『アプリケーション・サーバーの始動』を参照してください。
12. マイグレーションする追加の各管理対象ノードについては、ステップ 1 から 11 を繰り返します。
13. 互換性オプション (デフォルト) を選択していて、すべてのノードが完全に WebSphere Process Server バージョン 6.1 にマイグレーションされたら、convertScriptCompatibility スクリプトを実行して、バージョン 6.1 デプロイメント・マネージャーから後方互換性を除去します。

convertScriptCompatibility コマンドを bin ディレクトリーから発行します。

- **UNIX** **Linux** `install_root/bin/convertScriptCompatibility.sh`

- **Windows** `install_root%bin%convertScriptCompatibility.bat`

convertScriptCompatibility コマンドを参照してください。

結果

非クラスター管理対象ノードはマイグレーションされました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

9 ページの『マイグレーション・ウィザード』

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連タスク

11 ページの『マイグレーション・ウィザードの実行』

コマンド行ツールを使用した管理対象ノードのマイグレーション

コマンド行ツールを使用して、非クラスター管理対象ノードを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

36 ページの『手動による Business Process Choreographer データベースのアップグレード』

Business Process Choreographer が構成されているサーバーをマイグレーションした後は、サーバーを始動する前に、関連付けられた Business Process Choreographer データベースのスキーマをアップグレードする必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければなりません。

32 ページの『マイグレーションのためのデータベースのアップグレード』
マイグレーションと同時に、一部の WebSphere Process Server コンポーネントのデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、`manageprofiles` コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

115 ページの『管理対象ノードのロールバック』

restoreConfig および **wsadmin** コマンドを使用して、マイグレーション済みの WebSphere Process Server バージョン 6.1 管理対象ノードを、マイグレーション前の状態にロールバックすることができます。ロールバックする各管理対象ノードに対して、管理対象ノードそれ自体と、デプロイメント・マネージャーにあるマスター・リポジトリに加えた対応する変更をロールバックする必要があります。

111 ページの『事後マイグレーション構成検査』

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマ

ンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の *WBIPostUpgrade* コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

関連情報

アプリケーション・サーバーの始動

convertScriptCompatibility コマンド

startManager コマンド

backupConfig コマンド

stopServer コマンド

コマンド行ツールを使用した非クラスター管理対象ノードのマイグレーション

コマンド行ツールを使用して、非クラスター管理対象ノードを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。
- 古いバージョンの WebSphere Process Server で作成された統合プロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。
- マイグレーションしようとする管理対象ノードを管理するデプロイメント・マネージャーが、既に新しいバージョンの WebSphere Process Server にマイグレーションされていて、実行中です。

注: WebSphere Process Server バージョン 6.0.x 管理対象ノードをバージョン 6.1 管理対象ノードにマイグレーションするには、まずバージョン 6.0.x デプロイメント・マネージャーをバージョン 6.1 デプロイメント・マネージャーにマイグレーションする必要があります。詳しくは、50 ページの『デプロイメント・マネージャーのマイグレーション』を参照してください。このトピックの指示に従って続行する前に、デプロイメント・マネージャーのマイグレーションを完了してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、`backupConfig` コマンドまたは望ましいバックアップ・ユーティリティーを使用して、管理対象ノードの構成をバックアップします。『`backupConfig` コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

このタスクについて

古いバージョンのデプロイメント・マネージャーを新しいバージョンの WebSphere Process Server にマイグレーションすれば、新しいバージョンのデプロイメント・マネージャーはデフォルトでは互換モードで実行され、このとき新旧両方のバージョンの WebSphere Process Server を管理できます。例えばマイグレーションの後で、バージョン 6.1 デプロイメント・マネージャーはバージョン 6.0.x とバージョン 6.1 のリリースのノードの両方を管理できます。バージョン 6.0.x デプロイメント・マネージャーの管理対象ノードは、バージョン 6.0.x 管理対象ノードとしてバージョン 6.1 デプロイメント・マネージャー内で実行されています。リリースが混在したセルを使用する上での制限については、他の WebSphere 製品のインストール済み環境との『共存』を参照してください。

バージョン 6.0.x WebSphere Process Server 管理対象ノード (バージョン 6.1 デプロイメント・マネージャーによって管理されるサーバー) を、バージョン 6.1 管理対象ノードにそれぞれ順次マイグレーションしていきます。すべてのバージョン 6.0.x 管理対象ノードのマイグレーションが終わったら、`convertScriptCompatibility` スクリプトを使用して、バージョン 6.0.x 管理スクリプトの後方互換性をサポートするノードから、バージョン 6.1 のみをサポートするノードに、デプロイメント・マネージャーを変更します。 `convertScriptCompatibility` コマンドを参照してください。

注: このリンクの指示に従って、`convertScriptCompatibility` コマンドを使用するとき、`WASPostUpgrade` コマンドではなく `WBIPPostUpgrade` コマンドを使用してください。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

プロシージャ

1. Linux または UNIX システムでは `root` として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション対象のノード上でバージョン 6.0.x サーバーが実行中の場合、そのサーバーを停止します。影響を受けるサーバーのプロファイルの `profile_dir/bin` ディレクトリで `stopServer` コマンドを使用するか、プロファイルのファースト・ステップ・コンソールでサーバーを停止します。

`stopServer` コマンドについて詳しくは、『`stopServer` コマンド』を参照してください。

Linux 例えば、Linux システムで以下のコマンドを実行して、server1 を停止します。

```
./stopServer.sh server1
```

セキュリティが有効になっている場合、stopServer コマンドで `-username` および `-password` パラメーターを指定します。入力するユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーである必要があります。

Windows Windows システムで以下のコマンドを実行して、server1 を停止します。

```
stopServer.bat server1
```

Windows オペレーティング・システム上では、セキュリティが有効になっていても、サーバーが Windows サービスとして実行されていれば、`-username` および `-password` パラメーターを指定する必要はありません。この場合、パラメーターは、Windows サービスがシステムのシャットダウンに使用するスクリプトに自動的に渡されます。

注: サーバーはマイグレーション中に停止していても実行中でも構いません。サーバーを停止せずにマイグレーションできます。ただし、新しいサーバー (新しいバージョンの WebSphere Process Server を実行するサーバー) を始動するには、古いサーバーを停止する必要があります。また、サーバー構成をマイグレーションするのに、サーバーを稼働させておく必要はありません。マイグレーション・ツールはサーバーが停止していても、すべての構成データを取り出すことができます。

3. マイグレーション対象のノードのノード・エージェントを停止します。プラットフォームに応じて、以下のいずれかのコマンドを実行し、ノード・エージェント・プロセスを停止します (ここで、`profile_root` は統合ノードのインストール・ディレクトリを表します)。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:
`profile_root/bin/stopNode.sh`

- **Windows** **Windows** プラットフォームの場合:
`profile_root%bin%stopNode.bat`

セキュリティが使用可能な場合は、代わりに以下のコマンドのいずれかを使用します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:
`profile_root/bin/stopNode.sh -username user_ID -password password`

- **Windows** **Windows** プラットフォームの場合:
`profile_root%bin%stopNode.bat -username user_ID -password password`

注: ノードはマイグレーション中に停止していても実行中でも構いません。管理対象ノードは停止せずに (ノード・エージェントを停止せずに) マイグレーションできます。ただし、新しいノード (新しいバージョンの WebSphere Process Server を実行するノード) を始動するには、古いノードを停止する必要があります。また、ノード構成をマイグレーションするのに、ノードを稼働させてお

く必要はありません。マイグレーション・ツールはノードが停止していても、すべての構成データを取り出すことができます。

4. 新規プロファイルを作成します。プロファイル管理ツールまたは `manageprofiles` コマンドを使用して、新しいバージョンの WebSphere Process Server で新規プロファイルを作成します。

バージョン 6.1 スタンドアロン・サーバー・プロファイルまたはカスタム・プロファイルのいずれかをターゲットとして作成できます。カスタム・プロファイルを作成する場合、マイグレーション前にノードを統合しないでください。マイグレーション・ツールがマイグレーション中にノードをフェデレートします。

ノード名、セル名、および拡張レベルがバージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。

注: マイグレーションが正常に行われるためには、バージョン 6.0.x からバージョン 6.1 にマイグレーションする各ノードについて、同じノード名とセル名を使用する必要があります。

ヒント: マイグレーションする前に、仮想ホスト情報の変更などのセル・レベルの変更を新しいバージョン 6.1 ノードに行った場合、これらの変更はマイグレーション中に失われます。したがって、ノードがマイグレーションされるまで待つから、このような変更を行う必要があります。そうしない場合、仮想ホストおよびホスト別名情報への変更などの新しいセルへのすべての変更を、デプロイメント・マネージャー上で実行される管理コンソールを使用して、マイグレーション後に手動で再び行う必要があります。

5. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

インストールのルート・ディレクトリー

`-currentWebSphereDirectory` パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

`-backupDirectory` パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

`-oldProfile` パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

`-profileName` パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

`-replacePorts` および `-portBlock` パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

- バージョン 6.1 デプロイメント・マネージャーが稼働中であるか確認してください。

注: ノードが実行中でも停止中でも、バージョン 6.0.x ノードはマイグレーションできます。いずれの状態でも、マイグレーション・ツールはすべての構成データを取り出すことができます。ただし、インストールするバージョン 6.1 ノードを開始するには、バージョン 6.0.x ノードを先に停止する必要があるため、ここで停止するのが妥当です。

- マイグレーション・バックアップ・ディレクトリー名と既存の WebSphere Process Server ディレクトリー名を指定して、WBIPreUpgrade コマンドを実行します。WBIPreUpgrade ツールは、`install_root` および `profile_root` ディレクトリーから、指定したバックアップ・ディレクトリーに、選択されたファイルを保存します。詳しくは、14 ページの『WBIPreUpgrade コマンド』を参照してください。
- マイグレーション・バックアップ・ディレクトリーを指定して、WBIPostUpgrade コマンドを実行します。WBIPostUpgrade ツールによって、バックアップ・ディレクトリー内の環境が、新しい WebSphere Process Server スタンドアロン・サーバー・インストール済み環境にリストアされます。詳しくは、17 ページの『WBIPostUpgrade コマンド』を参照してください。
- サーバーおよびノード・エージェントを停止します (まだ停止していない場合)。サーバーが停止していない場合、ステップ 2 で説明するようにサーバーを停止します。ノード・エージェントが停止していない場合、ステップ 3 で説明するようにノード・エージェントを停止します。
- Business Process Choreographer が構成されているサーバーをマイグレーションしており、その関連データベースを手動でアップグレードする必要がある場合は、このステップをここで実行してください。36 ページの『手動による Business Process Choreographer データベースのアップグレード』を参照してください。
- ノード・エージェントを再始動します。ノード・エージェントを始動するには、`profile_root\bin\startNode` コマンドを実行します (ここで、`profile_root` は管理対象ノードのインストール・ディレクトリーを表します)。

- Linux** **UNIX** Linux および UNIX プラットフォームの場合:

```
profile_root/bin/startNode.sh
```

- Windows** Windows プラットフォームの場合:

```
profile_root\bin\startNode.bat
```

- このノード上で実行中のサーバーを始動します。startServer コマンド、管理コンソール、またはプロファイルのファースト・ステップ・コンソールを使用して、各サーバーを始動します。詳しくは、『アプリケーション・サーバーの始動』を参照してください。
- マイグレーションする追加の管理対象ノードごとに、ステップ 1 から 12 を繰り返します。

注: ステップ 7 を再度実行 (WBIPreUpgrade を実行) する必要があるのは、最初に WBIPreUpgrade を実行した後にバージョン 6.0.x システムが再構成された場合のみです。

14. 互換性オプション (デフォルト) を選択していて、すべてのノードが完全に WebSphere Process Server バージョン 6.1 にマイグレーションされたら、convertScriptCompatibility スクリプトを実行して、バージョン 6.1 デプロイメント・マネージャーから後方互換性を除去します。

convertScriptCompatibility コマンドを bin ディレクトリーから発行します。

- **UNIX** **Linux** `install_root/bin/convertScriptCompatibility.sh`
- **Windows** `install_root\bin\convertScriptCompatibility.bat`

convertScriptCompatibility コマンドを参照してください。

結果

非クラスター管理対象ノードはマイグレーションされました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

9 ページの『マイグレーション・ウィザード』

バージョン間マイグレーション・ウィザードは、WebSphere Process Server の旧バージョンから新バージョンへのマイグレーションを支援するグラフィカル・インターフェースです。

関連タスク

61 ページの『マイグレーション・ウィザードを使用した非クラスター管理対象ノードのマイグレーション』

マイグレーション・ウィザードを使用して、非クラスター管理対象ノードを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

36 ページの『手動による Business Process Choreographer データベースのアップグレード』

Business Process Choreographer が構成されているサーバーをマイグレーションした後は、サーバーを始動する前に、関連付けられた Business Process Choreographer データベースのスキーマをアップグレードする必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければなりません。

32 ページの『マイグレーションのためのデータベースのアップグレード』

マイグレーションと同時に、一部の WebSphere Process Server コンポーネント

のデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、`manageprofiles` コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

115 ページの『管理対象ノードのロールバック』

restoreConfig および **wsadmin** コマンドを使用して、マイグレーション済みの WebSphere Process Server バージョン 6.1 管理対象ノードを、マイグレーション前の状態にロールバックすることができます。ロールバックする各管理対象ノードに対して、管理対象ノードそれ自体と、デプロイメント・マネージャーにあるマスター・リポジトリに加えた対応する変更をロールバックする必要があります。

111 ページの『事後マイグレーション構成検査』

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

関連情報

アプリケーション・サーバーの始動

`convertScriptCompatibility` コマンド

`startManager` コマンド

`backupConfig` コマンド

`stopServer` コマンド

クラスタのマイグレーション

クラスタのマイグレーションは、特別な手順に従って、クラスタ・メンバーを含む各プロファイルを順番にマイグレーションすることによって行います。クラスタ・サービスのダウン時間を最小限に抑えるには、追加のステップを実行してください。

クラスタをマイグレーションする場合は、各プロファイルをマイグレーションする前に、クラスタに含まれるすべてのノード・エージェントとサーバーを停止する必要があります。また、クラスタごとに WBIProfileUpgrade スクリプトを実行する必要があります。必要に応じて、以下のサブトピックの中から適当なものを選択してください。

関連タスク

50 ページの『デプロイメント・マネージャーのマイグレーション』

WebSphere Process Server デプロイメント・マネージャーを、ニーズに応じていくつかの方法から選択してマイグレーションします。

クラスタのマイグレーション

クラスタをマイグレーションするには、そのクラスタのメンバーを含む各プロファイルを一度に 1 つずつマイグレーションします。マイグレーションには、非クラスタ環境には不要な追加のステップが必要になります。

始める前に

注: クラスタ内で、バージョン 6.0.x メンバーとバージョン 6.1 メンバーを同時に稼働させることはできません。最初にバージョン 6.1 クラスタ・メンバーを開始する前に、すべてのバージョン 6.0.x クラスタ・メンバーを停止する必要があります。また、いったんバージョン 6.1 クラスタ・メンバーを開始したら、そのクラスタでバージョン 6.0.x クラスタ・メンバーを開始しないでください。

新しいバージョン (例えば、バージョン 6.1) へのマイグレーションを希望する古いバージョンの WebSphere Process Server (例えば、バージョン 6.0.x) 上で実行している、1 つ以上のクラスタを含む既存のセルが必要です。さらに、新しいバージョンの WebSphere Process Server がインストールされている必要があります。

このタスクについて

以下のステップに従うことで、新しいバージョンの WebSphere Process Server のクラスタ機能を保持できます。

プロシージャ

1. デプロイメント・マネージャーをマイグレーションします。 50 ページの『デプロイメント・マネージャーのマイグレーション』にリストされた一連の手順の 1 つに従い、このタスクを完了します。
2. 新しいデプロイメント・マネージャーが稼働していることを確認します。
3. 関係するプロファイルを識別します。
 - a. クラスタ・メンバーを含む古いバージョンのプロファイルを識別します。

- b. このプロファイルが関与する他のクラスターを識別します。つまり、このプロファイルで、他のクラスターのメンバーであるサーバーが定義されている場合、そのクラスターを識別します。
- c. ステップ3bで識別されたクラスターのいずれかに対してクラスター・メンバーを提供する、同じセルのすべての他のプロファイルを識別します。
- d. ステップ3cで識別されるプロファイルによって定義されるすべてのノード・エージェントおよびプロセス・サーバーを識別します。

ステップ 3c で識別されるすべてのプロファイルと、ステップ 3d で識別される対応するすべてのノード・エージェントおよびサーバーがマイグレーションに含まれます。

- 4. ステップ3dで識別した、すべてのノード・エージェントおよびサーバーを停止します。
- 5. ステップ3cで識別した各プロファイルを 1 つずつマイグレーションしますが、新規エージェントまたはサーバーを**始動しないでください**。そのためには、77 ページの『マイグレーション・ウィザードを使用したクラスター・メンバーのマイグレーション』または 82 ページの『コマンド行ツールを使用したクラスター・メンバーのマイグレーション』にリストされた手順の 1 つに従います。
- 6. WebSphere Process Server バージョン 6.1 デプロイメント・マネージャー・プロファイルをホストするシステムで、`install_dir/util` ディレクトリーにナビゲートします。このディレクトリーには `WBIPProfileUpgrade` スクリプトである `WBIPProfileUpgrade.ant` があります。
- 7. マイグレーション済みのプロファイルで定義された各クラスターについて、`WBIPProfileUpgrade` スクリプトを実行します。ステップ3bで定義された各クラスターについて `WBIPProfileUpgrade` を実行します。`WBIPProfileUpgrade` の実行についての説明は、22 ページの『`WBIPProfileUpgrade` スクリプト』を参照してください。
- 8. `Business Process Choreographer` が構成されているサーバーをマイグレーションしており、その関連データベースを手動でアップグレードする必要がある場合は、このステップをここで実行してください。36 ページの『手動による `Business Process Choreographer` データベースのアップグレード』を参照してください。
- 9. クラスターのメンバーである新規の (マイグレーション済みの) ノード・エージェントおよびサーバーをすべて始動します。

結果

クラスターは新しいバージョンのWebSphere Process Serverにマイグレーションされました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで `Business Process Choreographer` が構成されている場合は、132 ページの『`Business Process Choreographer` に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要のある追加タスクに注意してください。

関連タスク

50 ページの『デプロイメント・マネージャーのマイグレーション』

WebSphere Process Server デプロイメント・マネージャーを、ニーズに応じていくつかの方法から選択してマイグレーションします。

60 ページの『非クラスター管理対象ノードのマイグレーション』

WebSphere Process Server 管理対象ノードを、ニーズに応じていくつかの方法から選択してマイグレーションします。

87 ページの『最小限のダウン時間でのクラスターのマイグレーション』

ダウン時間を最小限にしてクラスターをマイグレーションするには、クラスターに寄与するプロファイルの約半分を最初にマイグレーションしてから、残りの半分をマイグレーションします。最初のプロファイルのセットをマイグレーションした後に、クラスター・マイグレーションに必要な追加のステップを実行します。

36 ページの『手動による Business Process Choreographer データベースのアップグレード』

Business Process Choreographer が構成されているサーバーをマイグレーションした後は、サーバーを始動する前に、関連付けられた Business Process Choreographer データベースのスキーマをアップグレードする必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければなりません。

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』

WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

111 ページの『事後マイグレーション構成検査』

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

関連資料

22 ページの『WBIPProfileUpgrade スクリプト』

クラスターをマイグレーションする場合とその他の特殊な状況において、WBIPProfileUpgrade スクリプトを使用して WebSphere Process Server プロファイルのアプリケーションと構成の設定を更新します。

マイグレーション・ウィザードを使用したクラスター・メンバーのマイグレーション:

マイグレーション・ウィザードを使用して、クラスター・メンバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

注: 以下の説明は、ご使用のクラスター内のすべてのサーバーをマイグレーションするための手順の一部です。ここで説明するステップを実行する前に、75 ページの『クラスターのマイグレーション』または 87 ページの『最小限のダウン時間でのクラスターのマイグレーション』の説明に従ってください。

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。
- 古いバージョンの WebSphere Process Server で作成された統合プロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。
- マイグレーションしようとする管理対象ノードを管理するデプロイメント・マネージャーが、既に新しいバージョンの WebSphere Process Server にマイグレーションされていて、実行中です。

注: WebSphere Process Server バージョン 6.0.x 管理対象ノードをバージョン 6.1 管理対象ノードにマイグレーションするには、まずバージョン 6.0.x デプロイメント・マネージャーをバージョン 6.1 デプロイメント・マネージャーにマイグレーションする必要があります。詳しくは、50 ページの『デプロイメント・マネージャーのマイグレーション』を参照してください。このトピックの指示に従って続行する前に、デプロイメント・マネージャーのマイグレーションを完了してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、`backupConfig` コマンドまたは望ましいバックアップ・ユーティリティーを使用して、管理対象ノードの構成をバックアップします。『`backupConfig` コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

このタスクについて

古いバージョンのデプロイメント・マネージャーを新しいバージョンの WebSphere Process Server にマイグレーションすれば、新しいバージョンのデプロイメント・マネージャーはデフォルトでは互換モードで実行され、このとき新旧両方のバージョンの WebSphere Process Server を管理できます。例えばマイグレーションの後で、バージョン 6.1 デプロイメント・マネージャーはバージョン 6.0.x とバージョン

6.1 のリリースのノードの両方を管理できます。バージョン 6.0.x デプロイメント・マネージャーの管理対象ノードは、バージョン 6.0.x 管理対象ノードとしてバージョン 6.1 デプロイメント・マネージャー内で実行されています。リリースが混在したセルを使用する上での制限については、他の WebSphere 製品のインストール済み環境との『共存』を参照してください。

バージョン 6.0.x WebSphere Process Server 管理対象ノード (バージョン 6.1 デプロイメント・マネージャーによって管理されるサーバー) を、バージョン 6.1 管理対象ノードにそれぞれ順次マイグレーションしていきます。すべてのバージョン 6.0.x 管理対象ノードのマイグレーションが終わったら、**convertScriptCompatibility** スクリプトを使用して、バージョン 6.0.x 管理スクリプトの後方互換性をサポートするノードから、バージョン 6.1 のみをサポートするノードに、デプロイメント・マネージャーを変更します。 **convertScriptCompatibility** コマンドを参照してください。

注: このリンクの指示に従って、**convertScriptCompatibility** コマンドを使用するとき、**WASPostUpgrade** コマンドではなく **WBIPostUpgrade** コマンドを使用してください。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

プロシージャ

1. Linux または UNIX システムでは **root** として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション対象のノード上でバージョン 6.0.x サーバーが実行中の場合、そのサーバーを停止します。影響を受けるサーバーのプロファイルの `profile_dir/bin` ディレクトリで **stopServer** コマンドを使用するか、プロファイルのファースト・ステップ・コンソールでサーバーを停止します。

stopServer コマンドについて詳しくは、『**stopServer** コマンド』を参照してください。

Linux 例えば、Linux システムで以下のコマンドを実行して、`server1` を停止します。

```
./stopServer.sh server1
```

セキュリティが有効になっている場合、**stopServer** コマンドで `-username` および `-password` パラメーターを指定します。入力するユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーである必要があります。

Windows Windows システムで以下のコマンドを実行して、`server1` を停止します。

```
stopServer.bat server1
```

Windows オペレーティング・システム上では、セキュリティが有効になっていても、サーバーが Windows サービスとして実行されていれば、`-username` お

および `-password` パラメーターを指定する必要はありません。この場合、パラメーターは、Windows サービスがシステムのシャットダウンに使用するスクリプトに自動的に渡されます。

注: サーバーはマイグレーション中に停止していても実行中でも構いません。サーバーを停止せずにマイグレーションできます。ただし、新しいサーバー (新しいバージョンの WebSphere Process Server を実行するサーバー) を始動するには、古いサーバーを停止する必要があります。また、サーバー構成をマイグレーションするのに、サーバーを稼働させておく必要はありません。マイグレーション・ツールはサーバーが停止していても、すべての構成データを取り出すことができます。

3. マイグレーション対象のノードのノード・エージェントを停止します。プラットフォームに応じて、以下のいずれかのコマンドを実行し、ノード・エージェント・プロセスを停止します (ここで、`profile_root` は統合ノードのインストール・ディレクトリーを表します)。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:

```
profile_root/bin/stopNode.sh
```

- **Windows** **Windows** プラットフォームの場合:

```
profile_root%bin%stopNode.bat
```

セキュリティーが使用可能な場合は、代わりに以下のコマンドのいずれかを使用します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:

```
profile_root/bin/stopNode.sh -username user_ID -password password
```

- **Windows** **Windows** プラットフォームの場合:

```
profile_root%bin%stopNode.bat -username user_ID -password password
```

注: ノードはマイグレーション中に停止していても実行中でも構いません。管理対象ノードは停止せずに (ノード・エージェントを停止せずに) マイグレーションできます。ただし、新しいノード (新しいバージョンの WebSphere Process Server を実行するノード) を始動するには、古いノードを停止する必要があります。また、ノード構成をマイグレーションするのに、ノードを稼働させておく必要はありません。マイグレーション・ツールはノードが停止していても、すべての構成データを取り出すことができます。

4. オプション: マイグレーション・ウィザード以外のツールを使用して、新しいバージョン 6.1 プロファイルを作成します。

バージョン 6.1 スタンドアロン・サーバー・プロファイルまたはカスタム・プロファイルのいずれかをターゲットとして作成できます。カスタム・プロファイルを作成する場合、マイグレーション前にノードを統合しないでください。マイグレーション・ツールがマイグレーション中にノードをフェデレートします。

注: マイグレーションが正常に行われるためには、バージョン 6.0.x からバージョン 6.1 にマイグレーションする各ノードについて、同じノード名とセル名を使用する必要があります。

ヒント: マイグレーションする前に、仮想ホスト情報の変更などのセル・レベルの変更を新しいバージョン 6.1 ノードに行った場合、これらの変更はマイグレーション中に失われます。したがって、ノードがマイグレーションされるまで待ってから、このような変更を行う必要があります。そうしない場合、仮想ホストおよびホスト別名情報への変更などの新しいセルへのすべての変更を、デプロイメント・マネージャー上で実行される管理コンソールを使用して、マイグレーション後に手動で再び行う必要があります。

5. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

6. バージョン 6.1 デプロイメント・マネージャーが稼働中であるか確認してください。

注: ノードが実行中でも停止中でも、バージョン 6.0.x ノードはマイグレーションできます。いずれの状態でも、マイグレーション・ツールはすべての構成データを取り出すことができます。ただし、インストールするバージョン 6.1 ノードを開始するには、バージョン 6.0.x ノードを先に停止する必要がありますので、ここで停止するのが妥当です。

7. マイグレーション・ウィザードを起動します。

マイグレーション・ウィザードは、以下のいずれかの方法で起動します。

- WebSphere Process Server のファースト・ステップ・コンソールから「マイグレーション・ウィザード」を選択する。
- `install_dir/bin` ディレクトリーに保存されている以下のスクリプトの 1 つ (これはオペレーティング・システムによって異なります) を実行する。
 - a.   UNIX および Linux システムの場合:
`wbi_migration.sh`
 - b.  Windows システムの場合: `wbi_migration.bat`

マイグレーション・ウィザードは、構成およびアプリケーションをバージョン 6.0.x 管理対象ノードからバージョン 6.1 管理対象ノードにコピーします。すべ

てのデータのマイグレーションが終わったら、ウィザードはバージョン 6.1 管理対象ノードをデプロイメント・マネージャー・セルに統合します。

8. マイグレーションするクラスター・メンバーごとに、ステップ 1 から 7 (65 ページ) を繰り返します。
9. 互換性オプション (デフォルト) を選択していて、すべてのノードが完全に WebSphere Process Server バージョン 6.1 にマイグレーションされたら、convertScriptCompatibility スクリプトを実行して、バージョン 6.1 デプロイメント・マネージャーから後方互換性を除去します。

convertScriptCompatibility コマンドを bin ディレクトリーから発行します。

- **UNIX** **Linux** `install_root/bin/convertScriptCompatibility.sh`
- **Windows** `install_root¥bin¥convertScriptCompatibility.bat`

convertScriptCompatibility コマンドを参照してください。

結果

クラスター・メンバー・プロファイルはマイグレーションされました。

75 ページの『クラスターのマイグレーション』のステップ 6 から 9 または 87 ページの『最小限のダウン時間でのクラスターのマイグレーション』のステップ 7 から 12 を実行して、クラスターのマイグレーションを完了します。

関連タスク

『コマンド行ツールを使用したクラスター・メンバーのマイグレーション』
コマンド行ツールを使用して、クラスター・メンバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

コマンド行ツールを使用したクラスター・メンバーのマイグレーション:

コマンド行ツールを使用して、クラスター・メンバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

注: 以下の説明は、ご使用のクラスター内のすべてのサーバーをマイグレーションするための手順の一部です。ここで説明するステップを実行する前に、75 ページの『クラスターのマイグレーション』または 87 ページの『最小限のダウン時間でのクラスターのマイグレーション』の説明に従ってください。

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。
- 古いバージョンの WebSphere Process Server で作成された統合プロファイルが同じシステムに存在する。

- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。
- マイグレーションしようとする管理対象ノードを管理するデプロイメント・マネージャーが、既に新しいバージョンの WebSphere Process Server にマイグレーションされていて、実行中です。

注: WebSphere Process Server バージョン 6.0.x 管理対象ノードをバージョン 6.1 管理対象ノードにマイグレーションするには、まずバージョン 6.0.x デプロイメント・マネージャーをバージョン 6.1 デプロイメント・マネージャーにマイグレーションする必要があります。詳しくは、50 ページの『デプロイメント・マネージャーのマイグレーション』を参照してください。このトピックの指示に従って続行する前に、デプロイメント・マネージャーのマイグレーションを完了してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、`backupConfig` コマンドまたは望ましいバックアップ・ユーティリティを使用して、管理対象ノードの構成をバックアップします。『`backupConfig` コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

このタスクについて

古いバージョンのデプロイメント・マネージャーを新しいバージョンの WebSphere Process Server にマイグレーションすれば、新しいバージョンのデプロイメント・マネージャーはデフォルトでは互換モードで実行され、このとき新旧両方のバージョンの WebSphere Process Server を管理できます。例えばマイグレーションの後で、バージョン 6.1 デプロイメント・マネージャーはバージョン 6.0.x とバージョン 6.1 のリリースのノードの両方を管理できます。バージョン 6.0.x デプロイメント・マネージャーの管理対象ノードは、バージョン 6.0.x 管理対象ノードとしてバージョン 6.1 デプロイメント・マネージャー内で実行されています。リリースが混在したセルを使用する上での制限については、他の WebSphere 製品のインストール済み環境との『共存』を参照してください。

バージョン 6.0.x WebSphere Process Server 管理対象ノード (バージョン 6.1 デプロイメント・マネージャーによって管理されるサーバー) を、バージョン 6.1 管理対象ノードにそれぞれ順次マイグレーションしていきます。すべてのバージョン 6.0.x 管理対象ノードのマイグレーションが終わったら、`convertScriptCompatibility` スクリプトを使用して、バージョン 6.0.x 管理スクリプトの後方互換性をサポートするノードから、バージョン 6.1 のみをサポートするノードに、デプロイメント・マネージャーを変更します。 `convertScriptCompatibility` コマンドを参照してください。

注: このリンクの指示に従って、convertScriptCompatibility コマンドを使用するとき、WASPostUpgrade コマンドではなく WBIPostUpgrade コマンドを使用してください。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

プロシージャー

1. Linux または UNIX システムでは root として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション対象のノード上でバージョン 6.0.x サーバーが実行中の場合、そのサーバーを停止します。影響を受けるサーバーのプロファイルの `profile_dir/bin` ディレクトリで `stopServer` コマンドを使用するか、プロファイルのファースト・ステップ・コンソールでサーバーを停止します。

`stopServer` コマンドについて詳しくは、『`stopServer` コマンド』を参照してください。

Linux 例えば、Linux システムで以下のコマンドを実行して、`server1` を停止します。

```
./stopServer.sh server1
```

セキュリティが有効になっている場合、`stopServer` コマンドで `-username` および `-password` パラメーターを指定します。入力するユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーである必要があります。

Windows Windows システムで以下のコマンドを実行して、`server1` を停止します。

```
stopServer.bat server1
```

Windows オペレーティング・システム上では、セキュリティが有効になっていても、サーバーが Windows サービスとして実行されていれば、`-username` および `-password` パラメーターを指定する必要はありません。この場合、パラメーターは、Windows サービスがシステムのシャットダウンに使用するスクリプトに自動的に渡されます。

注: サーバーはマイグレーション中に停止していても実行中でも構いません。サーバーを停止せずにマイグレーションできます。ただし、新しいサーバー(新しいバージョンの WebSphere Process Server を実行するサーバー)を始動するには、古いサーバーを停止する必要があります。また、サーバー構成をマイグレーションするのに、サーバーを稼働させておく必要はありません。マイグレーション・ツールはサーバーが停止していても、すべての構成データを取り出すことができます。

3. マイグレーション対象のノードのノード・エージェントを停止します。プラットフォームに応じて、以下のいずれかのコマンドを実行し、ノード・エージェント・プロセスを停止します(ここで、`profile_root` は統合ノードのインストール・ディレクトリを表します)。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:

```
profile_root/bin/stopNode.sh
```

- **Windows** **Windows** プラットフォームの場合:

```
profile_root%bin%stopNode.bat
```

セキュリティーが使用可能な場合は、代わりに以下のコマンドのいずれかを使用します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:

```
profile_root/bin/stopNode.sh -username user_ID -password password
```

- **Windows** **Windows** プラットフォームの場合:

```
profile_root%bin%stopNode.bat -username user_ID -password password
```

注: ノードはマイグレーション中に停止していても実行中でも構いません。管理対象ノードは停止せずに (ノード・エージェントを停止せずに) マイグレーションできます。ただし、新しいノード (新しいバージョンの WebSphere Process Server を実行するノード) を始動するには、古いノードを停止する必要があります。また、ノード構成をマイグレーションするのに、ノードを稼働させておく必要はありません。マイグレーション・ツールはノードが停止していても、すべての構成データを取り出すことができます。

4. 新規プロファイルを作成します。プロファイル管理ツールまたは `manageprofiles` コマンドを使用して、新しいバージョンの WebSphere Process Server で新規プロファイルを作成します。

バージョン 6.1 スタンドアロン・サーバー・プロファイルまたはカスタム・プロファイルのいずれかをターゲットとして作成できます。カスタム・プロファイルを作成する場合、マイグレーション前にノードを統合しないでください。マイグレーション・ツールがマイグレーション中にノードをフェデレートします。

ノード名、セル名、および拡張レベルが バージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。

注: マイグレーションが正常に行われるためには、バージョン 6.0.x からバージョン 6.1 にマイグレーションする各ノードについて、同じノード名とセル名を使用する必要があります。

ヒント: マイグレーションする前に、仮想ホスト情報の変更などのセル・レベルの変更を新しい バージョン 6.1 ノードに行った場合、これらの変更はマイグレーション中に失われます。したがって、ノードがマイグレーションされるまで待つから、このような変更を行う必要があります。そうしない場合、仮想ホストおよびホスト別名情報への変更などの新しいセルへのすべての変更を、デプロイメント・マネージャー上で実行される管理コンソールを使用して、マイグレーション後に手動で再び行う必要があります。

5. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

- バージョン 6.1 デプロイメント・マネージャーが稼働中であるか確認してください。

注: ノードが実行中でも停止中でも、バージョン 6.0.x ノードはマイグレーションできます。いずれの状態でも、マイグレーション・ツールはすべての構成データを取り出すことができます。ただし、インストールするバージョン 6.1 ノードを開始するには、バージョン 6.0.x ノードを先に停止する必要があるため、ここで停止するのが妥当です。

- マイグレーション・バックアップ・ディレクトリー名と既存の WebSphere Process Server ディレクトリー名を指定して、WBIPreUpgrade コマンドを実行します。WBIPreUpgrade ツールは、*install_root* および *profile_root* ディレクトリーから、指定したバックアップ・ディレクトリーに、選択されたファイルを保存します。詳しくは、14 ページの『WBIPreUpgrade コマンド』を参照してください。
- マイグレーション・バックアップ・ディレクトリーを指定して、WBIPostUpgrade コマンドを実行します。WBIPostUpgrade ツールによって、バックアップ・ディレクトリー内の環境が、新しい WebSphere Process Server スタンドアロン・サーバー・インストール済み環境にリストアされます。詳しくは、17 ページの『WBIPostUpgrade コマンド』を参照してください。
- ステップ 1 から 8 までを繰り返します (ステップ 7 は実行しない場合があります)。

注: ステップ 7 を再度実行 (WBIPreUpgrade を実行) する必要があるのは、最初に WBIPreUpgrade を実行した後にバージョン 6.0.x システムが再構成された場合のみです。

- 互換性オプション (デフォルト) を選択していて、すべてのノードが完全に WebSphere Process Server バージョン 6.1 にマイグレーションされたら、*convertScriptCompatibility* スクリプトを実行して、バージョン 6.1 デプロイメント・マネージャーから後方互換性を除去します。

convertScriptCompatibility コマンドを *bin* ディレクトリーから発行します。

- **UNIX** **Linux** `install_root/bin/convertScriptCompatibility.sh`
- **Windows** `install_root\bin\convertScriptCompatibility.bat`

`convertScriptCompatibility` コマンドを参照してください。

結果

クラスター・メンバー・プロファイルはマイグレーションされました。

75 ページの『クラスターのマイグレーション』のステップ 6 から 9 または『最小限のダウン時間でのクラスターのマイグレーション』のステップ 7 から 12 を実行して、クラスターのマイグレーションを完了します。

関連タスク

77 ページの『マイグレーション・ウィザードを使用したクラスター・メンバーのマイグレーション』

マイグレーション・ウィザードを使用して、クラスター・メンバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

最小限のダウン時間でのクラスターのマイグレーション

ダウン時間を最小限にしてクラスターをマイグレーションするには、クラスターに寄与するプロファイルの約半分を最初にマイグレーションしてから、残りの半分をマイグレーションします。最初のプロファイルのセットをマイグレーションした後、クラスター・マイグレーションに必要な追加のステップを実行します。

始める前に

注: クラスター内で、バージョン 6.0.x メンバーとバージョン 6.1 メンバーを同時に稼働させることはできません。最初にバージョン 6.1 クラスター・メンバーを開始する前に、すべてのバージョン 6.0.x クラスター・メンバーを停止する必要があります。また、いったんバージョン 6.1 クラスター・メンバーを開始したら、そのクラスターでバージョン 6.0.x クラスター・メンバーを開始しないでください。

新しいバージョン (例えば、バージョン 6.1) へのマイグレーションを希望する古いバージョンの WebSphere Process Server (例えば、バージョン 6.0.x) 上で実行している、1 つ以上のクラスターを含む既存のセルが必要です。さらに、新しいバージョンの WebSphere Process Server がインストールされている必要があります。

このタスクについて

以下のステップに従うことで、新しいバージョンの WebSphere Process Server のクラスター機能を保持できます。

プロシージャ

1. デプロイメント・マネージャーをマイグレーションします。50 ページの『デプロイメント・マネージャーのマイグレーション』にリストされた一連の手順の 1 つに従い、このタスクを完了します。
2. 新しいデプロイメント・マネージャーが稼働していることを確認します。
3. 関係するプロファイルを識別します。

- a. クラスタ・メンバーを含む古いバージョンのプロファイルを識別します。
- b. このプロファイルが関与する他のクラスタを識別します。つまり、このプロファイルで、他のクラスタのメンバーであるサーバーが定義されている場合、そのクラスタを識別します。
- c. ステップ3bで識別されたクラスタのいずれかに対してクラスタ・メンバーを提供する、同じセルのすべての他のプロファイルを識別します。
- d. ステップ3cで識別されるプロファイルによって定義されるすべてのノード・エージェントおよびプロセス・サーバーを識別します。

ステップ 3c で識別されるすべてのプロファイルと、ステップ 3d で識別される対応するすべてのノード・エージェントおよびサーバーがマイグレーションに含まれます。

4. ステップ3 (75 ページ)で特定されたプロファイルの完全なセットから、プロファイルの 2 つのグループを定義します。プロファイルを約半分に分割します (プロファイルの総数が奇数の場合、片方のグループはもう一方のグループよりも 1 つ多くなります)。一方のサーバーのセットを実行している間に他方のセットをマイグレーションできるため、クラスタ内のすべてのサーバーが停止する時間を削減できます。
5. マイグレーションする最初のプロファイルのセットによって定義された、すべてのノード・エージェントおよびサーバーを停止します。
6. 最初のセットのプロファイルをそれぞれ 1 つずつマイグレーションしますが、新規エージェントまたはサーバーを始動しないでください。77 ページの『マイグレーション・ウィザードを使用したクラスタ・メンバーのマイグレーション』または 82 ページの『コマンド行ツールを使用したクラスタ・メンバーのマイグレーション』にリストされた手順の 1 つに従います。
7. 残りのノード・エージェントおよびサーバーを停止します。これらは 2 番目のプロファイルのセットで定義されたものになります。このアクションを開始した時点から、クラスタ・サービスが利用不可になります。
8. WebSphere Process Server バージョン 6.1 デプロイメント・マネージャー・プロファイルをホストするシステムで、`install_dir\util` ディレクトリにナビゲートします。このディレクトリには `WBIPProfileUpgrade` スクリプトである `WBIPProfileUpgrade.ant` があります。
9. 既にマイグレーション済みのプロファイルで定義された各クラスタについて、`WBIPProfileUpgrade` スクリプトを実行します。(つまり、ステップ3 (75 ページ)で定義された各クラスタについて `WBIPProfileUpgrade` を実行します。) `WBIPProfileUpgrade` の実行についての説明は、22 ページの『`WBIPProfileUpgrade` スクリプト』を参照してください。
10. Business Process Choreographer が構成されているサーバーをマイグレーションしており、その関連データベースを手動でアップグレードする必要がある場合は、このステップをここで実行してください。36 ページの『手動による Business Process Choreographer データベースのアップグレード』を参照してください。
11. 新規の (マイグレーション済みの) ノード・エージェントおよびサーバーをすべて始動します。これは、既にマイグレーション済みのプロファイルに対応するノード・エージェントおよびサーバーです。

12. 2 番目のプロファイルのセットにある各プロファイルをマイグレーションします。最初のセットと同様に、マイグレーションするには、77 ページの『マイグレーション・ウィザードを使用したクラスター・メンバーのマイグレーション』または 82 ページの『コマンド行ツールを使用したクラスター・メンバーのマイグレーション』にリストされた手順の 1 つに従います。今回は、それぞれの管理対象ノードのマイグレーションを続行しながら、マイグレーション済みノード・エージェントおよびサーバーを始動できます。

結果

クラスターは新しいバージョンの WebSphere Process Server にマイグレーションされました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要のある追加タスクに注意してください。

関連タスク

50 ページの『デプロイメント・マネージャーのマイグレーション』

WebSphere Process Server デプロイメント・マネージャーを、ニーズに応じていくつかの方法から選択してマイグレーションします。

60 ページの『非クラスター管理対象ノードのマイグレーション』

WebSphere Process Server 管理対象ノードを、ニーズに応じていくつかの方法から選択してマイグレーションします。

75 ページの『クラスターのマイグレーション』

クラスターをマイグレーションするには、そのクラスターのメンバーを含む各プロファイルを一度に 1 つずつマイグレーションします。マイグレーションには、非クラスター環境には不要な追加のステップが必要になります。

36 ページの『手動による Business Process Choreographer データベースのアップグレード』

Business Process Choreographer が構成されているサーバーをマイグレーションした後は、サーバーを始動する前に、関連付けられた Business Process Choreographer データベースのスキーマをアップグレードする必要があります。データ・ソース用に定義されているデータベース・ユーザーに、データベース・スキーマを変更するための十分な権限がない場合は、手動でアップグレードしなければなりません。

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』
WebSphere Process Server の古いバージョンからのマイグレーション時に問題が
発生した場合は、このページのトラブルシューティングのヒントを参照してくだ
さい。

