

バージョン 6.2.0



WebSphere Process Server のマイグレーション





お願い

本書および本書で紹介する製品をご使用になる前に、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 2、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。

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 6.2.0  
Migrating WebSphere Process Server

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2009.1

© Copyright International Business Machines Corporation 2006, 2008.

---

## PDF ブックおよびインフォメーション・センター

PDF ブックは、印刷およびオフラインでの参照用に提供されています。最新情報は、オンラインのインフォメーション・センターを参照してください。

セットとして、PDF ブックには、インフォメーション・センターと同一の内容が含まれます。

PDF 資料は、バージョン 6.0 またはバージョン 6.1 など、インフォメーション・センターのメジャー・リリースの後の四半期以内にご利用いただけます。

PDF 資料の更新頻度は、インフォメーション・センターより低いですが、Redbooks® よりも頻繁に更新されます。通常、PDF ブックはブックに十分な変更が累積されたときに更新されます。

PDF ブックの外部にあるトピックへのリンクを選択すると、Web 上のインフォメーション・センターに移動します。PDF ブックの外部にあるターゲットへのリンクには、そのターゲットが PDF ブックと Web ページのどちらなのかを示すアイコンによるマークが付いています。

表 1. 本書の外部にあるトピックへのリンクのプレフィックスとなるアイコン

アイコン	説明
	<p data-bbox="540 254 1325 281">インフォメーション・センターのページを含む、Web ページへのリンク。</p> <p data-bbox="540 310 1427 407">インフォメーション・センターへのリンクは、ターゲット・トピックが新しい場所に移動した場合でもその機能を保つように、間接参照ルーティング・サービスを経由します。</p> <p data-bbox="540 436 1427 604">ローカルのインフォメーション・センターでリンク先ページを見つけたい場合は、リンクのタイトルを検索することができます。あるいは、トピック ID を検索することもできます。検索の結果、タイプが異なる製品についてのトピックがいくつか見つかった場合は、検索結果の「<b>グループ別 (Group by)</b>」コントロールを使用して、表示するトピック・インスタンスを識別できます。以下に例を示します。</p> <ol data-bbox="540 619 1427 905" style="list-style-type: none"> <li>1. リンク URL をコピーします。例えば、リンクを右クリックして「リンク先をコピーする (Copy link location)」を選択します。例: <code>http://www14.software.ibm.com/webapp/wsbroker/redirect?version=wbpm620&amp;product=wesb-dist&amp;topic=tins_apply_service</code></li> <li>2. <code>&amp;topic=</code> の後のトピック ID をコピーします。例: <code>tins_apply_service</code></li> <li>3. ローカル・インフォメーション・センターの検索フィールドに、トピック ID を貼り付けます。文書機能がローカルにインストールされている場合は、検索結果にそのトピックが表示されます。以下に例を示します。</li> </ol> <div data-bbox="581 911 1427 1108" style="border: 1px solid black; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p data-bbox="589 926 818 947">1 result(s) found for</p> <p data-bbox="589 972 1102 1018">Group by: None   Platform   Version   Product Show Summary</p> <p data-bbox="589 1043 1346 1089">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</p> </div> <ol data-bbox="540 1142 1183 1169" style="list-style-type: none"> <li>4. 検索結果のリンクをクリックしてトピックを表示します。</li> </ol>
	<p data-bbox="540 1184 799 1211">PDF ブックへのリンク。</p>

# 目次

PDF ブックおよびインフォメーション・センター . . . . .	iii
------------------------------------	-----

## 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
WebSphere Process Server に関する事前マイグレーションの考慮事項 . . . . .	4
Business Process Choreographer に関する事前マイグレーションの考慮事項 . . . . .	13
WebSphere が提供するビジネス・スペースに関する事前マイグレーションの考慮事項 . . . . .	16
バージョン間マイグレーション用ツール . . . . .	19
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 . . . . .	29
製品構成のマイグレーション時の構成マッピング . . . . .	31
WebSphere アプリケーションのマイグレーション . . . . .	38
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 . . . . .	39
手動による共通データベースのアップグレード . . . . .	40
手動による Business Process Choreographer データベースのアップグレード . . . . .	43
手動による ビジネス・スペース データベースのアップグレード . . . . .	54
スタンドアロン・サーバーのマイグレーション . . . . .	57
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 . . . . .	58
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 . . . . .	63
リモート・システムへのマイグレーション . . . . .	67
32 ビット Windows プラットフォームから 64 ビット Windows プラットフォームへのマイグレーション . . . . .	72
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 . . . . .	74
Network Deployment 環境のマイグレーション . . . . .	80
デプロイメント・マネージャーのマイグレーション . . . . .	81
非クラスター管理対象ノードのマイグレーション . . . . .	93
クラスターのマイグレーション . . . . .	108
ネットワーク・デプロイメント環境でのビジネス・ルール・マネージャーのマイグレーション . . . . .	133
デプロイメント・ターゲットへのビジネス・ルール・マネージャーのマイグレーション . . . . .	134
マイグレーションの検査 . . . . .	135
環境のロールバック . . . . .	137
デプロイメント・セルのロールバック . . . . .	137
管理対象ノードのロールバック . . . . .	140

WebSphere Process Server の事後マイグレーション・タスク . . . . .	143
Business Process Choreographer の事後マイグレーション・タスク . . . . .	145
WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク . . . . .	149
Cloudscape データベースのマイグレーション . . . . .	150
Cloudscape v10.1.x 自動マイグレーションの検査 . . . . .	153
手動による Cloudscape のアップグレード . . . . .	157
UDDI レジストリーのマイグレーション . . . . .	162
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 . . . . .	165

## 第 2 章 その他の WebSphere 製品からのマイグレーション . . . . . 177

WebSphere InterChange Server または WebSphere Business Integration Server Expressからのマイグレーション . . . . .	178
事前マイグレーションの考慮事項 . . . . .	178
reposMigrate コマンドを使用した WebSphere InterChange Server または WebSphere Business Integration Server Express の成果物のマイグレーション . . . . .	186
事後マイグレーションの考慮事項 . . . . .	189
WebSphere Business Integration データ・ハンドラーのサポート . . . . .	205
サポートされる WebSphere InterChange Server または WebSphere Business Integration Server Express API . . . . .	207
WebSphere InterChange Server または WebSphere Business Integration Server Express からマイグレーションする場合の制限事項 . . . . .	230
WebSphere InterChange Server または WebSphere Business Integration Server Express からのマイグレーションのトラブルシューティング . . . . .	231
WebSphere Studio Application Developer Integration Edition からのマイグレーション . . . . .	234
WebSphere MQ Workflow からのマイグレーション . . . . .	235

## 第 3 章 使用すべきでないフィーチャー 237

## 第 4 章 マイグレーションのトラブルシューティング . . . . . 253

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 . . . . .	253
WebSphere InterChange Server または WebSphere Business Integration Server Express からのマイグレーションのトラブルシューティング . . . . .	264

サポートされる WebSphere InterChange Server  
または WebSphere Business Integration Server  
Express API に対するロギングとトレースの有効  
化 . . . . . 265  
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 . . . . . 266

特記事項. . . . . **269**

---

# 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 6.2 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 6.1.x が Microsoft® Windows® XP で稼働している場合は、その WebSphere Process Server 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 6.2 にデータをマイグレーションできます。しかし、Windows XP で稼働する WebSphere Process Server バージョン 6.1.x から、AIX® で稼働する WebSphere Process Server バージョン 6.2 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。(このマイグレーションについては、74 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 6.2 にマイグレーションすることができます。

表 2. サポートされるバージョン間マイグレーション・シナリオ

現在のインストール・バージョン	新バージョン
WebSphere Process Server バージョン 6.0.2.x	WebSphere Process Server バージョン 6.2
WebSphere Process Server バージョン 6.1.x	WebSphere Process Server バージョン 6.2
WebSphere Enterprise Service Bus バージョン 6.0.2.x	WebSphere Process Server バージョン 6.2
WebSphere Enterprise Service Bus バージョン 6.1.x	WebSphere Process Server バージョン 6.2

注: バージョン 6.0.2.x より前のバージョンの WebSphere Process Server からマイグレーションする場合は、最初にバージョン 6.0.2.x にマイグレーションしてから、その後でバージョン 6.2 に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 6.0.2.x へのマイグレーションについての詳細は、WebSphere Process Server バージョン 6.0.2.x インフォメーション・センターを参照してください。

## 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、WebSphere Process Server または WebSphere Application Server 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持されます。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 バージョン 6.1.x またはバージョン 6.0.2.x からバージョン 6.2 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。

## プロファイル・タイプ: WebSphere Process Server、WebSphere Enterprise Service Bus、および WebSphere Application Server プロファイル

WebSphere Process Server、WebSphere Enterprise Service Bus、および WebSphere Application Server プロファイルの 3 つのタイプのプロファイルを WebSphere Process Server バージョン 6.2 にマイグレーションすることができます。

**重要:** 以下の定義は、マイグレーションされるプロファイルを作成するのに使用された可能性のあるツールとしてプロファイル管理ツールに触れていますが、マイグ

レーション中にプロファイル管理ツールを使用して新規ターゲット・プロファイルを作成することはできません。マイグレーション中、新規ターゲット・プロファイルを作成するのに使用するのはマイグレーション・ウィザードまたはコマンド行ツールのみです。