111 ページの『事後マイグレーション構成検査』
マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を
変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があ
ります。

関連資料

22 ページの『WBIPprofileUpgrade スクリプト』
クラスターをマイグレーションする場合とその他の特殊な状況において、
WBIPprofileUpgrade スクリプトを使用して WebSphere Process Server プロファイ
ルのアプリケーションと構成の設定を更新します。

マイグレーション・ウィザードを使用したクラスター・メンバーのマイグレーショ ン:

マイグレーション・ウィザードを使用して、クラスター・メンバーを WebSphere
Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

注: 以下の説明は、ご使用のクラスター内のすべてのサーバーをマイグレーション
するための手順の一部です。ここで説明するステップを実行する前に、75 ページの
『クラスターのマイグレーション』または 87 ページの『最小限のダウン時間でのク
ラスタのマイグレーション』の説明に従ってください。

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認
してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。
- 古いバージョンの WebSphere Process Server で作成された統合プロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。
- マイグレーションしようとする管理対象ノードを管理するデプロイメント・マネージャーが、既に新しいバージョンの WebSphere Process Server にマイグレーションされていて、実行中です。

注: WebSphere Process Server バージョン 6.0.x 管理対象ノードをバージョン 6.1 管理対象ノードにマイグレーションするには、まずバージョン 6.0.x デプロイメント・マネージャーをバージョン 6.1 デプロイメント・マネージャーにマイグレーションする必要があります。詳しくは、50 ページの『デプロイメント・マネー

ジャーのマイグレーション』を参照してください。このトピックの指示に従って
続行する前に、デプロイメント・マネージャーのマイグレーションを完了してく
ださい。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを
確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデ
ータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、
backupConfig コマンドまたは望ましいバックアップ・ユーティリティーを使用し
て、管理対象ノードの構成をバックアップします。『backupConfig コマンド』を
参照してください。このバックアップした構成の正しい名前と場所をメモしてお
いてください。

このタスクについて

古いバージョンのデプロイメント・マネージャーを新しいバージョンの WebSphere
Process Server にマイグレーションすれば、新しいバージョンのデプロイメント・マ
ネージャーはデフォルトでは互換モードで実行され、このとき新旧両方のバージョ
ンの WebSphere Process Server を管理できます。例えばマイグレーションの後で、
バージョン 6.1 デプロイメント・マネージャーはバージョン 6.0.x とバージョン
6.1 のリリースのノードの両方を管理できます。バージョン 6.0.x デプロイメント・
マネージャーの管理対象ノードは、バージョン 6.0.x 管理対象ノードとして バージ
ョン 6.1 デプロイメント・マネージャー内で実行されています。リリースが混在し
たセルを使用する上での制限については、他の WebSphere 製品のインストール済み
環境との『共存』を参照してください。

バージョン 6.0.x WebSphere Process Server 管理対象ノード (バージョン 6.1 デプ
ロイメント・マネージャーによって管理されるサーバー) を、バージョン 6.1 管理
対象ノードにそれぞれ順次マイグレーションしていきます。すべての バージョン
6.0.x 管理対象ノードのマイグレーションが終わったら、**convertScriptCompatibility**
スクリプトを使用して、バージョン 6.0.x 管理スクリプトの後方互換性をサポート
するノードから、バージョン 6.1 のみをサポートするノードに、デプロイメント・
マネージャーを変更します。 **convertScriptCompatibility** コマンドを参照してくださ
い。

注: このリンクの指示に従って、**convertScriptCompatibility** コマンドを使用する
とき、**WASPostUpgrade** コマンドではなく **WBIPostUpgrade** コマンドを使用してく
ださい。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、
134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照
してください。

プロシージャ

1. Linux または UNIX システムでは root として、Windows システムでは管理者
グループのメンバーとしてログオンします。
2. マイグレーション対象のノード上でバージョン 6.0.x サーバーが実行中の場合、
そのサーバーを停止します。 影響を受けるサーバーのプロファイルの

`profile_dir/bin` ディレクトリーで `stopServer` コマンドを使用するか、プロファイルのファースト・ステップ・コンソールでサーバーを停止します。

`stopServer` コマンドについて詳しくは、『`stopServer` コマンド』を参照してください。

Linux 例えば、Linux システムで以下のコマンドを実行して、`server1` を停止します。

```
./stopServer.sh server1
```

セキュリティーが有効になっている場合、`stopServer` コマンドで `-username` および `-password` パラメーターを指定します。入力するユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーである必要があります。

Windows Windows システムで以下のコマンドを実行して、`server1` を停止します。

```
stopServer.bat server1
```

Windows オペレーティング・システム上では、セキュリティーが有効になっていても、サーバーが Windows サービスとして実行されていれば、`-username` および `-password` パラメーターを指定する必要はありません。この場合、パラメーターは、Windows サービスがシステムのシャットダウンに使用するスクリプトに自動的に渡されます。

注: サーバーはマイグレーション中に停止していても実行中でも構いません。サーバーを停止せずにマイグレーションできます。ただし、新しいサーバー (新しいバージョンの WebSphere Process Server を実行するサーバー) を始動するには、古いサーバーを停止する必要があります。また、サーバー構成をマイグレーションするのに、サーバーを稼働させておく必要はありません。マイグレーション・ツールはサーバーが停止していても、すべての構成データを取り出すことができます。

3. マイグレーション対象のノードのノード・エージェントを停止します。プラットフォームに応じて、以下のいずれかのコマンドを実行し、ノード・エージェント・プロセスを停止します (ここで、`profile_root` は統合ノードのインストール・ディレクトリーを表します)。

• **Linux** **UNIX** Linux および UNIX プラットフォームの場合:

```
profile_root/bin/stopNode.sh
```

• **Windows** Windows プラットフォームの場合:

```
profile_root¥bin¥stopNode.bat
```

セキュリティーが使用可能な場合は、代わりに以下のコマンドのいずれかを使用します。

• **Linux** **UNIX** Linux および UNIX プラットフォームの場合:

```
profile_root/bin/stopNode.sh -username user_ID -password password
```

• **Windows** Windows プラットフォームの場合:

```
profile_root¥bin¥stopNode.bat -username user_ID -password password
```

注: ノードはマイグレーション中に停止していても実行中でも構いません。管理対象ノードは停止せずに (ノード・エージェントを停止せずに) マイグレーションできます。ただし、新しいノード (新しいバージョンの WebSphere Process Server を実行するノード) を始動するには、古いノードを停止する必要があります。また、ノード構成をマイグレーションするのに、ノードを稼働させておく必要はありません。マイグレーション・ツールはノードが停止していても、すべての構成データを取り出すことができます。

4. オプション: マイグレーション・ウィザード以外のツールを使用して、新しいバージョン 6.1 プロファイルを作成します。

バージョン 6.1 スタンドアロン・サーバー・プロファイルまたはカスタム・プロファイルのいずれかをターゲットとして作成できます。カスタム・プロファイルを作成する場合、マイグレーション前にノードを統合しないでください。マイグレーション・ツールがマイグレーション中にノードをフェデレートします。

注: マイグレーションが正常に行われるためには、バージョン 6.0.x からバージョン 6.1 にマイグレーションする各ノードについて、同じノード名とセル名を使用する必要があります。

ヒント: マイグレーションする前に、仮想ホスト情報の変更などのセル・レベルの変更を新しいバージョン 6.1 ノードに行った場合、これらの変更はマイグレーション中に失われます。したがって、ノードがマイグレーションされるまで待ってから、このような変更を行う必要があります。そうしない場合、仮想ホストおよびホスト別名情報への変更などの新しいセルへのすべての変更を、デプロイメント・マネージャー上で実行される管理コンソールを使用して、マイグレーション後に手動で再び行う必要があります。

5. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

6. バージョン 6.1 デプロイメント・マネージャーが稼働中であるか確認してください。

注: ノードが実行中でも停止中でも、バージョン 6.0.x ノードはマイグレーションできます。いずれの状態でも、マイグレーション・ツールはすべての構成データを取り出すことができます。ただし、インストールするバージョン 6.1 ノードを開始するには、バージョン 6.0.x ノードを先に停止する必要がありますので、ここで停止するのが妥当です。

7. マイグレーション・ウィザードを起動します。

マイグレーション・ウィザードは、以下のいずれかの方法で起動します。

- WebSphere Process Server のファースト・ステップ・コンソールから「**マイグレーション・ウィザード**」を選択する。
- `install_dir/bin` ディレクトリに保存されている以下のスクリプトの 1 つ (これはオペレーティング・システムによって異なります) を実行する。

- a. **UNIX** **Linux** UNIX および Linux システムの場合:

`wbi_migration.sh`

- b. **Windows** Windows システムの場合: `wbi_migration.bat`

マイグレーション・ウィザードは、構成およびアプリケーションをバージョン 6.0.x 管理対象ノードからバージョン 6.1 管理対象ノードにコピーします。すべてのデータのマイグレーションが終わったら、ウィザードはバージョン 6.1 管理対象ノードをデプロイメント・マネージャー・セルに統合します。

8. マイグレーションするクラスター・メンバーごとに、ステップ 1 から 7 (65 ページ) を繰り返します。
9. 互換性オプション (デフォルト) を選択していて、すべてのノードが完全に WebSphere Process Server バージョン 6.1 にマイグレーションされたら、`convertScriptCompatibility` スクリプトを実行して、バージョン 6.1 デプロイメント・マネージャーから後方互換性を除去します。

`convertScriptCompatibility` コマンドを `bin` ディレクトリから発行します。

- **UNIX** **Linux** `install_root/bin/convertScriptCompatibility.sh`

- **Windows** `install_root\bin\convertScriptCompatibility.bat`

`convertScriptCompatibility` コマンドを参照してください。

結果

クラスター・メンバー・プロファイルはマイグレーションされました。

75 ページの『クラスターのマイグレーション』のステップ 6 から 9 または 87 ページの『最小限のダウン時間でのクラスターのマイグレーション』のステップ 7 から 12 を実行して、クラスターのマイグレーションを完了します。

関連タスク

82 ページの『コマンド行ツールを使用したクラスター・メンバーのマイグレーション』

コマンド行ツールを使用して、クラスター・メンバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

コマンド行ツールを使用したクラスター・メンバーのマイグレーション:

コマンド行ツールを使用して、クラスター・メンバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

始める前に

注: 以下の説明は、ご使用のクラスター内のすべてのサーバーをマイグレーションするための手順の一部です。ここで説明するステップを実行する前に、75 ページの『クラスターのマイグレーション』または 87 ページの『最小限のダウン時間でのクラスターのマイグレーション』の説明に従ってください。

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- 古いバージョンの WebSphere Process Server が存在するのと同じ物理コンピューター・システム上にマイグレーションする場合、WebSphere Process Server の新規バージョンを同じシステムにインストールしている。
- 古いバージョンの WebSphere Process Server で作成された統合プロファイルが同じシステムに存在する。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。
- マイグレーションしようとする管理対象ノードを管理するデプロイメント・マネージャーが、既に新しいバージョンの WebSphere Process Server にマイグレーションされていて、実行中です。

注: WebSphere Process Server バージョン 6.0.x 管理対象ノードをバージョン 6.1 管理対象ノードにマイグレーションするには、まずバージョン 6.0.x デプロイメント・マネージャーをバージョン 6.1 デプロイメント・マネージャーにマイグレーションする必要があります。詳しくは、50 ページの『デプロイメント・マネージャーのマイグレーション』を参照してください。このトピックの指示に従って続行する前に、デプロイメント・マネージャーのマイグレーションを完了してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、`backupConfig` コマンドまたは望ましいバックアップ・ユーティリティーを使用して、管理対象ノードの構成をバックアップします。『`backupConfig` コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

このタスクについて

古いバージョンのデプロイメント・マネージャーを新しいバージョンの WebSphere Process Server にマイグレーションすれば、新しいバージョンのデプロイメント・マネージャーはデフォルトでは互換モードで実行され、このとき新旧両方のバージョン

ンの WebSphere Process Server を管理できます。例えばマイグレーションの後で、バージョン 6.1 デプロイメント・マネージャーはバージョン 6.0.x とバージョン 6.1 のリリースのノードの両方を管理できます。バージョン 6.0.x デプロイメント・マネージャーの管理対象ノードは、バージョン 6.0.x 管理対象ノードとしてバージョン 6.1 デプロイメント・マネージャー内で実行されています。リリースが混在したセルを使用する上での制限については、他の WebSphere 製品のインストール済み環境との『共存』を参照してください。

バージョン 6.0.x WebSphere Process Server 管理対象ノード (バージョン 6.1 デプロイメント・マネージャーによって管理されるサーバー) を、バージョン 6.1 管理対象ノードにそれぞれ順次マイグレーションしていきます。すべてのバージョン 6.0.x 管理対象ノードのマイグレーションが終わったら、**convertScriptCompatibility** スクリプトを使用して、バージョン 6.0.x 管理スクリプトの後方互換性をサポートするノードから、バージョン 6.1 のみをサポートするノードに、デプロイメント・マネージャーを変更します。convertScriptCompatibility コマンドを参照してください。

注: このリンクの指示に従って、convertScriptCompatibility コマンドを使用するとき、WASPostUpgrade コマンドではなく WBIPPostUpgrade コマンドを使用してください。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

プロシージャー

1. Linux または UNIX システムでは root として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. マイグレーション対象のノード上でバージョン 6.0.x サーバーが実行中の場合、そのサーバーを停止します。影響を受けるサーバーのプロファイルの *profile_dir/bin* ディレクトリで stopServer コマンドを使用するか、プロファイルのファースト・ステップ・コンソールでサーバーを停止します。

stopServer コマンドについて詳しくは、『stopServer コマンド』を参照してください。

Linux 例えば、Linux システムで以下のコマンドを実行して、server1 を停止します。

```
./stopServer.sh server1
```

セキュリティが有効になっている場合、stopServer コマンドで -username および -password パラメーターを指定します。入力するユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーである必要があります。

Windows Windows システムで以下のコマンドを実行して、server1 を停止します。

```
stopServer.bat server1
```

Windows オペレーティング・システム上では、セキュリティーが有効になっていても、サーバーが Windows サービスとして実行されていれば、`-username` および `-password` パラメーターを指定する必要はありません。この場合、パラメーターは、Windows サービスがシステムのシャットダウンに使用するスクリプトに自動的に渡されます。

注: サーバーはマイグレーション中に停止していても実行中でも構いません。サーバーを停止せずにマイグレーションできます。ただし、新しいサーバー (新しいバージョンの WebSphere Process Server を実行するサーバー) を始動するには、古いサーバーを停止する必要があります。また、サーバー構成をマイグレーションするのに、サーバーを稼働させておく必要はありません。マイグレーション・ツールはサーバーが停止していても、すべての構成データを取り出すことができます。

3. マイグレーション対象のノードのノード・エージェントを停止します。プラットフォームに応じて、以下のいずれかのコマンドを実行し、ノード・エージェント・プロセスを停止します (ここで、`profile_root` は統合ノードのインストール・ディレクトリを表します)。

- **Linux** **UNIX** Linux および UNIX プラットフォームの場合:

```
profile_root/bin/stopNode.sh
```

- **Windows** Windows プラットフォームの場合:

```
profile_root%bin%stopNode.bat
```

セキュリティーが使用可能な場合は、代わりに以下のコマンドのいずれかを使用します。

- **Linux** **UNIX** Linux および UNIX プラットフォームの場合:

```
profile_root/bin/stopNode.sh -username user_ID -password password
```

- **Windows** Windows プラットフォームの場合:

```
profile_root%bin%stopNode.bat -username user_ID -password password
```

注: ノードはマイグレーション中に停止していても実行中でも構いません。管理対象ノードは停止せずに (ノード・エージェントを停止せずに) マイグレーションできます。ただし、新しいノード (新しいバージョンの WebSphere Process Server を実行するノード) を始動するには、古いノードを停止する必要があります。また、ノード構成をマイグレーションするのに、ノードを稼働させておく必要はありません。マイグレーション・ツールはノードが停止していても、すべての構成データを取り出すことができます。

4. 新規プロファイルを作成します。プロファイル管理ツールまたは `manageprofiles` コマンドを使用して、新しいバージョンの WebSphere Process Server で新規プロファイルを作成します。

バージョン 6.1 スタンドアロン・サーバー・プロファイルまたはカスタム・プロファイルのいずれかをターゲットとして作成できます。カスタム・プロファイルを作成する場合、マイグレーション前にノードを統合しないでください。マイグレーション・ツールがマイグレーション中にノードをフェデレートします。

ノード名、セル名、および拡張レベルが バージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。

注: マイグレーションが正常に行われるためには、バージョン 6.0.x からバージョン 6.1 にマイグレーションする各ノードについて、同じノード名とセル名を使用する必要があります。

ヒント: マイグレーションする前に、仮想ホスト情報の変更などのセル・レベルの変更を新しい バージョン 6.1 ノードに行った場合、これらの変更はマイグレーション中に失われます。したがって、ノードがマイグレーションされるまで待つから、このような変更を行う必要があります。そうしない場合、仮想ホストおよびホスト別名情報への変更などの新しいセルへのすべての変更を、デプロイメント・マネージャー上で実行される管理コンソールを使用して、マイグレーション後に手動で再び行う必要があります。

5. マイグレーション・ウィザードに必要な、以下にリストされた事前情報をあらかじめ把握してください。

インストールのルート・ディレクトリー

-currentWebSphereDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

マイグレーションのバックアップ・ディレクトリー名

-backupDirectory パラメーターについては、14 ページの『WBIPreUpgrade コマンド』を参照してください。

ソース・プロファイル名

-oldProfile パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ターゲット・プロファイル名

-profileName パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

ポート値割り当て (オプション)

-replacePorts および -portBlock パラメーターについては、17 ページの『WBIPostUpgrade コマンド』を参照してください。

6. バージョン 6.1 デプロイメント・マネージャーが稼働中であるか確認してください。

注: ノードが実行中でも停止中でも、バージョン 6.0.x ノードはマイグレーションできます。いずれの状態でも、マイグレーション・ツールはすべての構成データを取り出すことができます。ただし、インストールする バージョン 6.1 ノードを開始するには、バージョン 6.0.x ノードを先に停止する必要がありますので、ここで停止するのが妥当です。

7. マイグレーション・バックアップ・ディレクトリー名と既存の WebSphere Process Server ディレクトリー名を指定して、WBIPreUpgrade コマンドを実行します。WBIPreUpgrade ツールは、*install_root* および *profile_root* ディレクト

リーから、指定したバックアップ・ディレクトリーに、選択されたファイルを保存します。詳しくは、14 ページの『WBIPreUpgrade コマンド』を参照してください。

8. マイグレーション・バックアップ・ディレクトリーを指定して、WBIPostUpgrade コマンドを実行します。WBIPostUpgrade ツールによって、バックアップ・ディレクトリー内の環境が、新しい WebSphere Process Server スタンドアロン・サーバー・インストール済み環境にリストアされます。詳しくは、17 ページの『WBIPostUpgrade コマンド』を参照してください。
9. ステップ 1 から 8 までを繰り返します (ステップ 7 は実行しない場合があります)。

注: ステップ 7 を再度実行 (WBIPreUpgrade を実行) する必要があるのは、最初に WBIPreUpgrade を実行した後にバージョン 6.0.x システムが再構成された場合のみです。

10. 互換性オプション (デフォルト) を選択していて、すべてのノードが完全に WebSphere Process Server バージョン 6.1 にマイグレーションされたら、convertScriptCompatibility スクリプトを実行して、バージョン 6.1 デプロイメント・マネージャーから後方互換性を除去します。

convertScriptCompatibility コマンドを bin ディレクトリーから発行します。

- **UNIX** **Linux** `install_root/bin/convertScriptCompatibility.sh`
- **Windows** `install_root\bin\convertScriptCompatibility.bat`

convertScriptCompatibility コマンドを参照してください。

結果

クラスター・メンバー・プロファイルはマイグレーションされました。

75 ページの『クラスターのマイグレーション』のステップ 6 から 9 または 87 ページの『最小限のダウン時間でのクラスターのマイグレーション』のステップ 7 から 12 を実行して、クラスターのマイグレーションを完了します。

関連タスク

77 ページの『マイグレーション・ウィザードを使用したクラスター・メンバーのマイグレーション』

マイグレーション・ウィザードを使用して、クラスター・メンバーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

リモート・システムへのマイグレーション

マイグレーション・ツールを使用して、あるシステムの古いバージョンから、別のリモート・システム上の新しいバージョンの WebSphere Process Server にマイグレーションします。(スタンドアロン・サーバーのみ。)

始める前に

注: この手順は、スタンドアロン・サーバーの場合のみサポートされます。

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- ご使用の新規 (リモート) システムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、`backupConfig` コマンドまたは望ましいバックアップ・ユーティリティを使用して、既存の構成をバックアップします。『`backupConfig` コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

1 ページの『マイグレーションの概要』および3 ページの『事前マイグレーションの考慮事項』を参照してください。

通常では、マイグレーション・ツールを使用して、同じシステム上で古いバージョンから新しいバージョンにアップグレードできます。ただし、シナリオによっては、あるマシン上にある古いバージョン構成を、別のシステム上にある WebSphere Process Server の新しいバージョンにマイグレーションすることが必要な場合もあります。このシナリオの 1 つとして考えられるのは、新しいバージョンに基づく新しいシステムを環境にインストールしたけれど、既存の古い構成を他のシステムからマイグレーションする必要があるような場合です。

マイグレーションするノードがデプロイメント・セルの一部の場合、この手順を続行する前に、50 ページの『マイグレーション・ウィザードを使用したデプロイメント・マネージャーのマイグレーション』または56 ページの『コマンド行ツールを使用したデプロイメント・マネージャーのマイグレーション』で説明されているように、古いデプロイメント・マネージャーを新しい WebSphere Process Server バージョンにマイグレーションします。デプロイメント・マネージャーのリリース・レベルは、管理対象のセル内のノードのレベルと同じか、それよりも高くする必要があります。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

このタスクについて

WBIPreUpgrade コマンドは既存の古いバージョンの構成を、マイグレーション固有のバックアップ・ディレクトリーに保存します。**WBIPostUpgrade** コマンドはこのディレクトリーを使用して、古い構成設定を新しいバージョンの環境に追加します。

プロシージャ

1. マイグレーション後のバージョン (例えば バージョン 6.1) の WebSphere Process Server CD-ROM および (DVD メディアを所有する場合は) WebSphere Process Server DVD を用意します。

WebSphere Process Server 6.1 の場合、以下を用意する必要があります。

- WebSphere Process Server バージョン 6.1 のディスク 1 (2 枚のうち) または WebSphere Process Server バージョン 6.1 DVD
- WebSphere Process Server バージョン 6.1、WebSphere Application Server Network Deployment Supplements

製品インストール・メディアから、WebSphere Process Server の古いバージョン (バージョン 6.0.x) をホスティングするシステム上のディレクトリーへ、ファイルをコピーする必要があります。これらのディレクトリー内のファイルは、WebSphere Process Server の新しいバージョンをインストールせずに、**WBIPreUpgrade** コマンドを実行するために使用できる特殊な環境を構成します。

2. WebSphere Process Server の古いバージョン (バージョン 6.0.x) をホスティングするローカル・システム上に、`migration_copy` という新規ディレクトリーを作成します。(このディレクトリーには任意の名前を付けることができます。ここでは、説明を分かりやすくするため「`migration_copy`」を使用します。)
3. 製品メディアから、ローカルの バージョン 6.0.x システム上に作成したばかりのディレクトリー `migration_copy` にファイルをコピーします。以下にリストするディレクトリー全体をコピーします。
 - a. WebSphere Process Server バージョン 6.1 のディスク 1 CD-ROM (2 枚のうち) または WebSphere Process Server バージョン 6.1 DVD のいずれかの `migration` というディレクトリー。
 - b. WebSphere Process Server バージョン 6.1、WebSphere Application Server Network Deployment Supplements CD の `migration` というディレクトリー。

注: このディレクトリー内のファイルと、WebSphere Process Server バージョン 6.1 メディアの `migration` ディレクトリー内のファイルを、ご使用のシステム上の単一の `migration` ディレクトリーに結合する必要があります。

- c. WebSphere Process Server バージョン 6.1、WebSphere Application Server Network Deployment Supplements CD の `JDK` というディレクトリー。

これで、システム上に以下のディレクトリー構造ができます。

```
migration_copy/  
  migration/  
    JDK/
```

4. `migration_copy/migration/bin` ディレクトリーにナビゲートします。
5. `migration_copy/bin` ディレクトリーから **WBIPreUpgrade** スクリプトを使用して現在の構成を保存します。詳しくは、14 ページの『WBIPreUpgrade コマンド』を参照してください。

古いバージョンをホスティングするシステム上にある、マイグレーション固有のバックアップ・ディレクトリーに構成を保存します。例えば、次のようになります。

Linux

UNIX

Linux または UNIX の場合:

```
./WBIPreUpgrade.sh /filepath/migration_specific_backup /opt/IBM/WBI602
```

Windows Windows の場合:

```
WBIPreUpgrade C:%filepath%migration_specific_backup  
"C:%Program Files%IBM%WBI602"
```

WBIPreUpgrade コマンドは、画面に状況を表示し、*migration_specific_backup* ディレクトリーのログ・ファイルに状況を出力します。ASCII ログ・ファイル名は **WBIPreUpgrade** というテキストで始まり、日付とタイム・スタンプを含みます。

6. *migration_specific_backup* ディレクトリーを古いバージョンの WebSphere Process Server システムから新しいバージョンのシステムにコピーします。

ftp コマンド、共用ストレージ、または他の手段を使用して、ディレクトリーを新しいマシンにコピーします。

7. 新しいバージョンの WebSphere Process Server をインストールします。

インストール情報については、WebSphere Process Serverのインストールおよび構成を参照してください。

8. プロファイル管理ツールまたは **manageprofiles** コマンドを使用して、新しいバージョンの WebSphere Process Server で新規プロファイルを作成します。ノード名、セル名、および拡張レベルがバージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。プロファイルの作成については、『プロファイルの作成』を参照してください。
9. 古いバージョンの WebSphere Process Server 構成から新しいバージョンの構成に、**WBIPostUpgrade** コマンドを使用して構成を追加します。新規インストール環境の *install_root/bin* ディレクトリー内で **WBIPostUpgrade** コマンドを使用して、古いバージョンの構成 (ステップ6で新しいシステムのディレクトリーにコピーしたもの) を新しいバージョンの構成に追加します。

詳しくは、17 ページの『WBIPostUpgrade コマンド』を参照してください。

Linux **UNIX** Linux または UNIX の場合:

```
./WBIPostUpgrade.sh /filepath/migration_specific_backup
```

Windows Windows の場合:

```
WBIPostUpgrade C:%filepath%migration_specific_backup
```

WBIPostUpgrade ツールは、*migration_specific_backup/WBIPostUpgrade.log* ファイルに、デプロイする各エンタープライズ Bean に固有の詳細な情報を記録します。

10. 新しい WebSphere Process Server バージョンの管理コンソールを使用して、構成を変更します。
 - a. セキュリティー要件に一致するように、ユーザー ID とパスワードを変更します。

古いバージョンをホスティングするシステムで使用中的のものと同一でない場合、ユーザー ID およびパスワードの変更が必要ことがあります。

- b. その他のシステム固有の情報を変更します。

構成は、他のソフトウェア製品を参照したり、または新しいシステムに存在しない構成を参照することがあります。例えば、古いシステムにデータベースがあるとします。古いシステムのデータベースを指すよう、データ・ソースを変更します。

- 11. 必要に応じて、WebSphere Process Server で使用されているデータベースを手動で更新します。通常では、新しいバージョンの WebSphere Process Server で必要なデータベースの変更は、自動的に適用されます。サーバーが最初に始動されるたびに、データベース・テーブルが新規スキーマ・バージョンにマイグレーションされます。ただし、サーバーがデータベース・スキーマにアクセスするための十分な権限を持たないか、その他のデータベース固有の要件が満たされていない場合、データベースを手動で更新する必要があります。詳しくは、32 ページの『マイグレーションのためのデータベースのアップグレード』を参照してください。

結果

古いバージョンから、新しいバージョンをホスティングするリモート・システムに、WebSphere Process Server をマイグレーションしました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要がある追加タスクに注意してください。

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

1 ページの『マイグレーションの概要』

WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからマイグレーションします。

関連タスク

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』
WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

111 ページの『事後マイグレーション構成検査』
マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、`manageprofiles` コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

105 ページの『サポート対象外となったオペレーティング・システムからのマイグレーション』

マイグレーション・ツールを使用して、新しいバージョンでサポートされないオペレーティング・システム上で実行中の以前の WebSphere Process Server リリースをマイグレーションします。(スタンドアロン・サーバーのみ。)

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された `backupDirectory` から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の `profile_root/installedApps` ディレクトリーに追加します。

関連情報

WebSphere Process Serverのインストールおよび構成

このセクションでは、IBM WebSphere Process Server のインストールの準備、インストールの実行、および構成を行う方法について説明します。これらの説明は、Linux、i5/OS、UNIX、および Windows システムを対象としています。

サポート対象外となったオペレーティング・システムからのマイグレーション

マイグレーション・ツールを使用して、新しいバージョンでサポートされないオペレーティング・システム上で実行中の以前の WebSphere Process Server リリースをマイグレーションします。(スタンドアロン・サーバーのみ。)

始める前に

注: この手順は、スタンドアロン・サーバーの場合のみサポートされます。

マイグレーション・プロセスを開始する前に、以下の条件が満たされているか確認してください。

- 使用中のシステムが、WebSphere Process Server の新規バージョンのハードウェア要件およびソフトウェア要件をすべて満たしている。
- マイグレーションするプロファイルおよびそのバックアップのための十分なディスク・スペースが使用可能である。ディスク・スペース要件については、3 ページの『事前マイグレーションの考慮事項』を参照してください。

マイグレーション・プロセスを開始する前に、以下のタスクが完了していることを確認してください。

- バージョン 6.0.x の WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
- マイグレーションの後に以前の状態にリストアできるようにするためには、`backupConfig` コマンドまたは望ましいバックアップ・ユーティリティーを使用して、既存の構成をバックアップします。『`backupConfig` コマンド』を参照してください。このバックアップした構成の正しい名前と場所をメモしておいてください。

1 ページの『マイグレーションの概要』および 3 ページの『事前マイグレーションの考慮事項』を参照してください。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

プロシージャ

1. マイグレーション後のバージョン (例えば バージョン 6.1) の WebSphere Process Server CD-ROM および (DVD メディアを所有する場合は) WebSphere Process Server DVD を用意します。

WebSphere Process Server 6.1 の場合、以下を用意する必要があります。

- WebSphere Process Server バージョン 6.1 のディスク 1 (2 枚のうち) または WebSphere Process Server バージョン 6.1 DVD
- WebSphere Process Server バージョン 6.1、WebSphere Application Server Network Deployment Supplements

製品インストール・メディアから、WebSphere Process Server の古いバージョン (バージョン 6.0.x) をホスティングするシステム上のディレクトリーへ、フ

ファイルをコピーする必要があります。これらのディレクトリー内のファイルは、WebSphere Process Server の新しいバージョンをインストールせずに、**WBIPreUpgrade** コマンドを実行するために使用できる特殊な環境を構成します。

2. WebSphere Process Server の古いバージョン (バージョン 6.0.x) をホスティングするローカル・システム上に、`migration_copy` という新規ディレクトリーを作成します。(このディレクトリーには任意の名前を付けることができます。ここでは、説明を分かりやすくするため「`migration_copy`」を使用します。)
3. 製品メディアから、ローカルのバージョン 6.0.x システム上に作成したばかりのディレクトリー `migration_copy` にファイルをコピーします。以下にリストするディレクトリー全体をコピーします。
 - a. WebSphere Process Server バージョン 6.1 のディスク 1 CD-ROM (2 枚のうち) または WebSphere Process Server バージョン 6.1 DVD のいずれかの `migration` というディレクトリー。
 - b. WebSphere Process Server バージョン 6.1、WebSphere Application Server Network Deployment Supplements CD の `migration` というディレクトリー。

注: このディレクトリー内のファイルと、WebSphere Process Server バージョン 6.1 メディアの `migration` ディレクトリー内のファイルを、ご使用のシステム上の単一の `migration` ディレクトリーに結合する必要があります。

- c. WebSphere Process Server バージョン 6.1、WebSphere Application Server Network Deployment Supplements CD の `JDK` というディレクトリー。

これで、システム上に以下のディレクトリー構造ができます。

```
migration_copy/  
  migration/  
    JDK/
```

4. `migration_copy/migration/bin` ディレクトリーにナビゲートします。
5. `migration_copy/bin` ディレクトリーから **WBIPreUpgrade** スクリプトを使用して現在の構成を保存します。詳しくは、14 ページの『WBIPreUpgrade コマンド』を参照してください。

古いバージョンをホスティングするシステム上にある、マイグレーション固有のバックアップ・ディレクトリーに構成を保存します。例えば、次のようになります。

Linux

UNIX

Linux または UNIX の場合:

```
./WBIPreUpgrade.sh /filepath/migration_specific_backup /opt/IBM/WBI602
```

Windows

Windows の場合:

```
WBIPreUpgrade C:¥filepath¥migration_specific_backup  
"C:¥Program Files¥IBM¥WBI602"
```

WBIPreUpgrade コマンドは、画面に状況を表示し、`migration_specific_backup` ディレクトリーのログ・ファイルに状況を出力します。ASCII ログ・ファイル名は `WBIPreUpgrade` というテキストで始まり、日付とタイム・スタンプを含みます。

6. 構成のすべてのサーバー・ノードを停止して、古い WebSphere Process Server リリースをシャットダウンします。
7. バックアップ・ディレクトリーを圧縮し (TAR や ZIP などのユーティリティーを使用)、FTP を使用してファイルを別のシステムにコピーします。
8. ホスト名を同じにして、新しいオペレーティング・システムをインストールします。

可能な場合、システム名およびパスワードを古いシステムと同じままにします。マイグレーションするアプリケーションに関連したデータベース・ファイルがあれば、前のシステムの場所と同じパスに配置します。一般的には、パスは同じにするようにします。パスまたは名前の変更が必要な場合、後のステップで説明するように、**WBIPostUpgrade** コマンドを実行する前に管理コンソールで変更を加えます。

9. FTP を使用してバックアップ・ディレクトリーを他のシステムからコピーし、解凍します。
10. 新しいバージョンの WebSphere Process Server をインストールします。

WebSphere Process Serverのインストールおよび構成を参照してください。

11. プロファイル管理ツールまたは **manageprofiles** コマンドを使用して、新しいバージョンの WebSphere Process Server で新規プロファイルを作成します。ノード名、セル名、および拡張レベルがバージョン 6.0.x プロファイルと同じ (さらに可能な場合はホスト名も同じ) である新規プロファイルを作成する必要があります。異なるホスト名を使用した場合、マイグレーション後に管理コンソールを使用して、ホスト名が構成されているその他の場所でそのホスト名を手動で更新する必要があります。プロファイルの作成については、『プロファイルの作成』を参照してください。
12. **WBIPostUpgrade** コマンドを、新規バージョンの *install_root/bin* ディレクトリーから実行します。

ステップ 9 で作成したバックアップ・ディレクトリーのコピーを指定します。正しいコマンド構文については、17 ページの『WBIPostUpgrade コマンド』を参照してください。以下に例を示します。

```
install_root¥bin¥WBIPostUpgrade wbi_installation¥migration
```

13. 必要に応じて、WebSphere Process Server で使用されているデータベースを手動で更新します。通常では、新しいバージョンの WebSphere Process Server で必要なデータベースの変更は、自動的に適用されます。サーバーが最初に始動されるときに、データベース・テーブルが新規スキーマ・バージョンにマイグレーションされます。ただし、サーバーがデータベース・スキーマにアクセスするための十分な権限を持たないか、その他のデータベース固有の要件が満たされていない場合、データベースを手動で更新する必要があります。詳しくは、32 ページの『マイグレーションのためのデータベースのアップグレード』を参照してください。

結果

WebSphere Process Server をサポートするオペレーティング・システムに構成をマイグレーションしました。

次のタスク

マイグレーションが正常に実行されたことを確認します。サーバーで Business Process Choreographer が構成されている場合は、132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』を参照してください。最後に、111 ページの『事後マイグレーション構成検査』で説明されている検査を実行します。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要がある追加タスクに注意してください。

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

1 ページの『マイグレーションの概要』

WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからマイグレーションします。

関連タスク

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』

WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

111 ページの『事後マイグレーション構成検査』

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

99 ページの『リモート・システムへのマイグレーション』

マイグレーション・ツールを使用して、あるシステムの古いバージョンから、別のリモート・システム上の新しいバージョンの WebSphere Process Server にマイグレーションします。(スタンドアロン・サーバーのみ。)

プロファイルの作成

新規の WebSphere Enterprise Service Bus または WebSphere Process Server プロファイルを作成する方法について説明します。プロファイルの作成は、manageprofiles コマンドを使用してコマンド行から行うことも、プロファイル管理ツールのグラフィカル・ユーザー・インターフェース (GUI) を使用して対話式に行うこともできます。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインス

インストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

関連情報



WebSphere Process Serverのインストールおよび構成

このセクションでは、IBM WebSphere Process Server のインストールの準備、インストールの実行、および構成を行う方法について説明します。これらの説明は、Linux、i5/OS、UNIX、および Windows システムを対象としています。

マイグレーションの検査

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

始める前に

マイグレーションされたサーバーが始動していることを確認してください。

プロシージャ

1. WBIPostUpgrade コマンドおよび WBIPreUpgrade.ant スクリプトのマイグレーション・ログ・ファイルを確認します。
 - a. *backupDirectory/logs/WBIPostUpgradetimestamp.log* ファイルで、以下のメッセージのいずれかがあるかどうかを調べます。(*backupDirectory* は、マイグレーション中に、マイグレーションされたデータがまず保管され、後で取り出されるディレクトリーであり、マイグレーション・ウィザードか、WBIPreUpgrade または WBIPostUpgrade コマンドで指定されています。)
 - MIGR0259I: マイグレーションは正常に完了しました。
 - MIGR0271W: マイグレーションは、1 つ以上の警告を伴って、正常に完了しました。
 - b. *backupDirectory/logs/WBIPreUpgrade.anttimestamp.log* ファイルで、「BUILD SUCCESSFUL」というメッセージがあるかどうかを調べます。

これらのログ・ファイルの両方で、上記のメッセージによって成功したことが示された場合に、マイグレーションが正常に行われたと見なすことができます。
2. プロファイルのログ・ファイルを確認します。
 - a. マイグレーションされたプロファイルに対応する *profile_root/logs/server_name* ディレクトリーにナビゲートします。
 - b. SystemOut.log ファイルを調べて、致命的エラーがないことを確認します。

- c. SystemErr.log ファイルを調べて、致命的エラーがないことを確認します。
3. logs ディレクトリーのログ・ファイルを確認します。例えば、スタンドアロン・サーバーのログは /WebSphere/V6R1/AppServer/profiles/default/logs ディレクトリーで確認します。
4. 管理コンソールで操作を確認します。
 - a. 管理コンソール (Integrated Solutions Console) を開きます。
 - b. ナビゲーション・パネルから「アプリケーション」>「エンタープライズ・アプリケーション」を選択します。
 - c. 右側のパネルで、リストされているすべてのアプリケーションが開始していること (緑の「開始済み」アイコンで示される) を確認します。
 - d. ナビゲーション・パネルから「リソース」>「JDBC」>「ビジネス・インテグレーション・データ・ソース (Business Integration Data Sources)」を選択します。
 - e. このパネルにリストされている WebSphere Process Server データ・ソースごとに、チェック・ボックスを選択してから、「テスト接続」を選択します。
 - f. データ・ソースごとに、「ノード Dmgr1Node1 にあるサーバー Dmgr1 上のデータ・ソース WPS_DataSource のテスト接続が成功しました。」に類似したメッセージが返されます。

次のタスク

マイグレーションが正常に行われた場合、サーバーの使用を開始できます。マイグレーションが正常に完了しなかった場合は、134 ページの『バージョン間のマイグレーションのトラブルシューティング』でトラブルシューティングの情報を参照してください。

関連タスク

111 ページの『環境のロールバック』

WebSphere Process Server バージョン 6.1 環境へのマイグレーション後に、バージョン 6.0.x 環境にロールバックできます。これによって、構成はマイグレーション前の状態に戻ります。環境のロールバック後に、マイグレーション・プロセスを再開できます。

134 ページの『バージョン間のマイグレーションのトラブルシューティング』

WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

関連情報

 [エンタープライズ・アプリケーションの管理](#)

コンソールの「エンタープライズ・アプリケーション」ページ (「アプリケーション」>「エンタープライズ・アプリケーション」とクリックして表示) を使用して、サーバーにインストールされているエンタープライズ・アプリケーションの表示および管理を行います。

事後マイグレーション構成検査

マイグレーション後に、いくつかの構成設定を確認する必要があります。設定を変更するか、さらにバージョン 6.1 サーバーを構成しなければならない場合があります。

始める前に

サーバーまたはクラスターのマイグレーションを完了し、マイグレーションが正常に行われたことを確認済みである必要があります。

このタスクについて

ご使用の環境に該当する場合は、以下の検査を実行します。

- バージョン 6.0.x で使用していた Lightweight Third Party Authentication (LTPA) セキュリティー設定を検査して、バージョン 6.1 セキュリティーが適切に設定されているか確認します。
- logs ディレクトリーの WBIPostUpgrade.log ファイルを調べ、マイグレーション・ツールによってマイグレーションされなかった JSP オブジェクトの詳細を確認します。

バージョン 6.1 が、JSP オブジェクトの構成レベルをサポートしていない場合、マイグレーション・ツールは出力の際にオブジェクトを認識して、ログに記録します。

- ご使用の Java 仮想マシンの設定を見直して、推奨ヒープ・サイズを使用していることを確認してください。『Java 仮想マシン設定』を参照してください。このリンクの情報は、WebSphere Process Server サーバーと WebSphere Application Server のサーバーに適用されます。
- Cloudscape データベースの自動マイグレーションの結果を検証し、ツールによって自動的にマイグレーションされていない Cloudscape データベースがあれば手動でマイグレーションします。詳しくは、118 ページの『Cloudscape データベースのマイグレーション』を参照してください。

環境のロールバック

WebSphere Process Server バージョン 6.1 環境へのマイグレーション後に、バージョン 6.0.x 環境にロールバックできます。これによって、構成はマイグレーション前の状態に戻ります。環境のロールバック後に、マイグレーション・プロセスを再開できます。

このタスクについて

一般に、マイグレーションを行っても、旧リリースの構成は何も変わりません。ただし、最小限の変更が行われる場合もあります。例えば、デプロイメント・マネージャーや管理対象ノードなどの変更で、これらの変更は元に戻すことが可能です。

以下のサブトピックで、このような場合について詳細に説明しています。

関連概念

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

関連タスク

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

デプロイメント・セルのロールバック

restoreConfig および **wsadmin** コマンドを使用して、マイグレーション済みの WebSphere Process Server バージョン 6.1 デプロイメント・セルを、バージョン 6.0.x にロールバックすることができます。これによって、構成はマイグレーション前の状態に戻ります。デプロイメント・セルをロールバックした後、マイグレーション・プロセスを再開できます。

始める前に

バージョン 6.0.x のデプロイメント・セルをマイグレーションする場合、マイグレーション後に以前の状態にロールバックできるようにするには、以下の操作を実行します。

1. WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
2. **backupConfig** コマンドまたは望ましいバックアップ・ユーティリティを使用して、既存の構成をバックアップします。

- **backupConfig** コマンドまたは望ましいユーティリティを実行して、バージョン 6.0.x デプロイメント・マネージャー構成をバックアップします。

重要: このバックアップした構成の正しい名前と場所をメモしておいてください。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『**backupConfig** コマンド』を参照してください。

- **backupConfig** コマンドまたは望ましいユーティリティを実行して、バージョン 6.0.x 管理対象ノード構成をバックアップします。

重要: これらのバックアップした各構成の正しい名前と場所をメモしておいてください。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『**backupConfig** コマンド』を参照してください。

3. デプロイメント・セルをマイグレーションします。

プロシージャ

1. WebSphere Process Server バージョン 6.1 環境で現在実行中のサーバーをすべて停止します。
2. バージョン 6.1 デプロイメント・マネージャーにマイグレーションしたとき、以前のデプロイメント・マネージャーを使用不可にすることを選択した場合、以下のいずれか 1 つの操作を実行します。

- a. **backupConfig** コマンドまたは望ましいバックアップ・ユーティリティを使用して、以前のデプロイメント・マネージャーの構成をバックアップした場合、**restoreConfig** コマンドまたは望ましいユーティリティを実行して、デプロイメント・マネージャーのバージョン 6.0.x 構成をリストアします。

重要: デプロイメント・マネージャーをマイグレーションした直前に作成した同じバックアップ構成をリストアするようにしてください。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『restoreConfig コマンド』を参照してください。

- b. 以前のデプロイメント・マネージャー構成をバックアップしなかった場合、**wsadmin** コマンドを使用して、バージョン 6.1 からロールバックする必要があるデプロイメント・マネージャーのバージョン 6.0.x `profile_root/bin` ディレクトリーから `migrationDisablementReversal.jacl` スクリプトを実行してください。

Linux 例えば、Linux 環境では以下のパラメーターを使用します。

```
./wsadmin.sh -f migrationDisablementReversal.jacl -conntype NONE
```

ヒント: `migrationDisablementReversal.jacl` スクリプトの実行に問題がある場合、スクリプト内のステップを手動で実行してみてください。

- 1) 以下のディレクトリーに移動します。

```
profile_root/config/cells/cell_name/nodes/node_name
```

ここで、`node_name` はロールバック対象のデプロイメント・マネージャー・ノードの名前です。

- 2) `serverindex.xml_disabled` ファイルがこのディレクトリーに表示された場合、以下の操作を実行します。

- a) `serverindex.xml` ファイルを削除するか名前変更します。

- b) `serverindex.xml_disabled` ファイルを `serverindex.xml` に名前変更します。

3. ロールバックが必要なデプロイメント・セルの管理対象ノードそれぞれについて、以下のいずれか 1 つの操作を実行します。

- a. **backupConfig** コマンドまたは望ましいバックアップ・ユーティリティを使用して、以前の管理対象ノードの構成をバックアップした場合、**restoreConfig** コマンドまたは望ましいユーティリティを実行して、管理対象ノードのバージョン 6.0.x 構成をリストアします。

重要: 管理対象ノードをマイグレーションした直前に作成した同じバックアップ構成をリストアするようにしてください。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『restoreConfig コマンド』を参照してください。

- b. 以前の管理対象ノード構成をバックアップしなかった場合、**wsadmin** コマンドを使用して、管理対象ノードのバージョン 6.0.x `profile_root/bin` ディレクトリーから `migrationDisablementReversal.jacl` スクリプトを実行してください。

Linux

例えば、Linux 環境では以下のパラメーターを使用します。

```
./wsadmin.sh -f migrationDisablementReversal.jacl -conntype NONE
```

ヒント: migrationDisablementReversal.jacl スクリプトの実行に問題がある場合、スクリプト内のステップを手動で実行してみてください。

1) 以下のディレクトリーに移動します。

```
profile_root/config/cells/cell_name/nodes/node_name
```

ここで、*node_name* はロールバックする管理対象ノードの名前です。

2) serverindex.xml_disabled ファイルがこのディレクトリーに表示された場合、以下の操作を実行します。

a) serverindex.xml ファイルを削除するか名前変更します。

b) serverindex.xml_disabled ファイルを serverindex.xml に名前変更します。

4. バージョン 6.1 デプロイメント・マネージャーが実行しているときに管理対象ノードも実行中の場合、管理対象ノードを同期化します。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『wsadmin ツールによるノードの同期化』を参照してください。

5. バージョン 6.1 へのマイグレーション中に、インストールしたアプリケーションを以前のリリースと同じ場所に保持するよう選択したとき、バージョン 6.1 のアプリケーションで以前のリリースとの互換性のないものがある場合は、互換性のあるアプリケーションをインストールしてください。

6. バージョン 6.1 プロファイルを削除します。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『プロファイルの削除』を参照してください。

7. データベースをロールバックします。(アップグレードされた WebSphere Process Server コンポーネントをサポートするデータベースに対して、マイグレーション・ツールによって自動で、または手動で、マイグレーション・プロセスを開始する前に作成したバックアップをリストアします。)

8. ロールバックしたデプロイメント・マネージャーとその管理対象ノードを、バージョン 6.0.x 環境で開始します。

結果

構成はマイグレーション前の状態に戻ります。

次のタスク

マイグレーション・プロセスを再開する必要がある場合は、ここで再開できます。

関連タスク

115 ページの『管理対象ノードのロールバック』

restoreConfig および **wsadmin** コマンドを使用して、マイグレーション済みの WebSphere Process Server バージョン 6.1 管理対象ノードを、マイグレーション前の状態にロールバックすることができます。ロールバックする各管理対象ノード

ドに対して、管理対象ノードそれ自体と、デプロイメント・マネージャーにあるマスター・リポジトリに加えた対応する変更をロールバックする必要があります。

50 ページの『マイグレーション・ウィザードを使用したデプロイメント・マネージャーのマイグレーション』

マイグレーション・ウィザードを使用して、デプロイメント・マネージャーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

 コマンド行ツールを使用したデプロイメント・マネージャーのマイグレーション

コマンド行ツールを使用して、デプロイメント・マネージャーを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

関連情報

restoreConfig コマンド

backupConfig コマンド

wsadmin ツールによるノードの同期化

プロファイルの削除

管理対象ノードのロールバック

restoreConfig および **wsadmin** コマンドを使用して、マイグレーション済みの WebSphere Process Server バージョン 6.1 管理対象ノードを、マイグレーション前の状態にロールバックすることができます。ロールバックする各管理対象ノードに対して、管理対象ノードそれ自体と、デプロイメント・マネージャーにあるマスター・リポジトリに加えた対応する変更をロールバックする必要があります。

始める前に

バージョン 6.0.x の管理対象ノードをマイグレーションする場合、マイグレーション後に以前の状態にロールバックできるようにするには、以下の操作を実行します。

1. WebSphere Process Server コンポーネントをサポートするデータベースをバックアップします。
2. **backupConfig** コマンドまたは望ましいバックアップ・ユーティリティを使用して、既存の構成をバックアップします。
 - **backupConfig** コマンドまたは望ましいユーティリティを実行して、バージョン 6.0.x デプロイメント・マネージャー構成をバックアップします。

重要: このバックアップした構成の正しい名前と場所をメモしておいてください。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『**backupConfig** コマンド』を参照してください。

- **backupConfig** コマンドまたは望ましいユーティリティを実行して、バージョン 6.0.x 管理対象ノード構成をバックアップします。

重要: このバックアップした構成の正しい名前と場所をメモしておいてください。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『 backupConfig コマンド』を参照してください。

3. 管理対象ノードをマイグレーションします。

必要な場合、マイグレーションしたばかりの管理対象ノードをロールバックすることができます。

重要: ロールバックするバージョン 6.0.x 管理対象ノードをマイグレーションする前の状態のバージョン 6.1 デプロイメント・マネージャー構成のバックアップ・コピーを持たない場合、この項目で説明する手順は使用できず、112 ページの『デプロイメント・セルのロールバック』で説明するようにセル全体をロールバックする必要があります。

このタスクについて

別の管理対象ノードのロールバックに進む前に、マイグレーション済みの管理対象ノードごとに、バックアップおよびロールバック操作をすべて実行する必要があります。

プロシージャ

1. データベースをロールバックします。(アップグレードされた WebSphere Process Server コンポーネントをサポートするデータベースに対して、マイグレーション・ツールによって自動で、または手動で、マイグレーション・プロセスを開始する前に作成したバックアップをリストアします。)
2. バージョン 6.1 環境で現在実行中のサーバーをすべて停止します。
3. 以前の構成をリストアします。
 - a. **restoreConfig** コマンドまたは望ましいユーティリティーを実行して、バージョン 6.1 デプロイメント・マネージャー構成をリストアします。

重要: 管理対象ノードをマイグレーションした直前に作成した同じバックアップ構成をリストアするようにしてください。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『 restoreConfig コマンド』を参照してください。

- b. 以下のアクションのいずれかを実行して、管理対象ノードのバージョン 6.0.x 構成をリストアします。
 - **restoreConfig** コマンドまたは望ましいユーティリティーを実行して、バージョン 6.0.x 構成をリストアします。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『 restoreConfig コマンド』を参照してください。

- **wsadmin** コマンドを使用して、管理対象ノードのバージョン 6.0.xprofile_root/bin ディレクトリーから migrationDisablementReversal.jacl スクリプトを実行してください。

Linux 例えば、Linux 環境では以下のパラメーターを使用します。

```
./wsadmin.sh -f migrationDisablementReversal.jacl -conntype NONE
```

ヒント: migrationDisablementReversal.jacl スクリプトの実行に問題がある場合、スクリプト内のステップを手動で実行してみてください。

1) 以下のディレクトリーに移動します。

```
profile_root/config/cells/cell_name/nodes/node_name
```

ここで、*node_name* はロールバックする管理対象ノードの名前です。

2) serverindex.xml_disabled ファイルがこのディレクトリーに表示された場合、以下の操作を実行します。

a) serverindex.xml ファイルを削除するか名前変更します。

b) serverindex.xml_disabled ファイルを serverindex.xml に名前変更します。

- バージョン 6.1 デプロイメント・マネージャーを始動します。
- 管理対象ノードを同期化します。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『wsadmin ツールによるノードの同期化』を参照してください。

- バージョン 6.1 へのマイグレーション中に、インストールしたアプリケーションを以前のリリースと同じ場所に保持するよう選択したとき、バージョン 6.1 のアプリケーションで以前のリリースとの互換性のないものがある場合は、互換性のあるアプリケーションをインストールしてください。
- バージョン 6.1 の管理対象プロファイルを削除します。

WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『プロファイルの削除』を参照してください。

- バージョン 6.1 環境で、ロールバックされた管理対象ノードを開始します。

結果

構成はマイグレーション前の状態に戻ります。

次のタスク

マイグレーション・プロセスを再開する必要がある場合は、ここで再開できます。

関連タスク

112 ページの『デプロイメント・セルのロールバック』

restoreConfig および **wsadmin** コマンドを使用して、マイグレーション済みの WebSphere Process Server バージョン 6.1 デプロイメント・セルを、バージョン 6.0.x にロールバックすることができます。これによって、構成はマイグレーション前の状態に戻ります。デプロイメント・セルをロールバックした後、マイグレーション・プロセスを再開できます。

61 ページの『マイグレーション・ウィザードを使用した非クラスター管理対象ノードのマイグレーション』

マイグレーション・ウィザードを使用して、非クラスター管理対象ノードを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

コマンド行ツールを使用した管理対象ノードのマイグレーション

コマンド行ツールを使用して、非クラスター管理対象ノードを WebSphere Process Server の古いバージョンから新しいバージョンにマイグレーションします。

関連情報

restoreConfig コマンド

backupConfig コマンド

wsadmin ツールによるノードの同期化

プロファイルの削除

Cloudscape データベースのマイグレーション

マイグレーション・ツールを使用して WebSphere Process Server バージョン 6.1 にマイグレーションした後、Cloudscape データベースの自動マイグレーションの結果を検査して、ツールによって自動的にマイグレーションされていない Cloudscape データベース・インスタンスがあれば、手動でマイグレーションします。

始める前に

1 ページの『マイグレーションの概要』および 3 ページの『事前マイグレーションの考慮事項』を参照してください。

ヒント:

- Cloudscape データベースをマイグレーションする前に、Cloudscape データベースを使用するアプリケーションをホスティングしているサーバーがシャットダウンされているか確認します。シャットダウンされていないサーバーがあると、Cloudscape のマイグレーションに失敗します。
- マイグレーション・ツールを実行する前に、デバッグ・マイグレーション・トレース がアクティブであるか確認します。デフォルトでは、このトレース機能は無効になっています。デバッグ・マイグレーション・トレースが無効の場合に再び有効化するには、以下のいずれか 1 つのトレース・オプションを設定します。
 - all traces*=all
 - com.ibm.ws.migration.WASUpgrade=all

このタスクについて

WebSphere Process Server バージョン 6.1 では Cloudscape バージョン 10.1 が必要です。

Cloudscape バージョン 10.1 は純粋な Java データベース・サーバーで、Apache Derby ランタイムを、IBM ソフトウェア・サポートの完全なサービスを使用できる機能と結合したものです。Cloudscape バージョン 10.1 に関する包括的な情報については、Cloudscape 製品の Web ページを参照してください。

マイグレーションするときの問題のトラブルシューティングのヘルプについては、134 ページの『バージョン間のマイグレーションのトラブルシューティング』を参照してください。

プロシージャ

1. Cloudscape データベース・インスタンスの自動マイグレーションを検査します。

WebSphere Process Server バージョン 6.0.x から バージョン 6.1 にマイグレーションすると、マイグレーション・ツールは、UDDI レジストリーなどの一部の内部コンポーネントによって、組み込みのフレームワークを経由してアクセスされるデータベース・インスタンスを自動的にアップグレードします。またこのツールでは、ご使用のアプリケーションが組み込みフレームワークを経由してアクセスする Cloudscape インスタンスもアップグレードしようとしています。マイグレーション・ツールを実行した後、これらのマイグレーションの結果を検査する必要があります。

120 ページの『Cloudscape v10.1.x 自動マイグレーションの検査』を参照してください。

2. 必要に応じて、Cloudscape データベース・インスタンスを手動でマイグレーションします。

バージョン 6.1 マイグレーション・ツールは、Cloudscape ネットワーク・サーバー・フレームワークを経由してアプリケーションとの間で処理を行うデータベース・インスタンスの自動マイグレーションを試行しません。このように除外することで、WebSphere Process Server からアクセスするデータベース・インスタンスと同じデータベース・インスタンスにアクセスするサード・パーティー・アプリケーションを破損するリスクがなくなります。

Cloudscape ネットワーク・サーバー・フレームワーク経由でアクセスするデータベース・インスタンスと、自動マイグレーションに失敗した Cloudscape インスタンスを手動でマイグレーションする方法についての詳細は、125 ページの『手動による Cloudscape のアップグレード』を参照してください。

3. UDDI レジストリーが Cloudscape ネットワーク・サーバー・フレームワーク上のデータベースを使用する場合、UDDI レジストリーを手動でマイグレーションします。

129 ページの『UDDI レジストリーのマイグレーション』を参照してください。

関連概念

24 ページの『旧バージョンからマイグレーションするときのデータ処理方法』WebSphere Process Server のバージョン間マイグレーション・ツールは、さまざまなデータ・セット (エンタープライズ・アプリケーション・データ、構成データ、およびシステム・アプリケーション・データ) をそれぞれ異なる方法で処理します。

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

1 ページの『マイグレーションの概要』

WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからマイグレーションします。

関連タスク

32 ページの『マイグレーションのためのデータベースのアップグレード』
マイグレーションと同時に、一部の WebSphere Process Server コンポーネントのデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

『Cloudscape v10.1.x 自動マイグレーションの検査』

WebSphere Process Server バージョン 6.1.x を使用するには、バージョン v10.1.x 以上の Cloudscape を実行する必要があります (Cloudscape v10.1.x は、Apache Derby バージョン 10.1 のコード・ベースで構成されていることに注意してください。) WebSphere Process Server バージョン 6.1 のアップグレード中に、マイグレーション・ツールは、UDDI レジストリーなどの一部の内部コンポーネントによって、組み込みフレームワークを経由してアクセスされるデータベース・インスタンスを自動的にアップグレードします。また、このツールは、使用中のアプリケーションが組み込みフレームワークを経由してアクセスする Cloudscape インスタンスもアップグレードしようとしています。これらのバックエンド・データベースについて、マイグレーション結果を検査する必要があります。

129 ページの『UDDI レジストリーのマイグレーション』

ほとんどのシナリオにおいて、既存の UDDI レジストリーのマイグレーションは、現行レベルの WebSphere Process Server にマイグレーションするとき自動的に実行されます。ただし、既存の UDDI レジストリーでネットワーク Cloudscape データベースを使用しているか、または DB2 UDDI バージョン 2 データベースを使用している場合、手動でのステップがいくつか必要になります。

関連情報

IBM Cloudscape 製品 Web ページ

Cloudscape マイグレーション文書

Cloudscape v10.1.x 自動マイグレーションの検査

WebSphere Process Server バージョン 6.1.x を使用するには、バージョン v10.1.x 以上の Cloudscape を実行する必要があります (Cloudscape v10.1.x は、Apache Derby バージョン 10.1 のコード・ベースで構成されていることに注意してください。) WebSphere Process Server バージョン 6.1 のアップグレード中に、マイグレーション・ツールは、UDDI レジストリーなどの一部の内部コンポーネントによって、組み込みフレームワークを経由してアクセスされるデータベース・インスタンスを自動的にアップグレードします。また、このツールは、使用中のアプリケーションが組み込みフレームワークを経由してアクセスする Cloudscape インスタンスもアップグレードしようとしています。これらのバックエンド・データベースについて、マイグレーション結果を検査する必要があります。

始める前に

Cloudscape v10.1.x を実動データベースとして使用しないでください。これは開発およびテスト目的としてのみ使用してください。

詳細情報: 新しいバージョンの Cloudscape は、IBM Quality Assurance (QA) および各国語サポート (NLS) などの追加のサービスを Derby ランタイムと組み合わせています。Cloudscape v10.1.x オープン・ソース・コード・ベースについての情報は、Cloudscape 製品 Web ページを参照してください。

マイグレーション・ツールは、組み込みフレームワークのみを経由してアクセスする Cloudscape データベース・インスタンスをアップグレードしようとします。Derby ネットワーク・サーバー・フレームワークでサーバーとの間で処理を行う Cloudscape インスタンスは、手動でアップグレードする必要があります。(125 ページの『手動による Cloudscape のアップグレード』を参照してください。) この要件によって、ネットワーク・サーバー・フレームワークを使用して、WebSphere Process Server と同じデータベース・インスタンスにアクセスするサード・パーティー・アプリケーションを破損するリスクがなくなります

他のアプリケーションはネットワーク・サーバー上の Cloudscape にアクセスできませんが、これは、このフレームワークによって接続ソフトウェアの基盤がデータベースに提供されるためです。組み込みフレームワークでは提供されません。Cloudscape ネットワーク・サーバーは複数の Java 仮想マシン (JVM) (またはサーバー) と同時に相互作用できますが、組み込みフレームワーク上の Cloudscape は単一の JVM としか作業できません。WebSphere Process Server のクラスターまたは共存インプリメンテーションには、ネットワーク・サーバーが必要です。詳しくは、『IBM Cloudscape インフォメーション・センター』を参照してください。

このタスクについて

組み込みフレームワークを経由してアプリケーションがアクセスするデータベース・インスタンスの場合、自動マイグレーションは完全に成功するか、完全に失敗するか、または警告を出して成功するかのいずれかです。マイグレーションで警告メッセージが出された場合、ご使用のデータを含む Cloudscape v10.1.x データベースが作成されますが、以下に示す構成済みのロジックなどの設定がすべてマイグレーションされるわけではありません。

- キー
- チェック
- ビュー
- トリガー
- 別名
- ストアード・プロシージャ

部分的に成功したマイグレーションと完全に成功したマイグレーションを識別するには、一般のアップグレード後のログと個別のデータベース・ログの両方を確認して、自動マイグレーションの結果を検査する必要があります。これらのタスクを実行すると、部分的にマイグレーションされたデータベースと、自動マイグレーションが完全に失敗したデータベースをトラブルシューティングするための重要な診断データが取得できます。最終的には、これらのデータベースは手動プロセスでマイグレーションします。

プロシージャ

1. それぞれの新しい WebSphere Process Serverバージョン 6.1.x プロファイルのアップグレード後のログを開きます。 ログのパス名は `install_rootprofiles/profileName/logs/WASPostUpgrade.timestamp.log` です。
2. アップグレード後のログを検査し、データベース・エラー・メッセージがあるかどうか調べます。 これらの例外は、データベース・マイグレーションの失敗を示すものです。以下の行はアップグレード後のログ内容の例で、データベースのエラー・コードは DSRA7600E です。(マイグレーション・ツールでは、すべてのデータベースの例外に接頭部 DSRA が付いています。)

```
MIGR0344I: Processing configuration file /opt/WebSphere51/AppServer/cloudscape/db2j.properties.
```

```
MIGR0344I: Processing configuration file /opt/WebSphere51/AppServer/config/cells/migr06/applications/MyBankApp.ear/deployments/MyBankApp/deployment.xml.
```

```
DSRA7600E: Cloudscape migration of database instance /opt/WebSphere61/Express/profiles/default/databases/_opt_WebSphere51_AppServer_bin_DefaultDB failed, reason: java.sql.SQLException: Failure creating target db
```

```
MIGR0430W: Cloudscape Database /fvt/temp/51BaseXExpress/PostUpgrade50BaseFVTTest9/testRun/pre/websphere_backup/bin/DefaultDB failed to migrate <new database name>
```

重要: WebSphere 内部コンポーネント (つまり、アプリケーションの 1 つではなく、WebSphere Process Server を構成するコンポーネント) によってアクセスされる Cloudscape インスタンスのマイグレーション障害メッセージが表示された場合、IBM WebSphere Process Server サポートに連絡してください。

3. バックエンドの Cloudscape データベースにそれぞれ対応する個々のデータベース・マイグレーション・ログを開きます。これらのログには、一般的なアップグレード後のログと同じタイム・スタンプが付いています。このログには、一般的なアップグレード後のログにリストされるものよりも詳細なエラーが表示されるほか、一般的なログに記載されていないエラーも記述されます。

それぞれのデータベース・ログのパス名は `WAS_HOME/profiles/profileName/logs/myFullDbPathName_migrationLogtimestamp.log` です。

4. 各データベース・マイグレーション・ログを検査して、エラーがないか調べます。マイグレーションが完全に正常な場合、ログには以下のテキストに似たメッセージが表示されます。

```
MIGR0429I: Cloudscape Database F:%temp%51BaseXExpress%PostUpgrade50BaseFVTTest2%testRun%pre%websphere_backup%bin%DefaultDB was successfully migrated. See log C:%WebSphere61%Express%profiles%default%logs%DefaultDB_migrationLogSun-Dec-18-13.31.40-CST-2005.log
```

そうでない場合、ログには以下の例のような形式のエラー・メッセージが表示されます。

```
connecting to source db <jdbc:db2j:/fvt/temp/51BaseXExpress/PostUpgrade50BaseFVTTest9/testRun/pre/websphere_backup/bin/DefaultDB>
```

```
connecting to source db <jdbc:db2j:/fvt/temp/51BaseXExpress/PostUpgrade50BaseFVTTest9/testRun/pre/websphere_backup/bin/DefaultDB> took 0.26 seconds
```

```
creating target db <jdbc:derby:/opt/WebSphere61/Express/profiles/default/databases/_opt_WebSphere51_AppServer_bin_DefaultDB>
```

```
ERROR: An error occurred during migration. See debug.log for more details.
```

```
shutting down databases
```

```
shutting down databases took 0.055 seconds
```

5. マイグレーション・エラーに関する詳細なデータについては、データベース・マイグレーション・ログに対応するデバッグ・ログを確認してください。

WebSphere Application Server マイグレーション・ユーティリティーは、デフォルトでデバッグ・マイグレーション・トレース をトリガーします。このトレース機能は、データベース・デバッグ・ログを生成します。デバッグ・ログの絶対パス名は `WAS_HOME/profiles/profileName/logs/myFullDbPathName_migrationDebugtimestamp.log` です。

以下の行は、デバッグ・テキストのサンプルです。この行には、前述のデータベース・マイグレーション・ログ・データのサンプルに記載されているエラーの、詳細な例外データが表示されています。

```
java.sql.SQLException: Database_opt_WebSphere51_AppServer_bin_DefaultDB already exists.
Aborting migration
at com.ibm.db2j.tools.migration.MigrateFrom51Impl.go(Unknown Source)
at com.ibm.db2j.tools.migration.MigrateFrom51Impl.doMigrate(Unknown Source)
at com.ibm.db2j.tools.MigrateFrom51.doMigrate(Unknown Source)
at com.ibm.ws.adapter.migration.CloudscapeMigrationUtility.migr
```

結果

- WebSphere Process Server マイグレーション・ユーティリティーは、アプリケーションによってアクセスされるデータベース・インスタンスを正常にマイグレーションしたかどうかにかかわらず、Cloudscape JDBC 構成を変更します。このツールは、Cloudscape JDBC プロバイダー・クラス・パス、データ・ソースのインプリメンテーション・クラス、およびデータ・ソースのヘルパー・クラスを変更します。以下の表に、これらの変更内容を示します。

表 1. 新しいクラス情報

クラス・タイプ	古い値	新しい値
JDBC プロバイダー・クラス・パス	<code>\${CLOUDSCAPE_JDBC_DRIVER_PATH}/db2j.jar</code>	<code>\${DERBY_JDBC_DRIVER_PATH}/derby.jar</code> <ul style="list-style-type: none"> • ここで、<code>DERBY_JDBC_DRIVER_PATH</code> は Cloudscape JDBC プロバイダーを定義する WebSphere 環境変数です。 • また、<code>derby.jar</code> は JDBC ドライバー・クラス・ファイルのベース名です (使用中の環境では、絶対パス名で JDBC ドライバー・クラス・ファイルを参照します。)
データ・ソースのインプリメンテーション・クラス: 接続プール	<code>com.ibm.db2j.jdbc.DB2jConnectionPool DataSource</code>	<code>org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource</code>
データ・ソースのインプリメンテーション・クラス: XA	<code>com.ibm.db2j.jdbc.DB2jXADataSource</code>	<code>org.apache.derby.jdbc.EmbeddedXADataSource</code>
データ・ソースのヘルパー・クラス	<code>com.ibm.websphere.rsadapter.CloudscapeDataStoreHelper</code>	<code>com.ibm.websphere.rsadapter.DerbyDataStoreHelper</code>

また、`db2j.properties` ファイルでは以下が変更されます。

- `WAS_HOME/cloudscape/dbj.properties` の名前が `WAS_HOME/derby/derby.properties` に変更されます
- ファイル上で、プロパティ名が `db2j.drda.*` から `derby.drda.*` に変更されます。
- データベースのマイグレーションが部分的または完全に成功した場合、以下の例に従ってデータベースの場所と名前が変更されます。
 - 古いデータベース名: `c:¥temp¥mydb`
 - 新しいデータベース名: 新しい名前には、古いデータベースのパス名全体と、マイグレーション・タイム・スタンプを結合したハッシュ・コードが含まれません。新しい名前には、古いデータベース名とタイム・スタンプがそのまま含まれます。例:
`install_root¥profiles¥profile_name¥databases¥my_db_hashCode_timestamp`

パス名の正確なメモ: 部分的なマイグレーションと失敗したマイグレーションの両方の場合、ログ・メッセージには、手動マイグレーションの実行に使用する必要がある正しい新旧のデータベース・パス名が含まれています。これらの新しいパス名を正しくメモします。

部分的なマイグレーションの場合、Cloudscape についての専門的な知識がある場合に限り、新しい v10.1.x データベースのトラブルシューティングを試行するようにしてください。そうでない場合、新しいデータベースを削除します。自動マイグレーションに完全に失敗した各データベースについて実行するのと同様に、元のデータベースで手動マイグレーション手順を実行します。詳しくは、125 ページの『手動による Cloudscape のアップグレード』を確認してください。

正しくマイグレーションした Cloudscape インスタンスの場合、セルを有効範囲とする新しいデータ・ソースは、バージョン 6.0.2 以降の WebSphere Process Server を実行するノードのみで使用できることに注意してください。以前のバージョンの製品は、新しい Cloudscape をサポートしません。バージョン 6.0.2 より前のノード上のアプリケーションが Cloudscape 10.1.x データ・ソースにアクセスしようとする、サーバーはランタイム例外を発行します。

関連タスク

125 ページの『手動による Cloudscape のアップグレード』

WebSphere Process Server バージョン 6.1 のアップグレード中に、マイグレーション・ツールは、組み込みフレームワークのみを経由してアクセスする Cloudscape のインスタンスをアップグレードしようとします。(新しいバージョンの Cloudscape はバージョン 10.1.x で、Derby をベースにしています。) 自動アップグレードでは、ネットワーク・サーバー・フレームワークを経由してアプリケーションとの間で処理を行う Cloudscape インスタンスは除外されます。このように除外することで、WebSphere Process Server と同じデータベース・インスタンスにアクセスするサード・パーティー・アプリケーションを破損するリスクがなくなります。ネットワーク・サーバー・フレームワーク経由でアクセスするデータベース・インスタンスは手動でアップグレードする必要があります。自動マイグレーションに失敗したデータベースについても、同じ操作を行います。

118 ページの『Cloudscape データベースのマイグレーション』

マイグレーション・ツールを使用して WebSphere Process Server バージョン 6.1 にマイグレーションした後、Cloudscape データベースの自動マイグレーションの結果を検査して、ツールによって自動的にマイグレーションされていない Cloudscape データベース・インスタンスがあれば、手動でマイグレーションします。

129 ページの『UDDI レジストリーのマイグレーション』

ほとんどのシナリオにおいて、既存の UDDI レジストリーのマイグレーションは、現行レベルの WebSphere Process Server にマイグレーションするときに自動的に実行されます。ただし、既存の UDDI レジストリーでネットワーク Cloudscape データベースを使用しているか、または DB2 UDDI バージョン 2 データベースを使用している場合、手動でのステップがいくつか必要になります。

関連情報

IBM Cloudscape 製品 Web ページ

Cloudscape マイグレーション文書

手動による Cloudscape のアップグレード

WebSphere Process Server バージョン 6.1 のアップグレード中に、マイグレーション・ツールは、組み込みフレームワークのみを経由してアクセスする Cloudscape のインスタンスをアップグレードしようとします。(新しいバージョンの Cloudscape はバージョン 10.1.x で、Derby をベースにしています。) 自動アップグレードでは、ネットワーク・サーバー・フレームワークを経由してアプリケーションとの間で処理を行う Cloudscape インスタンスは除外されます。このように除外することで、WebSphere Process Server と同じデータベース・インスタンスにアクセスするサード・パーティー・アプリケーションを破損するリスクがなくなります。ネットワーク・サーバー・フレームワーク経由でアクセスするデータベース・インスタンスは手動でアップグレードする必要があります。自動マイグレーションに失敗したデータベースについても、同じ操作を行います。

始める前に

Cloudscape v10.1.x を実動データベースとして使用しないでください。これは開発およびテスト目的としてのみ使用してください。

詳細情報: 新しいバージョンの Cloudscape は、IBM Quality Assurance (QA) および各国語サポート (NLS) などの追加のサービスを Derby ランタイムと組み合わせています。

- Cloudscape v10.1.x オープン・ソース・コード・ベースについての情報は、Cloudscape 製品 Web ページを参照してください。
- Cloudscape v10.1.x と v5.1.60x (および v5.1.60x 以前のバージョン) との非互換性についての情報は、「Migrating IBM Cloudscape to Version 10」を参照してください。

組み込みフレームワーク経由でアクセスする Cloudscape のインスタンスについては、どのインスタンスが自動アップグレード・プロセスで完全に失敗したか、また部分的にアップグレードされたかを判別してください。120 ページの『Cloudscape v10.1.x 自動マイグレーションの検査』のトピックでは、データベース・エラーおよび診断データを、各種マイグレーション・ログから見つける方法が説明されています。ログ・メッセージには、手動マイグレーションの実行に使用する必要がある正しい新旧のデータベース・パス名が含まれています。これらの新しいパス名を正しくメモします。

自動マイグレーション・プロセス中に部分的にアップグレードされたデータベースについて、マイグレーション・エラーのリスクを最小限にするには、新しいデータベースを削除します。ログ診断データに従って元のデータベースをトラブルシューティングした後、元のデータベースで手動マイグレーションを実行します。

このタスクについて

以下のセクションでは、両方のフレームワーク (組み込みフレームワークとネットワーク・サーバー・フレームワーク) を経由してアクセスする Cloudscape インスタンスをマイグレーションするためのステップを示します。Cloudscape ネットワー

ク・サーバー・フレームワークにのみ適用されるステップには、それを示すマークが付けられています。マイグレーションのベスト・プラクティスとして、使用するユーザー ID が以下のいずれかの権限を持つようにしてください。

- Cloudscape インスタンスにアクセスするサーバーの管理者
- データベース・インスタンスにアクセスできる umask

権限がない場合、データベース・インスタンスが読み取り専用であることを示すランタイム・エラーが表示されることがあります。

プロシージャ

1. **ネットワーク・サーバー・フレームワークのみ:** Cloudscape データベースのすべてのクライアントが Cloudscape v10.1.x をサポートできるようにします。データベースの WebSphere Process Server クライアントが実行する WebSphere Process Server は、バージョン 6.0.1.x またはそれ以上である必要があります。

ノードが混在したセルの場合、WebSphere Process Server バージョン 6.0.1.x 以降のノードのみが、Cloudscape 10.1.x にアクセスするために、マイグレーション後に作成するデータ・ソースを使用できます。以前のバージョンの製品は、新しい Cloudscape をサポートしません。WebSphere Process Server のバージョン 6.0.1.x より前のノード上のアプリケーションが、セルを有効範囲とする Cloudscape 10.1.x データ・ソースにアクセスしようとすると、WebSphere Process Server はランタイム例外を発行します。

2. **ネットワーク・サーバー・フレームワークのみ:** データベースをオフラインにします。クライアントはマイグレーション・プロセス中にデータベースにアクセスできません。
3. WebSphere Process Server が提供するサンプルの Cloudscape マイグレーション・スクリプト (db2jmigrate.bat または db2jmigrate.sh) を検査します。両方のスクリプトについて、パスは `install_root%derby%bin%embedded%...` です。使用中の環境の要件に応じて、スクリプトを変更できます。スクリプトで利用できるオプションについては、「Migrating IBM Cloudscape to Version 10」を参照してください。例えば、オプション `-DB2j.migrate.ddlFile=filename` を使用して、新しいデータベースの DDL ファイルを指定できます。
4. マイグレーション・スクリプトを実行するときにデータベース・デバッグ・ログを生成するには、デバッグ・マイグレーション・トレース がアクティブであるか確認します。デフォルトでは、このトレース機能は有効になっています。デバッグ・トレースが使用不可になっている場合、再アクティブ化します。
 - a. 管理コンソールでトレース・オプションを設定するには、コンソール・ナビゲーション・ツリーで「トラブルシューティング」 > 「ロギングおよびトレース」をクリックします。
 - b. サーバー名を選択します。
 - c. 「ログ・レベルの詳細の変更 (Change Log Level Details)」をクリックします。
 - d. オプション: 「すべてのコンポーネント」が有効にされている場合、これをオフにして特定のコンポーネントを使用可能に設定することが必要な場合もあります。
 - e. オプション: コンポーネントまたはグループ名を選択します。詳細については、WebSphere Application Server Network Deployment バージョン 6.1 イン

フォメーション・センターの『ログ・レベル設定』を参照してください。選択されたサーバーが実行中でない場合、グラフィック・モードで個別のコンポーネントを表示できません。

- f. トレース・ストリング・ボックスにトレース・ストリングを入力します。この場合、以下のいずれか 1 つを入力します。

- all traces*=all
- com.ibm.ws.migration.WASUpgrade=all

トレースについては、WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの『トレースによる処理』を読んでください。

- g. 「適用」を選択し、次に「OK」を選択します。

5. スクリプトを実行するときに、古いデータベース名と、新しいデータベース名のマイグレーション後の絶対パスを指定します。例:

```
E:¥WebSphere¥ProcServer¥derby¥bin¥embedded>db2jMigrate.bat myOldDB
```

myNewDB 自動マイグレーションで生成されたログには、古いデータベースとターゲット・データベースの両方を指定するための正確なパス名が記載されています。新しいデータベースを指定するには、このターゲット・データベース名を使用する必要があります。なぜなら、マイグレーション済みの Cloudscape データ・ソース (WebSphere Process Server マイグレーション・ユーティリティによって更新されたもの) は、このターゲット・データベース名を指すようになっているためです。ログ・メッセージにターゲット・データベース名がどのように表示されるかを以下のサンプル・テキストで示します。

```
Cloudscape migration of database instance C:¥temp¥migration2¥profiles¥Srv01¥
installedApps\ghongellNode01Cell\DynamicQuery.ear\EmployeeFinderDB to
new database instance C:\WebSphere\ProcServer
\profiles\Srv01\databases\C_WAS602_ProcServer_profiles_ProcSrv01_
installedApps_ghongellNode01Cell_DynamicQuery.ear_
EmployeeFinderDB failed, reason: java.sql.SQLException:
Failure creating target db
```

ネットワーク・サーバー・フレームワーク経由でアクセスする Cloudscape のインスタンスの場合は、新しいデータベースに付ける任意の名前を入力します。新しいデータベース名を指すように、既存のデータ・ソースを変更するようにしてください。

6. マイグレーション・プロセスが終了したら、データベース・マイグレーション・ログを確認して、結果を検査してください。各データベース・マイグレーション・ログのパス名は、*install_root/logs/derby/myFulldbName_migrationLog.log* です。

マイグレーションが正常な場合、データベース・マイグレーション・ログには、以下のテキストに似たメッセージが表示されます。

```
Check E:¥WebSphere¥ProcServer¥derby¥myOldDB_migrationLog.log for progress
Migration Completed Successfully
E:¥WebSphere¥ProcServer¥derby¥bin¥embedded>
```

そうでない場合、ログには以下のような形式のエラー・メッセージが表示されます。

```

Check E:\WebSphere\ProcServer\derby\my01ddb_migrationLog.log for progress
ERROR: An error occurred during migration. See debug.log for more details.
ERROR XMG02: Failure creating target db
java.sql.SQLException: Failure creating target db
    at com.ibm.db2j.tools.migration.MigrationState.getCurrSQLException(Unknown
    Source)
    at com.ibm.db2j.tools.migration.MigrateFrom51Impl.handleException(Unknown
    Source)
    at com.ibm.db2j.tools.migration.MigrateFrom51Impl.go(Unknown Source)
    at com.ibm.db2j.tools.migration.MigrateFrom51Impl.main(Unknown Source)
    at com.ibm.db2j.tools.MigrateFrom51.main(Unknown Source)

```

7. マイグレーション・エラーに関する詳細なデータについては、データベース・マイグレーション・ログに対応するデバッグ・ログを確認してください。デバッグ・ログ・ファイルの絶対パス名は `install_root/logs/derby/myFulldbName_migrationDebug.log` です。

以下の行は、デバッグ・テキストのサンプルです。

```

sourceDBURL=jdbc:db2j:E:\WebSphere\my01ddb
newDBURL=jdbc:derby:e:\tempo\myNewDB
ddlOnly=false
connecting to source db <jdbc:db2j:E:\WebSphere\my01ddb>
connecting to source db <jdbc:db2j:E:\WebSphere\my01ddb> took 0.611 seconds
creating target db <jdbc:derby:e:\tempo\myNewDB>
creating target db <jdbc:derby:e:\tempo\myNewDB> took 6.589 seconds
initializing source db data structures
initializing source db data structures took 0.151 seconds
recording DDL to create db <E:\WebSphere\my01ddb>
recording DDL to create db <E:\WebSphere\my01ddb> took 5.808 seconds

```

結果

前のステップに示したように、データベース・マイグレーション・ログには Migration Completed Successfully メッセージか、またはマイグレーション失敗の例外を含むメッセージのいずれかが表示されます。

次のタスク

- マイグレーションに失敗したデータベースについては、ログに記録されたエラー・データに従ってトラブルシューティングします。その後、マイグレーション・スクリプトを再実行します。
- 正常にアップグレードしたデータベースに組み込みフレームワークを経由してアクセスするには、新しいデータベース名を指すようにデータ・ソースを変更します。
- 正常にアップグレードしたデータベースにネットワーク・サーバー・フレームワークを経由してアクセスするには、DB2 Universal JDBC ドライバーまたは Derby Client JDBC ドライバーのいずれかを使用できます。
 - 既存の JDBC 構成で DB2 Universal JDBC ドライバーを使用し続ける場合、新しいデータベース名を指すようにデータ・ソースを変更します。
 - XA データ・ソースをサポートできる Derby Client JDBC ドライバーを使用する場合、新しい Derby Client JDBC ドライバー・クラスおよび新しいデータ・ソースのインプリメンテーション・クラスを使用するよう JDBC プロバイダーを変更します。次に、正しい Derby データ・ソースのヘルパー・クラスを使用し、かつ新しいデータベース名を指すように、既存のそれぞれのデータ・ソースを再構成します。

新しいすべてのクラス名については、WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターのトピック [ベンダー固有データ・ソースで最低限必要な設定を確認してください](#)。

- `install_root/dbscripts/component_name/Derby`ディレクトリーでデータベース・アップグレード・スクリプトを実行して、データベース・テーブルおよびスキーマを WebSphere Process Server バージョン 6.1 レベルにアップグレードします。詳しくは、32 ページの『マイグレーションのためのデータベースのアップグレード』を参照してください。

関連タスク

120 ページの『Cloudscape v10.1.x 自動マイグレーションの検査』

WebSphere Process Server バージョン 6.1.x を使用するには、バージョン v10.1.x 以上の Cloudscape を実行する必要があります (Cloudscape v10.1.x は、Apache Derby バージョン 10.1 のコード・ベースで構成されていることに注意してください。) WebSphere Process Server バージョン 6.1 のアップグレード中に、マイグレーション・ツールは、UDDI レジストリーなどの一部の内部コンポーネントによって、組み込みフレームワークを経由してアクセスされるデータベース・インスタンスを自動的にアップグレードします。また、このツールは、使用中のアプリケーションが組み込みフレームワークを経由してアクセスする Cloudscape インスタンスもアップグレードしようとします。これらのバックエンド・データベースについて、マイグレーション結果を検査する必要があります。

『UDDI レジストリーのマイグレーション』

ほとんどのシナリオにおいて、既存の UDDI レジストリーのマイグレーションは、現行レベルの WebSphere Process Server にマイグレーションするとき自動的に実行されます。ただし、既存の UDDI レジストリーでネットワーク Cloudscape データベースを使用しているか、または DB2 UDDI バージョン 2 データベースを使用している場合、手動でのステップがいくつか必要になります。

32 ページの『マイグレーションのためのデータベースのアップグレード』

マイグレーションと同時に、一部の WebSphere Process Server コンポーネントのデータベース・スキーマをアップグレードする必要があります。これは自動的に実行させることができますが、場合によっては手動でスキーマをアップグレードしなければなりません。

関連情報

[IBM Cloudscape 製品 Web ページ](#)

[Cloudscape マイグレーション文書](#)

[Migrating IBM Cloudscape to Version 10](#)

[ログ・レベル設定](#)

[トレースの使用](#)

[ベンダー固有データ・ソースで最低限必要な設定](#)

UDDI レジストリーのマイグレーション

ほとんどのシナリオにおいて、既存の UDDI レジストリーのマイグレーションは、現行レベルの WebSphere Process Server にマイグレーションするとき自動的に実行されます。ただし、既存の UDDI レジストリーでネットワーク Cloudscape デー

データベースを使用しているか、または DB2 UDDI バージョン 2 データベースを使用している場合、手動でのステップがいくつか必要になります。

始める前に

WebSphere Process Server のインストール済み環境をマイグレーションします。アプリケーションをマイグレーションするオプションを選択して、UDDI レジストリー・アプリケーションがマイグレーションされるようにします。

このタスクについて

既存の UDDI レジストリーで Oracle、組み込み Cloudscape、または DB2 UDDI バージョン 3 データベースを使用している場合、手動マイグレーションを実行する必要はありません。WebSphere Process Server をマイグレーションして、マイグレーション後に UDDI ノードを最初に始動したときに、マイグレーションは自動的に実行されます。

既存の UDDI レジストリーでネットワーク Cloudscape データベースを使用しているか、または DB2 UDDI バージョン 2 データベースを使用している場合、レジストリーをマイグレーションするには手動ステップをいくつか実行する必要があります。