### WebSphere Process Server プロファイル

WebSphere Process Server プロファイルは、「default.wbiserver」、「dmgr.wbiserver」、または「managed.wbiserver」のプロファイル・テンプレートの 1 つを使用して作成されたプロファイルです。プロファイル管理ツールを使用するとき、これは、「環境の選択」ページの「**WebSphere Process Server**」を選択することを意味します。

### WebSphere Enterprise Service Bus

WebSphere Enterprise Service Bus プロファイルは、「default.esbserver」、「dmgr.esbserver」、または「managed.esbserver」のプロファイル・テンプレートの 1 つを使用して作成されたプロファイルです。プロファイル管理ツールを使用するとき、これは、「環境の選択」ページの「**WebSphere Enterprise Service Bus**」を選択することを意味します。

### WebSphere Application Server プロファイル

WebSphere Application Server プロファイルは、「default」、「dmgr」、または「managed」のプロファイル・テンプレートの 1 つを使用して作成されたプロファイルです。プロファイル管理ツールを使用するとき、これは、「環境の選択」ページで「**Application Server**」、「**Deployment Manager**」、または「**カスタム・プロファイル**」のうちから 1 つを選択することを意味します。

注: WebSphere Process Server ネットワーク・デプロイメント環境では、Deployment Manager は WebSphere Process Server プロファイルで作成する必要があります。

注: WebSphere Process Server クライアント・インストールで作成できるのは、WebSphere Application Server プロファイルのみです。

表 3. マイグレーション中にサポートされるプロファイル

プロファイル・タイプ	プロファイルが作成された環境	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マイグレーション可能か。	コマンド行ツールを使用してプロファイルをマイグレーション可能か。
WebSphere Process Server	WebSphere Process Server - サーバー・インストール 注: これには、WebSphere Process Server クライアント・インストール・オプションが含まれます。	はい	はい

表 3. マイグレーション中にサポートされるプロファイル (続き)

プロファイル・タイプ	プロファイルが作成された環境	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マイグレーション可能か。	コマンド行ツールを使用してプロファイルをマイグレーション可能か。
WebSphere Enterprise Service Bus	WebSphere Process Server - サーバー・インストール 注: これには、WebSphere Process Server クライアント・インストール・オプションが含まれます。	はい	はい
WebSphere Application Server	WebSphere Process Server - クライアント・インストール	いいえ	はい
WebSphere Application Server	WebSphere Process Server - サーバー・インストール	いいえ	はい

#### 関連概念

10 ページの『ターゲット・プロファイルの考慮事項』

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、ここに示すターゲット・プロファイルの考慮事項について知っておく必要があります。



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。

#### 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。

## WebSphere Process Server に関する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 6.2 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、および考慮事項が適用されます。

- 『WebSphere Process Server インストール要件』
- 『拡張』
- 『バックアップ・ディレクトリー』
- 6 ページの『セル構成』
- 6 ページの『クラスター』
- 6 ページの『データベース』
- 6 ページの『HTTP トランスポート』
- 6 ページの『Java/JDK (Java Development Kit)』
- 7 ページの『JNI (Java Native Interface)』
- 7 ページの『マイグレーション・ツール』
- 8 ページの『プロファイル』
- 8 ページの『環境のロールバック』
- 8 ページの『ストレージ』

## WebSphere Process Server インストール要件

- WebSphere Process Server のバージョン 6.2 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、ほかの 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。

## 拡張

- バージョン 6.0.2.x またはバージョン 6.1.x プロファイルをバージョン 6.2 プロファイルにマイグレーションできるのは、それらが同じ拡張レベルにある場合に限られます。
- 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 用に拡張されていれば、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。

## バックアップ・ディレクトリー

- 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。このディレクトリーで使用可能なスペースのサイズは、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である必要があります。前のプロファイルは、WebSphere Process Server または WebSphere Application Server プロファイルです。

注: バージョン 6.0.2.x からマイグレーションするとき、WebSphere Process Server の前のインストール済み環境にあった既存のプロファイルはすべてバックアップされます。しかし、バージョン 6.1.x からマイグレーションするときは、一度にバックアップされるプロファイルは 1 つだけです。

## セル構成

- デプロイメント・マネージャーまたは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 6.2のセル名がバージョン 6.1.x および 6.0.2.x のセル名と一致する必要があります。

新規のセル名で WebSphere Process Server または WebSphere Application Server 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。

## クラスター

- クラスターのメンバーで、WebSphere Process Server の異なるバージョン (6.0.2.x、6.1.x、6.2) を実行することはできません。異なるバージョンを実行するサーバーが含まれているクラスターを構成した場合は、最初のバージョン 6.2 のクラスター・メンバーを開始する前に、WebSphere Process Server の以前のバージョンを実行するすべてのメンバーを停止する必要があります。また、一度バージョン 6.2 のクラスター・メンバーを開始した後は、そのクラスター内でバージョン 6.1.x および 6.0.2.x のクラスター・メンバーを開始しないでください。

## データベース

- Cloudscape® または Derby 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。

## HTTP トランスポート

WebSphere Process Server バージョン 6.0.2.x 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・トランスポート・チェーンに変換されます。

注: これは、バージョン 6.0.2.x からのマイグレーションにのみ適用します。

バージョン 6.2 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

## Java/JDK (Java™ Development Kit)

- バージョン 6.0.2.x からマイグレーションする場合、および 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、したがって WebSphere Process Server バージョン 6.2 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調べてください。

注: このタスクは、バージョン 6.1.x からマイグレーションする場合には不要です。

『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。

## JNI (Java Native Interface)

- **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 6.2 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 6.2 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 6.2 の 64 ビット環境では実行できません。

## マイグレーション・ツール

注: これは、バージョン 6.0.2.x からのマイグレーションにのみ適用します。

- マイグレーション・ツールを使用して WebSphere Process Server バージョン 6.2 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。
  - WebSphere Application Server バージョン 6.0.2.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、WebSphere Process Server バージョン 6.2 セキュリティーが適切に設定されているか確認します。

『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。

WebSphere Application Server 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。

150 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。

## プロファイル

- ターゲット・プロファイルの作成について詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。
- プロファイル・タイプについて詳しくは、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。

## 環境のロールバック

- ノードをWebSphere Process Server バージョン 6.2 にマイグレーションしてから、バージョン 6.1.x および 6.0.2.x に戻す必要があることが明らかになった場合は、137 ページの『環境のロールバック』を参照してください。

## ストレージ

- バージョン 6.2 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 WebSphere Process Server または WebSphere Application Server 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、WBIPreUpgrade コマンド行ユーティリティー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 WebSphere Process Server または WebSphere Application Server 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、WBIPostUpgrade コマンド行ユーティリティーを参照してください。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

### 関連タスク

137 ページの『環境のロールバック』

WebSphere Process Server バージョン 6.2 環境へのマイグレーション後に、バージョン 6.1.x および 6.0.2.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。

150 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 6.2 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



WBIPreUpgrade コマンド行ユーティリティー

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。



WBIPostUpgrade コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された `backupDirectory` に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

### 関連情報



ほかの 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 6.2 のインストール済み環境は、同一システム上で任意のバージョンの WebSphere Enterprise Service Bus、WebSphere Process Server、および精選された WebSphere 製品と共存できます。

-  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

## ターゲット・プロファイルの考慮事項

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、ここに示すターゲット・プロファイルの考慮事項について知っておく必要があります。

マイグレーション用に正しいターゲット・プロファイルを作成することは、マイグレーション・プロセスの重要な部分です。プロファイルをマイグレーションするとき、新規のターゲット・プロファイルを作成することを推奨します。マイグレーション・ターゲット・プロファイルを作成するとき、ノード名、セル名、および拡張レベルが WebSphere Process Server バージョン 6.1.x および 6.0.2.x プロファイル、または WebSphere Application Server 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) であるプロファイルを作成する必要があります。

**ヒント:** あらかじめターゲット・プロファイルを作成済みの場合、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 6.2 ノードに行わないでください。これらの変更はマイグレーション中に失われます。代わりに、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います。そうしない場合、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、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、マイグレーション後に手動で再び行う必要があります。

マイグレーション・ウィザードおよびコマンド行ツールには共に組み込みオプションがあり、容易に新規ターゲット・プロファイルを作成できます。

## マイグレーション・ウィザード

マイグレーション・ウィザードを使用する場合、マイグレーション元のプロファイルがどのようなタイプであれ、マイグレーション・ウィザード・パネルの「**新規プロファイルを作成する**」オプションを選択して、WebSphere Process Server バージ

ョン 6.2 に新規プロファイルをマイグレーション用に作成します。このオプションにより、マイグレーションのための正しい構成設定の新規プロファイルが作成できます。

**注:** マイグレーション・ウィザードは、WebSphere Application Server プロファイルのマイグレーションをサポートしていません。WebSphere Process Server のフル・インストール、または WebSphere Process Server のクライアント・インストールでは、WebSphere Application Server プロファイルをマイグレーションする際、コマンド行ツールを使用する必要があります。

**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 i5/OS プラットフォームまたは 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、コマンド行ツールを使用します。