- UDDI レジストリーで DB2 UDDI バージョン 2 データベースを使用している場合、『バージョン 3 の UDDI レジストリーへのマイグレーション』およびサブトピックのステップに従ってください。
- UDDI レジストリーでネットワーク Cloudscape データベースを使用している場合、以下のステップを実行してください。
 1. 異なるレベルの WebSphere Process Server のサーバーを持つクラスターがある場合、WebSphere Process Server バージョン 6.1 のサーバー上で UDDI レジストリーが実行するようにします。例えば、2 つのノードにまたがるクラスターがあって、UDDI レジストリーを実行しているサーバーが バージョン 6.1 である場合、1 つのノードを WebSphere Process Server バージョン 6.1 にアップグレードして、別のノードを以前のレベルに維持することができます。
 2. 関連する UDDI ノードを初期化します。初期化プロセスによって、一部の UDDI レジストリーのマイグレーションが実行されます。
 3. データベース管理者として、*install_root/cloudscape/lib* から以下のコマンドを入力します。

```
java -cp db2j.jar;db2jtools.jar com.ibm.db2j.tools.ij
connect 'jdbc:db2j:uddi_cloudscape_database_path';

run 'install_root/UDDIReg/databaseScripts/uddi30crt_drop_triggers
_cloudscape.sql';

quit;

cd install_root/derby/migration

java -cp db2j.jar;db2jmigration.jar;../lib/derby.jar
com.ibm.db2j.tools.MigrateFrom51
jdbc:db2j:uddi_cloudscape_database_path
```

ここで、

- `uddi_cloudscape_database_path` は既存の Cloudscape データベースの絶対パスで、例えば `install_root/profiles/profile_name/databases/com.ibm.uddi/UDDI30` となります。
- `install_root` は、WebSphere Process Server のインストール済み環境のルート・ディレクトリーです。

結果

UDDI データベースおよびデータ・ソースがマイグレーションされ、UDDI ノードがアクティブになります。

注: WebSphere Process Server をマイグレーションすると、プロファイルのアップグレード後のログは、UDDI データベースのマイグレーションが部分的に完了していて、トリガー、別名、および保管ステートメントのステップが欠落していることを示します。デバッグ機能を最初に有効化すると、データベースのデバッグ・ログに、トリガーの作成に失敗したことが示されます。これらのメッセージは無視してください。UDDI ノードが始動すれば、データベースのマイグレーションは完了しています。これらのログ・ファイルについては、120 ページの『Cloudscape v10.1.x 自動マイグレーションの検査』を参照してください。ログに他のエラーが表示された場合も、このトピックを参照してください。

UDDI データベースのマイグレーションが正常に完了した場合、サーバー・ログに以下のメッセージが表示されます。

```
CWUDQ0003I: UDDI registry migration has completed
```

以下のエラーが表示された場合、予期しないエラーがマイグレーション時に発生しています。UDDI レジストリー・ノードはアクティブにされていません。問題がないかエラー・ログを確認して、問題が修正できない場合、<http://www.ibm.com/software/support> の IBM ソフトウェア・サポートの Web サイトを参照してください。

```
CWUDQ0004W: UDDI registry not started due to migration errors
```

関連タスク

120 ページの『Cloudscape v10.1.x 自動マイグレーションの検査』

WebSphere Process Server バージョン 6.1.x を使用するには、バージョン v10.1.x 以上の Cloudscape を実行する必要があります (Cloudscape v10.1.x は、Apache Derby バージョン 10.1 のコード・ベースで構成されていることに注意してください。) WebSphere Process Server バージョン 6.1 のアップグレード中に、マイグレーション・ツールは、UDDI レジストリーなどの一部の内部コンポーネントによって、組み込みフレームワークを経由してアクセスされるデータベース・インスタンスを自動的にアップグレードします。また、このツールは、使用中のアプリケーションが組み込みフレームワークを経由してアクセスする Cloudscape インスタンスもアップグレードしようとします。これらのバックエンド・データベースについて、マイグレーション結果を検査する必要があります。

125 ページの『手動による Cloudscape のアップグレード』

WebSphere Process Server バージョン 6.1 のアップグレード中に、マイグレーション・ツールは、組み込みフレームワークのみを経由してアクセスする Cloudscape のインスタンスをアップグレードしようとします。(新しいバージョン

ンの Cloudscape はバージョン 10.1.x で、Derby をベースにしています。) 自動アップグレードでは、ネットワーク・サーバー・フレームワークを経由してアプリケーションとの間で処理を行う Cloudscape インスタンスは除外されます。このように除外することで、WebSphere Process Server と同じデータベース・インスタンスにアクセスするサード・パーティー・アプリケーションを破損するリスクがなくなります。ネットワーク・サーバー・フレームワーク経由でアクセスするデータベース・インスタンスは手動でアップグレードする必要があります。自動マイグレーションに失敗したデータベースについても、同じ操作を行います。

118 ページの『Cloudscape データベースのマイグレーション』

マイグレーション・ツールを使用して WebSphere Process Server バージョン 6.1 にマイグレーションした後、Cloudscape データベースの自動マイグレーションの結果を検査して、ツールによって自動的にマイグレーションされていない Cloudscape データベース・インスタンスがあれば、手動でマイグレーションします。

関連情報

バージョン 3 の UDDI レジストリーへのマイグレーション

IBM ソフトウェア・サポート Web サイト

Business Process Choreographer に関するマイグレーションの考慮事項

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要のある追加タスクに注意してください。

混合セルの制約事項

セルのマイグレーション・プロセス中に、バージョン 6.0.x レベルとバージョン 6.1 レベルの両方で、セルが同時にノードを実行している場合、以下の点に注意してください。

- デプロイメント・マネージャーをバージョン 6.1 にマイグレーションした後は、バージョン 6.0.x レベルのままのセルのノードで Business Process Choreographer アプリケーション (BPEL アプリケーションまたはヒューマン・タスク) を、インストール、更新、またはアンインストールすることはできません。
- デプロイメント・マネージャーをバージョン 6.1 にマイグレーションした後は、バージョン 6.0.x レベルのままのセルのノードで Business Process Choreographer を構成することはできません。
- バージョン 6.1 クラスターを保有し、そのクラスター上に Business Process Choreographer を構成済みの場合、同じセルの 6.0.x ノードに新規のクラスター・メンバーを作成してはいけません。

マイグレーション後のタスク

ご使用の環境によっては、実動環境で WebSphere Process Server バージョン 6.1 を使用する前に、以下のタスクを実行しなければならない場合があります。

- WebSphere Process Server バージョン 6.0.1 が Business Process Choreographer Observer サンプルを使用していた場合は、サンプルを除去してください。
『Business Process Choreographer Observer Sample バージョン 6.0.1 の除去 (Removing the Business Process Choreographer Observer Sample Version 6.0.1)』を

参照してください。このサンプルはマイグレーションされません。バージョン 6.0.2 またはバージョン 6.1 では、Business Process Choreographer Observer はサンプルではありません。

- 最初にユーザーを認証せずに、Business Process Choreographer API を使用するクライアントを作成済みの場合、API を使用する前に、ログインを実行するようにクライアントを変更する必要があります。マイグレーション後、J2EE ロール BPEAPIUser および TaskAPIUser は、値 Everyone に設定されます。これによって、アプリケーション・セキュリティーが有効な場合に 6.0.x がログインを要求しないようにして、後方互換性を保つことができます。クライアントを修正した後、これらのロールを値 AllAuthenticated に変更して、認証されていないユーザーが API にアクセスすることを防ぎます。新規のインストールの場合、これらのロールのデフォルト値は AllAuthenticated です。

これを行うには、次のようにします。

1. 管理コンソールを開き、「アプリケーション」>「エンタープライズ・アプリケーション」を選択します。
 2. 右のパネルで、BPEContainer_name を選択します。ここで、name は、nodeName_serverName か clusterName のいずれかで、ユーザーが Business Process Choreographer をサーバー上またはクラスター上のどちらに構成したかによって決まります。(名前の左のチェック・ボックスではなく、名前を選択します。)
 3. 右のパネルの、「詳細プロパティ」の下の、「ユーザー/グループ・マッピングへのセキュリティー・ロール」を選択します。
 4. J2EE BPEAPIUser ロールのマッピングを「Everyone」から「All authenticated」に変更します。
 5. 「OK」を選択します。
 6. TaskContainer_name エンタープライズ・アプリケーションの TaskAPIUser ロールについて、これらのステップを繰り返します。
 7. 変更を保管して、Business Process Choreographer を構成したサーバーまたはクラスターを再始動します。
- *install_root/ProcessChoreographer/Staff* ディレクトリーにある、デフォルトの XSL 変換ファイル (EverybodyTransformation.xml、LDAPTransformation.xml、SystemTransformation.xml、 UserRegistryTransformation.xml) に何らかの変更を適用した場合、マイグレーション後にその変更を WebSphere Process Server バージョン 6.1 バージョンのこれらのファイルに再度適用する必要があります。*install_root/ProcessChoreographer/Staff* ディレクトリーにあるカスタム XSL 変換ファイルは自動的にマイグレーションされます。その他のディレクトリーにあるカスタム XSL 変換ファイルは、手動でコピーする必要がありますが、バージョン 6.0.x スタッフ・プラグイン構成 (WebSphere Process Server バージョン 6.1 では担当者ディレクトリー構成と呼ばれるようになりました) に指定された変換ファイル・パスの正確な値によって異なります。

関連概念

3 ページの『事前マイグレーションの考慮事項』

WebSphere Process Server の新バージョンへのマイグレーション・プロセスを開始する前に、ここに示す考慮事項について考慮してください。

関連タスク

『バージョン間のマイグレーションのトラブルシューティング』
WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

関連情報

 Business Process Choreographer Observer Sample バージョン 6.0.1 の除去 (Removing the Business Process Choreographer Observer Sample Version 6.0.1)

 エンタープライズ・アプリケーションの管理

コンソールの「エンタープライズ・アプリケーション」ページ（「アプリケーション」>「エンタープライズ・アプリケーション」とクリックして表示）を使用して、サーバーにインストールされているエンタープライズ・アプリケーションの表示および管理を行います。

バージョン間のマイグレーションのトラブルシューティング

WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

- 構成をマイグレーションする前に、バージョン 6.1 マイグレーション・ウィザードを使用してプロファイルを作成しているときに、以下のプロファイル作成エラー・メッセージが表示される可能性があります。

```
profileName: profileName cannot be empty  
profilePath: Insufficient disk space
```

これらのエラー・メッセージは、スペースなどの誤った文字を含むプロファイル名を入力した場合に表示される可能性があります。マイグレーション・ウィザードを再実行して、プロファイル名にスペース、引用符、他の特殊文字などの誤った文字が含まれていないことを確認してください。

- WebSphere Process Server の旧バージョンから バージョン 6.1 へのマイグレーション中に問題が発生する場合は、ログ・ファイルや他の入手可能な情報を確認してください。

1. ログ・ファイルを見つけて手掛かりを探してください。

- *migration_backup_directory*/WBIPreUpgrade.time_stamp.log
- *profile_root*/log/WASPostUpgrade.time_stamp.log
- *install_root*/logs/clientupgrade.time_stamp.log
- *profile_root*/logs/bpeupgrade.log
- *migration_backup_directory*/WBIProfileUpgrade.ant.timestamp.log

2. 以下のディレクトリーで、「MIGR0259I: マイグレーションは正常に完了しました。」または「MIGR0271W: マイグレーションは、1 つ以上の警告を伴って、正常に完了しました。」を探します。

- *migration_backup_directory*/WBIPreUpgrade.time_stamp.log
- *profile_root*/logs/WASPostUpgrade.time_stamp.log
- *install_root*/logs/clientupgrade.time_stamp.log

「MIGR0286E: マイグレーションは完了できませんでした。」が表示されている場合、ログ・ファイルに記録されているエラー・メッセージに基づいて、すべての問題を訂正するようにしてください。エラーを訂正したら、製品のインストール・ルートの bin ディレクトリーからコマンドを再実行します。

3. アクセスしようとしているリソースをホスティングしているサーバーのサービス・ログの Application Server Toolkit (AST) に組み込まれている Log and Trace Analyzer を開いて、エラー・メッセージおよび警告メッセージを参照します。

Application Server Toolkit でのコンポーネントのデバッグを参照してください。

4. WebSphere Process Server で dumpNameSpace コマンドを実行し、出力をパイピング、リダイレクト、および詳細出力にして、出力を見やすくします。

このコマンドを実行すると、WebSphere Process Server 名前空間のすべてのオブジェクトがディレクトリー・パスとオブジェクト名を含めて表示されます。

5. クライアントがアクセスする必要があるオブジェクトが表示されない場合は、管理コンソールを使用して、以下の状態を確認します。
 - ターゲット・リソースをホスティングしているサーバーが開始していること。
 - ターゲット・リソースをホスティングしている Web モジュールまたは Enterprise JavaBean コンテナが稼働していること。
 - ターゲット・リソースの JNDI 名が正しく指定されていること。

これらのステップのいずれでも問題を解決できない場合は、IBM サポートとの連絡方法なども記載されている追加のトラブルシューティング・リソースについて、トラブルシューティングおよびサポートを参照してください。

- マイグレーション・プロセス時に、WBIPreUpgrade コマンドまたは WBIPostUpgrade コマンドを使用中、問題が発生することがあります。
 - WBIPreUpgrade コマンドの使用中に、問題が発生する可能性があります。
 - 「見つかりません」または「そのようなファイルまたはディレクトリーがありません (No such file or directory)」というメッセージが返されます。

この問題は、WBIPreUpgrade コマンドを WebSphere Process Server バージョン 6.1 *install_root/bin* ディレクトリー以外のディレクトリーから実行しようとする場合に発生する可能性があります。WBIPreUpgrade スクリプトが、バージョン 6.1 *install_root/bin* ディレクトリーに存在することを確認し、このファイルをその場所から起動します。

- DB2 JDBC ドライバーおよび DB2 JDBC ドライバー (XA) が、管理コンソールに表示されているサポートされる JDBC プロバイダーのドロップダウン・リスト内で見つかりません。

管理コンソールには、推奨されない JDBC プロバイダー名が表示されなくなりました。管理コンソールで使用されている新しい JDBC プロバイダー名は、より説明的で、紛らわしさが解消されています。新しいプロバイダー名と推奨されないプロバイダー名は、名前だけが異なっています。

推奨されない名前は、マイグレーション上の理由で (例えば、既存の JACL スクリプトなどのために) `jdbc-resource-provider-templates.xml` ファイルに引き続き残されています。ただし、JACL スクリプトでは、新しい JDBC プロバイダー名を使用するようお勧めします。

- 以下のメッセージを受け取ります。

```
MIGR0108E: The specified WebSphere directory does not contain a WebSphere version that can be upgraded.
```

これは、`WBIPreUpgrade` コマンドで、誤ったディレクトリーを指定した場合に発生する可能性があります。

`WBIPreUpgrade` コマンドを参照してください。

- `WBIPostUpgrade` コマンドを使用中に、問題が発生する可能性があります。

- 「見つかりません」または「そのようなファイルまたはディレクトリーがありません (No such file or directory)」というメッセージが返されます。

この問題は、`WBIPostUpgrade` コマンドを WebSphere Process Server バージョン 6.1 `install_root/bin` 以外のディレクトリーから実行しようとする場合に発生する可能性があります。`WBIPostUpgrade` スクリプトが、バージョン 6.1 `install_root/bin` ディレクトリーに存在することを確認して、このファイルをその場所から起動します。

- セル内で統合ノードをマイグレーションすると、以下のエラー・メッセージを受け取ります。

```
MIGR0304I: The previous WebSphere environment is being restored.  
com.ibm.websphere.management.exception.RepositoryException:  
com.ibm.websphere.management.exception.ConnectorException: ADMC0009E:  
The system failed to make the SOAP RPC call: invoke  
MIGR0286E: The migration failed to complete.
```

接続タイムアウトは、統合ノードの `WBIPostUpgrade` マイグレーション・ステップ中に、統合ノードが `Deployment Manager` から構成の更新を検索しようとするときに発生します。バージョン 6.1 にマイグレーションする構成に以下のいずれかの要素が含まれている場合、構成全体のコピーにかかる時間が、接続タイムアウトより長くなる可能性があります。

- 小規模アプリケーションが多数ある
- 大規模アプリケーションがいくつかある
- 非常に大規模なアプリケーションが 1 つある

これが発生した場合は、タイムアウト値を変更してから、`WBIPostUpgrade` コマンドを実行して統合ノードをマイグレーションしてください。

1. 統合ノードのマイグレーション先のプロファイルが含まれているバージョン 6.1 ディレクトリー内の以下の場所に移動します。

`profile_root/properties`

2. このディレクトリー内の `soap.client.props` ファイルを開き、`com.ibm.SOAP.requestTimeout` プロパティの値を見つけます。これは、秒単位のタイムアウト値です。デフォルト値は 180 秒です。

3. `com.ibm.SOAP.requestTimeout` の値を変更して、構成をマイグレーションできるように十分に大きくします。例えば、以下のように入力すると、タイムアウト値は 30 分になります。

```
com.ibm.SOAP.requestTimeout=1800
```

注: タイムアウト値には、必要を満たす最小の値を選択してください。選択したタイムアウトの少なくとも 3 倍の長さの待機時間を見込んでください。つまり、ファイルをバックアップ・ディレクトリーにダウンロードする時間、マイグレーション済みのファイルをデプロイメント・マネージャーにアップロードする時間、およびデプロイメント・マネージャーとマイグレーション済みのノード・エージェントとを同期化する時間です。

4. `WBIPreUpgrade` コマンドで作成したバックアップ・ディレクトリーの、以下のロケーションに移動します。

```
migration_backup_directory/profiles/default/properties
```

5. このディレクトリー内の `soap.client.props` ファイルを開き、`com.ibm.SOAP.requestTimeout` プロパティーの値を見つけます。
6. `com.ibm.SOAP.requestTimeout` の値を バージョン 6.1 ファイルで使用しているのと同じ値に変更します。

- 「Unable to copy document to temp file」というエラー・メッセージが表示されます。以下に例を示します。

```
MIGR0304I: The previous WebSphere environment is being restored.  
com.ibm.websphere.management.exception.DocumentIOException: Unable to copy  
document to temp file:  
cells/sunblade1Network/applications/LARGEApp.ear/LARGEApp.ear
```

ファイル・システムに空きがない可能性があります。ファイル・システムに空きがない場合、一部のスペースを消去して `WBIPostUpgrade` コマンドを再実行してください。

- 以下のメッセージを受け取ります。

```
MIGR0108E: The specified WebSphere directory does not contain a WebSphere  
version that can be upgraded.
```

このエラーの原因として、以下のような理由が存在すると考えられます。

- `WBIPreUpgrade` コマンドまたは `WBIPostUpgrade` コマンドの起動時に誤ったディレクトリーが指定されていた。
- `WBIPreUpgrade` コマンドが実行されなかった。

- 以下のエラー・メッセージを受け取ります。

```
MIGR0253E: The backup directory migration_backup_directory does not exist.
```

このエラーの原因として、以下のような理由が存在すると考えられます。

- `WBIPreUpgrade` コマンドが、`WBIPostUpgrade` コマンドの前に実行されなかった。

1. エラー・メッセージに示されているバックアップ・ディレクトリーが存在するかどうかを確認します。
2. 存在しない場合は、`WBIPreUpgrade` コマンドを実行します。

`WBIPreUpgrade` コマンドを参照してください。

3. `WBIPostUpgrade` コマンドを再実行します。

- 誤ったバックアップ・ディレクトリーが指定されていた可能性がある。

例えば、ディレクトリーが、`WBIPreUpgrade` コマンドの実行後に削除されたバージョン 6.0.x ツリーのサブディレクトリーであり、

`WBIPostUpgrade` コマンドの実行前に、製品の旧バージョンがアンインストールされたということが考えられます。

1. エラー・メッセージに示されているディレクトリー構造全体が存在するかどうかを判別します。

2. 可能であれば、正しいマイグレーション・バックアップ・ディレクトリー全体を指定して、`WBIPreUpgrade` コマンドを再実行します。

3. バックアップ・ディレクトリーが存在しない場合で、旧バージョンが削除されている場合は、バックアップ・リポジトリーまたは XML 構成ファイルから旧バージョンを再ビルドします。

4. `WBIPreUpgrade` コマンドを再実行します。

- `WBIPostUpgrade` コマンドの実行後に、`WBIPreUpgrade` をもう一度実行しなければならなくなりました。

Deployment Manager または管理対象ノードのマイグレーションの過程で、`WBIPostUpgrade` が旧環境を無効にする可能性があります。`WBIPostUpgrade` の実行後に、`WBIPreUpgrade` を旧インストールに対してもう一度実行する場合、旧 `install_root/bin` ディレクトリーに存在する

`migrationDisablementReversal.jacl` スクリプトを実行する必要があります。この JACL スクリプトを実行すると、バージョン 6.0.x 環境はもう一度有効な状態になり、`WBIPreUpgrade` を実行して有効な結果を出すことができますようになります。

スクリプト記述について詳しくは、スクリプト記述入門を参照してください。ここで説明されているスクリプト記述は、WebSphere Process Server で使用可能です。

- 統合マイグレーションが、メッセージ `MIGR0405E` で失敗します。

統合マイグレーションの一環として Deployment Manager で実行されたマイグレーションが失敗しました。このエラーが発生した詳しい理由については、Deployment Manager ノードの `...DeploymentManagerProfile/temp` ディレクトリーの下にあるフォルダー `your_node_name_migration_temp` を開いてください。以下に例を示します。

```
/websphere61/procserver/profiles/dm_profile/temp/nodeX_migration_temp
```

Deployment Manager ノード上のこのノードのマイグレーションに関するログや他のすべての情報は、このフォルダーに置かれています。このフォルダーは、このシナリオに関連した IBM サポートでも必要になります。

- WebSphere Process Server バージョン 6.1 アプリケーションがマイグレーション中に失われる。

統合マイグレーション中に、バージョン 6.1 アプリケーションのいずれかがインストールに失敗する場合、それらのアプリケーションは構成の同期化中

に失われます。これが発生する理由は、WBIPostUpgrade の最終手順の 1 つで、syncNode コマンドが実行されるためです。この結果、Deployment Manager ノードの構成がダウンロードされ、統合ノードの構成が上書きされます。アプリケーションのインストールが失敗すると、それらのアプリケーションは Deployment Manager ノードの構成に含まれなくなります。この問題を解決するには、マイグレーション後にアプリケーションを手動でインストールしてください。標準 のバージョン 6.1 アプリケーションの場合、*install_root/installableApps* ディレクトリーにあります。

マイグレーション中に失われたアプリケーションを手動でインストールするには、wsadmin コマンドを使用して、マイグレーション・ツールがバックアップ・ディレクトリーに作成した *install_application_name.jacl* スクリプトを実行します。

Linux 例えば、Linux 環境では以下のパラメーターを使用します。

```
./wsadmin.sh -f migration_backup_directory/install_application_name.jacl  
-conntype NONE
```

Wsadminツールを参照してください。

- WebSphere Process Server バージョン 6.1 アプリケーションのインストールが失敗する。

WBIPostUpgrade の完了後に、wsadmin コマンドを使用して、アプリケーションを手動でインストールします。

マイグレーション中にインストールが失敗したアプリケーションを手動でインストールするには、wsadmin コマンドを使用して、マイグレーション・ツールがバックアップ・ディレクトリーに作成した *install_application_name.jacl* スクリプトを実行します。

Linux 例えば、Linux 環境では以下のパラメーターを使用します。

```
./wsadmin.sh -f migration_backup_directory/install_application_name.jacl  
-conntype NONE
```

『Wsadmin ツール』または WBIPostUpgrade コマンドを参照してください。

- **Solaris** マイグレーション・ウィザードを使用して、Solaris x64 プロセッサー・ベースのシステム上でプロファイル WebSphere Process Server バージョン 6.0.x から バージョン 6.1.x にマイグレーションする場合は、WBIPostUpgrade ステップ中にマイグレーションが失敗する可能性があります。

*profile_root/logs/WASPostUpgrade.time_stamp.log*内に、以下のようなメッセージが記録される場合があります。

```
MIGR0327E: A failure occurred with stopNode.  
MIGR0272E: The migration function cannot complete the command.
```

WebSphere Process Server バージョン 6.0.x は、Java 仮想マシン (JVM) を 32 ビット・モードで使用します。WebSphere Process Server バージョン 6.1.x のマイグレーション・ウィザードは、WBIPostUpgrade.sh スクリプトを呼び出しま

す。このスクリプトは、サーバーがバージョン 6.0.x ノードを停止すると、バージョン 6.0.x 用の JVM を 64 ビット・モードで実行しようとしています。

以下のアクションを実行して、不完全なプロファイルを除去し、WebSphere Process Server がバージョン 6.0.x プロファイルを正しくマイグレーションできるようにします。

1. コマンド行で、*install_root/bin* ディレクトリーに移動します。

例えば、以下のコマンドを入力します。

```
cd /opt/IBM/WebSphere/Procserver/bin
```

2. *install_root/bin* ディレクトリー内で *WBIPostUpgrade.sh* スクリプトを見つけ、バックアップ・コピーを作成します。
3. *WBIPostUpgrade.sh* または *WBIPostUpgrade.bat* ファイルをエディターで開き、以下のアクションを実行します。
 - a. 以下のコード行を見つけます。

UNIX

Linux

```
"$binDir" /setupCmdLine.sh
```

Windows

```
call "%~dp0setupCmdLine.bat" %*
```

- b. 前のステップで特定したコードの後ろに、以下のコード行を挿入します。

```
JVM_EXTRA_CMD_ARGS=""
```
 - c. 変更を保管します。
4. *WASPostUpgrade.sh* または *WASPostUpgrade.bat* ファイルについて、ステップ 2 から 4 を繰り返します。
 5. 以下のコマンドを使用して、マイグレーション・プロセス中に作成された不完全なバージョン 6.1.x プロファイルを削除します。

```
install_root/bin/manageprofiles.sh -delete -profileName profile_name
```
 6. 前のステップで除去したバージョン 6.1.x プロファイルの *profile_root* ディレクトリーを削除します。
 7. マイグレーション・ウィザードを再実行します。
- マイグレーション・プロセスで、バージョン 6.0.x 構成に存在するエンタープライズ・アプリケーションを新しいバージョン 6.1 構成にインストールするオプションを選択すると、マイグレーションのアプリケーション・インストール・フェーズでエラー・メッセージが表示される場合があります。

バージョン 6.0.x 構成に存在するアプリケーションのデプロイメント情報が誤っている可能性があります。その場合、WebSphere Process Server の旧ランタイムで十分に検証されなかったために XML 文書が誤っているという場合がほとんどです。ランタイムのアプリケーション・インストール検証プロセスが改善されているため、これらの誤った形式の EAR ファイルのインストールが失敗します。このため、*WBIPostUpgrade* のアプリケーション・インストール・フェーズで障害が発生し、「E:」エラー・メッセージが生成されます。これは「致命的な」マイグレーション・エラーと見なされます。

アプリケーションのインストール中に、マイグレーションがこのような方法で失敗する場合、以下のいずれかを実行してください。

- バージョン 6.0.x アプリケーションの問題を修正してから、再マイグレーションする。
- マイグレーションを続行し、これらのエラーを無視する。

この場合、マイグレーション・プロセスでは、障害が起こったアプリケーションはインストールされませんが、他のすべてのマイグレーション手順は完了します。

後で、アプリケーションの問題を修正してから、管理コンソールまたはインストール・スクリプトを使用して新しいバージョン 6.1 構成に手動でインストールできます。

- WebSphere Process Server バージョン 6.0.1.3 より古いバージョン 6.0.x ノードを含んでいるか、これらのノードと相互運用するバージョン 6.1 セルにマイグレーションすると、クラスター機能に障害が発生する場合があります。

これらのバージョン 6.0.x サーバーを始動すると、以下の問題が発生する可能性があります。

- First Failure Data Capture (FFDC) ログに `ClassNotFoundException` エラー・メッセージが記録される場合があります。この例外は `RuleEtiquette.runRules` メソッドからスローされ、以下のような形式になっています。

```
Exception = java.lang.ClassNotFoundException
Source = com.ibm.ws.cluster.selection.SelectionAdvisor.<init>
probeid = 133
Stack Dump = java.lang.ClassNotFoundException: rule.local.server
at java.net.URLClassLoader.findClass(URLClassLoader.java(Compiled Code))
at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:106)
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.Class.forName1(Native Method)
at java.lang.Class.forName(Class.java(Compiled Code))
at com.ibm.ws.cluster.selection.rule.RuleEtiquette.runRules(RuleEtiquette.java:154)
at com.ibm.ws.cluster.selection.SelectionAdvisor.handleNotification(SelectionAdvisor.java:153)
at com.ibm.websphere.cluster.topography.DescriptionFactory$Notifier.run(DescriptionFactory.java:257)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1462)
```

- 以下のような形式の `java.io.IOException` が記録される場合があります。

```
Exception = java.io.IOException
Source = com.ibm.ws.cluster.topography.DescriptionManagerA.update probeid = 362
Stack Dump = java.io.IOException
at com.ibm.ws.cluster.topography.ClusterDescriptionImpl.importFromStream(ClusterDescriptionImpl.java:916)
at com.ibm.ws.cluster.topography.DescriptionManagerA.update(DescriptionManagerA.java:360)
Caused by: java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java(Compiled Code))
at java.io.DataInputStream.readUTF(DataInputStream.java(Compiled Code))
at com.ibm.ws.cluster.topography.KeyRepositoryImpl.importFromStream(KeyRepositoryImpl.java:193)
```

マイグレーション中にバージョン 6.1 クラスター情報がセル全体に配布されません。バージョン 6.0.1.3 以降ではない WebSphere Process Server バージョン 6.0.x ノードは、この情報を読み取ることができません。

この問題を回避するには、Deployment Manager をバージョン 6.1 にマイグレーションする前に、バージョン 6.1 セルに含まれるか、このセルと相互運用されるすべてのバージョン 6.0.x ノードをバージョン 6.0.1.3 以降にアップグレードします。

- 管理対象ノードをバージョン 6.1 にマイグレーションした後、アプリケーション・サーバーが始動しない場合があります。

アプリケーション・サーバーを始動しようとする、以下の例のようなエラーが発生する場合があります。

```
[5/11/06 15:41:23:190 CDT] 0000000a SystemErr R
    com.ibm.ws.exception.RuntimeError:
com.ibm.ws.exception.RuntimeError: org.omg.CORBA.INTERNAL:
    CREATE_LISTENER_FAILED_4
vmcid: 0x49421000 minor code: 56 completed: No
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.bootServerContainer(WsServerImpl.java:198)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.start(WsServerImpl.java:139)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.main(WsServerImpl.java:460)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServer.main(WsServer.java:59)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:64)
[5/11/06 15:41:23:197 CDT] 0000000a SystemErr R at
sun.reflect.DelegatingMethodAccessorImpl.invoke
    (DelegatingMethodAccessorImpl.java:43)
```

管理対象ノードのサーバーが listen するポート番号を変更します。例えば、Deployment Manager がポート 9101 で ORB_LISTENER_ADDRESS を listen している場合、管理対象ノードのサーバーはポート 9101 で ORB_LISTENER_ADDRESS を listen してはいけません。この例のような問題を解決するには、以下の手順を実行します。

1. 管理コンソールで、「アプリケーション・サーバー」 → 「*server_name*」 → 「ポート」 → 「ORB_LISTENER_ADDRESS」をクリックします。
 2. ORB_LISTENER_ADDRESS のポート番号を使用されていない番号に変更します。
- 管理対象ノードのバージョン 6.1 へのマイグレーション時に同期に失敗すると、サーバーが始動しない場合があります。

管理対象ノードをバージョン 6.1 にマイグレーションすると、以下のようなメッセージが記録される場合があります。

```
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0111E: Program exiting with error:
    com.ibm.websphere.management.exception.AdminException: ADMU0005E:
    Error synchronizing repositories
ADMU0211I: Error details may be seen in the file:
    /opt/WebSphere/61AppServer/profiles/AppSrv02/logs/syncNode.log
MIGR0350W: Synchronization with the deployment manager using the SOAP protocol
    failed.
MIGR0307I: The restoration of the previous WebSphere Application Server
    environment is complete.
MIGR0271W: Migration completed successfully, with one or more warnings.
```

これらのメッセージは、以下のことを示しています。

- Deployment Manager の構成レベルがバージョン 6.1 になっている。
- これからマイグレーションする管理対象ノードの構成レベルが (アプリケーションノードの構成レベルも含めて)、Deployment Manager のリポジトリでバージョン 6.1 になっている。
- syncNode 操作を完了しなかった場合は、管理対象ノードがまったく完了していない。

以下のアクションを実行して、この問題を解決します。

1. ノード上で syncNode コマンドを再実行し、ノードを Deployment Manager と同期化します。

syncNodeコマンドを参照してください。

2. GenPluginCfg コマンドを実行します。

GenPluginCfgコマンドを参照してください。

次のタスク

問題がリストされていない場合は、IBM サポートにお問い合わせください。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要がある追加タスクに注意してください。

関連タスク

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリに追加します。

関連情報

Application Server Toolkit でのコンポーネントのデバッグ

Wsadmin ツール

syncNode コマンド

GenPluginCfg コマンド



トラブルシューティングおよびサポート

ご使用の **IBM** ソフトウェアに関する問題の理解、特定、および解決を支援するため、トラブルシューティングおよびサポート情報では **IBM** 製品で提供されている問題判別リソースの使用について説明しています。

スクリプト記述入門

第 2 章 以前の WebSphere 製品からのマイグレーション

WebSphere Process Server より前に存在した特定の IBM 製品からアプリケーションおよび構成データをマイグレーションできます。

以下の先行製品から WebSphere Process Server へのマイグレーションがサポートされています。

- WebSphere InterChange Server バージョン 4.2.0 以降。詳しくは、146 ページの『WebSphere InterChange Server からのマイグレーション』を参照してください。
- WebSphere Business Integration Server Foundation バージョン 5.1 および 5.1.1。詳しくは、203 ページの『WebSphere Studio Application Developer Integration Edition からのソース成果物のマイグレーション』を参照してください。
- WebSphere MQ Workflow バージョン 3.6。詳しくは、204 ページの『WebSphere MQ Workflow からのマイグレーション』を参照してください。

注: また、WebSphere Process Server に、特定のバージョンの WebSphere Enterprise Service Bus および WebSphere Application Server からマイグレーションできます。また、前のバージョンの WebSphere Process Server 自体からもマイグレーションできます。これらの製品からのマイグレーションについて詳しくは、1 ページの『第 1 章 WebSphere Process Server および WebSphere Enterprise Service Bus の以前のバージョンからのマイグレーション』を参照してください。

i5/OS これらの先行製品は i5/OS 上でサポートされていませんでしたが、これらの製品のモジュールは、それぞれのプラットフォームで WebSphere Process Server バージョン 6.1 にマイグレーションできます (reposMigrate コマンドなどのマイグレーション・ツールが使用可能)。それによって、i5/OS オペレーティング・システムで稼働する WebSphere Process Server バージョン 6.1 にデプロイできます。

前の製品から WebSphere Process Server へマイグレーションする場合 (例えば、WebSphere InterChange Server から WebSphere Process Server へ)、マイグレーション手順で、マイグレーション・ツールを使用して、ソース成果物を新規 WebSphere Process Server バージョンの成果物に変換する必要があります。

WebSphere Integration Developer には、既存のアプリケーション・ソース成果物を WebSphere Process Server 成果物にマイグレーションするためのマイグレーション・ツールが含まれています。これらのツールは、WebSphere Integration Developer の「ファイル」>「インポート...」ウィザードから利用できます。WebSphere Process Server のコマンド行から、WebSphere InterChange Server からのマイグレーションを支援するために設計されたマイグレーション・ツールも利用できます。

IBM developerWorks® の「テクニカル・ライブラリー」(<http://www.ibm.com/developerworks>) でも、マイグレーションに役立つ記事を見つけることができます。

WebSphere InterChange Server からのマイグレーション

WebSphere InterChange Server 4.2.2 以降から WebSphere Process Server 6.1 にマイグレーションするには、WebSphere Integration Developer ウィザードまたは WebSphere Process Server の reposMigrate コマンドを使用します。

このタスクについて

このバージョンの WebSphere InterChange Server の場合...	実行内容
WebSphere InterChange Server バージョン 4.2.2 以降。	WebSphere Integration Developer からマイグレーション・ウィザードを使用して、WebSphere InterChange Server のすべての成果物を WebSphere Process Server の配置可能な成果物にマイグレーションし、その成果物を WebSphere Integration Developer のアクティブなワークスペース内のプロジェクトに置きます。あるいは、reposMigrate コマンドを使用して、WebSphere InterChange Server のすべての成果物を WebSphere Process Server の配置可能な成果物にマイグレーションし、必要に応じて、WebSphere Process Server に直接デプロイすることができます。
WebSphere InterChange Server の 4.2.2 より古いバージョン	まず WebSphere InterChange Server 4.2.2 以降にマイグレーションしてから、WebSphere Process Server にマイグレーションします。

関連情報

Migrating WebSphere InterChange Server using the Migration wizard

WebSphere Integration Developer インフォメーション・センター

事前マイグレーションの考慮事項

WebSphere InterChange Server 成果物を WebSphere Process Server にマイグレーションする作業を容易にするために、WebSphere InterChange Server 用の統合成果物を開発するための以下のガイドラインを検討してください。

以下の推奨事項は、ガイドとしてのみ使用されることを想定しています。これらのガイドラインから逸脱することが必要な場合も考えられます。このような場合には、成果物のマイグレーションに必要な再加工作業の量を最小限にするため、逸脱の範囲を制限するように注意する必要があります。ここで概説するガイドラインは、WebSphere InterChange Server 成果物の開発に関する一般的な推奨事項をすべて網羅しているわけではありません。その範囲は、将来成果物を容易にマイグレーションできるかどうかに影響する可能性のある考慮事項に絞られています。

関連概念

200 ページの『WebSphere InterChange Server からのマイグレーションのトラブルシューティング』

マイグレーションで発生する問題の解決策と、ロギングとトレースをオンにする方法について説明します。

関連資料

158 ページの『事後マイグレーションの考慮事項』

アプリケーションが WebSphere InterChange Server から WebSphere Process Server にマイグレーションされた場合は、WebSphere Process Server と WebSphere InterChange Server のアーキテクチャーの間には違いがあるため、マイグレーションされたアプリケーションが WebSphere Process Server において意図したとおりに機能するように、一部の領域に特別の注意を払う必要があります。

事前マイグレーションの考慮事項: 一般的な開発

WebSphere InterChange Server モジュールを開発して、将来の WebSphere Process Server へのマイグレーションを容易にするには、以下の推奨プラクティスに従ってください。

大半のインテグレーション成果物の開発に幅広く適用される、いくつかの考慮事項があります。一般的に、最もスムーズにマイグレーションできるのは、WebSphere InterChange Server ツールが提供する機能を利用し、ツールで実行されるメタデータ・モデルに合致している成果物です。また、大きな拡張および外部依存がある成果物は、マイグレーション時に行わなければならない手操作による介入が増える可能性があります。

一般的に、IBM では、以下を実行するようにお勧めします。

- システムおよびコンポーネント設計の文書化
- 開発ツールを使用したインテグレーション成果物の編集
- 推奨プラクティスを利用した、ツールおよび Java 断片によるルールの定義

インテグレーション・ソリューションは、WebSphere InterChange Server が提供するプログラミング・モデルおよびアーキテクチャーに従うことが重要です。

WebSphere InterChange Server 内の各インテグレーション・コンポーネントは、アーキテクチャー内で明確に定義された役割を果たします。このモデルからの逸脱が大きい場合、WebSphere Process Server 上で適切な成果物にコンテンツをマイグレーションすることが、さらに難しくなります。

将来のマイグレーション・プロジェクトを確実に成功させるもう 1 つの一般的なプラクティスは、システム設計を文書化することです。機能設計、サービスの品質の要件、プロジェクト間で共用される成果物の依存関係、デプロイメント中に行われた設計上の決定事項など、インテグレーション・アーキテクチャーおよび設計を必ず記録してください。これは、マイグレーション中のシステム分析を支援し、再作業の手間を最小限に抑えます。

成果物定義の作成、構成、および変更については、提供されている開発ツールのみを使用してください。成果物のメタデータを手動で操作すること (例えば、XML ファイルを直接編集すること) は避けてください。マイグレーションする成果物を破損する可能性があります。

コラボレーション・テンプレート、マップ、共通コード・ユーティリティ、およびその他のコンポーネント内で Java コードを開発する場合、IBM では以下を推奨します。

- 公開済みの API のみを使用する。
- Activity Editor を使用する。

- EIS へのアクセスにアダプターを使用する。
- Java 断片コードで外部依存を避ける。
- 移植性のために J2EE 開発プラクティスに従う。
- スレッドを spawn しない。また、スレッド同期プリミティブを使用しない。その必要がある場合は、マイグレーションの際に、非同期 Bean に変換する必要があります。
- java.io.* を使用したディスク I/O を行わない。データの保管には JDBC を使用する。
- ソケット I/O、クラス・ロード、ネイティブ・ライブラリーのロードなど、EJB コンテナ用に予約済みの関数を実行しない。その必要がある場合は、マイグレーションの際に、これらの断片が EJB コンテナ関数を使用するように手動変換する必要があります。

WebSphere InterChange Server 製品資料で公開済みの API のみを、成果物に対して使用します。これらの概要は、WebSphere InterChange Server の開発ガイドに記載されています。公開済みの WebSphere InterChange Server API 用に、互換性 API が WebSphere Process Server で提供されています。WebSphere InterChange Server には使用可能な多くの内部インターフェースがありますが、IBM では、これらの使用をお勧めしません。これらのインターフェースは、将来もサポートされる保証がないからです。

マップおよびコラボレーション・テンプレート内でビジネス・ロジックおよび変換ルールを設計する場合は、フィールド開発共通コード・ユーティリティ・ライブラリーの使用を避けるようにしてください (WebSphere InterChange Server のクラスパスに Java アーカイブ (*.jar) ファイルとして含まれています)。これらは手動でマイグレーションする必要があるためです。

可能な限り広い範囲で Activity Editor ツールを使用してください。これによって、ロジックはメタデータによって記述されるため、新規の成果物に変換しやすくなります。ツールで再使用したい操作には、可能な限り Activity Editor の「マイ・コレクション」フィーチャーを使用してください。

開発が必要なあらゆる Java コード断片で、コードを可能な限り単純かつアトミックにすることを IBM ではお勧めします。Java コードにおける高度化のレベルは、基本の評価、操作および計算を伴うスクリプト記述、データ・フォーマット付け、型変換などと同等である必要があります。より広範な、または高度なアプリケーション・ロジックが必要な場合は、WebSphere Application Server で稼働している EJB を使用してロジックをカプセル化すること、および Web サービス呼び出しを使用して、WebSphere InterChange Server から起動することを検討してください。個別にマイグレーションしなければならないサード・パーティーまたは外部のライブラリーではなく、標準の JDK ライブラリーを使用してください。また、単一のコード断片内にすべての関連ロジックをまとめてください。接続およびトランザクションのコンテキストが複数のコード断片に及ぶようなロジックの使用は避けてください。データベース操作の場合、例えば、接続の取得、トランザクションの開始と終了、接続のリリースに関するコードは 1 つのコード断片に含める必要があります。

一般的に、Enterprise Information System (EIS) とのインターフェースを目的として設計されるコードは、マップまたはコラボレーション・テンプレート内ではなく、

アダプター内に配置してください。これは、アーキテクチャー設計で一般的に推奨されるプラクティスです。また、これはサード・パーティー・ライブラリーに関する前提条件、およびコード内の関連した考慮事項 (接続管理や想定される Java Native Interface (JNI) 実装など) の回避にも役立ちます。

適切な例外処理を使用して、可能な限り安全なコードを作成してください。また、現在 J2SE 環境で実行中であっても、J2EE アプリケーション・サーバー環境内で実行できる互換性を備えたコードを作成してください。静的変数の回避、スレッドの spawn、およびディスク I/O など、J2EE の開発プラクティスに従ってください。これらは、一般的に準拠すべき優れたプラクティスであるだけでなく、移植性を向上させることができます。