## コマンド行ツール

マイグレーション・コマンド行ツールを使用する場合、マイグレーション元のプロファイルがどのようなタイプであれ、マイグレーション目的で WebSphere Process Server バージョン 6.2 のインストール済み環境に新規プロファイルを作成するために `WBIPostUpgrade` を呼び出すとき、**-createTargetProfile** を使用します。マイグレーションのコマンド行ツールの完全な説明は、`WBIPreUpgrade` コマンド行ユーティリティー および `WBIPostUpgrade` コマンド行ユーティリティー を参照してください。

**注:** マイグレーション・フレームワークの外部にマイグレーションのターゲット・プロファイルを作成する場合、オプションで `manageprofiles` コマンドを使用することができます。一例として、マイグレーション・タスクを実行しているときに、ターゲット・プロファイルを準備する場合です。

**注:** i5/OS i5/OS プラットフォームでマイグレーションする場合、以下の考慮事項を知っておく必要があります。

- ターゲットのプロファイル名は、マイグレーションされるソース・プロファイルのプロファイル名に一致する必要があります。
- `WBIPostUpgrade` スクリプトで `-createTargetProfile` を指定する場合、ターゲット・プロファイルがマイグレーションされた後、`install_root/bin/enableJVM` スクリプトを呼び出してターゲット・プロファイルの `JVM` タイプを必要な値に設定することができます。`WBIPostUpgrade` スクリプトにより、ターゲット・プロファイルがマイグレーション・プロセス用ソース・プロファイルと同じ `JVM` タイプで確実に生成されます。
- `manageprofiles` を使用して手動でターゲット・プロファイルを生成する場合、ターゲット・プロファイルの `JVM` タイプがソース・プロファイルの `JVM` タイプと一致することを確認する必要があります。これは、`WBIPostUpgrade` スクリプトを起動する前に、`install_root/bin/enableJVM` スクリプトを起動してターゲット・プロファイルの `JVM` タイプを設定することで可能です。
- `enableJVM` コマンドの `[-profile target_profile_name]` オプションを使用して、製品インストールのすべてのプロファイルが指定したタイプに切り替わるのを防ぐ必要があります。

## コマンド行ツールを使用して作成したプロファイルの例

注: 以下の例では、作成するプロファイルのタイプにより、-templatePath に以下のパラメーターの 1 つを使用します。

- **WebSphere Process Server** スタンドアロン・サーバー: <WAS\_HOME>/profileTemplates/default.wbiserver
- **WebSphere Process Server** デプロイメント・マネージャー: <WAS\_HOME>/profileTemplates/dmgr.wbiserver
- **WebSphere Process Server** カスタム・ノード: <WAS\_HOME>/profileTemplates/managed.wbiserver
- **WebSphere Application Server** スタンドアロン・サーバー: <WAS\_HOME>/profileTemplates/default
- **WebSphere Application Server** カスタム・ノード: <WAS\_HOME>/profileTemplates/managed

### 例: WebSphere Process Server プロファイルの作成

- スタンドアロン・サーバーおよびデプロイメント・マネージャー:  
manageprofiles -create -profileName <profname> -templatePath <example: <WAS-HOME>/profileTemplates/(default)(dmgr).wbiserver> -profilePath <example: /QIBM/UserData/<WAS\_HOME>/profiles/<profname>> -winserviceCheck false -defaultPorts -createDefaultProfileForMigration true -dbDelayConfig true -omitAction defaultAppDeployAndConfig
- カスタム・ノード: manageprofiles -create -profileName <profname> -templatePath <example: <WAS-HOME>/profileTemplates/managed.wbiserver> -nodeName <nodename> -cellName <cellname> -hostName <hostname> -defaultPorts -createDefaultProfileForMigration true -dbDelayConfig true

### 例: WebSphere Application Server プロファイルの作成

- スタンドアロン・サーバーおよびデプロイメント・マネージャー:  
manageprofiles -create -profileName <profname> -templatePath <WAS\_HOME>/profileTemplates/default -profilePath <WAS\_HOME>/profiles/<profname> -cellName <cellname> -winserviceCheck false -defaultPorts -hostName <hostname> -omitAction defaultAppDeployAndConfig -nodeName <nodename> -enableAdminSecurity false
- カスタム・ノード: manageprofiles -create -profileName <profname> -templatePath <WAS\_HOME>/profileTemplates/managed -profilePath <WAS\_HOME>/profiles/<profname> -cellname <cellname> -hostName <hostname> -nodeName <nodename> -federateLater true

#### 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。

## Business Process Choreographer に関する事前マイグレーションの考慮事項

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

### データベース: バックアップ

データのマイグレーションに失敗した場合、バックアップからデータベースを復元する必要があるため、Business Process Choreographer データベースをアップグレードする前に、データベース全体をバックアップします。

### データベース: アップグレードとマイグレーションのスクリプト

Business Process Choreographer データベースをアップグレードするには、2 つまたは 3 つのスクリプトを以下の順序で実行する必要があります。

1. データベースのタイプとバージョンによっては、テーブル・スペースをアップグレードするスクリプトを実行する必要があります。
2. すべてのデータベースについて、スキーマをアップグレードするスクリプトを実行します。
3. すべてのデータベースについて、ランタイム・データを新しいスキーマにマイグレーションするスクリプトを実行する必要があります。また、z/OS<sup>®</sup> 用の DB2<sup>®</sup> UDB と DB2 の場合は、スクリプトによってデータが新しいテーブル・スペースにマイグレーションされます。

実動システムについて、手動で新しいデータベース・スキーマにアップグレードします。データベース・サーバーは稼働させておく必要がありますが、Business Process Choreographer を構成する WebSphere Process Server サーバーは、データのマイグレーションが完了するまで起動しないでください。

### データベース: 許可

各スクリプトで異なるデータベース権限が必要になるため、1 つのユーザー ID だけですべてのスクリプトを実行できるかどうか、または、データベース管理者がいずれかのスクリプトを実行する必要があるかどうかを確認します。

Linux<sup>®</sup>、UNIX<sup>®</sup>、および Windows 用の DB2 に対して `upgradeTablespaces SQL` スクリプトを実行するには、次の権限が必要です。

```
CREATE BUFFERPOOL および CREATE TABLESPACE
```

z/OS 用の DB2 に対して `upgradeTablespaces SQL` スクリプトを実行するには、次の権限が必要です。

```
CREATE TABLESPACE
```

`upgradeSchema SQL` スクリプトを実行するには、次の権限が必要です。

```
すべてのデータベース・タイプについて、CREATE TABLE、ALTER TABLE、DROP INDEX、CREATE INDEX、CREATE VIEW、および DROP VIEW が実行可能になっている必要があります。
```

`migrateDB.py` スクリプトを実行するには、次の権限が必要です。

- すべてのデータベース・タイプについて、SELECT、INSERT、UPDATE、CREATE VIEW、および DROP VIEW が実行可能になっている必要があります。
- i5/OS® 用の DB2® Universal Database™ を使用している場合は、\*ALLOBJ と \*SECADM の特殊権限を持つユーザー・プロファイルを使用していることを確認してください。
- Linux、UNIX、Windows、または z/OS 用の DB2 を使用している場合は、テーブル・スペースのマイグレーションで、CREATE TABLE、RENAME TABLE、CREATE INDEX、DROP INDEX、CREATE VIEW、および DROP VIEW の各権限がユーザー ID に設定されている必要があります。

マテリアライズド・ビューを構成した場合は、DROP TABLE 権限と CREATE TABLE 権限も必要になります。

## データ・マイグレーション: 時間要件とチューニング・オプション

データ量とデータベース・サーバーの能力によっては、データ・マイグレーションのステップに数時間かかることがあります (データベースのバックアップとデータベース・スキーマのアップグレードに必要な時間は除く)。所要時間を短縮するチューニング・オプションなど、データ・マイグレーションの実行に関する最新情報は、『技術情報 21327385』を参照してください。

## データ・マイグレーション: ストレージ要件

データ・マイグレーションによって古いテーブルから新しいテーブルにデータがマージされるため、一時的に余分なストレージが必要になります。念のため、データ全体のコピーの 2 倍のデータ量を保存できるだけのストレージ・スペースがデータベースにあることを確認してください。サーバーが起動可能な状態であることを確認したら、必要に応じて、どのテーブルを削除してストレージに空き容量を確保するかを確認します。

## データ・マイグレーション: マテリアライズド・ビュー

名前付きマテリアライズド・ビューに対してカスタム・テーブル定義ファイルを以前に使用していた場合、このビューはデータ・マイグレーション・スクリプトによってドロップされます。WebSphere Process Server で名前付きマテリアライズド・ビューを再作成できるのは、**customTableDefinition** がアクセスできるカスタム・テーブル定義 XML ファイルを指している場合だけです。WebSphere Process Server で名前付きマテリアライズド・ビューを再作成できるかどうかを確認するには、以下の操作を実行します。

1. WebSphere Process Server が稼働していることを確認します。
2. 管理コンソールで、「サーバー」 → 「アプリケーション・サーバー」 → 「*serverName*」または、「クラスター」 → 「*clusterName*」をクリックし、「ビジネス・インテグレーション」で「Business Process Choreographer」を展開して「Business Flow Manager」 → 「カスタム・プロパティ」をクリックします。

3. ビジネス・コンテナのカスタム・プロパティのリストで、**customTableDefinition** という名前のエントリーを探します。このエントリーにより、*path/customData.xml* など、カスタム・テーブル定義ファイルのファイル・システム・ロケーションが指定されます。
  4. XML ファイルが存在することを確認します。
    - スタンドアロン環境の場合、サーバー・ノード上に存在します。
    - クラスター環境の場合、クラスター・メンバーをホストする各ノード上に存在します。
- 注: XML ファイルのファイル・システム・ロケーションに *\${WAS\_INSTALL\_ROOT}* などの WebSphere 変数が定義されている場合は、マイグレーション中にこの変数の値が変更されることがあります。場合によっては、マイグレーションされたサーバーやクラスターを起動する前に、XML ファイルを新しいロケーションにコピーする必要があります。
5. WebSphere Process Server から XML ファイルにアクセスできることを確認します。

### マイグレーション中: 混合セルの制約事項

WebSphere Process Server をバージョン 6.1.x からバージョン 6.2 にマイグレーションする際に、新しいレベルと事前マイグレーションのレベルでセル内のノードが同時に実行されている場合は、次の点に注意してください。

- デプロイメント・マネージャーが既に最新バージョンにマイグレーションされている場合は、セル内のノードに対して、以下に示す事前マイグレーション・レベルのアクションを実行することはできません。
  - Business Process Choreographer の構成。
  - ビジネス・プロセスやヒューマン・タスクなどが定義されたアプリケーションのインストール、更新、またはアンインストール。
- Business Process Choreographer が構成されている最新バージョンのクラスターを使用している場合は、事前マイグレーション・レベルのノード上で新しいクラスター・メンバーを作成する必要はありません。

### マイグレーション後: Query API によって返される結果において発生する可能性がある影響

両方の作業項目テーブルのマージ後、WORK\_ITEM\_T テーブルには新しいエントリーが格納されます。すべての新しいエントリーには、固有の作業項目 ID (WIID) も設定されます。したがって、Query API に対する一部の照会で、異なる結果が返されます。例えば、WORK\_ITEM ビュー内の異なる WIID のカウント数について、実際よりも高い値が返されることがあります。ただし、WORK\_ITEM ビュー内のエントリーの合計数には影響はありません。

#### 関連概念

4 ページの『WebSphere Process Server に関する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

43 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 が構成されたサーバーまたはクラスターのマイグレーション後、Business Process Choreographer データベースのスキーマを手動でアップグレードし、サーバーまたはクラスター・メンバーを始動する前にデータをマイグレーションする必要があります。

45 ページの『Business Process Choreographer データベース・スキーマのアップグレード』

Business Process Choreographer が構成されているサーバーまたはクラスターをマイグレーションまたはアップグレードした後、関連する 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。

49 ページの『Business Process Choreographer ランタイム・データのマイグレーション』

Business Process Choreographer が構成されたサーバーまたはクラスターをマイグレーションまたはアップグレードした後、サーバーまたはクラスター・メンバーを始動する前にデータをマイグレーションする必要があります。

145 ページの『Business Process Choreographer の事後マイグレーション・タスク』

サーバーまたはクラスターで Business Process Choreographer を実行する場合、追加のタスクをいくつか実行してからサーバーやクラスターを起動する必要があります。

165 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。

## WebSphere が提供するビジネス・スペース に関する事前マイグレーションの考慮事項

WebSphere Process Server をバージョン 6.1.2 からバージョン 6.2 にマイグレーションする前に、ここで説明するビジネス・スペースの考慮事項に注意してください。

WebSphere Process Server をバージョン6.1.2 からバージョン 6.2 にマイグレーションする場合、以下に示す規則、制限事項、考慮事項がビジネス・スペースに対して適用されます。

17 ページの『混合ノード』

17 ページの『ターゲット・プロファイル構成』

17 ページの『ビジネス・スペースで構成された複数の製品のマイグレーション』

## 混合ノード

ビジネス・スペースで構成された 1 つ以上のクラスターの混合ノード・マイグレーションは、ビジネス・スペースで構成されたクラスターを最後にマイグレーションする場合のみサポートされます。混合ノード・マイグレーションの場合、新しいバージョンのデプロイメント・マネージャーにより、WebSphere Process Server の古いバージョンと新しいバージョンの両方をマイグレーション中に管理することができます。例えば、バージョン 6.2 のデプロイメント・マネージャーにより、バージョン 6.1.2 とバージョン 6.2 の両方のノードをマイグレーション後に管理することができます。以前のデプロイメント・マネージャーの管理対象ノードは、バージョン 6.2 のデプロイメント・マネージャーでバージョン 6.1.2 の管理対象ノードとして稼働します。

ビジネス・スペースの最初のバージョンは 6.1.2 です。ビジネス・スペースをバージョン 6.1.2 からバージョン 6.2 へ段階的にマイグレーションする場合は、ビジネス・スペースをマイグレーションする前に、製品のアプリケーションとバイナリーをマイグレーションする必要があります。この順序は非常に重要です。これは、製品に同梱されているウィジェット・エンドポイント・コードは ビジネス・スペースのバージョン 6.1.2 では機能しますが、ビジネス・スペースバージョン 6.2 では、バージョン 6.1.2 の製品に同梱されているウィジェット・エンドポイント・コードが機能しないためです。マイグレーションされた製品を共存させて使用するには、ビジネス・スペース をマイグレーションする前に、その他の製品をマイグレーションする必要があります。

## ターゲット・プロファイル構成

ビジネス・スペースが構成されている状態で WebSphere Process Server のバージョン 6.1.2 からマイグレーションする場合は、ターゲット・プロファイルを作成します。その際、このプロファイルはビジネス・スペース用には構成しません。ビジネス・スペースに対してターゲット・プロファイルを構成しないのは、ビジネス・スペースのデータベースが新しく作成されないようにするためです。『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク (Postmigration tasks for Business Space powered by WebSphere)』で説明されているステップを実行する際に、ビジネス・スペースのバージョン 6.2 で使用できるように、ビジネス・スペースのバージョン 6.1.2 のデータベース表を更新します。

ビジネス・スペース用に構成せずにターゲット・プロファイルを作成するには、マイグレーション・ウィザードで「新規プロファイルを作成する」を選択するか、WBIPostUpgrade コマンドを呼び出すときにパラメーター `-createTargetProfile` を使用します。

## ビジネス・スペースで構成された複数の製品のマイグレーション

WebSphere Process Server のほかに WebSphere Business Monitor や WebSphere Business Modeler Publishing Server などの製品もマイグレーションする場合は、追加のステップが必要になることがあります。詳しくは、各製品のマイグレーションに関する資料を参照してください。

## ビジネス・スペースのターゲット・プロファイルの考慮事項

ビジネス・スペースを使用している場合、WebSphere Process Server プロファイルを構成して、マイグレーション後にビジネス・スペースを処理する必要があります。ビジネス・スペースの複数インスタンスを実行している場合、WebSphere Process Server を処理するビジネス・スペースの各インスタンスのプロファイルを構成する必要があります。

ビジネス・スペースを追加するのに必要なアクションは、マイグレーション元の環境によって異なります。この環境は、以下の表にソース・プロファイルとしてリストされています。ビジネス・スペースを持つ WebSphere Process Server の最初のリリースは、バージョン 6.1.2 です。

マイグレーション後、ビジネス・スペースを処理するために WebSphere Process Server プロファイルを構成するのに必要なアクションを以下の表に詳述します。取る必要のあるアクションは、マイグレーション元の WebSphere Process Server のバージョン、およびソース・プロファイルが構成済みのビジネス・スペースを持っているかどうかによって変わります。

表 4. ビジネス・スペース用にプロファイルを構成するのに必要なアクション

ソース・プロファイル	ビジネス・スペース用に新規ターゲット・プロファイルを構成するのに必要なアクション
<ul style="list-style-type: none"><li>バージョン 6.0.2.x (このバージョンにはビジネス・スペースは存在しません)</li><li>バージョン 6.1.0.x (このバージョンにはビジネス・スペースは存在しません)</li><li>バージョン 6.1.2.x (構成済みのビジネス・スペースはなし)</li></ul>	<ol style="list-style-type: none"><li>マイグレーション中、ビジネス・スペース用に構成されないようにターゲット・プロファイルを作成します。ターゲット・プロファイルについて詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。</li><li>マイグレーションに続き、WebSphere Application Server 管理コンソールを使用してビジネス・スペース用マイグレーション済みプロファイルを構成します。これを行って、ビジネス・スペース・アプリケーションを追加し、ビジネス・スペースのバージョン 6.2 データベース・アプリケーションを作成し、ビジネス・スペースのバージョン 6.2 データベースを作成します。詳しくは、『管理コンソールを使用したビジネス・スペースの構成』を参照してください。</li><li>必要に応じて、手動でエンドポイントを構成します。</li></ol>

表 4. ビジネス・スペース用にプロファイルを構成するのに必要なアクション (続き)

ソース・プロファイル	ビジネス・スペース用に新規ターゲット・プロファイルを構成するのに必要なアクション
バージョン 6.1.2.x (構成済みビジネス・スペースあり)	<ol style="list-style-type: none"> <li>1. マイグレーション中、ビジネス・スペース用に構成されないようにターゲット・プロファイルを作成します。ターゲット・プロファイルについて詳しくは、10ページの『ターゲット・プロファイルの考慮事項』を参照してください。</li> <li>2. マイグレーションに続き、マイグレーション後タスクを実行して、ビジネス・スペースバージョン 6.2 による使用に備えてビジネス・スペースバージョン 6.1.2 データベースを更新し、エンドポイントを構成します。エンドポイントの構成に関する情報は、『管理コンソールでビジネス・スペース・ウィジェットのエンドポイントを使用可能にする』、および『リモート・エンドポイントに対してビジネス・スペース・ウィジェットを手動で使用可能にする』を参照してください。</li> </ol>