事前マイグレーションの考慮事項: 共通コード・ユーティリティー

IBM では、WebSphere InterChange Server 環境内のインテグレーション成果物が使用する共通コード・ユーティリティー・ライブラリーの開発を避けることをお勧めします。インテグレーション成果物間でコードの再使用が必要な場合、IBM では、Activity Editor ツールの「マイ・コレクション」フィーチャーを活用することをお勧めします。また、WebSphere Application Server で稼働している EJB を使用してロジックをカプセル化すること、および Web サービス呼び出しを使用して、WebSphere InterChange Server から起動することを検討してください。

共通コード・ユーティリティー・ライブラリーが WebSphere Process Server 上で適切に実行可能な場合は、カスタム・ユーティリティーのマイグレーションを責任を持って行って下さい。

事前マイグレーションの考慮事項: データベース接続プール

マップまたはコラボレーション・テンプレート内の WebSphere InterChange Server データベース接続プールは、WebSphere Process Server 内の標準 JDBC リソースとして扱われます。しかし、接続およびトランザクションの管理方法は、WebSphere InterChange Server と WebSphere Process Server とでは異なる場合があります。そのため、複数の Java 断片にまたがるデータベース・トランザクションをアクティブのままにするのは避けてください。

ユーザー定義のデータベース接続プールは、複数プロセス・インスタンスにまたがる簡単なデータ検索およびより優れた状態管理のために、マップおよびコラボレーション・テンプレート内で役立ちます。WebSphere InterChange Server 内のデータベース接続プールは、WebSphere Process Server 内の標準の JDBC リソースとして扱われ、基本機能は同じです。しかし、接続およびトランザクションの管理方法は異なる場合があります。

将来の移植性を最大化するために、コラボレーション・テンプレートまたはマップ内で複数の Java 断片ノードにまたがるデータベース・トランザクションをアクティブのままにするのを避けてください。例えば、接続の取得、トランザクションの開始と終了、接続のリリースに関するコードは 1 つのコード断片に含める必要があります。

事前マイグレーションの考慮事項: データベース競合の防止

少なくとも 2 秒の間隔をおいてイベントが発生するようにスケジュールすることで、データベース競合を防止します。

マイグレーション済みのアプリケーションが WebSphere Business Integration コンポーネントに対して複数のイベントを同時に発生させる場合、データベース競合またはデッドロックを引き起こす可能性があります。 WebSphere Process Server Application Scheduler (AppScheduler) により、複数のイベントがまったく同時に発生するようにスケジュールされると、この問題が生じます。デッドロックが発生すると、その原因となったイベントはロールバックされ、できるだけ早く再試行されます。このサイクルは、データベースへのアクセスを試みるスレッドそれぞれがそのデータベースを正常に更新するまで続けられます。

以下に例を示します。

```
AppScheduler E com.ibm.wbiserver.scheduler.AppSchedulerMB process CWLWS0021E:  
The AppSchedulerMB.process method has generated an exception.  
WSRdbXAResour E DSRA0304E: XAException occurred. XAException contents and  
  details are:  
The DB2 Error message is : Error executing a XAResource.end(), Server returned  
XA_RBDEADLOCK The DB2 Error code is : -4203  
The DB2 SQLState is : null
```

この問題の発生を防ぐには、デッドロックが起こらないように、十分な間隔を空けてイベントが発生するようにスケジュールします。IBM は、少なくとも 2 秒の間隔において発生するようイベントをスケジュールすることをお勧めしています。しかし、間隔に要する時間は、データベース・サイズ、ハードウェア、接続速度、その他の要因など、ご使用の環境でパフォーマンスに影響を与える要因によって異なります。

事前マイグレーションの考慮事項: ビジネス・オブジェクト

ビジネス・オブジェクトを開発する場合、付属のツールのみを使用して成果物を構成してください。また、データ属性として明示的なデータ・タイプおよびデータ長を使用し、文書化された API のみを使用してください。

WebSphere Process Server 内のビジネス・オブジェクトは、サービス・データ・オブジェクト (SDO) に基づいています。SDO は、強く型付けされたデータ属性を使用します。WebSphere InterChange Server およびアダプター内のビジネス・オブジェクトの場合、データ属性は強く型付けされていません。そのためユーザーが、非ストリング・データ属性に対してストリング・データ属性を指定する場合があります。WebSphere Process Server で問題を避けるには、データ・タイプを明示的に指定してください。

WebSphere Process Server 内のビジネス・オブジェクトは、コンポーネント間で渡される際に実行時に直列化される場合があるため、システム・リソースの使用率を最小限に留めるために、データ属性で求められる長さを明確にしておくことが重要です。このため、例えばストリング属性の最大長である 255 文字を使用しないでください。また、ゼロの長さ属性を指定しないで下さい。長さ属性は現在 デフォルトの 255 文字です。代わりに、属性に必要な長さを正確に指定してください。

XSD NCName ルールが、WebSphere Process Server 内のビジネス・オブジェクト属性名に適用されます。そのため、スペースや「:」をビジネス・オブジェクト属性名に使用しないでください。スペースまたは「:」を使用したビジネス・オブジェクト属性名は WebSphere Process Server 内では無効です。ビジネス・オブジェクト属性名を名前変更してから、マイグレーションを実行してください。

ビジネス・オブジェクトで配列を使用している場合、マップまたは関係内の配列に索引付けを行う際に、その配列の順序を基にすることはできません。WebSphere Process Server にマイグレーションされる構成では、索引の順序を保証しません (特にエントリーが削除されている場合)。

ビジネス・オブジェクト定義の編集には Business Object Designer ツールのみを使用し、統合成果物内のビジネス・オブジェクトに対しては、公開されている API のみを使用することが重要です。

事前マイグレーションの考慮事項: コラボレーション・テンプレート

WebSphere InterChange Server コラボレーション・テンプレートを開発する場合、WebSphere Process Server へのマイグレーションを最も円滑に行えるようにするために、以下のガイドラインに従ってください。

プロセスがメタデータで適切に記述されるようにするために、コラボレーション・テンプレートの作成および変更には常に Process Designer ツールを使用し、メタデータ・ファイルを直接編集することは避けてください。可能な限り Activity Editor ツールを使用して、メタデータを最大限に使用して必要なロジックを記述するようにします。

マイグレーションで必要になる可能性のある手動によるやり直しの作業を最小限に抑えるため、コラボレーション・テンプレートでは文書化された API のみを使用します。静的変数の使用は避けます。代わりに、非静的変数およびコラボレーション・プロパティを使用して、ビジネス・ロジックの要件に対応します。Java 断片で、Java のファイナル修飾子 (final)、一時的修飾子 (transient)、ネイティブ修飾子 (native) の使用は避けてください。これらは、コラボレーション・テンプレートのマイグレーションの結果作成された BPEL Java 断片内で、強制実行できません。

将来の移植性を最大化するために、ユーザー定義データベース接続プールに対して、明示的な接続リリース・コールおよび明示的なトランザクション・ブラケット (つまり、明示的なコミットおよび明示的なロールバック) の使用は避けてください。代わりに、コンテナ管理の暗黙的な接続クリーンアップおよび暗黙的なトランザクション・ブラケットを使用してください。また、コラボレーション・テンプレート内で、Java 断片ノード間のシステム接続およびトランザクションをアクティブのままにすることを避けてください。これは、ユーザー定義データベース接続プールと同様に、外部システムとのどのような接続にも適用されます。外部 EIS による操作は、アダプター内で管理される必要があります。また、データベース操作に関するコードは 1 つのコード断片に含まれる必要があります。コラボレーション内でこのようにする必要があるのは、コラボレーションが、BPEL ビジネス・プロセス・コンポーネントとしてレンダリングされたときに、割り込み可能なフローとして選択的にデプロイされる可能性がある場合です。この場合、プロセスはいくつかの個別のトランザクションから構成され、状態およびグローバル変数情報のみがアクティビティー間で渡されます。これらのプロセス・トランザクションにまたがるすべてのシステム接続または関連トランザクションのコンテキストは失われます。

W3C XML NCName 命名規則に従ってコラボレーション・テンプレート・プロパティに名前を付けます。WebSphere Process Server は、これらの規則に準拠した名前を受け入れます。マイグレーション先の BPEL プロパティ名では、許可されて

いない文字はすべて無効です。マイグレーションで生成される BPEL で構文エラーを避けるために、マイグレーションの前にプロパティを名前変更して、許可されていない文字を除去してください。

「this」を使用した参照変数を使用しないでください。例えば、「this.inputBusObj」ではなく、「inputBusObj」を使用します。

シナリオ・スコープ変数ではなく、クラス・レベルのスコープ・オン変数を使用してください。マイグレーション中にシナリオ・スコープは繰り越されません。

Java 断片で宣言されているすべての変数を、デフォルト値で初期化してください (例えば、「Object myObject = null;」)。必ずすべての変数を宣言中に初期化してから、マイグレーションを実行してください。

コラボレーション・テンプレートのユーザーによる変更が可能なセクションに Java インポート・ステートメントがないことを確認してください。コラボレーション・テンプレートの定義では、インポートする Java パッケージを指定するために、インポート・フィールドを使用します。

着信するビジネス・オブジェクト値を *triggeringBusObj* 変数に保管するように設定しないでください。WebSphere InterChange Server では、*triggeringBusObj* は読み取り専用で、その値を上書きすることができません。そのため、着信したビジネス・オブジェクト値はいずれも保管されません。*triggeringBusObj* がインバウンド・サービス呼び出しで着信ビジネス・オブジェクトの受信変数として使用されている場合、マイグレーション後、インバウンド・サービス呼び出しの動作が変わります。つまり、BPEL プロセス内で、インバウンド・サービス呼び出しからの着信値が *triggeringBusObj* に保管されている値を上書きします。

事前マイグレーションの考慮事項: マップ

WebSphere InterChange Server マップを開発する場合、WebSphere Process Server へのマイグレーションを最も円滑に行えるようにするために、以下のガイドラインに従ってください。

マップがメタデータで適切に記述されるようにするために、マップの作成および変更には常に Map Designer ツールを使用し、メタデータ・ファイルを直接編集することは避けてください。可能な限り Activity Editor ツールを使用して、メタデータを最大限に使用して必要なロジックを記述するようにします。

マップ内の子ビジネス・オブジェクトを参照する場合、子ビジネス・オブジェクトのサブマップを使用してください。

SET 内の「値」として、Java コードの使用は避けてください。これは WebSphere Process Server 内では有効でないためです。代わりに定数を使用してください。例えば、SET 値が "xml version=" + "1.0" + " encoding=" + "UTF-8" であった場合、これは WebSphere Process Server では有効ではありません。代わりに、"xml version=1.0 encoding=UTF-8" に変更してから、マイグレーションを実行してください。

マイグレーションで必要になる可能性のある手動によるやり直しの作業を最小限に抑えるため、マップでは文書化された API のみを使用します。静的変数の使用は避

けます。代わりに、非静的変数を使用します。カスタム・コードで、Java のファイナル修飾子 (final)、一時的修飾子 (transient)、ネイティブ修飾子 (native) の使用は避けてください。

ビジネス・オブジェクトで配列を使用している場合、マップ内の配列に索引付けを行う際、配列の順序に依存しないでください。WebSphere Process Server にマイグレーションされる構成では、索引の順序を保証しません (特にエンタリーが削除されている場合)。

将来の移植性を最大化するために、ユーザー定義データベース接続プールに対して、明示的な接続リリース・コールおよび明示的なトランザクション・ブラケット (つまり、明示的なコミットおよび明示的なロールバック) の使用は避けてください。代わりに、コンテナ管理の暗黙的な接続クリーンアップおよび暗黙的なトランザクション・ブラケットを使用してください。また、カスタム・マップ・ステップ内で、変換ノード境界を越えてシステム接続およびトランザクションをアクティブのままにすることを避けてください。これは、ユーザー定義データベース接続プールと同様に、外部システムとのどのような接続にも適用されます。外部 EIS による操作は、アダプター内で管理される必要があります。また、データベース操作に関するコードは、1 つのカスタム・ステップに含まれる必要があります。

マップで内部クラスを使用しないでください。マイグレーション・コマンド (reposMigrate) は、内部クラスをマイグレーションしないため、マップにそれらが含まれているとエラーが発生します。WebSphere InterChange Server リポジトリで、あるノード内に定義された内部クラスが、同じコラボレーション・テンプレート内の他のノードから参照される可能性があります。WebSphere Process Server では、BPEL コンポーネント内に定義された内部クラスを他のコンポーネントが使用することはできません。この制約事項のために、内部クラスは変換されず、手動で処理しなければなりません。推奨される変更方法としては、内部クラス・コードを外部クラスとしてライブラリーにパッケージするか、または内部クラス宣言を除去し、エラーがあれば解決した上で、BPEL 全体で必要な箇所にコードを配置するなどの方法があります。

事前マイグレーションの考慮事項: リレーションシップ

リレーションシップ定義はマイグレーションして、WebSphere Process Server で使用できます。また、リレーションシップ・テーブル・スキーマおよびインスタンス・データを WebSphere Process Server で再使用し、WebSphere InterChange Server と WebSphere Process Server 間で同時に共用できます。

リレーションシップの場合、関連コンポーネントを構成するために提供されたツールのみを使用し、インテグレーション成果物内のリレーションシップに対しては公開済みの API のみを使用してください。

リレーションシップ定義の編集には、Relationship Designer ツールのみを使用してください。また、WebSphere InterChange Server のみがリレーションシップ・スキーマを構成できるようにしてください。リレーションシップ・スキーマは、リレーションシップ定義のデプロイメントによって自動的に生成されます。データベース・ツールまたは SQL スクリプトによって、リレーションシップ・テーブル・スキーマを直接変更しないでください。

リレーションシップ・テーブル・スキーマ内のリレーションシップ・インスタンス・データを手動で変更する必要がある場合には、必ず Relationship Manager が提供している機能を使用してください。

インテグレーション成果物内のリレーションシップに対しては、公開済みの API のみを使用してください。

事前マイグレーションの考慮事項: アクセス・フレームワーク・クライアント

CORBA IDL インターフェース API を採用した新規クライアントを開発しないでください。これは、WebSphere Process Server でサポートされません。

reposMigrate コマンドを使用した WebSphere InterChange Server の成果物のマイグレーション

WebSphere InterChange Server の成果物から WebSphere Process Server の成果物へのマイグレーションには、**reposMigrate** コマンドを使用します。

始める前に

注: reposMigrate コマンドの機能は、サポート・ウィザード (グラフィカル・ユーザー・インターフェース) のある WebSphere Integration Developer から使用することもできます。詳しくは、WebSphere Integration Developer インフォメーション・センターを参照してください。

reposMigrate コマンドでは、入力として WebSphere InterChange Server リポジトリ JAR ファイルが必要です。この JAR ファイルは、マイグレーション対象のアプリケーションに関して、必要なものを完備していなければなりません。つまり、JAR ファイル内のいずれかの成果物によって参照されるすべての成果物も、JAR ファイル内に含まれていることが必要です。

生成されるリポジトリ JAR ファイルが必要なものを完備するように、サーバー・リポジトリをエクスポートする前に **-vr** オプションを指定して **repos_copy** コマンドを実行します。これにより、リポジトリが検証されます。リポジトリが有効である場合、repos_copy によって、「検証は成功しました。すべての依存関係が解決されました (Validation Succeeded. All Dependencies Resolved.)」という出力がコンソールに書き込まれます。リポジトリが無効である場合は、**repos_copy** によって、解決しなければならない依存関係のリストが出力されます。リポジトリをエクスポートする前に、依存関係を解決してください。

-o オプションを指定した WebSphere InterChange Server **repos_copy** コマンドを使用して、リポジトリ成果物をエクスポートし、リポジトリ JAR ファイルを作成します (個々のコンポーネントのエクスポート方法などの詳細については、WebSphere InterChange Server v4.3 の資料を参照してください)。

このタスクについて

reposMigrate コマンドは、JAR ファイルにある WebSphere InterChange Server のすべての成果物を WebSphere Process Server の配置可能な成果物に変換します。これらの成果物は、1 つ以上の JAR ファイルとして作成されたモジュールです。JAR ファイルは、コラボレーション・オブジェクトごと、およびマイグレーション済み

のコネクター定義ごとに 1 つずつ作成されます。ビジネス・オブジェクト、マップ、およびリレーションシップなど、その他の成果物では、生成された各 JAR ファイルに、入力 JAR ファイルから生成されたこれらすべての成果物のコピーが格納されます。マイグレーションされたコラボレーション・オブジェクトやコネクターがない場合は、共有されるすべての成果物から成るモジュールを格納した 1 つの JAR ファイルが作成されます。新しい JAR ファイルが作成された後、**serviceDeploy** コマンドを使用して、WebSphere Process Server にデプロイできる EAR ファイルを生成します。

WebSphere InterChange Server の成果物に対応する成果物が WebSphere Process Server がない場合、マイグレーション中に **Jython** スクリプトが生成されます。このスクリプトは、**wsadmin** コマンドを使用して実行し、元の WebSphere InterChange Server の成果物に対応する WebSphere Process Server の構成定義を作成することができます。

プロシージャ

1. WebSphere Process Server の配置可能な成果物に変換する、事前にエクスポートした WebSphere InterChange Server の成果物を格納する JAR ファイルを指定します。
2. コマンド行プロンプトから **reposMigrate** コマンドを起動します。WebSphere Process Server のコマンド・プロンプトで、必須の引数および必要なオプションの引数を指定したコマンドを入力します。詳しくは、156 ページの『**reposMigrate** コマンド』を参照してください。
3. 必要であれば、結果として生成された JAR ファイルを編集します。
4. **serviceDeploy** を実行して、JAR ファイルごとの配置可能な EAR ファイルを作成します。

注: WPS ランタイムでのマイグレーション済み ICS アプリケーションの処理に対するサポートは、**serviceDeploy** コマンドで使用されるデフォルトの命名規則に依存します。IBM では、**serviceDeploy** コマンドを使用して、マイグレーション済みプロジェクトをビルドするとき、**serviceDeploy -outputApplication** パラメーターを指定しないことをお勧めします。これにより、デフォルトの出力ファイル名が生成されます。

詳しくは、「リファレンス」の PDF ファイルの WebSphere Process Server の **serviceDeploy** コマンドを参照してください。

5. 管理コンソールまたは **wsadmin** コマンドを使用して、WebSphere Process Server に EAR ファイルをインストールします。**wsadmin** コマンドを使用して、**InstallAdministrativeObjects.py** スクリプトを実行します。これにより、WebSphere Process Server システム内に、JDBC データ・ソースや **WBIScheduler** エントリなどのすべてのターゲット・リソースに対応するリソースが作成されます。

例

reposMigrate コマンドを使用して、既存の WebSphere InterChange Server の成果物を、稼働中の WebSphere Process Server に直接マイグレーションすることができます。

1. WebSphere Process Server でコマンド・プロンプトを開きます。

2. 以下の必須パラメーターを指定して **reposMigrate** コマンドを発行します。

```
install_root/bin/reposMigrate SourceArtifactJAR OutputArtifactDirectory
```

reposMigrate コマンドは、生成された成果物を以下のように作成します。

- **reposMigrate** は、入力 JAR ファイル内の WebSphere InterChange Server コラボレーション・オブジェクトおよびコネクタ定義ごとに、マイグレーション済み成果物から JAR ファイルを作成します。
- ビジネス・オブジェクト、マップ、およびリレーションシップなど、その他の成果物では、生成された各 JAR ファイルに、入力 JAR ファイルから生成されたこれらすべての成果物のコピーが格納されます。入力にコラボレーション・オブジェクトやコネクタ定義がなかった場合は、共有されるすべての成果物で 1 つの JAR ファイルが作成されます。

reposMigrate コマンドのデフォルトの振る舞いでは、個々の成果物のマイグレーションでのエラーをログに記録して、残りの成果物のマイグレーションを続行します。実行が完了したら、出力メッセージをチェックして、エラーがないか確認する必要があります。出力を表示するには、ログ・ファイル・パラメーター (**-l***LogFile*) を使用して、出力を指定のファイルに転送します。このデフォルトの振る舞いをオーバーライドして、マイグレーションできない成果物が最初に見つかったときに、**reposMigrate** による処理を強制終了させるには、**-fh** (最初の障害時に一時停止) フラグを指定します。実行に失敗した後再試行するには、**reposMigrate** を最初から実行することができます。

関連資料

158 ページの『事後マイグレーションの考慮事項』

アプリケーションが WebSphere InterChange Server から WebSphere Process Server にマイグレーションされた場合は、WebSphere Process Server と WebSphere InterChange Server のアーキテクチャーの間には違いがあるため、マイグレーションされたアプリケーションが WebSphere Process Server において意図したとおりに機能するように、一部の領域に特別の注意を払う必要があります。

関連情報

Wsadmin ツール

WebSphere InterChange Server v4.3 の資料

WebSphere Integration Developer インフォメーション・センター

reposMigrate コマンド

reposMigrate コマンドの構文および使用法を、詳しく説明します。

目的

reposMigrate コマンドは、WebSphere InterChange Server の成果物を WebSphere Process Server の配置可能な成果物にマイグレーションします。

ロケーション

コマンド・ファイルは、*install_root/bin* ディレクトリにあります。コマンド・ファイルは、以下の名前のスクリプトです。

- **UNIX** **Linux** Linux および UNIX ベースのオペレーティング・システムの場合: reposMigrate.sh
- **Windows** Windows システムの場合: reposMigrate.bat
- **i5/OS** i5/OS オペレーティング・システムの場合: reposMigrate

構文

reposMigrate [-es] [-td *TemplateDirectory*] [-ml] [-fh] [-lv] [-wi] *SourceArtifactJAR* *OutputArtifactDirectory*

SourceArtifactJAR 引数および *OutputArtifactDirectory* 引数は必須です。

パラメーター

[-es]

オプションのパラメーターです。すべての非同期 WSDL メソッドに対して、イベント順序付けを使用可能にするように要求します。このオプションを指定しない場合のデフォルトは、マイグレーションにおいて、すべての WSDL メソッドでイベント順序付けを使用可能にしないことです。

[-td *TemplateDirectory*]

オプションのパラメーターです。指定したディレクトリーにあるすべてのアセンブリ・エディター・テンプレートをロードし、XML から Java への変換に使用するように要求します。このプロパティーのデフォルトは、標準アセンブリ・エディター・テンプレート v4.3.3 のみを、XML から Java への変換に使用することです。

[-ml]

オプションのパラメーターです。コラボレーション・テンプレート内に存在するすべてのループを維持するように要求します。このオプションを指定しない場合のデフォルトは、マイグレーションで、ループ解除を使用することです。ループについて詳しくは、166 ページの『コラボレーションのマイグレーション』を参照してください。

[-fh]

オプションのパラメーターです。デフォルトでは、特定の成果物の処理中にエラーが発生した場合、reposMigrate は JAR ファイル内の残りの成果物の処理を続行します。このオプションを設定した場合、エラーが検出されるとすぐに処理が停止します。エラーのある成果物、およびそれ以降のすべての成果物は処理されません。

[-lv]

オプションのパラメーターです。ログ・レベルを verbose に設定します。

[-wi]

オプションのパラメーターです。デフォルトでは、Java 変換の問題が見つかった場合、個々の成果物のマイグレーションは失敗します。このオプションを設定した場合、すべての Java 変換の問題は単に警告として取り扱われ、成果物はできる限り正常にマイグレーションされます。

SourceArtifactJAR

必須のパラメーターです。マイグレーションの対象となる WICS リポジトリー JAR ファイルを指定します。

OutputArtifactDirectory

必須のパラメーターです。生成されるモジュールの jar ファイルの格納先となる出力ディレクトリーを指定します。

例

Windows Windows システム上でのこの例は、WebSphere InterChange Server の既存の成果物をマイグレーションし、MigratedArtifacts ディレクトリーに格納します。Java 変換に関する警告メッセージは無視され、ログ・レベルが verbose に設定されます。

```
install_root%bin%reposMigrate.bat -wi -lv C:%inputRepos.jar  
C:%IBM%WebSphere%MigratedArtifacts
```

UNIX Linux UNIX ベースのシステム上でのこの例は、WebSphere InterChange Server の既存の成果物をマイグレーションし、MigratedArtifacts ディレクトリーに格納します。Java 変換に関する警告メッセージは無視され、ログ・レベルが verbose に設定されます。

```
install_root/bin/reposMigrate.sh -wi -lv /inputRepos.jar  
/opt/IBM/WebSphere/MigratedArtifacts
```

i5/OS システム上でのこの例は、WebSphere InterChange Server の既存の成果物をマイグレーションし、MigratedArtifacts ディレクトリーに格納します。Java 変換に関する警告メッセージは無視され、ログ・レベルが verbose に設定されます。

```
install_root/bin/reposMigrate -wi -lv /inputRepos.jar  
/home/user_name/MigratedArtifacts
```

ここで *user_name* は、reposMigrate スクリプトを起動する i5/OS ユーザー・プロフィールの名前です。

事後マイグレーションの考慮事項

アプリケーションが WebSphere InterChange Server から WebSphere Process Server にマイグレーションされた場合は、WebSphere Process Server と WebSphere InterChange Server のアーキテクチャーの間には違いがあるため、マイグレーションされたアプリケーションが WebSphere Process Server において意図したとおりに機能するように、一部の領域に特別の注意を払う必要があります。

ご使用のアプリケーションと環境に当てはまる場合は、以下のセクションで説明される情報に注意してください。

159 ページの『セキュリティ』

159 ページの『既存のデータベース接続、リリースシップ、およびスケジュール済みイベントの処理 (InstallAdministrativeObjects.py スクリプト)』

160 ページの『既存の WebSphere InterChange Server データベース接続プールの処理』

- 161 ページの『既存の WebSphere InterChange Server リレーションシップ・データベースの使用』
- 161 ページの『スケジュール済みイベントのマイグレーション』
- 162 ページの『Access Enterprise JavaBean (EJB) サポート』
- 163 ページの『DynamicSend API 構成』
- 163 ページの『BaseCollaboration.dynamicSend メソッド呼び出しの使用可能化』
- 165 ページの『イベント順序付けマイグレーション』
- 165 ページの『失敗したイベント』
- 166 ページの『マップのマイグレーション』
- 166 ページの『コラボレーションのマイグレーション』
- 168 ページの『BPEL 変数をマイグレーション後に定義する必要がある』
- 168 ページの『WebSphere Process Server での logError API E メール通知の使用可能化』
- 168 ページの『WebSphere Process Server での非同期呼び出しの処理』
- 169 ページの『Network Deployment のアップグレード後に AppScheduler の開始を可能にする』
- 169 ページの『WebSphere Process Server での相関値の処理』
- 170 ページの『マイグレーションされたアプリケーションのパッケージ化とデプロイ』

セキュリティー

ご使用のアプリケーションのセキュリティー・レベルを、WebSphere InterChange Server で稼働していたときと同じレベルに設定するには、追加のセキュリティー構成が必要です。この構成について詳しくは、172 ページの『WebSphere InterChange Server のマイグレーション後のグローバル・セキュリティーの構成』を参照してください。

既存のデータベース接続、リレーションシップ、およびスケジュール済みイベントの処理 (InstallAdministrativeObjects.py スクリプト)

マイグレーション中に Jython スクリプト InstallAdministrativeObjects.py が生成されます。このスクリプトには、3 つの目的があります。WebSphere Process Server 内に対応する成果物を持たない WebSphere InterChange Server スケジューラー・エントリーのマイグレーション、既存の DBConnection プールの使用、および既存のリレーションシップ・データベースの使用を可能にすることです。このスクリプトは、wsadmin コマンドを使用して実行することができ、元の WebSphere InterChange Server の成果物に対応する WebSphere Process Server 構成定義を作成します。共有成果物が組み込まれるすべての場所に、InstallAdministrativeObjects.py のコピーが組

み込まれます。つまり、スクリプトは、`reposMigrate` コマンドによって作成されるすべての JAR ファイル内に存在します。また、このスクリプトは、WebSphere Integration Developer でのインポート時に指定された共有ライブラリー・プロジェクト内に置かれます。`InstallAdministrativeObjects.py` スクリプトは、このスクリプトを必要とする成果物がない場合でも必ず生成されます。このスクリプトに変更を加えて項目の追加や削除を行った後、`wsadmin` コマンドを使用してスクリプトを実行することができます。

`wsadmin` コマンドの使用について詳しくは、『`wsadmin` ツール』を参照してください。

既存の WebSphere InterChange Server データベース接続プールの処理

WebSphere Process Server が使用する既存の WebSphere InterChange Server データベース接続プールを保存するために、`wsadmin` コマンドを使用して `InstallAdministrativeObjects.py` スクリプトを実行することにより、WebSphere Process Server 内に接続プールを作成できます。適切な JDBC プロバイダーが定義されていない場合、このスクリプトはデフォルトの JDBC プロバイダー・テンプレートを使用して JDBC プロバイダーを作成します。このようなデフォルトのテンプレートを使用することによる副次作用として、WebSphere Process Server により、空のサンプル・データ・ソース定義が作成されます。このサンプル・データ・ソースは使用されません。データ・ソースで必要なすべての情報を指定しているわけではないため、サーバーの始動時に例外が発生しないように、このサンプル・データ・ソースを削除する必要があります。

WebSphere InterChange Server 環境では、システム全体に対してリソースが定義されるのは 1 回のみです。これを WebSphere Process Server 環境でシミュレートするため、`InstallAdministrativeObjects.py` スクリプトによってリソースがセル・スコープで定義されます。WebSphere Process Server システム内には、デフォルトの JDBC プロバイダー・テンプレートから作成された JDBC プロバイダーを使用するために、WebSphere 変数がノード・スコープで事前に定義されています。ノードごとにこの変数をカスタマイズすることができるように、この変数はノード・スコープで定義されます。このようなスコープの不一致のため、以下のいずれかの操作を実行する必要があります。

- 作成された JDBC プロバイダーが必要とする WebSphere 変数をセル・スコープで定義する。
- `InstallAdministrativeObjects.py` スクリプトを実行してから、JDBC プロバイダーをノード・スコープに移動する。

どちらの WebSphere 変数が必要であるかを判別するには、管理コンソールを使用して、生成された JDBC プロバイダーを調べます。管理コンソールから、「環境」>「WebSphere 変数」を選択して、必要な変数を作成します。詳しくは、WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センターの、`Defining WebSphere variables` を参照してください。

生成された `InstallAdministrativeObjects.py` スクリプトに含まれる、JDBC コネクター・プールを生成するためのコードの例を以下に示します。

```
dsName = "sqls"
create_datasource(dsName, JNDI_PREFIX + dsName, DATASOURCE_DESCRIPTION,
MS_SQL_JDBC_PROVIDER_NAME, MS_SQL_JDBC_PROVIDER_TYPE, "icsadmin", "icsadmin",
4, 50, "qaxs17", "1433", "wicsrepos")
```

wsadmin コマンドについて詳しくは、『wsadmin ツール』を参照してください。

既存の WebSphere InterChange Server リレーションシップ・データベースの使用

WebSphere Process Server で既存の WebSphere InterChange Server リレーションシップ・データベースを使用するために、wsadmin コマンドで

InstallAdministrativeObjects.py スクリプトを使用して、WebSphere Process Server 内にデータ・ソースおよびリレーションシップの構成情報を作成することができます。通常、WebSphere Process Server は、マイグレーションされたリレーションシップの構成情報を、リレーションシップのデプロイ時に自動的に作成します。既存のデータベースを使用できるようにするため、InstallAdministrativeObjects.py スクリプトでは、既存の WebSphere InterChange Server リレーションシップ・データベース用のデータベース接続と、リレーションシップ構成情報を、WebSphere Process Server 内に作成する必要があります。マイグレーション済みコンポーネントをデプロイする前に、InstallAdministrativeObjects.py スクリプトを実行します。その後、WebSphere Process Server は、リレーションシップをデプロイするときに、スクリプトが生成した構成情報を使用します。

生成された InstallAdministrativeObjects.py スクリプトに含まれる、リレーションシップ・データベース接続を生成するためのコードの例を以下に示します。

```
dsName = "ContactR"
create_datasource(dsName, JNDI_PREFIX + dsName, DATASOURCE_DESCRIPTION,
MS_SQL_JDBC_PROVIDER_NAME, MS_SQL_JDBC_PROVIDER_TYPE, "icsadmin", "icsadmin",
-1, -1, "9.26.230.56", "1433", "wicsrepos")

create_relationship("ContactR", "jdbc/wbi60migration/ContactR", "false")
create_role("ContactR", "ID1", "", "null", "", "null")
create_attribute("ContactR", "ID1", "JtextEmployeeID")
create_role("ContactR", "ID2", "", "null", "", "null")
create_attribute("ContactR", "ID2", "EmployeeID")
create_role("ContactR", "ID3", "", "null", "", "null")
create_attribute("ContactR", "ID3", "EmployeeID")
```

wsadmin コマンドについて詳しくは、『wsadmin ツール』を参照してください。

スケジュール済みイベントのマイグレーション

WebSphere InterChange Server スケジューラー・エントリーに対応する WebSphere Process Server コンポーネントがないため、WebSphere InterChange Server スケジューラー・エントリーのマイグレーションを実現するには、既存の WebSphere InterChange Server リポジトリ JAR ファイルから関連データを抽出し、WebSphere Process Server 共通データベース内の WebSphere Process Server スケジューラー・テーブルに、対応するエントリーを作成します。データは、Jython スクリプト内でストリング形式で表されます。WebSphere Process Server データベース内にスケジューラー・エントリーを作成するには、wsadmin を使用して InstallAdministrativeObjects.py スクリプトを実行します。

生成された InstallAdministrativeObjects.py スクリプトに含まれる、スケジューラ
ー・エントリーを生成するためのコードの例を以下に示します。

```
create_scheduler_entry("true", "stop", "JDBCConnector", "Connector",  
"2006-09-07T10:44:29.000PDT", "undefined", 0, 0)  
create_scheduler_entry("true", "start", "JTextConnector", "Connector",  
"2006-09-07T10:47:06.000PDT", "undefined", 0, 0)  
create_scheduler_entry("true", "stop", "jtext_jdbcCollab", "Collaboration",  
"2006-09-07T10:48:10.000PDT", "undefined", 0, 0)  
create_scheduler_entry("true", "start", "jtext_jdbcCollab", "Collaboration",  
"2006-09-07T10:48:10.000PDT", "undefined", 0, 0)  
create_scheduler_entry(true, "START", "JDBCConnector", "Connector",  
"2006-10-22T12:34.56.789CDT", "MINUTES", 20, 0):
```

Access Enterprise JavaBean (EJB) サポート

WebSphere InterChange Server は、J2EE EJB (Enterprise JavaBeans™) プロトコルを
使用した、クライアント・コードによるコラボレーションの起動をサポートしま
す。このコラボレーションの起動方法のサポートを、「AccessEJB」または
「AccessEJB for EJB」のサポートと呼びます。WebSphere Process Server は、後方
互換性を確保するために AccessEJB のサポートを提供します。AccessEJB サポート
では、呼び出される SCA BPEL モジュールが、本書で説明する WebSphere
InterChange Server マイグレーション・ツール群によって生成されていることが前提
となっています。コラボレーション名およびポート名 (つまり、AccessEJB 用の入
力パラメーター) から SCA モジュール名へのマッピング、インターフェース、およ
びビジネス・オブジェクト・タイプでは、このマイグレーション・ツール群で使用
される規則を前提としています。WebSphere Process Server での AccessEJB サポート
は、AccessEJB.zip プロジェクト交換ファイルとして提供されます。このファイル
は、*install_root/HeritageAPI* ディレクトリーにあります。AccessEJB サポートは、
SCA BPEL モジュールを呼び出す SCA モジュール・プロジェクト
(DynamicRouting) を参照する EJB (AccessEJB) から成ります。この SCA BPEL モ
ジュールは、WebSphere InterChange Server で呼び出されていたコラボレーションの
マイグレーション版です。DynamicRouting モジュールは、セレクター・コンポーネ
ントを使用して、AccessEJB に渡されたコラボレーション名とポート名を基に正し
い SCA ターゲットを選択します。WebSphere Process Server で AccessEJB サポート
を使用可能にするには、以下のようにします。

1. AccessEJB 呼び出しのターゲットであるコラボレーションを格納する WebSphere
InterChange Server リポジトリを、WebSphere Integration Developer にインポ
ートします。
2. AccessEJB.zip プロジェクト交換ファイルを WebSphere Integration Developer に
インポートします。
3. DynamicRouting プロジェクトを開き、セレクター・テーブルを更新して
AccessEJB が呼び出すマイグレーション済みモジュールをセレクター・テー
ブルに挿入します。
4. AccessEJB が呼び出す、BPEL コンポーネントを含むマイグレーション済みプロ
ジェクトに移動して、BPEL モジュールを参照するエクスポートを
DynamicRouting プロジェクト上にドラッグします。
5. AccessEJB を使用してアクセス可能にする BPEL モジュールごとに、ステップ
3 と 4 を繰り返します。
6. プロジェクトをビルドし、WebSphere Process Server サーバーに配置します。

7. WebSphere Process Server サーバーのランタイム・クラスパスに、必要なデータ・ハンドラーがあることを確認します。
8. アクセス・クライアントが WebSphere Process Server を使用できるようにするため、アクセス・クライアントが WebSphere Process Server サーバーを指し示し、Access EJB の検索時に JNDI 名 `com/crossworlds/access/business/cwsession/CwSession` を使用することを確認します。

DynamicSend API 構成

WebSphere InterChange Server では、DynamicSend API を使用して、1 つのコラボレーションを別のコラボレーションから直接呼び出すことができます。呼び出すコラボレーションを事前に設定しておく必要はなく、実行時に動的に決定することができます。WebSphere Process Server での DynamicSend API のサポートでは、162 ページの『Access Enterprise JavaBean (EJB) サポート』で説明した DynamicRouting プロジェクトを使用します。指定した BPEL モジュールを起動できるように DynamicSend API を使用可能にするには、『BaseCollaboration.dynamicSend メソッド呼び出しの使用可能化』で説明した手順に従ってください。

BaseCollaboration.dynamicSend メソッド呼び出しの使用可能化

WebSphere InterChange Server の BaseCollaboration.dynamicSend メソッド呼び出しをマイグレーション後に正常に機能させるには、AccessEJB プロジェクト交換ファイルの DynamicRouting プロジェクトを変更する必要があります。これには、次の 2 つの主要な手順を実行する必要があります。

1. WebSphere InterChange Server リポジトリをマイグレーションする。
2. DynamicSend API を使用可能にする。

WebSphere InterChange Server リポジトリをマイグレーションするには、次の手順を実行します。

1. DynamicSend API を呼び出すコラボレーションが入っている WebSphere InterChange Server リポジトリを、WebSphere Integration Developer にインポートします。
2. DynamicSend API 呼び出しのターゲットになっているコラボレーションまたはコネクタが入っている WebSphere InterChange Server リポジトリを、WebSphere Integration Developer にインポートします。
3. すべてをビルドし、すべてのエラーを訂正します。

DynamicSend API を使用可能にするには、次の手順を実行します。

1. AccessEJB.zip プロジェクト交換ファイルを WebSphere Integration Developer にインポートします。
2. DynamicRouting プロジェクトを開き、WebSphere InterChange Server の共有ライブラリーを DynamicRouting プロジェクトの依存関係に追加します。
3. BaseCollaboration.dynamicSend メソッドが呼び出す、コンポーネントを含むマイグレーション済みモジュールに移動して、モジュールを参照するエクスポートを DynamicRouting プロジェクト上にドラッグします。「SCA バインディングを使用するインポート (Import with SCA Binding)」を選択し、「OK」をクリックします。

4. DynamicRouting アセンブリー・ダイアグラム・ウィンドウで、PreRoute_TargetCollab_TargetPort をコピーして貼り付け、新規作成されたコピーの名前を PreRoute_ModuleName_ExportName に変更します (コピーされたインポートの名前は PreRoute_TargetCollab_TargetPortCopy になります)。
5. PreRoute_ModuleName_ExportName で、右側に配置された、1.1 と表示した小さなボックスで表されている参照を左クリックします。右クリックして「削除」を選択します。
6. ステップ 3 で生成されたインポートに PreRoute_ModuleName_ExportName を接続します。Java WSDL 参照質問に「no」で応答します。
7. インポートの名前を ModuleName_ExportName に変更します。アセンブリー・ダイアグラムへの変更を保存します。
8. DynamicRouting プロジェクトのセクター・テーブルを更新し、DynamicSend API が呼び出すマイグレーション済みモジュールを含めます。
 - a. Java パースペクティブ・パッケージ・エクスプローラー・ビューに切り替えます。DynamicRouting/com.ibm を展開し、テキスト・エディターで RoutingSelector.selt を開きます。
 - b. OperationSelectionRecord ブロックをコピーし、そのブロック全体を既存のブロックの直後に貼り付けます。
 - c. 新規ブロックで、componentName="PreRoute_TargetCollab_TargetPort" を componentName="PreRoute_ModuleName_ExportName" に変更します。さらに新規ブロックで、value="TargetCollab_TargetPort" を value="ModuleName_ExportName" に変更します。


```
<OperationSelectionRecord>
  <SelectionKey>
    <SelectionKeyElement xsi:type="selt:StringSingletonKey" value="
      TargetCollab_TargetPort"/>
  </SelectionKey>
  <SelectionData xsi:type="selt:SCAInternalComponent"
    componentName="PreRoute_TargetCollab_TargetPort"/>
</OperationSelectionRecord>
<OperationSelectionRecord>
<SelectionKey>
<SelectionKeyElement xsi:type="selt:StringSingletonKey"
  value="ModuleName_ExportName"/>
</SelectionKey>
<SelectionData xsi:type="selt:SCAInternalComponent" componentName
  ="PreRoute_ModuleName_ExportName"/>
</OperationSelectionRecord>
```
 - d. RoutingSelector.selt を保存して閉じます。
9. インプリメンテーション・ファイルを生成します。
 - a. **com.ibm.sel** を展開し、PreRoute_TargetCollab_TargetPortImpl.java をコピーして同じロケーションに貼り付けます。新しく作成された Java ファイルに PreRoute_ModuleName_ExportNameImpl.java という名前を付けます。
 - b. PreRoute_ModuleName_ExportNameImpl.java を編集します。メソッド名を、locateService.TestB0InterfacePartner から locateService_InterfaceNamePartner に変更します (InterfaceName はメソッドです)。TestB0InterfacePartner を InterfaceNamePartner に変更します。

- c. `PreRoute_ModuleName_ExportNameImpl.java` で
「`locateService_TestB0InterfacePartner`」を検索し、その名前を
`locateService_InterfaceNamePartner` に変更します。
- 10. Business Integration パースペクティブに戻ります。DynamicRouting アセンブリー・ダイアグラムを開きます。「**PreRoute_ModuleName_ExportName**」をクリックします。「プロパティ」を開き、「実装 (Implementation)」を選択します。「**クラス: (Class:)**」フィールドに、`com.ibm.sel.PreRoute_ModuleName_ExportNameImpl` と入力します。
- 11. すべての変更を保存します。
- 12. BaseCollaboration.dynamicSend メソッドから呼び出す他のすべてのモジュールで、ステップ 3 から 11 を繰り返します。現在のところ、モジュールを DynamicRouting テーブルに追加して実行時にアクセスできるようにしなければ、「これらのモジュールを動的に検索する」方法はありません。
- 13. dynamicSend API を呼び出すプロジェクトの場合は、次の手順を実行します。
 - a. DynamicRouting モジュールから「RoutingPacket」インターフェースをコピーして貼り付けます。
 - b. dynamicSend メソッドを呼び出すコンポーネントで、コピーされたインターフェース「RoutingPacket」を Reference_Partners に追加し、名前を「RoutingPacketPartner」に変更します。
 - c. それを保存します。
 - d. アセンブリー・ダイアグラムを開きます。DynamicRouting から「RoutingInput」をドラッグします。「SCA バインディングを使用するインポート (Import with SCA Binding)」を選択し、「OK」をクリックします。名前を「Import1」から「DynamicRouting」に変更します。
 - e. dynamicSend API を呼び出すコンポーネントを削除してアセンブリー・ダイアグラム・ウィンドウに再ドラッグし、「RoutingPacketPartner」参照を「DynamicRouting」に接続し、他の参照を再接続します。
- 14. すべての変更を保存してビルドし、すべてのエラーを訂正します。すべてのモジュールを EAR ファイルにエクスポートします。