#### 関連タスク



プロファイル管理ツールを使用したビジネス・スペースの構成



管理コンソールを使用したビジネス・スペースの構成

## 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・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。

**注:** i5/OS オペレーティング・システムを使用する場合は、マイグレーション・ウィザードを使用できません。マイグレーション・コマンドを使用する必要があります。

注: マイグレーション・ウィザードは WebSphere Process Server プロファイルのみをサポートします。 WebSphere Application Server プロファイルがある場合、マイグレーション・コマンドを使用する必要があります。

## 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、クラスターを含む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、一部の状況ではデータベースを手動でアップグレードする必要があります。

WebSphere Process Server で提供された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るのは、以下の状況です。

- サーバー・プロセスが十分な許可を持たない場合 (つまり、該当するデータベースに対して十分な許可を持つ正しいユーザー ID を使用して構成されていない)
- デフォルトでないテーブル・スペースを使用した場合
- サーバーが Business Process Choreographer を実行する場合

### 関連タスク

39 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

#### WBIPreUpgrade コマンド行ユーティリティー

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。

#### WBIPostUpgrade コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された *backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

#### 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。

## 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。

**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 **i5/OS プラットフォーム** または **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。

**注:** マイグレーション・ウィザードは WebSphere Process Server プロファイルのみをサポートします。WebSphere Application Server プロファイルがある場合、マイグレーション・コマンドを使用する必要があります。

## 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、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・ディレクトリーのサブディレクトリー *profiles/profile\_name* および *websphere\_backup* に保存されます。

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選択したプロファイル環境を WebSphere Process Server の新しいプロファイル環境に復元します。

## 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。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
`wbi_migration.sh`

- **Windows** **Windows** プラットフォーム: `wbi_migration.bat`

**注:** マイグレーション・ウィザードを起動するとき、デフォルトのトレース設定 (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) をオプションで変更できます。デフォルトのトレース設定では、特定のクラスでのみトレースが有効になりますが、完全トレースを有効にするか、またはすべてのトレースを無効にするようにデフォルトを変更できます。

- 完全トレースを有効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
`wbi_migration.sh -W -migrationPanel.traceString="*=all=enabled"`

- **Windows** **Windows** プラットフォーム: `wbi_migration.bat -W -migrationPanel.traceString="*=all=enabled"`

- すべてのトレースを無効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
`wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`

- **Windows** **Windows** プラットフォーム: `wbi_migration.bat -W -migrationPanel.traceString="*=all=disabled"`

## 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** 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。すべてのマイグレーション・シナリオについてソース・プロファイルおよびターゲット・プロファイルのセル名が一致する必要があります。

### 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、`WBIPreUpgrade` コマンド行ユーティリティー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、`WBIPreUpgrade` コマンド行ユーティリティを参照してください。

### 管理セキュリティ・ユーザー名 (管理セキュリティが構成されている場合に必須)

`-username` パラメーターについては、`WBIPostUpgrade` コマンド行ユーティリティを参照してください。

### 管理セキュリティ・パスワード (管理セキュリティが構成されている場合に必須)

`-password` パラメーターについては、`WBIPostUpgrade` コマンド行ユーティリティ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、`WBIPostUpgrade` コマンド行ユーティリティ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、`WBIPostUpgrade` コマンド行ユーティリティ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、`WBIPostUpgrade` コマンド行ユーティリティを参照してください。

注: これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

### 関連資料



#### WBIPreUpgrade コマンド行ユーティリティ

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。



#### WBIPostUpgrade コマンド行ユーティリティ

WebSphere Process Server の `WBIPostUpgrade` コマンドを使用して、指定された `backupDirectory` に `WBIPreUpgrade` コマンドによって保存されたプロファイル構成を取得します。

### マイグレーション・ウィザードの実行:

AIX、HP-UX、Linux、Solaris、または Windows システムでマイグレーション・ウィザードを実行して、WebSphere Process Server をマイグレーションします。

### 始める前に

マイグレーション・ウィザードを起動するための手順を実行する前に行う必要がある手順を完了したことを確認してください。手順は、マイグレーションするのがスタンドアロン・サーバー、デプロイメント・マネージャー、非クラスター管理対象ノード、クラスターのいずれであるかによって異なります。

## このタスクについて

マイグレーション・ウィザードの実行は、WebSphere Process Server をバージョン 6.1.x および 6.0.2.x からバージョン 6.2 にマイグレーションするための長い手順の中の 1 ステップに過ぎません。

**注:** WebSphere Process Server 環境で作成された WebSphere Application Server プロファイルがある場合は、WebSphere Process Server のコマンド行ツールを使用してマイグレーションする必要があります。WebSphere Process Server 環境で作成された WebSphere Process Server プロファイルがある場合は、マイグレーション・ウィザードまたはコマンド行ツールのいずれかを使用してマイグレーションできます。

**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 **i5/OS プラットフォーム** または **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。

## 手順

1. 「WebSphere Process Server のマイグレーション・ウィザードへようこそ」画面で「ようこそ」パネルを読み、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. 「検出された WebSphere Process Server のバージョン」画面で、マイグレーション元となる旧バージョンの 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. 「ソース・プロファイルの選択」画面で、マイグレーションする旧バージョンの WebSphere Process Server のプロファイルを選択して「次へ」をクリックします。
4. 「ターゲット・プロファイルの選択」画面で、「**新規プロファイルを作成する**」を選択し、マイグレーション用の新しい空のターゲット・プロファイルを作成する必要があります。オプションで、有効なプロファイルのリストからバージョン 6.2 インストール環境のプロファイルを選択できますが、マイグレーション用に新しいターゲット・プロファイルを使用することを強くお勧めします。マイグレーション用のターゲット・プロファイルの作成については、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。
  - 「**新規プロファイルを作成する**」を選択した場合は、プロファイルの名前を入力し、ホスト名 (プロファイルがあるシステムの名前) は変更せずに、「次へ」をクリックします。

**注:** ホスト名は変更しないことをお勧めします。ただし、なんらかの理由でホスト名を変更した場合は、マイグレーション後に、ホスト名が構成されている他の場所でホスト名を手動で更新する必要があります。

- ターゲット・プロファイルのバックアップを検討してから「次へ」をクリックしてください。ターゲット・プロファイルの構成をバックアップするかどうかを決定する場合は、以下の内容を考慮してください。

- 既存のプロファイルを選択した場合は、ターゲット・プロファイルの構成のバックアップが必要になることがあります。backupConfig コマンドを使用してターゲット・プロファイルの構成をバックアップしてから、処理を進めてください。
  - 管理対象ノードをマイグレーションする場合は、マイグレーションによってデプロイメント・マネージャーの構成が変更されます。backupConfig コマンドを使用してデプロイメント・マネージャーの構成をバックアップしてから、処理を進めてください。
5. 「マイグレーション・バックアップ・ディレクトリー」画面で、旧バージョンからの構成のバックアップ・コピーを格納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ます。マイグレーション元の WebSphere Process Server のバージョンに応じて、以下のオプションが画面に表示されます。

• バージョン 6.0.2.x からのマイグレーション:

バージョン 6.0.2.x からバージョン 6.2 へのマイグレーション中に、WebSphere Process Server バージョン 6.0.2.x インストール環境の下に作成されたすべてのプロファイルがバックアップ操作によってバックアップされます。

- このプロファイルが WebSphere Process Server インストール済み環境でマイグレーションする最初のまたは唯一のプロファイルである場合は、「すべての既存プロファイルをバックアップしてください」チェック・ボックスを選択したままにします。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。
- この WebSphere Process Server インストール済み環境で別のプロファイルを既にマイグレーションしており、構成をまったく変更していない場合は、前の構成をバックアップしなくてもかまいません。その場合は、「すべての既存プロファイルをバックアップしてください」チェック・ボックスをクリアして、バックアップ・ディレクトリー名が前のマイグレーションで指定された名前と同じであることを確認します。

• バージョン 6.1.x からのマイグレーション:

バージョン 6.1.x からバージョン 6.2 へのマイグレーション中に、WebSphere Process Server バージョン 6.1.x インストール環境からマイグレーションする対象として指定されたプロファイルのみが、バックアップ操作によってバックアップされます。

- まだディレクトリーが存在しない場合は、マイグレーション中に作成されます。
- 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。

バックアップ・ディレクトリー情報を入力して「次へ」をクリックします。

6. 「アプリケーションのマイグレーション設定」画面で、マイグレーションされたアプリケーションを格納する場所を指定して「次へ」をクリックします。

注: 「アプリケーションのマイグレーション設定」画面は、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションしている場合に限って表示されます。

以下のオプションのいずれか 1 つを選択できます。

- 現在のアプリケーション・インストール・ディレクトリを保持します。

**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 6.0.2.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・インストール環境のデフォルト・ディレクトリにアプリケーションをインストールします。
- 以下のディレクトリにアプリケーションをインストールします。

ディレクトリ・フィールドでは、ディレクトリ・パスを入力するか、「参照」をクリックして検索することができます。ディレクトリ・パス `${USER_INSTALL_ROOT}¥installedApps` を使用することをお勧めします。

7. 「デプロイメント・マネージャー・オプション」画面では、オプションで「前のバージョンのデプロイメント・マネージャーを使用する」を選択できます。バージョン 6.2 のデプロイメント・マネージャーと競合しないように、このオプションのチェック・マークは外しておくことをお勧めします。

注: 「デプロイメント・マネージャー・オプション」画面は、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションしている場合に限って表示されます。

8. 「ポート値割り当て」画面で、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。

注: 「ポート値割り当て」画面は、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションしている場合に限って表示されます。

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。

- 前の (ソース) インストールに割り当てられたポート値を使用する。
- 指定されたターゲット・プロファイルに割り当て済みのポート値を使用する。
- 次の値で始まるポート値のブロックを定義する。

このオプションを選択した場合、割り当てる連続したポート番号のブロックの最初の値を入力します。

9. 「追加のマイグレーション・オプション」画面で、以下の情報を入力して「次へ」をクリックします。

**重要:** 「ターゲット・プロファイルの 6.0.x スクリプトをサポート (Support 6.0.x scripts in target profile)」チェック・ボックスは、バージョン 6.0.2.x からマイグレーションする場合のみ表示されます。

- 「ターゲット・プロファイルの 6.0.x スクリプトをサポート (Support 6.0.x scripts in target profile)」チェック・ボックス: バージョン 6.0.2.x からマイグレーションする際に、スクリプトの互換性をサポートしたままマイグレーションを行う場合は、このチェック・ボックスを選択します。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 6.0.2.x についての以下の構成定義が作成されます。

- トランSPORT
- ProcessDef
- 6.0.2 SSL
- 6.0.2 ORB サービス・スレッド・プール

これらは以下の バージョン 6.2 構成定義の代替となります。

- チャネル
- ProcessDefs
- バージョン 6.2 SSL
- バージョン 6.2 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 6.2 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 6.2) 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。

- 「セキュリティ資格認定を入力する」チェック・ボックス: 前のバージョンの WebSphere Process Server でセキュリティが有効でなかった場合は、

「**セキュリティ資格認定を入力する**」チェック・ボックスがクリアされていることを確認して「**次へ**」を選択します。

マイグレーション元となるバージョンの WebSphere Process Server で管理セキュリティが有効であった場合は、「**セキュリティ資格認定を入力する**」チェック・ボックスを選択し、対応するユーザー名とパスワードを「ユーザー名」フィールドと「パスワード」フィールドにそれぞれ入力する必要があります。

旧バージョンの WebSphere Process Server で構成されたユーザー名とパスワードがマイグレーション・ウィザードによって検出された場合は、「**セキュリティ資格認定を入力する**」チェック・ボックスが自動的に選択され、適切なユーザー名とパスワードが「ユーザー名」フィールドと「パスワード」フィールドにそれぞれ自動的に設定されます。その場合は、入力されたユーザー名が正しいことを確認して (パスワードはマスクされます)「**次へ**」を選択します。

前のバージョンでセキュリティが有効だったにも関わらず、マイグレーション・ウィザードがユーザー名とパスワードを検出できない場合、「**セキュリティ資格認定を入力する**」チェック・ボックスにはチェック・マークが表示されますが、「ユーザー名」フィールドと「パスワード」フィールドはそれぞれブランクのままになります。この場合は、ユーザー名とパスワード (前のバージョンで管理セキュリティに使用したものと同一のもの) を入力して「**次へ**」を選択します。

**注:** ウィザードがユーザー名とパスワードを検出できるかどうかは、マイグレーション元となる WebSphere Process Server のバージョンによって異なります。

10. 「マイグレーションの要約」画面の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。マイグレーション中に、以下のパネルが表示される場合があります。
  - 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画面に表示されます。
  - 画面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場合は、ウィザードに障害画面が表示されます。マイグレーションが部分的に正常な場合は、ウィザードに警告画面が表示されます。マイグレーションで問題が発生した場合は、問題を修正してからもう一度マイグレーションを実行してください。
  - 事後マイグレーションが正常に完了すると、正常に終了したことを通知する画面がウィザードに表示されます。
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。

## タスクの結果

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

21 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` コマンド行ユーティリティー

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。

 `WBIPostUpgrade` コマンド行ユーティリティー

WebSphere Process Server の `WBIPostUpgrade` コマンドを使用して、指定された `backupDirectory` に `WBIPreUpgrade` コマンドによって保存されたプロファイル構成を取得します。

### 関連情報

 Java 仮想マシン設定

 `convertScriptCompatibility` コマンド

---

## 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。
- 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 (インストール・ディレクトリー名以外) 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

ご使用のユーザー・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)。すべてのユーザー・アプリケーションは、新しいサーバーに自動的にマイグレーションされます。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: WebSphere Adapter のマイグレーションについては、IBM WebSphere Business Process Management バージョン 6.2 のインフォメーション・センターに収録されている WebSphere Integration Developer の資料で、ご使用のアダプターについての資料を参照してください。

注: 動的および静的起動の両方に単一の参照を使用する SCA モジュールがあり、その参照が JMS または HTTP バインディング付きのインポートにワイヤードされている場合、その JMS または HTTP バインディングは、動的な Web サービス起動を実行するのではなく、jms: または http: URL を使用する動的起動に使用されます。バージョン 6.1.2 の振る舞いを保持し、このシナリオで Web サービス呼び出しを続けるには、(MFC または POJO コンポーネントの) 呼び出しを行うときにモジュールを更新して bindingType を正しく設定し、Web サービス URL を示すようにするか、そうでなければ、WebSphere 変数の SCA\_USE\_WS\_FOR\_DYNAMIC\_INVOCATION を設定して、セミコロンで区切られたリスト (sca/myModule1;sca/myModule2 等) 中のモジュール名を含むようにします。

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 使用インストール・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

### 自動データベース・マイグレーション

バージョン 6.0.2.x からマイグレーションし、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、150 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 バージョン 6.2 によってサポートされています。

### 手動データベース・マイグレーション

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ごとにプロバイダ定義とデータ・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に行われます。

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。

Business Process Choreographer または構成済みビジネス・スペースがある場合は、手動でデータベースを更新する必要があります。

詳しくは、39 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

150 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 6.2 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 6.2 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 6.1 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 6.2 のスタンドアロン・サーバー・プロファイルにバージョン 6.1 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。

注: スタンドアロン・サーバーのプロファイルのみが独立したマシンにマイグレーションできます。

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 6.2 の環境に直接マップします。バージョン 6.0.2.x からマイグレーションするとき、WBIPostUpgrade の呼び出し中に `-portBlock` パラメーターが指定された場合は、バージョン 6.2 にマイグレーションする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 6.2 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 6.2 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 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。