イベント順序付けマイグレーション

WebSphere Process Server において、WebSphere InterChange Server で可能であったものと同様の形で、イベントの順序付けを行うための方法が提供されています。この主題について参考になる記事は、IBM developerWorks Web サイトから入手することができます。 <http://www.ibm.com/developerworks> にある「テクニカル・ライブラリー」で検索してください。

失敗したイベント

WebSphere Process Server での失敗したイベントの処理方法については、IBM developerWorks Web サイトの参考になる記事に説明があります。 <http://www.ibm.com/developerworks> にある「テクニカル・ライブラリー」で検索してください。

マップのマイグレーション

WebSphere InterChange Server のマイグレーションにより、WebSphere InterChange Server マップが WebSphere Process Server マップに変換されます。ビジネス・グラフ・マップとビジネス・オブジェクト・マップという 2 つの出力マップが生成されます。ビジネス・グラフ・マップは、ビジネス・オブジェクト・マップをサブマップとして呼び出します。ビジネス・グラフ・マップは、その名前と、呼び出すサブマップの名前を除けば、すべて同一です。これらのビジネス・グラフ・マップは、ビジネス・グラフ・レベルでのみ実行可能である、必要なマッピング手順を実行するためだけに存在します。ビジネス・オブジェクト・マップは、それぞれ固有であり、WebSphere InterChange Server マップをマイグレーションしたものです。

WebSphere InterChange Server 入力マップに、サポートされる WebSphere InterChange Server API ログ・メソッド用のカスタム・メッセージが含まれている場合、これらのメッセージは、プロパティ・ファイルに変換されます。

コラボレーションのマイグレーション

コラボレーション・テンプレート: WebSphere InterChange Server から WebSphere Process Server へのマイグレーション・ツールによって、WebSphere InterChange Server コラボレーション・テンプレートが WebSphere Process Server BPEL ファイルにマイグレーションされます。コラボレーション・テンプレートで定義されたトリガー・ポートごとに 1 つの BPEL ファイルが作成され、その名前は *CollaborationTemplateName_TriggeringPortName* という命名規則に基づくものになります。各 BPEL ファイルは、トリガー・ポートに関連付けられたビジネス・オブジェクト・タイプに基づくビジネス・オブジェクト・タイプを受け取ります。例えば、トリガー・ポートが Customer というビジネス・オブジェクト・タイプを取る場合、作成される BPEL ファイルでは、「TriggeringBusObj」変数タイプが Customer になります。

ループを維持する: WebSphere InterChange Server リポジトリをマイグレーションするときに、コラボレーション・テンプレートに存在していたループを維持するか、それらのループを解除するかを選択できます。ループを維持することを選択して、生成されたマイグレーション JAR ファイルを WebSphere Integrated Developer で表示した場合、WebSphere Integrated Developer ツールは現在ループをサポートしないため、検証エラーが表示されます。(WebSphere Process Server ではループをサポートしています。) WebSphere Integrated Developer で検証エラーを除去するには、BPEL ファイルを手動で更新する必要があります。BPEL ファイルを変更すると、WebSphere Integrated Developer BPEL エディター内からファイルを表示できなくなるという不都合が生じます。WebSphere Integrated Developer での検証エラーを除去するために必要な具体的な変更の実行方法について詳しくは、IBM developerWorks の記事「[Migrating WebSphere InterChange Server artifacts to WebSphere Process Server artifacts, Part 1: Migrating collaboration templates to BPEL](#)」の『Loop Unraveling』のセクションを参照してください。

ループを維持しない: ループを維持しないことを選択した場合、コラボレーション・テンプレート・ループは BPEL while ループに変換されます。変数を true に初期化すると、ループを少なくとも 1 回実行できます。ループを最初に実行した後、ループ変数が false に設定され、その後は通常のループ条件によって、ループを継

続いて実行するか終了するかが決まります。(注: このオプションの使用によって、WebSphere Integrated Developer 内部に検証エラーが発生することはありません。)

コラボレーション・オブジェクト: WebSphere InterChange Server から WebSphere Process Server へのマイグレーション・ツールによって、コラボレーション・オブジェクトがいくつかの Service Component Architecture (SCA) コンポーネントにマイグレーションされます。現在マイグレーションでは、以下のように、コラボレーション・テンプレートを参照するコラボレーション・オブジェクトをサポートします。

- サポートされるもの:
 - トリガー・ポートが 1 つ以上で、相関セットおよび非同期 in 呼び出しがない
 - トリガー・ポートがちょうど 1 つで、相関セットおよび非同期 in 呼び出しがある
- サポートされないもの:
 - マイグレーションでは、トリガー・ポートが 1 つ以上で、相関セットおよび非同期 in 呼び出しがある場合をサポートしません。この場合、結果として生成された成果物は、上述の最初の場合であるものとしてマイグレーションされます。さらに、欠落している SCA コンポーネントを手動で作成して、それらを適切に接続する必要があります。

SCA コンポーネント:

- エクスポート: コラボレーション・オブジェクトに関連付けられたコラボレーション・テンプレート内で定義されているトリガー・ポートのそれぞれに、エクスポートが作成されます。エクスポート名は *TriggeringPortName* です。
- BPEL へのエクスポート: データをエクスポートから BPEL ファイルへマップするインターフェース・マップが生成されます。インターフェース・マップ名は *Export_To_BPELname* です。トリガー・ポートがちょうど 1 つであり、コラボレーション・テンプレートに非同期 in 呼び出しがある場合、追加の SCA コンポーネントが作成されます。マイグレーションによって、インターフェース・マップが 1 つだけではなく 2 つ生成されます。1 つは同期呼び出し用、もう 1 つは非同期呼び出し用です。これら 2 つのインターフェース・マップのどちらに従うかを決定するには、Java コンポーネントを使用します。
- BPEL: すべてのトリガー・ポートで、エクスポートはインターフェース・マップに関連付けられ、そのインターフェース・マップは BPEL ファイルのインスタンスにマップされます。
- インポートする BPEL: トリガー・ポートでもそれ以外でも、すべてのポートに、BPEL ファイルをインポートへマップするインターフェース・マップがあります。インターフェース・マップ名は *BPEL_to_Port* です。
- インポート: 最後に、インポート・ファイルが作成されます。インポート名は *ConnectorName_BONameBG* です。

コラボレーション・テンプレートがどのように WebSphere Process Server BPEL ファイルにマイグレーションされるかについては、IBM developerWorks の記事「[Migrating WebSphere InterChange Server artifacts to WebSphere Process Server artifacts, Part 1: Migrating collaboration templates to BPEL](#)」を参照してください。

BPEL 変数をマイグレーション後に定義する必要がある

問題: WebSphere InterChange Server コラボレーション・テンプレートのポート定義で定義されていない変数が、パートナーの呼び出しで使用されています。マイグレーション後、変数は Business Process Execution Language (BPEL) 呼び出しで参照されますが、BPEL 変数としてセットアップされておらず、そのためモジュールに対して `serviceDeploy` コマンドを実行するときに、または WebSphere Integration Developer でモジュールを作成した後に、エラーのフラグが立てられます。**原因:** WebSphere Process Server の BPEL プロセスからパートナーを呼び出す場合は、呼び出しで使用されるすべてのオブジェクトを BPEL 変数として宣言し、使用されるオブジェクトのタイプを判別できるようにする必要があります。マイグレーション時には、コラボレーション・テンプレート内のポート宣言のみを調べて、宣言する必要がある BPEL 変数が判断されます。グローバル変数または ICS コラボレーション・テンプレート定義内の別の場所の断片で宣言される変数の場合は、マイグレーション・コードによってオブジェクト・タイプを確実に判別することはできないため、マイグレーションによって生成される BPEL ファイルでのオブジェクトに対しては BPEL 変数が宣言されません。**解決策:** マイグレーション後、呼び出し時に参照される変数を BPEL 変数として定義する必要があります。

WebSphere Process Server での logError API E メール通知の使用可能化

問題: WebSphere Process Server へのマイグレーション後、WebSphere InterChange Server の logError API が、WebSphere Interchange Server で構成されていたユーザーのリストに E メールを送信しません。**原因:** WebSphere InterChange Server では、指定されたユーザーのリストに対してエラー E メールを送信するように API 呼び出し logError を構成可能です。ただし、サーバー上で構成されるこのユーザー・リストにはマイグレーション・コードからアクセスできないため、WebSphere Process Server で手動設定する必要があります。**解決策:** WebSphere Process Server で WebSphere InterChange Server の logError の E メール通知機能を使用可能にするには、マイグレーションによって生成される各 BPEL ファイルで、`LOGERROR_EMAIL_LIST` という新規の BPEL 環境変数を作成します。ログ・エラーの E メールを受信する必要がある E メール・ユーザーのリストを使用してこの変数を設定してください。リスト内の名前はコンマで区切ります。

WebSphere Process Server での非同期呼び出しの処理

問題: 同じコネクタで `async-in` イベントとトリガー・イベントの両方を受信可能な状態になっている場合に、`async-in` イベントがトリガー・イベントとして動作します。**原因:** `async-in` イベントとトリガー・イベントの両方を同じコネクタで受信可能な状態になっている場合、マイグレーションされたアプリケーションはイベントのタイプを判別できません。デフォルトにより、このシナリオでマイグレーションされたアプリケーションでは、すべてのイベントがトリガー・イベントとして扱われます。**解決策:** イベントのタイプが `async-in` とトリガーのどちらなのかを判別するアプリケーション固有のロジックを、マイグレーションされたアプリケーションに追加する必要があります。同じコネクタでトリガー・イベントと `async-in` イベントを受信可能なマイグレーション済みモジュールは、JavaSelector と呼ばれるコンポーネントを持ちます。JavaSelector コンポーネントのインプリメンテーション・コードには、以下に示す `AsyncIn()` メソッドが含まれるようになります。このメソッド

は、イベントが `async-in` とトリガーのどちらなのかを検査するロジックを使用して更新する必要があります。このロジックは、各アプリケーションに固有のもので、処理されるイベントの性質に基づいています。

```
/** * Method generated to support async inbound service call routing */
public boolean isAsyncIn()
{ //Add custom code here
  //TODO
  return false;
}
```

Network Deployment のアップグレード後に AppScheduler の開始を可能にする

問題: WebSphere Process Server 6.0.1.x Network Deployment 構成を WebSphere Process Server 6.1 にマイグレーションした後、AppScheduler が、アップグレードされていない WebSphere Process Server 6.0.1.x サーバーとクラスター上で始動できません。次のような例外が生成されます。

```
WSVR0040E: addEjbModule failed for WBISchedulerEJB.jar
[class com.ibm.ws.runtime.component.
DeployedEJBModuleImpl] java.lang.NoClassDefFoundError:
com/ibm/wbiserver/scheduler/common/AppSchedulerException
```

原因: WebSphere Process Server 6.0.1.x Network Deployment 構成を WebSphere Process Server 6.1 にマイグレーションした後、AppScheduler アプリケーションが、WebSphere Process Server 6.0.1.x バージョンの `wbischedulercommon.jar` ファイルで `AppSchedulerException` クラスを探しますが、ローカル・システムの `install_root/lib` ディレクトリー内で見つけることができません。そのため、

```
java.lang.NoClassDefFoundError: com/ibm/wbiserver/scheduler/common/
```

`AppSchedulerException` 例外がスローされます。 **解決策:** WebSphere Process Server 6.0.1.x バージョンの `wbischedulercommon.jar` ファイルを、その JAR ファイルの WebSphere Process Server 6.1 または WebSphere Process Server 6.0.2.x バージョンに置き換えてください。新しい JAR ファイルは、WebSphere Process Server 6.1.x の `install_rootAppScheduler/lib` ディレクトリー、または WebSphere Process Server 6.0.2 の `install_root/lib` ディレクトリーから入手できます。その JAR ファイルを WebSphere Process Server 6.0.1.x の `lib` ディレクトリーにコピーして、既存の JAR ファイルと置き換えます。WebSphere Process Server は、`lib` ディレクトリー内のすべてのファイルを拡張子に関係なく JAR ファイルとして選択するので、既存の JAR ファイルの名前変更を行って `lib` ディレクトリーに残すことはしないでください。次に、サーバーまたはクラスターを再始動し、WebSphere Process Server が新しい JAR ファイルを選択するようにします。

WebSphere Process Server での相関値の処理

問題: WebSphere Process Server において、新規イベントが既存の相関値を使用しようとすると失敗します。そのような場合には、

```
CWWBE0074E: Correlation violation in activity 'null' for correlation set
'CorrelationSetA'java.sql.
SQLException: Could not insert new row - duplicate value in a UNIQUE INDEX column
```

というエラー・メッセージが表示されます。 **原因:** WebSphere InterChange Server でのコラボレーションまたはプロセス・インスタンスが完了すると、障害を処理する場合を除いて、そのインスタンスに関連するデータは削除されます。WebSphere

Process Server, では、プロセス・インスタンス関連データのパーシスタンスを Business Process Execution Language (BPEL) の「プロセスは完了後に自動的に削除 (Automatically delete the process after completion)」というオプションによって制御します。 WebSphere InterChange Server から WebSphere Process Server へのマイグレーション・ウィザードによって生成される BPEL ファイルでは、このオプションが選択されていません。そのため、プロセス・インスタンスが完了しても、手動でクリーンアップされるまではプロセス・インスタンス・データが残ります。プロセスが相関セットを定義する場合、プロセス・インスタンスによってロックされる相関値は、プロセスが完了した後であっても、プロセス・インスタンス・データが存続する限り残ります。結果として、同じ相関値を使用しようとする新規イベントは、前のプロセス・インスタンスのデータが存続する限り失敗することになります。この振る舞いは、 WebSphere InterChange Server の場合とは異なります。 WebSphere InterChange Server では、前のインスタンスが完了すると同時に、相関セット値が重複する新規イベントを処理できます。**解決策:** 相関セット値が重複する複数のイベントが存在する場合の WebSphere InterChange Server の振る舞いをシミュレートするために、 BPEL オプション「完了後にプロセスを自動的に削除 (Automatically delete the process after completion)」が選択されるようにして、プロセス・インスタンスが完了すると同時にプロセス・インスタンス・データが削除され、相関値がロック解除されるようにすることができます。このオプションを選択する前に、 WebSphere Process Server において障害が処理される方法を調べ、十分に理解してください。また、失敗したイベントの処理方針が、このオプションが設定された場合に自動的に削除されるデータに依存しないことを確認してください。

マイグレーションされたアプリケーションのパッケージ化とデプロイ

reposMigrate コマンドを使用して WebSphere InterChange Server リポジトリをマイグレーションした後、生成された JAR ファイルを WebSphere Process Server にデプロイするには、それらの JAR ファイルを EAR ファイルにパッケージする必要があります。これを実行するには、マイグレーションによって生成された各 JAR ファイルを WebSphere Integration Developer にインポートして、そのモジュールを EAR ファイルとしてエクスポートするか、または serviceDeploy コマンドを使用できます。serviceDeploy コマンドは、JAR ファイルを入力として受け入れ、配置可能な EAR ファイルを出力します。マイグレーション・コードを EAR ファイルにパッケージする際には、生成されたマイグレーション済み JAR ファイルをコンパイルします。このときに検証エラーが発生した場合、その原因として最も可能性が高いのは、サポートされない WebSphere InterChange Server API を使用したか、 WebSphere InterChange Server に存在していたが WebSphere Process Server クラス・パスにはまだ組み込まれていないサード・パーティー API を使用したことです。サポートされない API を削除し、サード・パーティーのクラスを WebSphere Process Server クラス・パスに追加します。

また、検証エラーは、事前マイグレーションの推奨される慣例に従わないために発生したか、その成果物に対してまだ実行する必要がある事後マイグレーション作業があることを示している可能性もあります。マイグレーション・エラーと同様に、各検証エラーは、エラーごとに処理する必要があります。推奨される事前マイグレーションの慣例に従っていなかった場合、リポジトリを更新して再度マイグレーションするか、または出力された成果物を編集して、問題を取り除くことができます。

その他の検証エラーは、これらの成果物が何もないところから作成されるものとして解決する必要があります。一般的な成果物のエラーおよびその解決策の概要を説明するバリデーターの資料を参照してください。自動化マイグレーションでは、必然的に、ユーザーのプログラムの意図を完全に明らかにすることはできません。できるだけ実際に近い推測をすることのみ可能です。そのため、検証エラーがない場合でも、マイグレーション済み成果物が意図したとおりに実行されない可能性があります。すべての成果物を検討して、各成果物の意図した目的が、マイグレーション済みの内容によって達成されることを確認する必要があります。

関連概念

199 ページの『WebSphereInterChange Server からマイグレーションする場合の制限事項』

WebSphere InterChange Server の特性の中には、WebSphere Process Server によって正確に再現されないものがあります。そのため、WebSphere InterChange server と同じように実行するために、マイグレーション後にアプリケーションを変更しなければならない場合があります。

200 ページの『WebSphere InterChange Server からのマイグレーションのトラブルシューティング』

マイグレーションで発生する問題の解決策と、ロギングとトレースをオンにする方法について説明します。

関連タスク

154 ページの『reposMigrate コマンドを使用した WebSphere InterChange Server の成果物のマイグレーション』

WebSphere InterChange Server の成果物から WebSphere Process Server の成果物へのマイグレーションには、**reposMigrate** コマンドを使用します。

関連資料

 [serviceDeploy コマンド](#)

serviceDeploy コマンドを使用して、Service Component Architecture (SCA) 対応モジュールをサーバーにインストール可能な Java アプリケーションとしてパッケージします。このコマンドは、wsadmin を使用してバッチ・インストールを実行する場合に有効です。

146 ページの『事前マイグレーションの考慮事項』

WebSphere InterChange Server 成果物を WebSphere Process Server にマイグレーションする作業を容易にするために、WebSphere InterChange Server 用の統合成果物を開発するための以下のガイドラインを検討してください。

関連情報

[Wsadmin ツール](#)

[WebSphere Integration Developer インフォメーション・センター](#)

[IBM developerWorks](#)

[Migrating WebSphere InterChange Server artifacts to WebSphere Process Server artifacts, Part 1: Migrating collaboration templates to BPEL](#)

[WebSphere 変数の定義 \(Defining WebSphere variables\)](#)

WebSphere InterChange Server のマイグレーション後のグローバル・セキュリティーの構成

WebSphere InterChange Server からマイグレーションされたプロジェクトを、WebSphere Process Server 環境で正常に実行できるようにするには、以下のセキュリティー構成ステップを追加で実行します。

始める前に

最初に、WebSphere Process Server のセキュリティーを構成する必要があります。『アプリケーションとその環境の保護』を参照してください。特に、『アダプターの保護』および『エンドツーエンド・セキュリティーの構築』に記載されたステップを必ず完了しておいてください。さらに、モジュールごとの EAR ファイルをインストールします。詳しくは、『セキュア・アプリケーションのデプロイ (インストール)』を参照してください。

このタスクについて

上記のタスクを実行すると、以下に示す構成ステップを実行できるようになります。

- メッセージ駆動型 Bean をアクティベーション・スペックにバインドする
- リソース参照をリソースにマップする
- セキュリティー・ロールをユーザーまたはグループにマップする (Common Based Event のモニター時のみ必要)
- RunAs ロールをマップする (Common Based Event のモニター時のみ必要)

注: セキュリティー・ロールをユーザーまたはグループにマップするステップと RunAs ロールをマップするステップは、EJB プロジェクトの EJB デプロイメント記述子に RunAs ロールが定義された場合のみ、管理コンソールから実行できます。アセンブリー・ツールを使用した RunAs ロールの定義については、WebSphere Application Server Network Deployment バージョン 6.1 インフォメーション・センター内の『アセンブリー・ツールを使用した RunAs ロールへのユーザーのマッピング (Mapping users to RunAs roles using an assembly tool)』を参照してください。

プロシージャ

1. メッセージ駆動型 Bean をアクティベーション・スペックにバインドします。
 - a. 管理コンソールで、「アプリケーション」>「エンタープライズ・アプリケーション」を選択します。
 - b. 右側のパネルで、インストールしたばかりのアプリケーションの名前を選択します。(名前の左側のチェック・ボックスではなく、名前を選択してください。)
 - c. 右側のパネルに戻り、「Enterprise Java Bean プロパティー」の下で、「**Message Driven Bean** リスナー・バインディング」を選択します。
 - d. インポートまたはエクスポート EJB (「_import」または「_export」で始まる EJB 名で示される) ごとに、「バインディング」列の下の「ActivationSpec 認証別名」フィールドで「**SCA_Auth_Alias**」を指定します。
 - e. 「OK」を選択し、「保管」を選択します。
2. リソース参照をリソースにマップします。

- a. 管理コンソールで、「アプリケーション」>「エンタープライズ・アプリケーション」を選択します。
 - b. 右側のパネルで、インストールしたばかりのアプリケーションの名前を選択します。(名前の左側のチェック・ボックスではなく、名前を選択してください。)
 - c. 右側のパネルの「参照」の下で、「リソース参照」を選択します。
 - d. javax.jms.ConnectionFactory の下の「認証方式の指定:」フィールドで、「デフォルト・メソッドの使用 (多対 1 のマッピング)」ラジオ・ボタンを選択します。
 - e. 「認証データ入力の選択 (Select authentication data entry)」プルダウン・メニューで、「SCA_Auth_Alias」を選択します。
 - f. すべてのモジュールを選択するには、チェック・ボックスにチェック・マークを付けます。
 - g. 「適用」を選択してから、「OK」を選択し、次に「保管」を選択します。
3. セキュリティー・ロールをユーザー・グループにマップします。
- a. 管理コンソールで、「アプリケーション」>「エンタープライズ・アプリケーション」を選択します。
 - b. 右側のパネルで、インストールしたばかりのアプリケーションの名前を選択します。(名前の左側のチェック・ボックスではなく、名前を選択してください。)
 - c. 右側のパネルの「詳細プロパティー」の下で、「ユーザー/グループ・マッピングへのセキュリティー・ロール」を選択します。
 - d. マップしたいロールの左側のチェック・ボックスを選択してから、「ユーザーの検索」を選択します。
 - e. 「検索」を選択して、ロールにマップできるユーザーのリストを表示し、正しいユーザー名を「選択済み:」列に移動します。
 - f. 「OK」を選択します。「ユーザー/グループ・マッピングへのセキュリティー・ロール」パネルが再度表示されます。
 - g. ロールに対応する「全員?」列および「全認証者?」列のチェック・ボックスのチェック・マークを外してから、「OK」をクリックし、次に「保管」をクリックします。
4. RunAs ロールをマップします。
- a. 管理コンソールで、「アプリケーション」>「エンタープライズ・アプリケーション」を選択します。
 - b. 右側のパネルで、インストールしたばかりのアプリケーションの名前を選択します。(名前の左側のチェック・ボックスではなく、名前を選択してください。)
 - c. 右側のパネルの「詳細プロパティー」の下で、「ユーザー RunAs ロール」を選択します。
 - d. ステップ 3 でマップしたロールの横にあるチェック・ボックスを選択します。
 - e. ステップ 3e で選択したユーザー名に対応するユーザー名をユーザー名フィールドに入力し、そのパスワードをパスワード・フィールドに入力します。
 - f. 「適用」を選択します。

- g. 「OK」を選択し、「保管」を選択します。

次のタスク

すべての EAR プロジェクトをインストールして構成した後、管理コンソールで「アプリケーション」>「エンタープライズ・アプリケーション」を選択し、インストールされたマイグレーション済みプロジェクトを開始します。それらが正常に開始した場合、イベントがサーバーで処理されるように、インバウンド・コネクターの 1 つを介してイベントを送信できるようになります。

関連タスク

アダプターの保護

WebSphere Process Server では、WebSphere Business Integration Adapters と WebSphere Adapters という 2 つのタイプのアダプターがサポートされています。ここでは、両タイプのアダプターのセキュリティーについて説明します。

セキュア・アプリケーションのデプロイ (インストール)

セキュリティー制約 (保護されたアプリケーション) を持つアプリケーションのデプロイは、セキュリティー制約なしのアプリケーションのデプロイとほぼ同じです。唯一の違いは、ユーザーとグループを保護されたアプリケーションのロールに割り当てることが必要な場合もあるという点です。なお、この保護されたアプリケーションでは、正しいアクティブ・ユーザー・レジストリーが必要になります。保護されたアプリケーションをインストールする場合は、ロールをアプリケーション内に事前に定義します。代行がアプリケーションに必要な場合は、RunAs ロールも定義し、有効なユーザー名とパスワードを指定する必要があります。

エンドツーエンド・セキュリティーの構築

構築可能なさまざまなエンドツーエンド・セキュリティーのシナリオがあります。これらの各シナリオでは、異なるセキュリティーの手順が必要になる可能性があります。ここでは、必要なセキュリティー・オプションを持つ数種類の標準的なシナリオを提供します。

関連情報

Wsadmin ツール

WebSphere InterChange Server v4.3 の資料

アセンブリー・ツールを使用した RunAs ロールへのユーザーのマッピング (Mapping users to RunAs roles using an assembly tool)

アプリケーションとその環境の保護

WebSphere Process Server 環境およびお客様のアプリケーションのセキュリティーは、非常に重要です。

WebSphere Business Integration データ・ハンドラーのサポート

データ・ハンドラー・サポート API によって、特定のデータ・ハンドラー方式を AccessEJB、WebSphere Process Server SCA Java コンポーネント、または WebSphere Process Server バインディングから起動できます。

WebSphere Process Server (バージョン 6.0.2.3 以上) は、データ・ハンドラー・サポート API (アプリケーション・プログラミング・インターフェース) を提供しており、これを使用して WebSphere Business Integration データ・ハンドラー方式を、AccessEJB、WebSphere Process Server SCA Java コンポーネント、または WebSphere Process Server バインディングから起動できます。Access EJB は、JService 呼び出しが入力ビジネス・オブジェクトを適切なマイグレーション済みモジュールに送付できるようにする EJB として複製されています。マイグレーション済みモジュール内の BPEL ファイルは、元の WebSphere InterChange Server ターゲット・コラボレーションの代わりに呼び出されます。

WebSphere Process Server バインディングは、データ・バインディングを起動し、データ変換を実行します。WebSphere Process Server は、ユーザー定義データ・バインディングを提供する機能と、複数の組み込みデータ・バインディングを提供します。ユーザー定義またはカスタムのデータ・バインディングを実装して、WebSphere Business Integration データ・ハンドラーを起動できます。

カスタム・データ・バインディング実装を提供することによって、データ・ハンドラー・サポート API を介して、WebSphere Business Integration データ・ハンドラーを活用できるようになります。データ・ハンドラー・サポート API は、既存の WebSphere Business Integration データ・ハンドラー・インターフェース・メソッドにラッパー・メソッドを提供し、これによって、WebSphere Business Integration ビジネス・オブジェクトと SDO 間の変換を実行できます。

データ・ハンドラー・サポート API

カスタム・データ・バインディング実装を提供することによって、データ・ハンドラー・サポート API を介して、WebSphere Business Integration データ・ハンドラーを活用できるようになります。この API は、カスタム・データ・バインディング、または Java コンポーネントから呼び出すことができる public メソッドのセットを定義します。これは、プロセス・サーバー・バインディングから、テキスト・ベースの WebSphere Business Integration データ・ハンドラーを起動する方法を提供します。API メソッドは以下のとおりです。

```
getSDOFromString(String inputString, String sdoName, String  
metaObjectName, String mimeType) (Returns dataObject)
```

```
getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)  
(Returns String)
```

Java クラス `com.ibm.wbi.datahandler.JavaConnectorUtilDH` を使用して、これらのメソッドにアクセスできます。IBM では、このクラスをデータ・バインディングまたは Java コンポーネントから使用することをお勧めします。既存のコードを保有している場合は、`AppSide_Connector.JavaConnectorUtil` クラスを使用できます。

使用法

データ・ハンドラー・サポート API で定義されているメソッドは、WebSphere Process Server バインディング、または Java コンポーネントから起動できます。しかし、データは通常 WebSphere Process Server 環境のバインディングで変換されるため、IBM では、データ・ハンドラー・サポート API のメソッドを Java コンポーネントではなく、カスタム・データ・バインディングから起動することを特にお勧めします。

制限

データ・ハンドラー・サポート API には、以下の制限事項があります。

- バイナリー変換メソッドはサポートされません。つまり、`getBytesFromSDO()`、`getStreamFromSDO()`、`getSDO(byte[])`、および同様の呼び出しはサポートされません。
- `setEncoding()`、`setLocale()` および `setOptions()` メソッドは、データ・ハンドラー・サポート API によって公開されません。
- 動的子メタ・オブジェクトはサポートされません。
- 新規オブジェクトの作成には WebSphere Business Integration Adapter ビジネス・オブジェクト・ツールを使用する必要があります。

関連資料

『サポートされる WebSphere InterChange Server API』

WebSphere Process Server および WebSphere Integration Developer で提供される WebSphere InterChange Server のソース成果物のマイグレーション・ツールに加えて、WebSphere Process Server は、WebSphere InterChange Server で提供されていた API の多くもサポートします。マイグレーション・ツールは、マイグレーション時に極力カスタム断片コードを保持することにより、これらの WebSphere InterChange Server API と連動して動作します。

関連情報

IBM WebSphere Business Integration Adapters/IBM WebSphere InterChange Server Data Handler Guide

サポートされる WebSphere InterChange Server API

WebSphere Process Server および WebSphere Integration Developer で提供される WebSphere InterChange Server のソース成果物のマイグレーション・ツールに加えて、WebSphere Process Server は、WebSphere InterChange Server で提供されていた API の多くもサポートします。マイグレーション・ツールは、マイグレーション時に極力カスタム断片コードを保持することにより、これらの WebSphere InterChange Server API と連動して動作します。

注: これらの API は、新しい WebSphere Process Server API を使用するための変更が可能になるまでは、マイグレーション済みの WebSphere InterChange Server アプリケーションをサポートするためだけに提供されます。

サポートされる WebSphere InterChange Server API を以下に示します。これらの API は、WebSphere InterChange Server で提供する機能と同様の機能を WebSphere Process Server において提供します。これらの API の機能の説明については、WebSphere InterChange Server v4.3 の資料を参照してください。

CwBiDiEngine

AppSide_Connector/

- BiDiBOTransformation(BusinessObject, String, String, boolean):BusinessObj
- BiDiBusObjTransformation(BusObj, String, String, boolean):BusObj
- BiDiStringTransformation(String, String, String):String

JavaConnectorUtil

AppSide_Connector/

- INFRASTRUCTURE_MESSAGE_FILE
- CONNECTOR_MESSAGE_FILE
- XRD_WARNING
- XRD_TRACE
- XRD_INFO
- XRD_ERROR
- XRD_FATAL
- LEVEL1
- LEVEL2
- LEVEL3
- LEVEL4
- LEVEL5
- createBusinessObject(String):BusinesObjectInterface
- createBusinessObject(String, Locale):BusinesObjectInterface
- createBusinessObject(String, String):BusinesObjectInterface
- createContainer(String):CxObjectContainerInterface
- generateMsg(int, int, int, int, int, Vector):String
- generateMsg(int, int, int, int, Vector):String
- getBlankValue():String
- getEncoding():String
- getIgnoreValue():String
- getLocale():String
- getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)
- getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)
- isBlankValue(Object):boolean
- isIgnoreValue(Object):boolean
- isTraceEnabled(int):boolean
- logMsg(String)
- logMsg(String, int)
- traceWrite(int, String)

JavaConnectorUtilDH

datahandler/

**wbi/
ibm/
com/**

- getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)
- getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)

BusObj

Collaboration/

- BusObj(DataObject)
- BusObj(String)
- BusObj(String, Locale)
- copy(BusObj)
- duplicate():BusObj
- equalKeys(BusObj):boolean
- equals(Object):boolean
- equalsShallow(BusObj):boolean
- exists(String):boolean
- get(int):Object
- get(String):Object
- getBoolean(String):boolean
- getBusObj(String):BusObj
- getBusObjArray(String):BusObjArray
- getCount(String):int
- getDouble(String):double
- getFloat(String):float
- getInt(String):int
- getKeys():String
- getLocale():java.util.Locale
- getLong(String):long
- getLongText(String):String
- getString(String):String
- getType():String
- getValues():String
- getVerb():String
- isBlank(String):boolean
- isKey(String):boolean
- isNull(String):boolean
- isRequired(String):boolean
- keysToString():String
- set(BusObj)

- set(int, Object)
- set(String, boolean)
- set(String, double)
- set(String, float)
- set(String, int)
- set(String, long)
- set(String, Object)
- set(String, String)
- setContent(BusObj)
- setDefaultAttrValues()
- setKeys(BusObj)
- setLocale(java.util.Locale)
- setVerb(String)
- setVerbWithCreate(String, String)
- setWithCreate(String, boolean)
- setWithCreate(String, BusObj)
- setWithCreate(String, BusObjArray)
- setWithCreate(String, double)
- setWithCreate(String, float)
- setWithCreate(String, int)
- setWithCreate(String, long):
- setWithCreate(String, Object)
- setWithCreate(String, String)
- toString():String
- validData(String, boolean):boolean
- validData(String, BusObj):boolean
- validData(String, BusObjArray):boolean
- validData(String, double):boolean
- validData(String, float):boolean
- validData(String, int):boolean
- validData(String, long):boolean
- validData(String, Object):boolean
- validData(String, String):boolean

BusObjArray

Collaboration/

- addElement(BusObj)
- duplicate():BusObjArray
- elementAt(int):BusObj
- equals(BusObjArray):boolean

- `getElements():BusObj[]`
- `getLastIndex():int`
- `max(String):String`
- `maxBusObjArray(String):BusObjArray`
- `maxBusObjs(String):BusObj[]`
- `min(String):String`
- `minBusObjArray(String):BusObjArray`
- `minBusObjs(String):BusObj[]`
- `removeAllElements()`
- `removeElement(BusObj)`
- `removeElementAt(int)`
- `setElementAt(int, BusObj)`
- `size():int`
- `sum(String):double`
- `swap(int, int)`
- `toString():String`

BaseDLM

DLM/

- `BaseDLM(BaseMap)`
- `getDBConnection(String):CwDBConnection`
- `getDBConnection(String, boolean):CwDBConnection`
- `getName():String`
- `getRelConnection(String):DtpConnection`
- `implicitDBTransactionBracketing():boolean`
- `isTraceEnabled(int):boolean`
- `logError(int)`
- `logError(int, Object[])`
- `logError(int, String)`
- `logError(int, String, String)`
- `logError(int, String, String, String)`
- `logError(int, String, String, String, String)`
- `logError(int, String, String, String, String, String)`
- `logError(String)`
- `logInfo(int)`
- `logInfo(int, Object[])`
- `logInfo(int, String)`
- `logInfo(int, String, String)`
- `logInfo(int, String, String, String)`
- `logInfo(int, String, String, String, String)`

- logInfo(int, String, String, String, String, String)
- logInfo(String)
- logWarning(int)
- logWarning(int, Object[])
- logWarning(int, String)
- logWarning(int, String, String)
- logWarning(int, String, String, String)
- logWarning(int, String, String, String, String)
- logWarning(int, String, String, String, String, String)
- logWarning(int, String, String, String, String, String, String)
- logWarning(String)
- raiseException(RuntimeEntityException)
- raiseException(String, int)
- raiseException(String, int, Object[])
- raiseException(String, int, String)
- raiseException(String, int, String, String)
- raiseException(String, int, String, String, String)
- raiseException(String, int, String, String, String, String)
- raiseException(String, int, String, String, String, String, String)
- raiseException(String, String)
- releaseRelConnection(boolean)
- trace(int, int)
- trace(int, int, Object[])
- trace(int, int, String)
- trace(int, int, String, String)
- trace(int, int, String, String, String)
- trace(int, int, String, String, String, String)
- trace(int, int, String, String, String, String, String)
- trace(int, String)
- trace(String)

CwDBConnection

CwDBConnection/

CxCommon/

- beginTransaction()
- commit()
- executePreparedSQL(String)
- executePreparedSQL(String, Vector)
- executeSQL(String)
- executeSQL(String, Vector)
- executeStoredProcedure(String, Vector)
- getUpdateCount():int

- hasMoreRows():boolean
- inTransaction():boolean
- isActive():boolean
- nextRow():Vector
- release()
- rollback()

CwDBConstants

CwDBConnection/

CxCommon/

- PARAM_IN - 0
- PARAM_INOUT - 1
- PARAM_OUT - 2

CwDBStoredProcedureParam

CwDBConnection/

CxCommon/

- CwDBStoredProcedureParam(int, Array)
- CwDBStoredProcedureParam(int, BigDecimal)
- CwDBStoredProcedureParam(int, boolean)
- CwDBStoredProcedureParam(int, Boolean)
- CwDBStoredProcedureParam(int, byte[])
- CwDBStoredProcedureParam(int, double)
- CwDBStoredProcedureParam(int, Double)
- CwDBStoredProcedureParam(int, float)
- CwDBStoredProcedureParam(int, Float)
- CwDBStoredProcedureParam(int, int)
- CwDBStoredProcedureParam(int, Integer)
- CwDBStoredProcedureParam(int, java.sql.Blob)
- CwDBStoredProcedureParam(int, java.sql.Clob)
- CwDBStoredProcedureParam(int, java.sql.Date)
- CwDBStoredProcedureParam(int, java.sql.Struct)
- CwDBStoredProcedureParam(int, java.sql.Time)
- CwDBStoredProcedureParam(int, java.sql.Timestamp)
- CwDBStoredProcedureParam(int, Long)
- CwDBStoredProcedureParam(int, String)
- CwDBStoredProcedureParam(int, String, Object)
- getParamType():int getValue():Object

DataHandler (抽象クラス)

DataHandlers/

crossworlds/

com/

- createHandler(String, String, String):DataHandler
- getBO(InputStream, Object):BusinessObjectInterface
- getBO(Object, BusinessObjectInterface, Object)
- getBO(Object, Object):BusinessObjectInterface
- getBO(Reader, BusinessObjectInterface, Object) (抽象メソッド)
- getBO(Reader, Object):BusinessObjectInterface (抽象メソッド)
- getBO(String, Object):BusinessObjectInterface
- getBOName(InputStream):String
- getBOName(Reader):String
- getBOName(String):String
- getBooleanOption(String):boolean
- getEncoding():String
- getLocale():Locale
- getOption(String):String
- getStreamFromBO(BusinessObjectInterface, Object):InputStream (抽象メソッド)
- getStringFromBO(BusinessObjectInterface, Object):String (抽象メソッド)
- setConfigMOname(String)
- setEncoding(String)
- setLocale(Locale)
- setOption(String, String)
- traceWrite(String, int)

NameHandler (抽象クラス)

DataHandlers/

crossworlds/

com/

- getBOName(Reader, String):String (抽象メソッド)

ConfigurationException (extends java.lang.Exception)

Exceptions/

DataHandlers/

crossworlds/

com/

MalformedDataException (extends java.lang.Exception)

Exceptions/

DataHandlers/

crossworlds/

com/

NotImplementedException (extends java.lang.Exception)

Exceptions/

DataHandlers/

crossworlds/

com/

BusinessObjectInterface

CxCommon/

- clone():Object
- dump():String
- getAppText():String
- getAttrCount():int
- getAttrDesc(int):CxObjectAttr
- getAttrDesc(String):CxObjectAttr
- getAttribute(String):Object
- getAttributeIndex(String):int
- getAttributeType(int):int
- getAttributeType(String):int
- getAttrName(int):String
- getAttrValue(int):Object
- getAttrValue(String):Object
- getBusinessObjectVersion():String
- getDefaultAttrValue(int):String
- getDefaultAttrValue(String):String
- getLocale():String
- getName():String
- getParentBusinessObject():BusinessObjectInterface
- getVerb():String
- getVerbAppText(String):String
- isBlank(int):boolean
- isBlank(String):boolean
- isIgnore(int):boolean
- isIgnore(String):boolean
- isVerbSupported(String):boolean
- makeNewAttrObject(int):Object
- makeNewAttrObject(String):Object
- setAttributeWithCreate(String, Object)
- setAttrValue(int, Object)
- setAttrValue(String, Object)
- setDefaultAttrValues()
- setLocale(Locale)
- setLocale(String)
- setVerb(String)

CxObjectAttr

CxCommon/

- BOOLEAN

- BOOLSTRING
- DATE
- DATESTRING
- DOUBLE
- DOUBSTRING
- FLOAT
- FLTSTRING
- INTEGER
- INTSTRING
- INVALID_TYPE_NUM
- INVALID_TYPE_STRING
- LONGTEXT
- LONGTEXTSTRING
- MULTIPLECARDSTRING
- OBJECT
- SINGLECARDSTRING
- STRING
- STRSTRING
- equals(Object):boolean
- getAppText():String
- getCardinality():String
- getDefault():String
- getMaxLength():int
- getName():String
- getRelationType():String
- getTypeName():String
- getTypeNum():String
- hasCardinality(String):boolean
- hasName(String):boolean
- hasType(String):boolean
- isForeignKeyAttr():boolean
- isKeyAttr():boolean
- isMultipleCard():boolean
- isObjectType():boolean
- isRequiredAttr():boolean
- isType(Object):boolean

CxObjectContainerInterface

CxCommon/

- getBusinessObject(int):BusinessObjectInterface

- getObjectCount():int
- insertBusinessObject(BusinessObjectInterface)
- removeAllObjects()
- removeBusinessObjectAt(int)
- setBusinessObject(int, BusinessObjectInterface)

DtpConnection

Dtp/

CxCommon/

- beginTran()
- commit()
- executeSQL(String)
- executeSQL(String, Vector)
- executeStoredProcedure(String, Vector)
- getUpdateCount():int
- hasMoreRows():boolean
- inTransaction():boolean
- isActive():boolean
- nextRow():Vector
- rollback()

DtpDataConversion

Dtp/

CxCommon/

- BOOL_TYPE - 4
- CANNOTCONVERT - 2
- DATE_TYPE - 5
- DOUBLE_TYPE - 3
- FLOAT_TYPE - 2
- INTEGER_TYPE - 0
- LONGTEXT_TYPE - 6
- OKTOCONVERT - 0
- POTENTIALDATALOSS - 1
- STRING_TYPE - 1
- UNKNOWN_TYPE - 999
- getType(double):int
- getType(float):int
- getType(int):int
- getType(Object):int
- isOKToConvert(int, int):int
- isOKToConvert(String, String):int
- toBoolean(boolean):Boolean

- toBoolean(Object):Boolean
- toDouble(double):Double
- toDouble(float):Double
- toDouble(int):Double
- toDouble(Object):Double
- toFloat(double):Float
- toFloat(float):Float
- toFloat(int):Float
- toFloat(Object):Float
- toInteger(double):Integer
- toInteger(float):Integer
- toInteger(int):Integer
- toInteger(Object):Integer
- toPrimitiveBoolean(Object):boolean
- toPrimitiveDouble(float):double
- toPrimitiveDouble(int):double
- toPrimitiveDouble(Object):double
- toPrimitiveFloat(double):float
- toPrimitiveFloat(int):float
- toPrimitiveFloat(Object):float
- toPrimitiveInt(double):int
- toPrimitiveInt(float):int
- toPrimitiveInt(Object):int
- toString(double):String
- toString(float):String
- toString(int):String
- toString(Object):String

DtpDate

Dtp/

CxCommon/

- DtpDate()
- DtpDate(long, boolean)
- DtpDate(String, String)
- DtpDate(String, String, String[], String[])
- addDays(int):DtpDate
- addMonths(int):DtpDate
- addWeekdays(int):DtpDate
- addYears(int):DtpDate
- after(DtpDate):boolean
- before(DtpDate):boolean