#### WebSphere Process Server バージョン 6.2 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¥¥ProcServer
            <EAR_file_name>
            {-nodeployejb
            -appname app_name
            -cluster cluster_name}"
```

## バージョン 6.1.x および 6.0.2.x ノードのバージョン 6.2 ノードへのマイグレーション

セルに属する WebSphere Process Server バージョン 6.1.x および 6.0.2.x のノードは、セルからそのノードを除去せずに WebSphere Process Server バージョン 6.2 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 6.1.x および 6.0.2.x からバージョン 6.2 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 バージョン 6.2 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 6.2 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 6.2 にマイグレーションされます。

セルは混合ノードを持つことができます。これは、セルはいくつかのバージョン 6.2 ノードおよびバージョン 6.1.x ノードを持つことができるということです。

**注:** バージョン 6.0.2.x からマイグレーションする場合、混合ノードはサポートされません。

## ポリシー・ファイル

WebSphere Process Server のバージョン 6.2 は、バージョン 6.1.x および 6.0.2.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 6.2 のポリシー・ファイル内のコメントはすべて保持されません。バージョン 6.1.x および 6.0.2.x のポリシーに含まれるコメントは、どれもバージョン 6.2 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 6.2 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 6.2 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 6.2 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 6.1.x および 6.0.2.x でインストールされたプロパティ・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 と JAR は、以下のようにマイグレーションされます。

#### 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 6.1.x および 6.0.2.x からバージョン 6.2 にマイグレーションされます。

- **バージョン 6.0.2.x からバージョン 6.2 へのマイグレーション:** このマイグレーションは、RAR および JAR ファイルを WAS\_INSTALL\_ROOT から WAS\_INSTALL\_ROOT、および USER\_INSTALL\_ROOT から USER\_INSTALL\_ROOT へコピーします。

- バージョン 6.1.x からバージョン 6.2 へのマイグレーション: このマイグレーションは、構成ファイルを以下のようにコピーします。
  - RAR または JAR を WebSphere Process Server インストールの一部としてインストールする場合、構成ファイルはマイグレーション・ターゲット・プロファイルにマイグレーションされ、RAR および JAR ファイルの新バージョンを示すように更新されます。
  - WebSphere Process Server をインストール後に RAR または JAR ファイルをインストールする場合、以下のようになります。
    - 先にインストールした WebSphere Process Server の下に RAR または JAR ファイルをインストールする場合は、構成ファイルのみがマイグレーションされるので、これらの RAR または JAR ファイルをマイグレーション・ターゲット・プロファイルにコピーするかインストールするかして、その構成が正しいことをサーバーを起動する前に確認します。
    - 先にインストールした WebSphere Process Server の外部に RAR または JAR ファイルをインストールする (推奨) 場合は、構成ファイルはマイグレーションされ、マイグレーション後に何もする必要はありません。

ただし、バージョン 6.1.x および 6.0.2.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 6.2 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

スタンドアロン・プロファイルのマイグレーション中、マイグレーションされる WebSphere Process Server サンプルはありません。すべてのバージョン 6.2 サンプルに対して、同等のバージョン 6.2 サンプルが使用可能です。

## セキュリティ

**注:** 以下のセキュリティ情報が適用されるのは、バージョン 6.0.2.xからマイグレーションする場合のみです。

WebSphere Process Serverバージョン 6.2 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 6.2 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 6.2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。

WebSphere Process Server バージョン 6.0.2.x からバージョン 6.2 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 6.2 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 6.2 のデフォルト構成に変換されます。バージョン 6.2 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 6.0.2.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 6.2 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 6.2 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 6.2 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、WBIPostUpgrade コマンド行ユーティリティ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 6.1.x および 6.0.2.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。デフォルトのサー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

4 ページの『WebSphere Process Server に関する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。

38 ページの『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。

108 ページの『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。

### 関連タスク

93 ページの『非クラスタ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

 [WBIPostUpgrade コマンド行ユーティリティー](#)

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された `backupDirectory` に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

### 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:** WebSphere Adapter のマイグレーションについては、IBM WebSphere Business Process Management バージョン 6.2 のインフォメーション・センターに収録されている WebSphere Integration Developer の資料で、ご使用のアダプターについての資料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 を基にしているので、同じ情報が適用されます。

WebSphere Adapter のマイグレーションについては、IBM WebSphere Business Process Management バージョン 6.2 のインフォメーション・センターに収録されている WebSphere Integration Developer の資料で、ご使用のアダプターについての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、データベース表が新規スキーマ・バージョンにマイグレーションされます。ただし、Business Process Choreographer またはビジネス・スペースが構成されている場合、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 6.2 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)
- WebSphere が提供するビジネス・スペース (デフォルト名 IBMBUSSP)

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。

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

150 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 6.2 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

#### WBIPreUpgrade コマンド行ユーティリティー

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。

#### WBIPostUpgrade コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された *backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

#### 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。

## 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。

### 手順

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。

- **i5/OS** **Linux** **UNIX** Linux、UNIX、およびi5/OS プラットフォームの場合: `install_root/dbscripts/component_name/database_type`
- **Windows** Windows 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 6.2 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 Universal Database for i5/OS	DB2iSeries
DB2 for z/OS バージョン 8.x	DB2zOSV8 - データベースの初期構成で DB2 z/OS v8 (長い表名を使用) を使用した場合、または DB2 z/OS v7 から DB2 z/OS v8 にアップグレードした場合は、このディレクトリーのスクリプトを使用します。
DB2 for z/OS バージョン 9.x	DB2zOSV9 - データベースの初期構成でDB2 z/OS v9 以降 (長い表名を使用) を使用した場合、または DB2 z/OS v7 から DB2 z/OS v9 にアップグレードした場合は、このディレクトリーのスクリプトを使用します。
Derby	Derby 注: 以前の WebSphere Process Server (6.0.2) 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。 150 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。その同じバージョン (6.0.2) 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 6.2 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、データベース構成を参照してください。

4. **upgradeSchema** という語で始まるデータベース用のマイグレーション・スクリプトの場所を探します。 ご使用のデータベースに応じて、`upgradeSchema.bat`、`upgradeSchema.sh`、またはこれら両方と、コンポーネント固有の SQL スクリプト (`upgradeSchema<Migrated WBI version>_<component>.sql` など) の場所を見つけます。 例えば、共通データベース・スキーマを WebSphere Process Server バージョン 6.0.2.x からバージョン 6.2 にアップグレードするとき、`dbType` が `DB2_UNIVERSAL` であるとする、以下のマイグレーション・スクリプトが必要になります。

- `install_root/dbscripts/CommonDB/DB2/upgradeSchema.bat`
- `install_root/dbscripts/CommonDB/DB2/upgradeSchema.sh`
- `install_root/dbscripts/CommonDB/DB2/upgradeSchemaTables.bat`
- `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_Recovery.sql`
- `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_relationshipService.sql`
- `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_governancerepository.sql`
- `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_DirectDeploy.sql`
- `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_CommonDB.sql`
- `install_root/dbscripts/CommonDB/DB2/upgradeSchema602_customization.sql`

注: マイグレーション元の WebSphere Process Server のバージョンに応じて、同じ `CommonDB/DB2` ディレクトリーに存在する他のスクリプトを使用してください。

5. 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、データベースがホストされるシステムにコピーします。
6. 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、各ディレクトリーのスクリプトも含まれます。スクリプトの名前は、`upgradeSchemannn_<component>.sql` という構文に従って付けられています。ここで、`nnn` は製品のバージョン番号に対応し、`component` は以下のいずれかに対応します。

- `CommonDB`
- `relationshipService`
- `governancerepository`
- `DirectDeploy`
- `customization`

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。

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。

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このステップを再試行します。

## タスクの結果

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 6.1.x またはバージョン 6.0.2.x サーバーを実行することはできません。

### 関連タスク

『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 が構成されたサーバーまたはクラスタのマイグレーション後、Business Process Choreographer データベースのスキーマを手動でアップグレードし、サーバーまたはクラスタ・メンバーを始動する前にデータをマイグレーションする必要があります。

## 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 が構成されたサーバーまたはクラスタのマイグレーション後、Business Process Choreographer データベースのスキーマを手動でアップグレードし、サーバーまたはクラスタ・メンバーを始動する前にデータをマイグレーションする必要があります。

### 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 (クラスタの場合はクラスタ内のサーバー) 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

実動システムの場合、マイグレーションされたサーバーを起動する前に、このサーバーがアクセスする Business Process Choreographer のデータベースに関連付けられたスキーマを更新する必要があります。クラスタの場合、マイグレーション済みクラスタ・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・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。

### 手順

1. DB2 for z/OS および OS/390® Version 7 を使用している状態で、DB2 for z/OS Version 8 または DB2 9 for z/OS にデータベースをアップグレードしていない場合は、DB2 for z/OS の文書を参照してアップグレードを実行してください。
2. データベースの文書に従い、Business Process Choreographer データベースをバックアップします。
3. 45 ページの『Business Process Choreographer データベース・スキーマのアップグレード』の説明に従い、データベース・スキーマをアップグレードします。
4. 49 ページの『Business Process Choreographer ランタイム・データのマイグレーション』の説明に従い、データベース内のランタイム・データをマイグレーションします。
5. サーバーまたはクラスターを始動し、Business Process Choreographer データベースが正常にマイグレーションされたことを確認します。サーバーが始動しない場合は、バックアップからデータベースを復元し、3 から処理を繰り返さなければならないことがあります。

## タスクの結果

これで、データベース・スキーマが更新され、データが新しいスキーマにマイグレーションされました。これは、データベースに対してバージョン 6.1.x またはバージョン 6.0.2.x サーバーを実行できない状態です。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

### 関連タスク

40 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。

94 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。

101 ページの『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。

108 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ブ

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。

121 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。

## Business Process Choreographer データベース・スキーマのアップグレード

Business Process Choreographer が構成されているサーバーまたはクラスターをマイグレーションまたはアップグレードした後、関連する 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。

### 手順

1. 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。

- i5/OS Linux UNIX Linux、UNIX、およびi5/OS プラットフォームの場合: `install_root/dbscripts/component_name/database_type`
- Windows Windows 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 6.2 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 Universal Database for i5/OS	DB2iSeries
DB2 for z/OS バージョン 8.x	DB2zOSV8 - データベースの初期構成で DB2 z/OS v8 (長い表名を使用) を使用した場合、または DB2 z/OS v7 から DB2 z/OS v8 にアップグレードした場合は、このディレクトリーのスクリプトを使用します。
DB2 for z/OS バージョン 9.x	DB2zOSV9 - データベースの初期構成で DB2 z/OS v9 以降 (長い表名を使用) を使用した場合、または DB2 z/OS v7 から DB2 z/OS v9 にアップグレードした場合は、このディレクトリーのスクリプトを使用します。

データベース・タイプ	ディレクトリー名
Derby	Derby 注: 以前の WebSphere Process Server (6.0.2) 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。150 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。その同じバージョン (6.0.2) 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 6.2 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、データベース構成を参照してください。

2. データベースと現行のスキーマのバージョンに必要なマイグレーション・スクリプトをコピーする準備をします。*schema\_version* は、バージョン 6.0.2 の場合は値 602、バージョン 6.1.0 の場合は値 610、バージョン 6.1.2 の場合は値 612 になります。必要なスクリプトを探します。ただし、この段階ではまだ実行しません。

#### Linux、UNIX、および Windows の DB2 の場合:

データベース・オブジェクトをアップグレードする前に、スクリプト `upgradeTablespacesschema_version.sql` と、以下に示すいずれかのアップグレード・スクリプトを使用して、テーブル・スペースをアップグレードします。

- `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。

#### i5/OS の DB2 の場合:

スクリプト `upgradeSchemaschema_version.sql` を使用して、スキーマ・オブジェクトをアップグレードします。

注: i5/OS の DB2 用の `upgradeTablespacesschema_version.sql` スクリプトはありません。

#### z/OS および OS/390 の DB2 の場合:

- データベースの初期構成で DB2 z/OS V8 または V9 を使用した場合は、次に示す 2 つのスクリプトを以下の順序で実行します。

- a. `upgradeTablespacesschema_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。
- データベースの初期構成で DB2 z/OS V7 を使用した場合は、次に示す 2 つのスクリプトを以下の順序で実行します。
  - a. `upgradeTablespacesschema_versionDB2zOSV7.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_versionDB2zOSV7.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 `createDbospace.sh` または `createDbospace.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。

注: デフォルトのテーブル・スペースにオブジェクトを新しく作成するための `upgradeSchemaschema_versionnonp.sql` スクリプトはありません。この場合、スキーマの自動アップグレードのみがサポートされます。

#### 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。
- `upgradeSchemaschema_versionUnicode.sql` は、`createSchemaUnicode.sql` スクリプトまたは `createDatabaseUnicode.sql` スクリプトを使用して UNICODE サポートのスキーマを作成しており、カスタム・スキーマ修飾子を使用する場合に使用します。

- upgradeSchemaSchema\_versionUnicodeNonp.sql は、createSchemaUnicode.sql スクリプトまたは createDatabaseUnicode.sql スクリプトを使用して UNICODE サポートのスキーマを作成しており、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。

3. 該当するスクリプトを、ディレクトリーからデータベースをホストするシステムにコピーします。
4. 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。スクリプトおよびデータベースによっては、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。例えば、DB2 Universal Database for i5/OS を使用している場合、SQL ステートメント内のすべての @SCHEMA@ プレースホルダーを、Business Process Choreographer データベースの集合名に置き換える必要があります。
5. 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。
6. DB2 Universal Database for i5/OS を使用している場合、IBM System i® 環境をセットアップして、ALTER テーブル・コマンドの実行時に送信される照会メッセージに対して自動的に応答する必要があります (通常の照会メッセージには、ユーザーからの応答が必要)。
  - a. i5/OS のコマンド行のウィンドウを開きます。
  - b. DSPJOB を入力して「2 ジョブ定義属性の表示 (Display job definition attributes)」オプションを選択し、「照会メッセージの応答 (Inquiry message reply)」の元の値を記録します。
  - c. 次に、以下のコマンドを入力します。
 