- calcDays(DtpDate):int
- calcWeekdays(DtpDate):int
- get12MonthNames():String[]
- get12ShortMonthNames():String[]
- get7DayNames():String[]
- getCWDate():String
- getDayOfMonth():String
- getDayOfWeek():String
- getHours():String
- getIntDay():int
- getIntDayOfWeek():int
- getIntHours():int
- getIntMilliseconds():int
- getIntMinutes():int
- getIntMonth():int
- getIntSeconds():int
- getIntYear():int
- getMaxDate(BusObjArray, String, String):DtpDate
- getMaxDateBO(BusObj[], String, String):BusObj[]
- getMaxDateBO(BusObjArray, String, String):BusObj[]
- getMinDate(BusObjArray, String, String):DtpDate
- getMinDateBO(BusObj[], String, String):BusObj[]
- getMinDateBO(BusObjArray, String, String):BusObj[]
- getMinutes():String
- getMonth():String
- getMSSince1970():long
- getNumericMonth():String
- getSeconds():String
- getShortMonth():String
- getYear():String
- set12MonthNames(String[], boolean)
- set12MonthNamesToDefault()
- set12ShortMonthNames(String[])
- set12ShortMonthNamesToDefault()
- set7DayNames(String[])
- set7DayNamesToDefault()
- toString():String
- toString(String):String
- toString(String, boolean):String

DtpMapService

Dtp/

CxCommon/

- runMap(String, String, BusObj[], CxExecutionContext):BusObj[]

DtpSplitString

Dtp/

CxCommon/

- DtpSplitString(String, String)
- elementAt(int):String
- firstElement():String
- getElementCount():int
- getEnumeration():Enumeration
- lastElement():String
- nextElement():String
- prevElement():String
- reset()

DtpUtils

Dtp/

CxCommon/

- padLeft(String, char, int):String
- padRight(String, char, int):String
- stringReplace(String, String, String):String
- truncate(double):int
- truncate(double, int):double
- truncate(float):int
- truncate(float, int):double
- truncate(Object):int
- truncate(Object, int):double

BusObjInvalidVerbException (extends InterchangeExceptions)

Exceptions/

CxCommon/

- getFormattedMessage()

IdentityRelationship

relationship/

utilities/

crossworlds/

com/

- addMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, Object, CxExecutionContext)

- foreignKeyLookup(String, String, BusObj, String, BusObj, String, CxExecutionContext)
- foreignKeyXref(String, String, String, BusObj, String, BusObj, String, CxExecutionContext)
- maintainChildVerb(String, String, String, BusObj, String, BusObj, String, CxExecutionContext, boolean, boolean)
- maintainCompositeRelationship(String, String, BusObj, Object, CxExecutionContext)
- maintainSimpleIdentityRelationship(String, String, BusObj, BusObj, CxExecutionContext)
- updateMyChildren(String, String, BusObj, String, String, String, String, CxExecutionContext)

MapExeContext

Dtp/

CxCommon/

- ACCESS_REQUEST - "SUBSCRIPTION_DELIVERY"
- ACCESS_RESPONSE - "ACCESS_RETURN_REQUEST"
- EVENT_DELIVERY - "SUBSCRIPTION_DELIVERY"
- SERVICE_CALL_FAILURE - "CONSUME_FAILED"
- SERVICE_CALL_REQUEST - "CONSUME"
- SERVICE_CALL_RESPONSE - "DELIVERBUSOBJ"
- getConnName():String
- getGenericBO():BusObj
- getInitiator():String
- getLocale():java.util.Locale
- getOriginalRequestBO():BusObj
- setConnName(String)
- setInitiator(String)
- setLocale(java.util.Locale)

Participant

RelationshipServices/

Server/

- Participant(String, String, int, BusObj)
- Participant(String, String, int, String)
- Participant(String, String, int, long)
- Participant(String, String, int, int)
- Participant(String, String, int, double)
- Participant(String, String, int, float)
- Participant(String, String, int, boolean)
- Participant(String, String, BusObj)
- Participant(String, String, String)

- Participant(String, String, long)
- Participant(String, String, int)
- Participant(String, String, double)
- Participant(String, String, float)
- Participant(String, String, boolean)
- getBoolean():boolean
- getBusObj():BusObj
- getDouble():double
- getFloat():float
- getInstanceId():int
- getInt():int
- getLong():long
- getParticipantDefinition():String
- getRelationshipDefinition():String
- getString():String INVALID_INSTANCE_ID
- set(boolean)
- set(BusObj)
- set(double)
- set(float)
- set(int)
- set(long)
- set(String)
- setInstanceId(int)
- setParticipantDefinition(String)
- setRelationshipDefinition(String)
- setParticipantDefinition(String)
- setRelationshipDefinition(String)

Relationship

RelationshipServices/

Server/

- addMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- addParticipant(Participant):int
- addParticipant(String, String, boolean):int
- addParticipant(String, String, BusObj):int
- addParticipant(String, String, double):int
- addParticipant(String, String, float):int
- addParticipant(String, String, int):int
- addParticipant(String, String, int, boolean):int
- addParticipant(String, String, int, BusObj):int
- addParticipant(String, String, int, double):int

- addParticipant(String, String, int, float):int
- addParticipant(String, String, int, int):int
- addParticipant(String, String, int, long):int
- addParticipant(String, String, int, String):int
- addParticipant(String, String, long):int
- addParticipant(String, String, String):int
- create(Participant):int
- create(String, String, boolean):int
- create(String, String, BusObj):int
- create(String, String, double):int
- create(String, String, float):int
- create(String, String, int):int
- create(String, String, long):int
- create(String, String, String):int
- deactivateParticipant(Participant)
- deactivateParticipant(String, String, boolean)
- deactivateParticipant(String, String, BusObj)
- deactivateParticipant(String, String, double)
- deactivateParticipant(String, String, float)
- deactivateParticipant(String, String, int)
- deactivateParticipant(String, String, long)
- deactivateParticipant(String, String, String)
- deactivateParticipantByInstance(String, String, int)
- deactivateParticipantByInstance(String, String, int, boolean)
- deactivateParticipantByInstance(String, String, int, BusObj)
- deactivateParticipantByInstance(String, String, int, double)
- deactivateParticipantByInstance(String, String, int, float)
- deactivateParticipantByInstance(String, String, int, int)
- deactivateParticipantByInstance(String, String, int, long)
- deactivateParticipantByInstance(String, String, int, String)
- deleteMyChildren(String, String, BusObj, String, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- deleteParticipant(Participant)
- deleteParticipant(String, String, boolean)
- deleteParticipant(String, String, BusObj)
- deleteParticipant(String, String, double)
- deleteParticipant(String, String, float)
- deleteParticipant(String, String, int)
- deleteParticipant(String, String, long)
- deleteParticipant(String, String, String)

- deleteParticipantByInstance(String, String, int)
- deleteParticipantByInstance(String, String, int, boolean)
- deleteParticipantByInstance(String, String, int, BusObj)
- deleteParticipantByInstance(String, String, int, double)
- deleteParticipantByInstance(String, String, int, float)
- deleteParticipantByInstance(String, String, int, int)
- deleteParticipantByInstance(String, String, int, long)
- deleteParticipantByInstance(String, String, int, String)
- getNewID(String):int
- maintainCompositeRelationship(String, String, BusObj, Object, CxExecutionContext)
- maintainSimpleIdentityRelationship(String, String, BusObj, BusObj, CxExecutionContext)
- retrieveInstances(String, boolean):int[]
- retrieveInstances(String, BusObj):int[]
- retrieveInstances(String, double):int[]
- retrieveInstances(String, float):int[]
- retrieveInstances(String, int):int[]
- retrieveInstances(String, long):int[]
- retrieveInstances(String, String):int[]
- retrieveInstances(String, String, boolean):int[]
- retrieveInstances(String, String, BusObj):int[]
- retrieveInstances(String, String, double):int[]
- retrieveInstances(String, String, float):int[]
- retrieveInstances(String, String, int):int[]
- retrieveInstances(String, String, long):int[]
- retrieveInstances(String, String, String):int[]
- retrieveInstances(String, String[], boolean):int[]
- retrieveInstances(String, String[], BusObj):int[]
- retrieveInstances(String, String[], double):int[]
- retrieveInstances(String, String[], float):int[]
- retrieveInstances(String, String[], int):int[]
- retrieveInstances(String, String[], long):int[]
- retrieveInstances(String, String[], String):int[]
- retrieveParticipants(String):Participant[]
- retrieveParticipants(String, String):Participant[]
- retrieveParticipants(String, String[]):Participant[]
- retrieveParticipants(String, int):Participant[]
- retrieveParticipants(String, String, int):Participant[]
- retrieveParticipants(String, String[], int):Participant[]

- updateMyChildren(String, String, BusObj, String, String, String, String, CxExecutionContext)
- updateParticipant(String, String, BusObj)
- updateParticipantByInstance(Participant)
- updateParticipantByInstance(String, String, int)
- updateParticipantByInstance(String, String, int, BusObj)

UserStoredProcedureParam

Dtp/

CxCommon/

- UserStoredProcedureParam(int, String, Object, String, String)
- getParamDataTypeJavaObj():String
- getParamDataTypeJDBC():int
- getParamIndex():int
- getParamIOType():String
- getParamName():String
- getParamValue():Object
- setParamDataTypeJavaObj(String)
- setParamDataTypeJDBC(int)
- setParamIndex(int)
- setParamIOType(String)
- setParamName(String)
- setParamValue(Object)
- PARAM_TYPE_IN - "IN"
- PARAM_TYPE_OUT - "OUT"
- PARAM_TYPE_INOUT - "INOUT"
- DATA_TYPE_STRING - "String"
- DATA_TYPE_INTEGER - "Integer"
- DATA_TYPE_DOUBLE - "Double"
- DATA_TYPE_FLOAT - "Float"
- DATA_TYPE_BOOLEAN - "Boolean"
- DATA_TYPE_TIME - "java.sql.Time"
- DATA_TYPE_DATE - "java.sql.Date"
- DATA_TYPE_TIMESTAMP - "java.sql.Timestamp"
- DATA_TYPE_BIG_DECIMAL - "java.math.BigDecimal"
- DATA_TYPE_LONG_INTEGER - "Long"
- DATA_TYPE_BINARY - "byte[]"
- DATA_TYPE_CLOB - "Clob"
- DATA_TYPE_BLOB - "Blob"
- DATA_TYPE_ARRAY - "Array"
- DATA_TYPE_STRUCT - "Struct"

- DATA_TYPE_REF - "Ref"

BaseCollaboration

Collaboration/

- BaseCollaboration(com.ibm.bpe.api.ProcessInstanceData)
- AnyException - "AnyException"
- AppBusObjDoesNotExist - "BusObjDoesNotExist"
- AppLogOnFailure - "AppLogOnFailure"
- AppMultipleHits - "AppMultipleHits"
- AppRequestNotYetSent - "AppRequestNotYetSent"
- AppRetrieveByContentFailed - "AppRetrieveByContent"
- AppTimeOut - "AppTimeOut"
- AppUnknown - "AppUnknown"
- AttributeException - "AttributeException"
- existsConfigProperty(String):boolean
- getConfigProperty(String):String
- getConfigPropertyArray(String):String[]
- getCurrentLoopIndex():int
- getDBConnection(String):CwDBConnection
- getDBConnection(String, boolean):CwDBConnection getLocale():java.util.Locale
- getMessage(int):String
- getMessage(int, Object[]):String
- getName():String
- implicitDBTransactionBracketing():boolean
- isCallerInRole(String):boolean
- isTraceEnabled(int):boolean
- JavaException - "JavaException"
- logError(int)
- logError(int, Object[])
- logError(int, String)
- logError(int, String, String)
- logError(int, String, String, String)
- logError(int, String, String, String, String)
- logError(int, String, String, String, String, String)
- logError(String)
- logInfo(int)
- logInfo(int, Object[])
- logInfo(int, String)
- logInfo(int, String, String)
- logInfo(int, String, String, String)

- logInfo(int, String, String, String, String)
- logInfo(int, String, String, String, String, String)
- logInfo(String)
- logWarning(int)
- logWarning(int, Object[])
- logWarning(int, String)
- logWarning(int, String, String)
- logWarning(int, String, String, String)
- logWarning(int, String, String, String, String)
- logWarning(int, String, String, String, String, String)
- logWarning(String)
- not(boolean):boolean ObjectException - "ObjectException"
- OperationException - "OperationException"
- raiseException(CollaborationException)
- raiseException(String, int)
- raiseException(String, int, Object[])
- raiseException(String, int, String)
- raiseException(String, int, String, String)
- raiseException(String, int, String, String, String)
- raiseException(String, int, String, String, String, String)
- raiseException(String, int, String, String, String, String, String)
- raiseException(String, String)
- ServiceCallException - "ConsumerException"
- ServiceCallTransportException - "ServiceCallTransportException"
- SystemException - "SystemException"
- trace(int, int)
- trace(int, int, Object[])
- trace(int, int, String)
- trace(int, int, String, String)
- trace(int, int, String, String, String)
- trace(int, int, String, String, String, String)
- trace(int, int, String, String, String, String, String)
- trace(int, String)
- trace(String)
- TransactionException - "TransactionException"

CxExecutionContext

CxCommon/

- CxExecutionContext()
- getContext(String):Object

- MAPCONTEXT - "MAPCONTEXT"
- setContext(String, Object)

CollaborationException

Collaboration/

- getMessage():String
- getMsgNumber():int
- getSubType():String
- getText():String
- getType():String
- toString():String

Filter

crossworlds/

com/

- Filter(BaseCollaboration)
- filterExcludes(String, String):boolean
- filterIncludes(String, String):boolean
- recurseFilter(BusObj, String, boolean, String, String):boolean
- recursePreReqs(String, Vector):int

Globals

crossworlds/

com/

- Globals(BaseCollaboration)
- callMap(String, BusObj):BusObj

SmartCollabService

crossworlds/

com/

- SmartCollabService()
- SmartCollabService(BaseCollaboration)
- doAgg(BusObj, String, String, String):BusObj
- doMergeHash(Vector, String, String):Vector
- doRecursiveAgg(BusObj, String, String, String):BusObj
- doRecursiveSplit(BusObj, String):Vector
- doRecursiveSplit(BusObj, String, boolean):Vector
- getKeyValues(BusObj, String):String
- merge(Vector, String):BusObj
- merge(Vector, String, BusObj):BusObj
- split(BusObj, String):Vector

StateManagement

**crossworlds/
com/**

- StateManagement()
- beginTransaction()
- commit()
- deleteBO(String, String, String)
- deleteState(String, String, String, int)
- persistBO(String, String, String, String, BusObj)
- recoverBO(String, String, String):BusObj
- releaseDBConnection()
- resetData()
- retrieveState(String, String, String, int):int
- saveState(String, String, String, String, int, int, double)
- setDBConnection(CwDBConnection)
- updateBO(String, String, String, String, BusObj)
- updateState(String, String, String, String, int, int)

EventKeyAttrDef

EventManagement/

CxCommon/

- EventKeyAttrDef()
- EventKeyAttrDef(String, String)
- public String keyName
- public String keyValue

EventQueryDef

EventManagement/

CxCommon/

- EventQueryDef()
- EventQueryDef(String, String, String, String, int)
- public String nameConnector
- public String nameCollaboration
- public String nameBusObj
- public String verb
- public int ownerType

FailedEventInfo

EventManagement/

CxCommon/

- FailedEventInfo()
- FailedEventInfo(String x6, int, EventKeyAttrDef[], int, int, String, String, int)
- public String nameOwner

- public String nameConnector
- public String nameBusObj
- public String nameVerb
- public String strTime
- public String strMessage
- public int wipIndex
- public EventKeyAttrDef[] strbusObjKeys
- public int nKeys
- public int eventStatus
- public String expirationTime
- public String scenarioName
- public int scenarioState

関連タスク

201 ページの『サポートされる WebSphere InterChange Server API のロギングおよびトレースの使用可能化』

サポートされる WebSphere InterChange Server API のロギングおよびトレースを、管理コンソールを通じて使用可能にします。

関連情報

WebSphere InterChange Server v4.3 の資料

WebSphereInterChange Server からマイグレーションする場合の制限事項

WebSphere InterChange Server の特性の中には、WebSphere Process Server によって正確に再現されないものがあります。そのため、WebSphere InterChange server と同じように実行するために、マイグレーション後にアプリケーションを変更しなければならない場合があります。

次のセクションでは、これらの制限事項および可能な解決策を説明します。

トランザクション・レベル

WebSphere InterChange Server のコラボレーションと WebSphere Process Server BPEL ファイルの間のトランザクション・レベルには、直接の対応関係はありません。そのため、WebSphere InterChange Server コラボレーションで指定されたトランザクション・レベルは無視され、デフォルトの BPEL トランザクション・レベルがマイグレーション後のアプリケーションで使用されます。希望する機能を得るためには、BPEL トランザクションについて理解し、マイグレーションしたアプリケーションをそれに合わせて調整する必要があります。

注: 処理中のトランザクションはマイグレーションされません。すべてのトランザクションを完了してから、マイグレーションを開始してください。

補正

WebSphere Process Server の補正は、WebSphere InterChange Server の補正と異なります。WebSphere Process Server が提供する新しいタイプの補正を評価し、アプリ

ケーションに最も適したタイプを選択する必要があります。

WebSphere Process Server で WebSphere InterChange Server API を使用する場合にはサポートされていないイベント要約と変更内容の要約

問題: マイグレーションされた WebSphere InterChange Server アプリケーションのイベント要約と変更内容の要約に、予期される情報が含まれていません。**原因:** WebSphere InterChange Server のビジネス・オブジェクト (BusObjs) は、変更内容の要約とイベント要約をサポートしません。WebSphere Process Server でサポートされる WebSphere InterChange Server の API は WebSphere InterChange Server タイプの BusObj と連動するため、それらの API を使用すると BusObj への変換が強制的に行われます。この操作が行われると、BusObj に変換された WebSphere Process Server DataObject に含まれるすべてのイベント要約と変更内容の要約の情報が失われます。WebSphere InterChange Server からのマイグレーションによって生成されたアプリケーションは、WebSphere Process Server で WebSphere InterChange Server の API を使用するため、コードを手動で更新して WebSphere InterChange Server の API の使用を停止するまでは、それらのアプリケーションでイベント要約と変更内容の要約を使用することはできません。**解決策:** WebSphere InterChange Server の API を一切使用しないようにするか、それらを WebSphere Process Server の API に変更します。

関連概念

『WebSphere InterChange Server からのマイグレーションのトラブルシューティング』

マイグレーションで発生する問題の解決策と、ロギングとトレースをオンにする方法について説明します。

関連資料

158 ページの『事後マイグレーションの考慮事項』

アプリケーションが WebSphere InterChange Server から WebSphere Process Server にマイグレーションされた場合は、WebSphere Process Server と WebSphere InterChange Server のアーキテクチャーの間には違いがあるため、マイグレーションされたアプリケーションが WebSphere Process Server において意図したとおりに機能するように、一部の領域に特別の注意を払う必要があります。

関連情報

 ビジネス・プロセスの補正

WebSphere InterChange Server からのマイグレーションのトラブルシューティング

マイグレーションで発生する問題の解決策と、ロギングとトレースをオンにする方法について説明します。

関連概念

199 ページの『WebSphereInterChange Server からマイグレーションする場合の制限事項』

WebSphere InterChange Server の特性の中には、WebSphere Process Server によって正確に再現されないものがあります。そのため、WebSphere InterChange

server と同じように実行するために、マイグレーション後にアプリケーションを変更しなければならない場合があります。

関連資料

158 ページの『事後マイグレーションの考慮事項』

アプリケーションが WebSphere InterChange Server から WebSphere Process Server にマイグレーションされた場合は、WebSphere Process Server と WebSphere InterChange Server のアーキテクチャーの間には違いがあるため、マイグレーションされたアプリケーションが WebSphere Process Server において意図したとおりに機能するように、一部の領域に特別の注意を払う必要があります。

146 ページの『事前マイグレーションの考慮事項』

WebSphere InterChange Server 成果物を WebSphere Process Server にマイグレーションする作業を容易にするために、WebSphere InterChange Server 用の統合成果物を開発するための以下のガイドラインを検討してください。

サポートされる WebSphere InterChange Server API のロギングおよびトレースの使用可能化

サポートされる WebSphere InterChange Server API のロギングおよびトレースを、管理コンソールを通じて使用可能にします。

このタスクについて

マイグレーション済みアプリケーションに、サポートされる WebSphere InterChange Server API が含まれている場合、トラブルシューティング用にそれらのロギングおよびトレースを使用可能にすることができます。

プロシージャ

1. 管理コンソールを起動します。
2. 左側の (ナビゲーション) パネルから、「トラブルシューティング」>「ログおよびトレース」を選択します。
3. 右側のパネルで、ロギングおよびトレースを使用可能にしたいサーバーの名前を選択します。
4. 右側のパネルの「一般プロパティ」の下で、「ログ・レベル詳細の変更 (Change Log Level Details)」を選択します。
5. 「ランタイム」タブを選択します。(「ランタイム」タブを選択すると、リアルタイムにこの変更を行うことができ、サーバーを再始動する必要はありません。)
6. パッケージの名前の後に =a11 を付加したものを、画面のボックス内のログに記録されるパッケージのリストに追加します。この新規の項目は、コロンを使用して既存の項目と分離します。例えば、CxCommon=a11 と指定できます。この場合、CxCommon は、一連のサポートされる WebSphere InterChange Server API のパッケージの名前です。a11 を指定すると、すべてのロギングおよびトレースが使用可能になります。API およびそのパッケージ名のリストについては、『WebSphere InterChange Server API』のサポートを参照してください。
7. 「適用」を選択します。

8. サーバーの再始動後にもこの構成を保持するには、「ランタイム変更も構成に保管する」チェック・ボックスを選択します。
9. 「OK」を選択します。
10. 次の画面が表示されたら、「保管」を選択して変更内容を保存します。

関連資料



『WebSphere InterChange Server API』のサポート

WebSphere Process Server および WebSphere Integration Developer で提供される WebSphere InterChange Server のソース成果物のマイグレーション・ツールに加えて、WebSphere Process Server は、WebSphere InterChange Server で提供されていた API の多くもサポートします。マイグレーション・ツールは、マイグレーション時に極力カスタム断片コードを保持することにより、これらの WebSphere InterChange Server API と連動して動作します。

マイグレーションされた BPEL ファイルで直列化可能ではないオブジェクトを直列化しようとしたときの失敗

マイグレーションで生成された BPEL ファイルで直列化が失敗する場合、失敗しないように BPEL ファイルを変更できる場合があります。

問題: マイグレーションによって生成される Business Process Execution Language (BPEL) ファイルのカスタム断片ノードで、直列化可能ではないオブジェクトを直列化しようとするために、直列化が失敗します。

原因: WebSphere InterChange Server では、コラボレーション・テンプレートが単一の Java クラスにコンパイルされます。WebSphere Process Server では、BPEL ファイル内の各ノードが別々の Java クラスにコンパイルされます。WebSphere InterChange Server では、変数を一度だけ宣言し、コラボレーション・テンプレートのさまざまな手順全体で共用することができます。マイグレーションされた BPEL ファイルでその振る舞いをシミュレートするには、コード断片で使用される各変数を断片の先頭で取得し、断片の終わりで保存する必要があります。WebSphere InterChange Server ポート定義で定義される変数は、BPEL 変数になります。これらの変数は、各断片の始まりに BusObj 変数に取得され (断片内で参照されている場合)、各断片の終わりに BPEL 変数に再び保存されます。例えば、断片の先頭での取得コードは、次のようになります。

```
BusObj tempBusObj = null;if (tempBusObj_var != null) { tempBusObj =  
    new BusObj(tempBusObj_var); };
```

また、断片の終わりでの保管コードは次のようになります。

```
if (tempBusObj == null) { tempBusObj_var = null; } else { tempBusObj_var =  
    tempBusObj.getBusinessGraph(); }
```

WebSphere InterChange Server の断片コードで使用される他の変数は直列化され、*CollabTemplateName_var* という名前で BPEL 変数のストリングとして保管されます。これらの変数は、各 BPEL 断片の先頭で非直列化され、参照元の各 BPEL 断片の終わりに直列化および保存されます。例えば、オブジェクトは次のようにして取得されます。

```
BusObj tempBusObj = (BusObj)BaseCollaboration.deserialize  
    (FrontEndCollab_var.getString("tempBusObj"));
```

また、オブジェクトは次のようにして保存されます。

```
FrontEndCollab_var.setString("tempBusObj", BaseCollaboration.serialize(tempBusObj));
```

直列化されるオブジェクト・タイプが直列化可能ではない場合は、BPEL を実行するときに直列化および非直列化の使用に失敗します。

解決策: マイグレーション後に BPEL ファイルを次のように変更します。

- Java で直列化可能ではない変数については、BPEL 断片を更新して直列化ステートメントと非直列化ステートメントを除去します。断片間で変数を共有する (各断片で再作成されるのではない) 必要がある場合は、別の方法を使用して、断片全体で変数の値を維持する必要があります。
- WebSphere InterChange Server のポート定義で宣言されていないのにパートナー呼び出しで使用されている BusObj タイプの変数に、BPEL 変数を手動で定義します。これが手動手順になる理由は、WebSphere Process Server で呼び出し時に使用される変数は強く型付けされている必要があるのに、マイグレーション・ツールでは WebSphere InterChange Server の断片からその型を正確に判別できないためです。

注: マイグレーション・ツールで使用される命名規則では、BPEL 変数に命名するときに、断片コードの変数の名前に `_var` を追加します。例えば、断片コードで `tempBusObj` と呼ばれる変数の場合、マイグレーション・ツールは、`tempBusObj_var` という名前の BPEL 変数を作成します。

- BPEL 変数として手動で宣言する必要がある変数の場合は、「BPEL 変数から取得/BPEL 変数へ保管」変数保持方式ではなく「非直列化/直列化」変数保持方式を使用するように、BPEL 断片コードを変更する必要があります。

WebSphere Studio Application Developer Integration Edition からのソース成果物のマイグレーション

WebSphere Studio Application Developer Integration Edition からソース成果物をマイグレーションするには、WebSphere Integration Developer で提供されるツールを使用します。

このタスクについて

WebSphere Integration Developer で使用可能なマイグレーション・ウィザードまたはコマンド行を使用して、WebSphere Application Server Developer Integration Edition サービス・プロジェクトをアクティブな WebSphere Integration Developer ワークスペース内のプロジェクトにマイグレーションします。詳しくは、WebSphere Integration Developer インフォメーション・センターを参照してください。

関連情報

WebSphere Integration Developer インフォメーション・センター

WebSphere MQ Workflow からのマイグレーション

WebSphere MQ Workflow からマイグレーションするには、WebSphere Integration Developer マイグレーション・ウィザードか、または WebSphere MQ Workflow 3.6 から WebSphere Process Server にマイグレーションするための特殊ユーティリティーを使用します。

このタスクについて

このバージョンの WebSphere MQ Workflow の場合...	実行内容
WebSphere MQ Workflow 3.6	WebSphere Integration Developer のマイグレーション・ウィザードまたは FDL2BPEL ユーティリティーを使用して、すべての WebSphere MQ Workflow の成果物を WebSphere Integration Developer の配置可能な成果物にマイグレーションします。
WebSphere MQ Workflow 3.5 以前	最初に WebSphere MQ Workflow バージョン 3.6 にマイグレーションする必要があります。

詳しくは、WebSphere Integration Developer インフォメーション・センターを参照してください。

関連情報

WebSphere Integration Developer インフォメーション・センター

第 3 章 使用すべきでないフィーチャー

このセクションでは、WebSphere Process Server バージョン 6.0、バージョン 6.1、および WebSphere Business Integration Server Foundation バージョン 5.1 などの製品の中で、使用すべきでないフィーチャーについてまとめています。他の WebSphere Application Server バージョン 5.1 および 6.x 製品の使用すべきでないフィーチャーについては、それらの製品の資料で説明されています。

廃止リスト

ここでは、以下のバージョンおよびリリースで使用すべきでないフィーチャーについて説明します。

- 206 ページの『WebSphere Process Server バージョン 6.1 で使用すべきでないフィーチャー』
- 210 ページの『WebSphere Process Server バージョン 6.0.2 で使用すべきでないフィーチャー』
- 212 ページの『WebSphere Process Server バージョン 6.0.1 で使用すべきでないフィーチャー』
- 212 ページの『WebSphere Process Server バージョン 6.0 で使用すべきでないフィーチャー』
- 216 ページの『WebSphere Business Integration Server Foundation バージョン 5.1.1 で使用すべきでないフィーチャー』
- 216 ページの『WebSphere Business Integration Server Foundation バージョン 5.1 で使用すべきでないフィーチャー』

以下のテーブルに、使用すべきでないものをバージョンおよびリリースごとにまとめました。各テーブルに、非推奨の影響のあるバージョンとリリース、および使用すべきでないもの（フィーチャー、API、スクリプト・インターフェース、ツール、ウィザード、公開された構成データ、命名 ID、定数など）を示しています。可能なところでは、推奨マイグレーション・アクションが提供されています。

WebSphere Process Server バージョン 6.1 で使用すべきでないフィーチャー

Container Manager Persistence over Anything (CMP/A)
<p>WebSphere Process Server に組み込まれている CMP/A サポートは推奨されません。これには、CMP/A、cmpdeploy.bat/sh コマンド行ツール、および以下のパブリック API を使用するためにカスタマイズされたアプリケーションのランタイム・サポートが含まれます。</p> <ul style="list-style-type: none">com.ibm.websphere.rsadapter.WSProceduralPushDownHelpercom.ibm.websphere.rsadapter.WSPushDownHelpercom.ibm.websphere.rsadapter.WSPushDownHelperFactorycom.ibm.websphere.rsadapter.WSRelationalPushDownHelper <p><u>推奨されるマイグレーション・アクション</u></p> <p>リレーショナル・データ・ソースを使用するように CMP エンティティ Bean を変換するか、または CMP エンティティ Bean を、サポートされた別のデータ・パーシスタンス・モデルに置き換えます。</p> <p>また、WebSphere Adapters を使用して、既存の CMP/A アプリケーションを置き換えることもできます。Adapter ツールは、サービス・インターフェースの作成に、「作成、取得、更新、および削除」というアーキテクチャーを使用しており、CMP/A が使用するアーキテクチャーと非常に似ています。</p>
JACL スクリプト (WebSphere Application Server バージョン 6.1 では非推奨)
<p>WebSphere Application Server における JACL スクリプトの非推奨と一貫性を保つため、WebSphere Process Server における JACL スクリプト・ファイルは非推奨です。</p> <p><u>推奨されるマイグレーション・アクション</u></p> <p>対応する .bat/sh ファイル、または wsadmin コマンドを使用して、同じ機能を実行してください。 注: 以下の Business Process Choreographer JACL スクリプトは非推奨ではありません。</p> <ol style="list-style-type: none"><install_root>%ProcessChoreographer%admin%bpcTemplates.jacl<install_root>%ProcessChoreographer%config%bpeconfig.jacl<install_root>%ProcessChoreographer%config%bpeunconfig.jacl<install_root>%ProcessChoreographer%config%bpeupgrade.jacl<install_root>%ProcessChoreographer%config%clientconfig.jacl
IBM Web Services Client for C++
<p>IBM Web Services Client for C++ は、独自のインストーラーを備えたスタンドアロン・アプリケーションですが、WebSphere Process Server メディアで配布されます。WebSphere Process Server は、このソフトウェアを使用せず、また依存関係もありますが、同様に本製品と共に配布される IBM Message Service Client for C/C++ は、このソフトウェアを使用し、依存関係があります。</p> <p><u>推奨されるマイグレーション・アクション</u></p> <p>GPL ライセンスの下で配布されているオープン・ソース製品である gSOAP (http://www.cs.fsu.edu/~engelen/soap.html) など、同じ機能を提供する、無償で入手可能なその他のツールのいずれかを使用してください。</p>
Business Process Choreographer

Container Manager Persistence over Anything (CMP/A)

汎用ビジネス・プロセス EJB API

- ProcessTemplateData の getAutoDelete() 関数は推奨されません。

推奨されるマイグレーション・アクション

対応するプロセス・テンプレートに対してどのように自動削除が処理されるかを照会するには、getAutoDeletionMode() メソッドを使用してください。

- 例外 SpecificFaultReplyException は推奨されません。

推奨されるマイグレーション・アクション

アクションは不要です。この例外は WSIF メッセージの処理にのみ必要で、この処理は現在サポートされていません。

汎用ビジネス・プロセス Webservice API - XML スキーマ・タイプ

複合タイプ ProcessTemplateType のエレメント autoDelete は推奨されません。

```
<xsd:element name="ProcessTemplate" type="tns:ProcessTemplateType"/>
<xsd:complexType name="ProcessTemplateType">
  <xsd:sequence>
    ...
    <xsd:element name="autoDelete" type="xsd:boolean" minOccurs="0"/>
  ...</xsd:sequence></xsd:complexType>
```

推奨されるマイグレーション・アクション

タイプ ProcessTemplateType のエレメント autoDeletionMode を使用してください。

```
<xsd:element name="ProcessTemplate" type="tns:ProcessTemplateType"/>
<xsd:complexType name="ProcessTemplateType">
  <xsd:sequence>
    ...
    <xsd:element name="autoDeletionMode" type="xsd:string" minOccurs="0"/>
  ...</xsd:sequence></xsd:complexType>
```

非推奨の ProcessContainer MBean の Observer DB Cleanup メソッド

以下のメソッドは推奨されません。

- public String observerForceRemoveInstanceData(String dataSourceName, String state, String templateName, String validFrom, String completedBefore)
- public String observerRemoveDeletedInstancesData(String dataSourceName, String completedBefore)
- public String observerRemoveInstanceDataOfTemplate(String dataSourceName, String templateName, String validFrom)

推奨されるマイグレーション・アクション

以下の新規メソッド (名前は同じで、パラメーター「dbSchemaName」が追加されている) を使用してください。

- public String observerForceRemoveInstanceData(String dataSourceName, String dbSchemaName, String state, String templateName, String validFrom, String completedBefore)
- public String observerRemoveDeletedInstancesData(String dataSourceName, String dbSchemaName, String completedBefore)
- public String observerRemoveInstanceDataOfTemplate(String dataSourceName, String dbSchemaName, String templateName, String validFrom)

Container Manager Persistence over Anything (CMP/A)

LDAP スタッフ解決プラグイン

LDAP スタッフ解決プラグインのスタッフ照会に関する属性評価仕様は、推奨されません。

```
<ldap:attribute name="attribute name"
                objectclass="LDAP object class"
                usage="simple">
</ldap:attribute>
```

推奨されるマイグレーション・アクション

LDAP オブジェクトごとに複数の属性をサポートする、結果オブジェクト評価仕様を使用してください。「user」照会の属性「objectclass」および「attribute」は、ユーザーごとの複数の結果属性をサポートする完全な結果オブジェクト評価仕様に置き換えられます。

汎用ヒューマン・タスク・マネージャー EJB API

- インターフェース Task の以下のフィールドは推奨されません。
 - STATE_FAILING
 - STATE_SKIPPED
 - STATE_STOPPED
 - STATE_TERMINATING
 - STATE_WAITING
 - STATE_PROCESSING_UNDO

推奨されるマイグレーション・アクション

インライン・ヒューマン・タスクのために、インライン・ヒューマン・タスクに関連したスタッフ・アクティビティの取得を使用し、汎用ビジネス・プロセス EJB API 内の ActivityInstanceData インターフェースで getExecutionState() メソッドを使用して、アクティビティ状態を確認します。

- インターフェース Task のフィールド KIND_WPC_STAFF_ACTIVITY は推奨されません。

推奨されるマイグレーション・アクション

Task インターフェースで isInline() メソッドを使用し、ビジネス・プロセス内でヒューマン・タスクがヒューマン・タスク (スタッフ) アクティビティに関連付けられているかどうかを判別します。

Container Manager Persistence over Anything (CMP/A)

非推奨の E メール担当者割り当て基準

エスカレーション・アクション「e-mail」を含むエスカレーションに使用される、E メール受信者の担当者割り当て基準 (スタッフ動詞) は推奨されません。バージョン 6.1 では必要なくなったためです。これは、以下の担当者割り当て基準に適用されます。

- 部門メンバーの E メール・アドレス
- グループ・メンバーの E メール・アドレス
- フィルターされたユーザーを除くグループ・メンバーの E メール・アドレス
- グループ検索の E メール・アドレス
- ロール・メンバーの E メール・アドレス
- ユーザーの E メール・アドレス
- ユーザー ID ごとのユーザーの E メール・アドレス

推奨されるマイグレーション・アクション

E メール・アドレスおよび設定済みの言語は、バージョン 6.1 の担当者割り当て基準の標準セットによって、ユーザー ID と共に解決されます。この非推奨情報は、カスタム XSLT 担当者割り当て基準のマッピング (スタッフ動詞) ファイルを作成するユーザーにとって、特に重要です。バージョン 6.0.2 タスク定義をデプロイしない場合、推奨されない担当者割り当て基準をサポートする必要はありません。バージョン 6.1 の場合は、担当者割り当て基準、「User Records by user ID」が導入されており、カスタム XSLT ファイルによるサポートが必要です。これは、E メール・アドレスをフォールバックとして解決するためです。

WebSphere Integration Developer 6.1 で、ソース成果物のマイグレーションを開始することで、既存のヒューマン・タスク定義内の推奨されない E メール担当者割り当て基準を除去できます。これを行うには、ご使用のバージョン 6.0.2 タスク定義を WebSphere Integration Developer 6.1 にインポートし、少し変更して (タスク記述にブランクを追加して再度削除するなど)、再度保管します。

BPC 内部メッセージング用の JMS プロバイダーとしての MQ の非推奨事項 (ビジネス・プロセス・コンテナおよびヒューマン・タスク・コンテナの構成)

MQSeries® を JMS プロバイダーとして使用するようビジネス・プロセス・コンテナおよびヒューマン・タスク・コンテナを構成することは、推奨されません。ビジネス・プロセス・コンテナおよびヒューマン・タスク・コンテナは、内部メッセージング (特に長時間稼働するプロセス・インスタンスのナビゲート) に JMS を使用します。

推奨されるマイグレーション・アクション

ビジネス・プロセス・コンテナおよびヒューマン・タスク・コンテナの構成中に、デフォルトの JMS メッセージング・プロバイダーを使用します。

ビジネス・オブジェクト

以下のビジネス・オブジェクト・メソッドは推奨されません。

- `com.ibm.websphere.bo.BOFactory.createClass(java.lang.Class interfaceClass);`
- `com.ibm.websphere.bo.BOType.getTypeByClass(java.lang.Class className);`

推奨されるマイグレーション・アクション

これらのメソッドがバージョン 6.1 で呼び出された場合、「機能はサポートされません」例外が発生します。

Common Event Infrastructure

Container Manager Persistence over Anything (CMP/A)
<p>ユーザー表示 Common Base Event の作成および編集は推奨されません。</p> <p>推奨されるマイグレーション・アクション</p> <p>現在はツールを使用して、モニター対象の発行イベントに含めるビジネス・オブジェクト・データを指定できます。</p>
zOS
<p>esb/messageLogger/qualifier で String オブジェクトを JNDI にバインドする要件は廃止予定です。</p> <p>推奨されるマイグレーション・アクション</p> <p>メッセージ・ロガー・プリミティブは、CommonDB データベースにメッセージ情報を保管するようになります。必要に応じて、プロファイル拡張フェーズ中に、ESB_MESSAGE_LOGGER_QUALIFIER という名前の WebSphere 変数が作成され、その値が、選択された CommonDB スキーマ修飾子の変数に設定されます。</p>
WebSphere InterChange Server
<p>『WebSphere InterChange Server API』のサポート にリストされている API (アプリケーション・プログラミング・インターフェース) は現在は非推奨ではなくなりました。</p> <p>注: これらの API は、以前に WebSphere Process Server バージョン 6.0.2 で非推奨でした。</p> <p>推奨されるマイグレーション・アクション</p> <p>これらの API は、マイグレーション済みの WebSphere InterChange Server コンポーネントを使用したアプリケーションにのみ使用する必要があります。その他のすべての場合には、WebSphere Process Server 用のサービス・データ・オブジェクトを使用する必要があります。</p>
WebSphere Enterprise Service Bus (WESB)
<p>WESB がセキュアな WSRR インスタンスと通信する際に使用される SSL レポートリーを識別する現在のメソッドは、推奨されていません。</p> <p>推奨されるマイグレーション・アクション</p> <p>新規プロパティが WSRR 定義に追加されており、同様のレポートリーの指定が可能です。</p>

WebSphere Process Server バージョン 6.0.2 で使用すべきでないフィーチャー

Human Task Manager
<p>タスク・コンテキスト変数 %htm:task.clientDetailURL% が不要になりました。このため非推奨になりました。</p> <p>推奨されるマイグレーション・アクション</p> <p>アクションは不要です。</p>
<p>TEL でのすべてのエスカレーション E メールに使用される標準の E メール実装が推奨されなくなり、これに代わって TEL での E メール定義用の固有のサポートが提供されています。</p> <p>推奨されるマイグレーション・アクション</p> <p>エスカレーションについては、カスタマイズ可能な E メール・フィーチャーを使用してください。</p>

バージョン 6.0 では非推奨であった以下のタスク・オブジェクト・メソッドが、非推奨ではなくなりました。

`getInputMessageTypeName()`

`getOutputMessageTypeName()`

推奨されるマイグレーション・アクション

これらのメソッドが使用できるようになりました。

Business Process Choreographer

Generic Business Process EJB API インターフェース `ActivityInstanceData`、`ProcessInstanceData`、および `ProcessTemplateData` において、メソッド `getProcessAdministrators()` は推奨されません。

推奨されるマイグレーション・アクション

これらに対応する以下のメソッドを使用してください。

- `HumanTaskManagerService` インターフェースの `getUsersInRole()` メソッドと組み合わせて使用する `getProcessAdminTaskID()`。以下に例を示します。

```
htm.getUsersInRole(actInstData.getProcessAdminTaskID(), WorkItem.REASON_ADMINISTRATOR)
```

- `HumanTaskManagerService` インターフェースの `getUsersInRole()` メソッドと組み合わせて使用する `getAdminTaskID()`。以下に例を示します。

```
htm.getUsersInRole(procInstData.getAdminTaskID(), WorkItem.REASON_ADMINISTRATOR)
```

- `HumanTaskManagerService` インターフェースの `getUsersInRole()` メソッドと組み合わせて使用する `getAdminTaskTemplateID()`。以下に例を示します。

```
htm.getUsersInRole(procTemplData.getAdminTaskTemplateID(), WorkItem.REASON_ADMINISTRATOR )
```

Generic Business Process EJB API の `BusinessFlowManagerService` インターフェースおよび Generic Task EJB API の `HumanTaskManagerService` インターフェースでは、以下のメソッドは推奨されません。

- `query(String storedQueryName, Integer skipTuples)`
- `query(String storedQueryName, Integer skipTuples, Integer threshold)`

推奨されるマイグレーション・アクション

これらに対応する以下のメソッドを使用してください。

- `query(String storedQueryName, Integer skipTuples, List parameters)`
- `query(String storedQueryName,Integer skipTuples, Integer threshold, List parameters)`

以下の JACL スクリプトは推奨されません。

- `deleteAuditLog.jacl`
- `deleteInvalidProcessTemplate.jacl`
- `deleteInvalidTaskTemplate.jacl`
- `queryNumberOfFailedMessages.jacl`
- `replayFailedMessages.jacl`
- `cleanupUnusedStaffQueryInstances.jacl`
- `refreshStaffQuery.jacl`

推奨されるマイグレーション・アクション

推奨されない各 JACL スクリプトについては、対応する Jython スクリプトが新しく提供されています。この Jython スクリプト (*.py) (<install_root>/ProcessChoreographer/admin ディレクトリー内にあります) を使用してください。

SCA 管理コマンド

以下のコマンド (wsadmin を介して使用される) は推奨されません。

- configSCAForServer
- configSCAForCluster

推奨されるマイグレーション・アクション

configSCAForServer の代わりに、同等の機能を持つ以下の 2 つのコマンドを使用してください。 :

- configSCAAsyncForServer
- [オプション; 必要な場合のみ使用] configSCAJMSForServer

configSCAForCluster の代わりに、同等の機能を持つ以下の 2 つのコマンドを使用してください。

- configSCAAsyncForCluster
- [オプション; 必要な場合のみ使用] configSCAJMSForCluster

WebSphere InterChange Server

注: これらの API は、バージョン 6.1 では非推奨ではありません。

『WebSphere InterChange Server API』のサポートにリストされている API (アプリケーション・プログラミング・インターフェース) は推奨されません。

推奨されるマイグレーション・アクション

WebSphere Process Server 用に作成されたコードでは、これらのインターフェースを使用すべきではありません。

IBM WebSphere InterChange Server Access for Enterprise JavaBeans (EJB) のサポートは推奨されません。

推奨されるマイグレーション・アクション

WebSphere Process Server 用に開発されたアプリケーションでは、Access for Enterprise JavaBeans を使用すべきではありません。

WebSphere Process Server バージョン 6.0.1 で使用すべきでないフィーチャー

WebSphere Process Server バージョン 6.0.1 には、使用すべきでないフィーチャーはありません。

WebSphere Process Server バージョン 6.0 で使用すべきでないフィーチャー

アプリケーション・プログラミング・モデルおよびコンテナ・サポート・フィーチャー

BRBeans コンポーネントは推奨されないため、ビジネス・ルールと差し替えられます。

推奨されるマイグレーション・アクション

ユーザーは、使用されているすべての BRBeans を手動で除去し、ビジネス・ルールに移行する必要があります。

バージョン 6 で、一部の BPEL ビジネス・プロセス・モデル構成体が構文的に変更されました。WebSphere Integration Developer バージョン 6.0 では、構文のみがサポートされます。これらの構成体のマイグレーションが可能です。

推奨されるマイグレーション・アクション

WebSphere Integration Developer 提供のマイグレーション・ウィザードを使用して、 WebSphere Business Integration Server Foundation バージョン 5.1 のサービス・プロジェクト (プロセス定義を含む) を WebSphere Process Server バージョン 6.0 にマイグレーションしてください。マイグレーション・ウィザードが完了したら、いくつかの手動ステップを実行してマイグレーションを完成させる必要があります。サービス・プロジェクトのマイグレーションの詳細については、WebSphere Integration Developer バージョン 6.0 のインフォメーション・センターを参照してください。

WebSphere Business Integration Server Foundation バージョン 5.1 には、取り消しサービスの入力用のオプションがあります。この取り消しサービスでは、出力データによってオーバーレイされる、補正可能なサービスの入力データをマージした結果のメッセージを暗黙的に提供します。BPEL が提供する拡張補正のサポートを前提として、この機能は推奨されません。

推奨されるマイグレーション・アクション

ビジネス・プロセスの BPEL 補正を使用してください。

Business Flow Manager の機能性の変更のため、WebSphere Process Server バージョン 6.0 の汎用プロセス API では、以下のメソッドは推奨されません。

- WorkList オブジェクトの名前が StoredQuery に変更されました。このため、BusinessFlowManager Bean で以下のメソッドは使用すべきではありません。該当する場合、WebSphere Process Server バージョン 6.0 を使用するメソッドを以下に示します。
 - newWorkList(String workListName, String selectClause, String whereClause, String orderByClause, Integer threshold, TimeZone timezone)
 代替りのメソッド: createStoredQuery(String storedQueryName, String selectClause, String whereClause, String orderByClause, Integer threshold, TimeZone timezone)
 - getWorkListNames()
 代替りのメソッド: getStoredQueryNames()
 - deleteWorkList(String workListName)
 代替りのメソッド: deleteStoredQuery(String storedQueryName)
 - getWorkList(String workListName)
 代替りのメソッド: getStoredQuery(String storedQueryName)
 - executeWorkList(String workListName)
 代替りのメソッド: query(String storedQueryName, Integer skipTuples)
 - getWorkListActions()
 サポートされません。
- WorkListData オブジェクトは推奨されません。
 代わりに、StoredQueryData を使用してください。
- ProcessTemplateData オブジェクトの以下のメソッドは、サポートされなくなりました。
 getInputMessageTypeSystemName()
 getOutputMessageTypeSystemName()
- ProcessInstanceData オブジェクトの以下のメソッドは、サポートされなくなりました。
 getInputMessageTypeSystemName()
 getOutputMessageTypeSystemName()
- ActivityInstanceData オブジェクトの以下のメソッドは、サポートされなくなりました。
 getInputMessageTypeSystemName()
 getOutputMessageTypeSystemName()
- ActivityServiceTemplateData オブジェクトの以下のメソッドは、サポートされなくなりました。
 getInputMessageTypeSystemName()

推奨されるマイグレーション・アクション

代替りのメソッドがある場合は、そのメソッドを使用してください。

Human Task Manager の機能性の変更のため、WebSphere Process Server バージョン 6.0 の汎用プロセス API では、以下のメソッドは推奨されません。

- HumanTaskManager Bean では、以下のメソッドは使用すべきではありません。WebSphere Process Server バージョン 6.0 で使用する代わりにのメソッドを以下に示します。

- createMessage(TKIID tkiid, String messageTypeName)

代わりに、createInputMessage(TKIID tkiid), createOutputMessage(TKIID tkiid), createFaultMessage(TKIID tkiid) の個別のメソッドを使用してください。

- createMessage(String tkiid, String messageTypeName)

代わりに、createInputMessage(String tkiid), createOutputMessage(String tkiid), createFaultMessage(String tkiid) の個別のメソッドを使用してください。

- Task オブジェクトで、以下のメソッドがサポートされなくなりました。

getInputMessageTypeNames()

getOutputMessageTypeNames()

推奨されるマイグレーション・アクション

代わりにのメソッドがある場合は、そのメソッドを使用してください。

以下のデータベース・ビューは推奨されません。

- 説明
- CUSTOM_PROPERTY

推奨されるマイグレーション・アクション

DESCRIPTION ビューの代わりに TASK_DESC ビューを、CUSTOM_PROPERTY ビューの代わりに TASK_CPROP ビューを使用してください。

Java コードの断片のプログラミング・モデル

- WebSphere Business Integration Server Foundation バージョン 5.1 では、インライン Java コードの断片 (アクティビティおよび条件) 内部の BPEL 変数に、getter メソッドおよび setter メソッドを通じてアクセスします。これらのメソッドはサポートされません。Java コードの断片内の BPEL 変数を表すために使用される WSIFMessage メソッドも、サポートされません。
- メソッド <typeOfP> getCorrelationSet<cs> Property<p>() は、スコープ・レベルで宣言された相関セットを考慮しないため、サポートされません。プロセス・レベルで宣言された相関セットにアクセスする場合のみ使用可能です。
- Java 断片アクティビティ内部のカスタム・プロパティにアクセスする WebSphere Business Integration Server Foundation バージョン 5.1 メソッドはサポートされません。
- 以下の getPartnerLink メソッドはサポートされません。スコープ・レベルで宣言されたパートナー・リンクを考慮していないため、プロセス・レベルで宣言されたパートナー・リンクにアクセスする場合にのみ使用可能です。

EndpointReference getPartnerLink();

EndpointReference getPartnerLink(int role);

void setPartnerLink(EndpointReference epr);

推奨されるマイグレーション・アクション

WebSphere Integration Developer 6.0 提供のマイグレーション・ウィザードを使用して、WebSphere Business Integration Server Foundation バージョン 5.1 のサービス・プロジェクト (プロセス定義を含む) を WebSphere Process Server バージョン 6.0 にマイグレーションしてください。マイグレーション・ウィザードが完了したら、いくつかの手動ステップを実行してマイグレーションを完成させる必要があります。サービス・プロジェクトのマイグレーションの詳細については、WebSphere Integration Developer バージョン 6.0 のインフォメーション・センターを参照してください。

アプリケーション・サービス・フィーチャー

拡張メッセージング・サービス・フィーチャー、およびすべての EMS/CMM API と SPI は推奨されません。

com/ibm/websphere/ems/CMMCorrelator
com/ibm/websphere/ems/CMMException
com/ibm/websphere/ems/CMMReplyCorrelator
com/ibm/websphere/ems/CMMRequest
com/ibm/websphere/ems/CMMResponseCorrelator
com/ibm/websphere/ems/ConfigurationException
com/ibm/websphere/ems/FormatException
com/ibm/websphere/ems/IllegalStateException
com/ibm/websphere/ems/InputPort
com/ibm/websphere/ems/OutputPort
com/ibm/websphere/ems/transport/jms/JMSRequest
com/ibm/websphere/ems/TimeoutException
com/ibm/websphere/ems/TransportException
com/ibm/ws/spi/ems/CMMFactory
com/ibm/ws/spi/ems/format/cmm/CMMFormatter
com/ibm/ws/spi/ems/format/cmm/CMMParser
com/ibm/ws/spi/ems/format/Formatter
com/ibm/ws/spi/ems/format/Parser
com/ibm/ws/spi/ems/transport/CMMReceiver
com/ibm/ws/spi/ems/transport/CMMReplySender
com/ibm/ws/spi/ems/transport/CMMSender
com/ibm/ws/spi/ems/transport/MessageFactory

推奨されるマイグレーション・アクション

拡張メッセージング・サービスとその関連ツールを使用する代わりに、標準の JMS API、またはそれと同等のメッセージング・テクノロジーを使用する必要があります。

WebSphere Business Integration Server Foundation バージョン 5.1.1 で使用すべきでないフィーチャー

WebSphere Business Integration Server Foundation バージョン 5.1.1 には、使用すべきでないフィーチャーはありません。

WebSphere Business Integration Server Foundation バージョン 5.1 で使用すべきでないフィーチャー

インストールおよびマイグレーション・ツール

WebSphere Studio Application Developer Integration Edition バージョン 5.0 またはそれ以前にモデルとなっていたビジネス・プロセスは、使用すべきではありません。

推奨されるマイグレーション・アクション

WebSphere Studio Application Developer Integration Edition バージョン 5.1 で提供されている **マイグレーション・オプション**を使用して、ビジネス・プロセスを BPEL 関連のプロセスにマイグレーションしてください。

バージョン 5.0 以前の WebSphere Studio Application Developer Integration Edition で作成されたビジネス・プロセスで使用される、いくつかの Business Process Choreographer API インターフェースおよびメソッド。

推奨されるマイグレーション・アクション

これらの API インターフェースおよびメソッドの詳細なリストが必要な場合は、Business Process Choreographer で提供される Javadoc を参照してください。

アプリケーション・プログラミング・モデルおよびコンテナ・サポート・フィーチャー

以下の public クラス、メソッド、および属性を含むビジネス・ルール Bean プログラミング・インターフェースは使用すべきではありません。

- public クラス:
 - com.ibm.websphere.brb.RuleImporter
 - com.ibm.websphere.brb.RuleExporter
- public メソッド:
 - getLocalRuleManager() on class com.ibm.websphere.brb.TriggerPoint
- Protected 属性:
 - ruleMgr on class com.ibm.websphere.brb.TriggerPoint

推奨されるマイグレーション・アクション

アクションは不要です。

以下の com.ibm.websphere.scheduler クラス・プログラミング・インターフェースの scheduler.Scheduler メソッドは使用すべきではありません。

```
public BeanTaskInfo createBeanTaskInfo();
public MessageTaskInfo createMessageTaskInfo();
```

推奨されるマイグレーション・アクション

以下のメソッドを使用します。

```
public Object createTaskInfo(Class taskInfoInterface) throws TaskInfoInvalid;
BeanTaskInfo ti = (BeanTaskInfo) Scheduler.createTaskInfo(BeanTaskInfo.class);
```

Web サービスのゲートウェイ・カスタマイズ API は使用すべきではありません。

推奨されるマイグレーション・アクション

アクションは不要です。ただし、可能であれば、フィルターなどの Web サービスのゲートウェイ固有のインターフェースではなく、XML ベースのリモート・プロシージャ・コール (JAX-RPC) のハンドラー用の Java API を使用してください。Web サービスのゲートウェイ API は、将来のリリースでは置き換えられる予定です。詳細については、WebSphere Business Integration Server Foundation インフォメーション・センターにあるトピック『JAX-RPC handlers - An alternative to gateway filters』を参照してください。

第 4 章 マイグレーションのトラブルシューティング

マイグレーション中に問題が発生する場合は、ここで説明する情報を参考にしてください。

バージョン間のマイグレーションのトラブルシューティング

WebSphere Process Server の古いバージョンからのマイグレーション時に問題が発生した場合は、このページのトラブルシューティングのヒントを参照してください。

- 構成をマイグレーションする前に、バージョン 6.1 マイグレーション・ウィザードを使用してプロファイルを作成しているときに、以下のプロファイル作成エラー・メッセージが表示される可能性があります。

```
profileName: profileName cannot be empty  
profilePath: Insufficient disk space
```

これらのエラー・メッセージは、スペースなどの誤った文字を含むプロファイル名を入力した場合に表示される可能性があります。マイグレーション・ウィザードを再実行して、プロファイル名にスペース、引用符、他の特殊文字などの誤った文字が含まれていないことを確認してください。

- WebSphere Process Server の旧バージョンからバージョン 6.1 へのマイグレーション中に問題が発生する場合は、ログ・ファイルや他の入手可能な情報を確認してください。

1. ログ・ファイルを見つけて手掛かりを探してください。

- *migration_backup_directory*/WBIPreUpgrade.time_stamp.log
- *profile_root*/log/WASPostUpgrade.time_stamp.log
- *install_root*/logs/clientupgrade.time_stamp.log
- *profile_root*/logs/bpeupgrade.log
- *migration_backup_directory*/WBIProfileUpgrade.ant.timestamp.log

2. 以下のディレクトリーで、「MIGR0259I: マイグレーションは正常に完了しました。」または「MIGR0271W: マイグレーションは、1 つ以上の警告を伴って、正常に完了しました。」を探します。

- *migration_backup_directory*/WBIPreUpgrade.time_stamp.log
- *profile_root*/logs/WASPostUpgrade.time_stamp.log
- *install_root*/logs/clientupgrade.time_stamp.log

「MIGR0286E: マイグレーションは完了できませんでした。」が表示されている場合、ログ・ファイルに記録されているエラー・メッセージに基づいて、すべての問題を訂正するようにしてください。エラーを訂正したら、製品のインストール・ルートの *bin* ディレクトリーからコマンドを再実行します。

3. アクセスしようとしているリソースをホスティングしているサーバーのサービス・ログの Application Server Toolkit (AST) に組み込まれている Log and Trace Analyzer を開いて、エラー・メッセージおよび警告メッセージを参照します。

Application Server Toolkit でのコンポーネントのデバッグを参照してください。

4. WebSphere Process Server で `dumpNameSpace` コマンドを実行し、出力をパイピング、リダイレクト、および詳細出力にして、出力を見やすくします。

このコマンドを実行すると、WebSphere Process Server 名前空間のすべてのオブジェクトがディレクトリー・パスとオブジェクト名を含めて表示されます。

5. クライアントがアクセスする必要があるオブジェクトが表示されない場合は、管理コンソールを使用して、以下の状態を確認します。
 - ターゲット・リソースをホスティングしているサーバーが開始していること。
 - ターゲット・リソースをホスティングしている Web モジュールまたは Enterprise JavaBean コンテナが稼働していること。
 - ターゲット・リソースの JNDI 名が正しく指定されていること。

これらのステップのいずれでも問題を解決できない場合は、IBM サポートとの連絡方法なども記載されている追加のトラブルシューティング・リソースについて、トラブルシューティングおよびサポートを参照してください。

- マイグレーション・プロセス時に、`WBIPreUpgrade` コマンドまたは `WBIPostUpgrade` コマンドを使用中、問題が発生することがあります。
 - `WBIPreUpgrade` コマンドの使用中に、問題が発生する可能性があります。
 - 「見つかりません」または「そのようなファイルまたはディレクトリーがありません (No such file or directory)」というメッセージが返されます。

この問題は、`WBIPreUpgrade` コマンドを WebSphere Process Server バージョン 6.1 `install_root/bin` ディレクトリー以外のディレクトリーから実行しようとする場合に発生する可能性があります。`WBIPreUpgrade` スクリプトが、バージョン 6.1 `install_root/bin` ディレクトリーに存在することを確認し、このファイルをその場所から起動します。

- DB2 JDBC ドライバーおよび DB2 JDBC ドライバー (XA) が、管理コンソールに表示されているサポートされる JDBC プロバイダーのドロップダウン・リスト内で見つかりません。

管理コンソールには、推奨されない JDBC プロバイダー名が表示されなくなりました。管理コンソールで使用されている新しい JDBC プロバイダー名は、より説明的で、紛らわしさが解消されています。新しいプロバイダー名と推奨されないプロバイダー名は、名前だけが異なっています。

推奨されない名前は、マイグレーション上の理由で (例えば、既存の JACL スクリプトなどのために) `jdbc-resource-provider-templates.xml` ファイルに引き続き残されています。ただし、JACL スクリプトでは、新しい JDBC プロバイダー名を使用するようお勧めします。

- 以下のメッセージを受け取ります。

MIGR0108E: The specified WebSphere directory does not contain a WebSphere version that can be upgraded.

これは、WBIPreUpgrade コマンドで、誤ったディレクトリーを指定した場合に発生する可能性があります。

WBIPreUpgrade コマンドを参照してください。

- WBIPostUpgrade コマンドを使用中に、問題が発生する可能性があります。
 - 「見つかりません」または「そのようなファイルまたはディレクトリーがありません (No such file or directory)」というメッセージが返されます。

この問題は、WBIPostUpgrade コマンドを WebSphere Process Server バージョン 6.1 *install_root*\bin 以外のディレクトリーから実行しようとする場合に発生する可能性があります。WBIPostUpgrade スクリプトが、バージョン 6.1 *install_root*\bin ディレクトリーに存在することを確認して、このファイルをその場所から起動します。

- セル内で統合ノードをマイグレーションすると、以下のエラー・メッセージを受け取ります。

```
MIGR0304I: The previous WebSphere environment is being restored.  
com.ibm.websphere.management.exception.RepositoryException:  
com.ibm.websphere.management.exception.ConnectorException: ADMC0009E:  
The system failed to make the SOAP RPC call: invoke  
MIGR0286E: The migration failed to complete.
```

接続タイムアウトは、統合ノードの WBIPostUpgrade マイグレーション・ステップ中に、統合ノードが Deployment Manager から構成の更新を検索しようとするときに発生します。バージョン 6.1 にマイグレーションする構成に以下のいずれかの要素が含まれている場合、構成全体のコピーにかかる時間が、接続タイムアウトより長くなる可能性があります。

- 小規模アプリケーションが多数ある
- 大規模アプリケーションがいくつかある
- 非常に大規模なアプリケーションが 1 つある

これが発生した場合は、タイムアウト値を変更してから、WBIPostUpgrade コマンドを実行して統合ノードをマイグレーションしてください。

1. 統合ノードのマイグレーション先のプロファイルが含まれている バージョン 6.1 ディレクトリー内の以下の場所に移動します。

profile_root/properties

2. このディレクトリー内の *soap.client.props* ファイルを開き、*com.ibm.SOAP.requestTimeout* プロパティーの値を見つけます。これは、秒単位のタイムアウト値です。デフォルト値は 180 秒です。
3. *com.ibm.SOAP.requestTimeout* の値を変更して、構成をマイグレーションできるように十分に大きくします。例えば、以下のように入力すると、タイムアウト値は 30 分になります。

```
com.ibm.SOAP.requestTimeout=1800
```

注: タイムアウト値には、必要を満たす最小の値を選択してください。選択したタイムアウトの少なくとも 3 倍の長さの待機時間を見込んでください。つまり、ファイルをバックアップ・ディレクトリーにダウンロード

ドする時間、マイグレーション済みのファイルをデプロイメント・マネージャーにアップロードする時間、およびデプロイメント・マネージャーとマイグレーション済みのノード・エージェントとを同期化する時間です。

4. `WBIPreUpgrade` コマンドで作成したバックアップ・ディレクトリーの、以下のロケーションに移動します。

```
migration_backup_directory/profiles/default/properties
```

5. このディレクトリー内の `soap.client.props` ファイルを開き、`com.ibm.SOAP.requestTimeout` プロパティーの値を見つけます。
6. `com.ibm.SOAP.requestTimeout` の値を バージョン 6.1 ファイルで使用しているのと同じ値に変更します。

- 「Unable to copy document to temp file」というエラー・メッセージが表示されます。以下に例を示します。

```
MIGR0304I: The previous WebSphere environment is being restored.  
com.ibm.websphere.management.exception.DocumentIOException: Unable to copy  
document to temp file:  
cells/sunblade1Network/applications/LARGEApp.ear/LARGEApp.ear
```

ファイル・システムに空きがない可能性があります。ファイル・システムに空きがない場合、一部のスペースを消去して `WBIPostUpgrade` コマンドを再実行してください。

- 以下のメッセージを受け取ります。

```
MIGR0108E: The specified WebSphere directory does not contain a WebSphere  
version that can be upgraded.
```

このエラーの原因として、以下のような理由が存在すると考えられます。

- `WBIPreUpgrade` コマンドまたは `WBIPostUpgrade` コマンドの起動時に誤ったディレクトリーが指定されていた。
- `WBIPreUpgrade` コマンドが実行されなかった。

- 以下のエラー・メッセージを受け取ります。

```
MIGR0253E: The backup directory migration_backup_directory does not exist.
```

このエラーの原因として、以下のような理由が存在すると考えられます。

- `WBIPreUpgrade` コマンドが、`WBIPostUpgrade` コマンドの前に実行されなかった。
 1. エラー・メッセージに示されているバックアップ・ディレクトリーが存在するかどうかを確認します。
 2. 存在しない場合は、`WBIPreUpgrade` コマンドを実行します。

`WBIPreUpgrade` コマンドを参照してください。

3. `WBIPostUpgrade` コマンドを再試行します。

- 誤ったバックアップ・ディレクトリーが指定されていた可能性がある。

例えば、ディレクトリーが、`WBIPreUpgrade` コマンドの実行後に削除されたバージョン 6.0.x ツリーのサブディレクトリーであり、`WBIPostUpgrade` コマンドの実行前に、製品の旧バージョンがアンインストールされたということが考えられます。

1. エラー・メッセージに示されているディレクトリー構造全体が存在するかどうかを判別します。
 2. 可能であれば、正しいマイグレーション・バックアップ・ディレクトリー全体を指定して、WBIPreUpgrade コマンドを再実行します。
 3. バックアップ・ディレクトリーが存在しない場合で、旧バージョンが削除されている場合は、バックアップ・リポジトリーまたは XML 構成ファイルから旧バージョンを再ビルドします。
 4. WBIPreUpgrade コマンドを再実行します。
- WBIPostUpgrade コマンドの実行後に、WBIPreUpgrade をもう一度実行しなければならなくなりました。

Deployment Manager または管理対象ノードのマイグレーションの過程で、WBIPostUpgrade が旧環境を無効にする可能性があります。WBIPostUpgrade の実行後に、WBIPreUpgrade を旧インストールに対してもう一度実行する場合、旧 `install_root/bin` ディレクトリーに存在する `migrationDisablementReversal.jacl` スクリプトを実行する必要があります。この JACL スクリプトを実行すると、バージョン 6.0.x 環境はもう一度有効な状態になり、WBIPreUpgrade を実行して有効な結果を出すことができるようになります。

スクリプト記述について詳しくは、スクリプト記述入門を参照してください。ここで説明されているスクリプト記述は、WebSphere Process Server で使用可能です。

- 統合マイグレーションが、メッセージ MIGR0405E で失敗します。

統合マイグレーションの一環として Deployment Manager で実行されたマイグレーションが失敗しました。このエラーが発生した詳しい理由については、Deployment Manager ノードの `...DeploymentManagerProfile/temp` ディレクトリーの下にあるフォルダー `your_node_name_migration_temp` を開いてください。以下に例を示します。

```
/websphere61/procserver/profiles/dm_profile/temp/nodeX_migration_temp
```

Deployment Manager ノード上のこのノードのマイグレーションに関係するログや他のすべての情報は、このフォルダーに置かれています。このフォルダーは、このシナリオに関連した IBM サポートでも必要になります。

- WebSphere Process Server バージョン 6.1 アプリケーションがマイグレーション中に失われる。

統合マイグレーション中に、バージョン 6.1 アプリケーションのいずれかがインストールに失敗する場合、それらのアプリケーションは構成の同期化中に失われます。これが発生する理由は、WBIPostUpgrade の最終手順の 1 つで、`syncNode` コマンドが実行されるためです。この結果、Deployment Manager ノードの構成がダウンロードされ、統合ノードの構成が上書きされます。アプリケーションのインストールが失敗すると、それらのアプリケーションは Deployment Manager ノードの構成に含まれなくなります。この問題を解決するには、マイグレーション後にアプリケーションを手動でインス

インストールしてください。標準 のバージョン 6.1 アプリケーションの場合、*install_root/installableApps* ディレクトリーにあります。

マイグレーション中に失われたアプリケーションを手動でインストールするには、*wsadmin* コマンドを使用して、マイグレーション・ツールがバックアップ・ディレクトリーに作成した *install_application_name.jacl* スクリプトを実行します。

Linux 例えば、Linux 環境では以下のパラメーターを使用します。

```
./wsadmin.sh -f migration_backup_directory/install_application_name.jacl  
-conntype NONE
```

Wsadminツールを参照してください。

- WebSphere Process Server バージョン 6.1 アプリケーションのインストールが失敗する。

WBIPostUpgrade の完了後に、*wsadmin* コマンドを使用して、アプリケーションを手動でインストールします。

マイグレーション中にインストールが失敗したアプリケーションを手動でインストールするには、*wsadmin* コマンドを使用して、マイグレーション・ツールがバックアップ・ディレクトリーに作成した *install_application_name.jacl* スクリプトを実行します。

Linux 例えば、Linux 環境では以下のパラメーターを使用します。

```
./wsadmin.sh -f migration_backup_directory/install_application_name.jacl  
-conntype NONE
```

『Wsadmin ツール』または WBIPostUpgrade コマンドを参照してください。

- **Solaris** マイグレーション・ウィザードを使用して、Solaris x64 プロセッサ・ベースのシステム上でプロファイル WebSphere Process Server バージョン 6.0.x から バージョン 6.1.x にマイグレーションする場合は、WBIPostUpgrade ステップ中にマイグレーションが失敗する可能性があります。

*profile_root/logs/WASPostUpgrade.time_stamp.log*内に、以下のようなメッセージが記録される場合があります。

```
MIGR0327E: A failure occurred with stopNode.  
MIGR0272E: The migration function cannot complete the command.
```

WebSphere Process Server バージョン 6.0.x は、Java 仮想マシン (JVM) を 32 ビット・モードで使用します。WebSphere Process Server バージョン 6.1.x のマイグレーション・ウィザードは、WBIPostUpgrade.sh スクリプトを呼び出します。このスクリプトは、サーバーが バージョン 6.0.x ノードを停止すると、バージョン 6.0.x 用の JVM を 64 ビット・モードで実行しようとしています。

以下のアクションを実行して、不完全なプロファイルを除去し、WebSphere Process Server が バージョン 6.0.x プロファイルを正しくマイグレーションできるようにします。

1. コマンド行で、*install_root/bin* ディレクトリーに移動します。

例えば、以下のコマンドを入力します。

```
cd /opt/IBM/WebSphere/Procserver/bin
```

2. `install_root/bin` ディレクトリー内で `WBIPostUpgrade.sh` スクリプトを見つけ、バックアップ・コピーを作成します。
3. `WBIPostUpgrade.sh` または `WBIPostUpgrade.bat` ファイルをエディターで開き、以下のアクションを実行します。
 - a. 以下のコード行を見つけます。

UNIX

Linux

```
"$binDir" /setupCmdLine.sh
```

Windows

```
call "%dp0setupCmdLine.bat" %*
```

- b. 前のステップで特定したコードの後ろに、以下のコード行を挿入します。

```
JVM_EXTRA_CMD_ARGS=""
```
 - c. 変更を保管します。
4. `WASPostUpgrade.sh` または `WASPostUpgrade.bat` ファイルについて、ステップ 2 から 4 を繰り返します。
5. 以下のコマンドを使用して、マイグレーション・プロセス中に作成された不完全なバージョン 6.1.x プロファイルを削除します。

```
install_root/bin/manageprofiles.sh -delete -profileName profile_name
```
6. 前のステップで除去したバージョン 6.1.x プロファイルの `profile_root` ディレクトリーを削除します。
7. マイグレーション・ウィザードを再実行します。
- マイグレーション・プロセスで、バージョン 6.0.x 構成に存在するエンタープライズ・アプリケーションを新しいバージョン 6.1 構成にインストールするオプションを選択すると、マイグレーションのアプリケーション・インストール・フェーズでエラー・メッセージが表示される場合があります。

バージョン 6.0.x 構成に存在するアプリケーションのデプロイメント情報が誤っている可能性があります。その場合、WebSphere Process Server の旧ランタイムで十分に検証されなかったために XML 文書が誤っているという場合がほとんどです。ランタイムのアプリケーション・インストール検証プロセスが改善されているため、これらの誤った形式の EAR ファイルのインストールが失敗します。このため、`WBIPostUpgrade` のアプリケーション・インストール・フェーズで障害が発生し、「E:」エラー・メッセージが生成されます。これは「致命的な」マイグレーション・エラーと見なされます。

アプリケーションのインストール中に、マイグレーションがこのような方法で失敗する場合、以下のいずれかを実行してください。

- バージョン 6.0.x アプリケーションの問題を修正してから、再マイグレーションする。
- マイグレーションを続行し、これらのエラーを無視する。

この場合、マイグレーション・プロセスでは、障害が起こったアプリケーションはインストールされませんが、他のすべてのマイグレーション手順は完了します。

後で、アプリケーションの問題を修正してから、管理コンソールまたはインストール・スクリプトを使用して新しいバージョン 6.1 構成に手動でインストールできます。

- WebSphere Process Server バージョン 6.0.1.3 より古いバージョン 6.0.x ノードを含んでいるか、これらのノードと相互運用するバージョン 6.1 セルにマイグレーションすると、クラスター機能に障害が発生する場合があります。

これらのバージョン 6.0.x サーバーを始動すると、以下の問題が発生する可能性があります。

- First Failure Data Capture (FFDC) ログに ClassNotFoundException エラー・メッセージが記録される場合があります。この例外は RuleEtiquette.runRules メソッドからスローされ、以下のような形式になっています。

```
Exception = java.lang.ClassNotFoundException
Source = com.ibm.ws.cluster.selection.SelectionAdvisor.<init>
probeid = 133
Stack Dump = java.lang.ClassNotFoundException: rule.local.server
at java.net.URLClassLoader.findClass(URLClassLoader.java(Compiled Code))
at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:106)
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.ClassLoader.loadClass(ClassLoader.java(Compiled Code))
at java.lang.Class.forName(Native Method)
at java.lang.Class.forName(Class.java(Compiled Code))
at com.ibm.ws.cluster.selection.rule.RuleEtiquette.runRules(RuleEtiquette.java:154)
at com.ibm.ws.cluster.selection.SelectionAdvisor.handleNotification(SelectionAdvisor.java:153)
at com.ibm.websphere.cluster.topography.DescriptionFactory$Notifier.run(DescriptionFactory.java:257)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:1462)
```

- 以下のような形式の java.io.IOException が記録される場合があります。

```
Exception = java.io.IOException
Source = com.ibm.ws.cluster.topography.DescriptionManagerA.update probeid = 362
Stack Dump = java.io.IOException
at com.ibm.ws.cluster.topography.ClusterDescriptionImpl.importFromStream(ClusterDescriptionImpl.java:916)
at com.ibm.ws.cluster.topography.DescriptionManagerA.update(DescriptionManagerA.java:360)
Caused by: java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java(Compiled Code))
at java.io.DataInputStream.readUTF(DataInputStream.java(Compiled Code))
at com.ibm.ws.cluster.topography.KeyRepositoryImpl.importFromStream(KeyRepositoryImpl.java:193)
```

マイグレーション中にバージョン 6.1 クラスター情報がセル全体に配布されません。バージョン 6.0.1.3 以降ではない WebSphere Process Server バージョン 6.0.x ノードは、この情報を読み取ることができません。

この問題を回避するには、Deployment Manager をバージョン 6.1 にマイグレーションする前に、バージョン 6.1 セルに含まれるか、このセルと相互運用されるすべてのバージョン 6.0.x ノードをバージョン 6.0.1.3 以降にアップグレードします。

- 管理対象ノードをバージョン 6.1 にマイグレーションした後、アプリケーション・サーバーが始動しない場合があります。

アプリケーション・サーバーを始動しようとする、以下の例のようなエラーが発生する場合があります。

```
[5/11/06 15:41:23:190 CDT] 0000000a SystemErr R
    com.ibm.ws.exception.RuntimeError:
com.ibm.ws.exception.RuntimeError: org.omg.CORBA.INTERNAL:
    CREATE_LISTENER_FAILED_4
vmcid: 0x49421000 minor code: 56 completed: No
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.bootServerContainer(WsServerImpl.java:198)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.start(WsServerImpl.java:139)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServerImpl.main(WsServerImpl.java:460)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
com.ibm.ws.runtime.WsServer.main(WsServer.java:59)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
[5/11/06 15:41:23:196 CDT] 0000000a SystemErr R at
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:64)
[5/11/06 15:41:23:197 CDT] 0000000a SystemErr R at
sun.reflect.DelegatingMethodAccessorImpl.invoke
    (DelegatingMethodAccessorImpl.java:43)
```

管理対象ノードのサーバーが listen するポート番号を変更します。例えば、Deployment Manager がポート 9101 で ORB_LISTENER_ADDRESS を listen している場合、管理対象ノードのサーバーはポート 9101 で ORB_LISTENER_ADDRESS を listen してはいけません。この例のような問題を解決するには、以下の手順を実行します。

1. 管理コンソールで、「アプリケーション・サーバー」 → 「*server_name*」 → 「ポート」 → 「ORB_LISTENER_ADDRESS」をクリックします。
 2. ORB_LISTENER_ADDRESS のポート番号を使用されていない番号に変更します。
- 管理対象ノードのバージョン 6.1 へのマイグレーション時に同期に失敗すると、サーバーが始動しない場合があります。

管理対象ノードをバージョン 6.1 にマイグレーションすると、以下のようなメッセージが記録される場合があります。

```
ADMU0016I: Synchronizing configuration between node and cell.
ADMU0111E: Program exiting with error:
    com.ibm.websphere.management.exception.AdminException: ADMU0005E:
    Error synchronizing repositories
ADMU0211I: Error details may be seen in the file:
    /opt/WebSphere/61AppServer/profiles/AppSrv02/logs/syncNode.log
MIGR0350W: Synchronization with the deployment manager using the SOAP protocol
    failed.
MIGR0307I: The restoration of the previous WebSphere Application Server
    environment is complete.
MIGR0271W: Migration completed successfully, with one or more warnings.
```

これらのメッセージは、以下のことを示しています。

- Deployment Manager の構成レベルがバージョン 6.1 になっている。

- これからマイグレーションする管理対象ノードの構成レベルが (アプリケーションも含めて)、Deployment Manager のリポジトリでバージョン 6.1 になっている。
- syncNode 操作を完了しなかった場合は、管理対象ノードがまったく完了していない。

以下のアクションを実行して、この問題を解決します。

1. ノード上で syncNode コマンドを再実行し、ノードを Deployment Manager と同期化します。

syncNodeコマンドを参照してください。

2. GenPluginCfg コマンドを実行します。

GenPluginCfgコマンドを参照してください。

次のタスク

問題がリストされていない場合は、IBM サポートにお問い合わせください。

関連概念

132 ページの『Business Process Choreographer に関するマイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働させている場合、いくつかの制限事項および実行する必要のある追加タスクに注意してください。

関連タスク

109 ページの『マイグレーションの検査』

ログ・ファイルを確認し、管理コンソールで操作を確認して、マイグレーションが正常に行われたことを検査します。

関連資料

14 ページの『WBIPreUpgrade コマンド』

WebSphere Process Server の WBIPreUpgrade コマンドを使用して、前にインストールされたバージョンの WebSphere Process Server の構成をマイグレーション固有のバックアップ・ディレクトリーに保存します。

17 ページの『WBIPostUpgrade コマンド』

WebSphere Process Server の WBIPostUpgrade コマンドは、WBIPreUpgrade コマンドによって作成された保存済み構成を、指定された *backupDirectory* から取り出すために使用します。WebSphere Process Server の WBIPostUpgrade コマンドは、このディレクトリーから構成を読み込んで、最新バージョンの WebSphere Process Server にマイグレーションし、マイグレーションされたすべてのアプリケーションを新規インストール用の *profile_root/installedApps* ディレクトリーに追加します。

関連情報

Application Server Toolkit でのコンポーネントのデバッグ

Wsadmin ツール

syncNode コマンド

GenPluginCfg コマンド

 [トラブルシューティングおよびサポート](#)

ご使用の IBM ソフトウェアに関する問題の理解、特定、および解決を支援するため、トラブルシューティングおよびサポート情報では IBM 製品で提供されている問題判別リソースの使用について説明しています。

スクリプト記述入門

WebSphere InterChange Server からのマイグレーションのトラブルシューティング

マイグレーションで発生する問題の解決策と、ロギングとトレースをオンにする方法について説明します。

関連概念

199 ページの『WebSphereInterChange Server からマイグレーションする場合の制限事項』

WebSphere InterChange Server の特性の中には、WebSphere Process Server によって正確に再現されないものがあります。そのため、WebSphere InterChange server と同じように実行するために、マイグレーション後にアプリケーションを変更しなければならない場合があります。

関連資料

158 ページの『事後マイグレーションの考慮事項』

アプリケーションが WebSphere InterChange Server から WebSphere Process Server にマイグレーションされた場合は、WebSphere Process Server と WebSphere InterChange Server のアーキテクチャーの間には違いがあるため、マイグレーションされたアプリケーションが WebSphere Process Server において意図したとおりに機能するように、一部の領域に特別の注意を払う必要があります。

146 ページの『事前マイグレーションの考慮事項』

WebSphere InterChange Server 成果物を WebSphere Process Server にマイグレーションする作業を容易にするために、WebSphere InterChange Server 用の統合成果物を開発するための以下のガイドラインを検討してください。

サポートされる WebSphere InterChange Server API のロギングおよびトレースの使用可能化

サポートされる WebSphere InterChange Server API のロギングおよびトレースを、管理コンソールを通じて使用可能にします。

このタスクについて

マイグレーション済みアプリケーションに、サポートされる WebSphere InterChange Server API が含まれている場合、トラブルシューティング用にそれらのロギングおよびトレースを使用可能にすることができます。

プロシージャ

1. 管理コンソールを起動します。
2. 左側の (ナビゲーション) パネルから、「トラブルシューティング」>「ログおよびトレース」を選択します。

3. 右側のパネルで、ロギングおよびトレースを使用可能にしたいサーバーの名前を選択します。
4. 右側のパネルの「一般プロパティ」の下で、「**ログ・レベル詳細の変更 (Change Log Level Details)**」を選択します。
5. 「ランタイム」タブを選択します。（「ランタイム」タブを選択すると、リアルタイムにこの変更を行うことができ、サーバーを再始動する必要はありません。）
6. パッケージの名前の後に `=all` を付加したものを、画面のボックス内のログに記録されるパッケージのリストに追加します。この新規の項目は、コロンを使用して既存の項目と分離します。例えば、`CxCommon=all` と指定できます。この場合、`CxCommon` は、一連のサポートされる `WebSphere InterChange Server API` のパッケージの名前です。 `all` を指定すると、すべてのロギングおよびトレースが使用可能になります。API およびそのパッケージ名のリストについては、『`WebSphere InterChange Server API`』のサポートを参照してください。
7. 「適用」を選択します。
8. サーバーの再始動後にもこの構成を保持するには、「**ランタイム変更も構成に保管する**」チェック・ボックスを選択します。
9. 「**OK**」を選択します。
10. 次の画面が表示されたら、「**保管**」を選択して変更内容を保存します。

関連資料



『`WebSphere InterChange Server API`』のサポート

WebSphere Process Server および WebSphere Integration Developer で提供される WebSphere InterChange Server のソース成果物のマイグレーション・ツールに加えて、WebSphere Process Server は、WebSphere InterChange Server で提供されていた API の多くもサポートします。マイグレーション・ツールは、マイグレーション時に極力カスタム断片コードを保持することにより、これらの WebSphere InterChange Server API と連動して動作します。

マイグレーションされた BPEL ファイルで直列化可能ではないオブジェクトを直列化しようとしたときの失敗

マイグレーションで生成された BPEL ファイルで直列化が失敗する場合、失敗しないように BPEL ファイルを変更できる場合があります。

問題: マイグレーションによって生成される Business Process Execution Language (BPEL) ファイルのカスタム断片ノードで、直列化可能ではないオブジェクトを直列化しようとするために、直列化が失敗します。

原因: WebSphere InterChange Server では、コラボレーション・テンプレートが単一の Java クラスにコンパイルされます。WebSphere Process Server では、BPEL ファイル内の各ノードが別々の Java クラスにコンパイルされます。WebSphere InterChange Server では、変数を一度だけ宣言し、コラボレーション・テンプレートのさまざまな手順全体で共用することができます。マイグレーションされた BPEL ファイルでその振る舞いをシミュレートするには、コード断片で使用される各変数を断片の先頭で取得し、断片の終わりで保存する必要があります。WebSphere InterChange Server ポート定義で定義される変数は、BPEL 変数になります。これ

らの変数は、各断片の始まりに BusObj 変数に取得され (断片内で参照されている場合)、各断片の終わりに BPEL 変数に再び保存されます。例えば、断片の先頭での取得コードは、次のようになります。

```
BusObj tempBusObj = null;if (tempBusObj_var != null) { tempBusObj =  
    new BusObj(tempBusObj_var); };
```

また、断片の終わりでの保管コードは次のようになります。

```
if (tempBusObj == null) { tempBusObj_var = null; } else { tempBusObj_var =  
    tempBusObj.getBusinessGraph(); }
```

WebSphere InterChange Server の断片コードで使用される他の変数は直列化され、*CollabTemplateName_var* という名前で BPEL 変数のストリングとして保管されます。これらの変数は、各 BPEL 断片の先頭で非直列化され、参照元の各 BPEL 断片の終わりに直列化および保存されます。例えば、オブジェクトは次のようにして取得されます。

```
BusObj tempBusObj = (BusObj)BaseCollaboration.deserialize  
    (FrontEndCollab_var.getString("tempBusObj"));
```

また、オブジェクトは次のようにして保存されます。

```
FrontEndCollab_var.setString("tempBusObj", BaseCollaboration.serialize(tempBusObj));
```

直列化されるオブジェクト・タイプが直列化可能ではない場合は、BPEL を実行するときに直列化および非直列化の使用に失敗します。

解決策: マイグレーション後に BPEL ファイルを次のように変更します。

- Java で直列化可能ではない変数については、BPEL 断片を更新して直列化ステートメントと非直列化ステートメントを除去します。断片間で変数を共有する (各断片で再作成されるのではない) 必要がある場合は、別の方法を使用して、断片全体で変数の値を維持する必要があります。
- WebSphere InterChange Server のポート定義で宣言されていないのにパートナー呼び出しで使用されている BusObj タイプの変数に、BPEL 変数を手動で定義します。これが手動手順になる理由は、WebSphere Process Server で呼び出し時に使用される変数は強く型付けされている必要があるのに、マイグレーション・ツールでは WebSphere InterChange Server の断片からその型を正確に判別できないためです。

注: マイグレーション・ツールで使用される命名規則では、BPEL 変数に命名するときに、断片コードの変数の名前に *_var* を追加します。例えば、断片コードで *tempBusObj* と呼ばれる変数の場合、マイグレーション・ツールは、*tempBusObj_var* という名前の BPEL 変数を作成します。

- BPEL 変数として手動で宣言する必要がある変数の場合は、「BPEL 変数から取得/BPEL 変数へ保管」変数保持方式ではなく「非直列化/直列化」変数保持方式を使用するように、BPEL 断片コードを変更する必要があります。

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。

〒106-8711
東京都港区六本木 3-2-12
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
577 Airport Blvd., Suite 800
Burlingame, CA 94010
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の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をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

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のめかしたり、保証することはできません。

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。(C) (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。(C) 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。

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

警告: 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

商標

IBM、IBM logo、Cloudscape、DB2、DB2 Universal Database、developerWorks、i5/OS、Informix、MQSeries、MVS、OS/390、RACF、WebSphere、z/OS は、International Business Machines Corporation の米国およびその他の国における商標です。

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

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc.の米国およびその他の国における商標です。

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。

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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

この製品には、Eclipse Project (<http://www.eclipse.org>) により開発されたソフトウェアが含まれています。



WebSphere Process Server for Multiplatforms バージョン 6.1.0



Printed in Japan