```
CHGJOB INQMSGRPY(*SYSRPLY)
ADDRPLYE SEQNBR(nn) MSGID(CPA32B2) CMPDTA(*NONE) RPY(I)
```

この nn は、システム応答リストの未使用のシーケンス番号を表します。
  - d. QShell セッションを開始します。
7. 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。例えば、DB2 Universal Database for i5/OS を使用してバージョン 6.1.0 からマイグレーションする場合、QShell で db2 -tvf upgradeSchema610.sql コマンドを発行します。

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報告された場合、報告されたエラーを修正してから、このステップを再度実行します。

8. DB2 Universal Database for i5/OS を使用している場合、「照会メッセージの応答 (Inquiry message reply)」の元の値を設定します。
  - a. i5/OS のコマンド行のウィンドウで以下のコマンドを入力して、応答リストの項目を一覧表示します。

```
WRKRPLYE
```

- b. ステップ 6c (48 ページ) で追加した応答を選択し、その横にオプション 4 (削除) を入力します。
- c. 次に、以下のコマンドを入力します。

```
CHGJOB INQMSGRPY(original_value)
```

## タスクの結果

これで、Business Process Choreographer データベース・スキーマが更新されました。

## 次のタスク

Business Process Choreographer データのマイグレーションを実行します。

## Business Process Choreographer ランタイム・データのマイグレーション

Business Process Choreographer が構成されたサーバーまたはクラスターをマイグレーションまたはアップグレードした後、サーバーまたはクラスター・メンバーを始動する前にデータをマイグレーションする必要があります。

## 始める前に

データのマイグレーションに関する最新情報は、『技術情報 21327385』を参照してください。

## 手順

1. DB2 for Linux、UNIX、Windows、または z/OS を使用している場合、カスタム作成した索引、ビュー、トリガーをドロップします。以下のテーブルを参照している項目はデータ・マイグレーションの影響を受けるため、それらの項目もドロップします。
  - PROCESS\_TEMPLATE\_B\_T
  - ACTIVITY\_TEMPLATE\_B\_T
  - SCOPED\_VARIABLE\_INSTANCE\_B\_T
  - CORRELATION\_SET\_INSTANCE\_B\_T
  - STAFF\_QUERY\_INSTANCE\_T
  - TASK\_TEMPLATE\_T
  - TASK\_INSTANCE\_T
2. 51 ページの『Business Process Choreographer データのマイグレーション・スクリプト』の説明に従い、データベース・マイグレーションのスクリプトを実行します。

**重要:** データ量とデータベース・サーバーの能力によっては、データのマイグレーションに数時間かかることがあります。マイグレーションに失敗した場合、失敗した場所からマイグレーションを再開して処理を継続するためのオプションが用意されています。処理を継続できないか、処理時間が長すぎるために途中で処理を停止した場合は、バックアップからデータベースを復元します。

3. データのマイグレーションが正常に実行されていることを確認します。以下に示すメッセージが wsadmin トレース・ファイルに書き込まれます。ただし、す

すべてのテーブルが並行してマイグレーションされるため、別のテーブルに対するメッセージがインターリーブされることがあります。

- a. データをマイグレーションする必要がない場合は、以下のメッセージが表示されます。

INFO: CWWBB0642I: 指定されたデータベースでは、データのマイグレーションは不要です。  
データのマイグレーションは、アクションなしで完了しました。

- b. カスタム・テーブルが存在する場合は、以下のメッセージが表示されます。

警告: カスタム・テーブルが構成されています。(Warning: Custom tables have been configured.)  
すぐにドロップして再作成する必要があります。(They must be dropped and re-created now.)

この場合、カスタム・テーブルをドロップしてスクリプトを再開します。

- c. マイグレーション・スクリプトの別のインスタンスが既に実行されている場合、以下のメッセージが表示されます。

CWWBB0654E: データのマイグレーションは既に開始されています。

これは、マイグレーション・スクリプトの複数のインスタンスが同時に実行されることを防止するためのメカニズムです。これまでに実行したスクリプトがすべてエラーとなって停止したことを確認してから問題を修正した場合は、`-force` オプションを使用してこの保護メカニズムを迂回することができます。このオプションの使用の詳細については、51 ページの『Business Process Choreographer データのマイグレーション・スクリプト』を参照してください。

- d. データのマイグレーションが開始されると、以下のメッセージが表示されます。

INFO: CWWBB0650I: データのマイグレーションを開始します。

- e. 作業項目データのマイグレーションの開始と終了を知らせる場合は、以下のメッセージが表示されます。

INFO: CWWBB0644I: 作業項目のマイグレーションを開始します。  
INFO: CWWBB0645I: 作業項目のマイグレーションが正常に完了しました。

作業項目データのマイグレーション中に、以下のように、進捗状況のおおよその割合が 2 分ごとに表示されます。

Nov 13, 2008 5:04:50 PM INFO: CWWBB0656I: 「作業項目のマイグレーション 23.56%」完了。  
(Nov 13, 2008 5:04:50 PM INFO: CWWBB0656I: 'Workitem migration 23.56%' completed.)

- f. データベースに対してテーブル・スペースのマイグレーションが必要な場合、開始と終了の際に以下のメッセージが表示されます。

INFO: CWWBB0646I: テーブル・スペースのマイグレーションを開始します。  
INFO: CWWBB0647I: テーブル・スペースのマイグレーションが正常に完了しました。

テーブル・スペースのマイグレーション中に、各マイグレーション処理の開始を示す以下のようなメッセージが表示されます。

INFO: CWWBB0657I: テーブルのマイグレーション「1/7」。(INFO: CWWBB0657I: Migrating table '1/7'.)

テーブル・スペースのマイグレーション中に、以下のように、進捗状況の割合が 2 分ごとに表示されます。

INFO: CWWBB0656I: 「テーブル 1/7 95.8%」完了。(INFO: CWWBB0656I: 'Table 1/7 95.8%' completed.)

処理が完了すると、以下のようなメッセージが表示されます。

INFO: CWWBB0656I: 「テーブル 1/7 100.0%」完了。(INFO: CWWBB0656I: 'Table 1/7 100.0%' completed.)

- g. エラーが発生したためにデータのマイグレーションを正常に終了できない場合は、以下のメッセージが表示されます。

SEVERE: CWWB0652E: データのマイグレーションは、エラーが発生して完了しました。

この場合は、有効なスタック・トレースを確認して問題の原因を修正します。問題を修正したら、ステップ 2 (49 ページ) の記述に従い、データ・マイグレーション・スクリプトを再実行します。スクリプトは、停止した位置から処理を継続します。

**注:** すべてのデータが正常にマイグレーションされるまで、Business Flow Manager と Human Task Manager を起動することはできません。また、Business Process Choreographer が構成されているサーバーを起動しようとすると、以下のメッセージが SystemOut.log ファイルに書き込まれます。

SEVERE: CWWB0653E: データのマイグレーションは開始されましたが、まだ完了していません。

- h. すべてのデータが正常にマイグレーションされると、以下のメッセージが表示されます。

INFO: CWWB0651I: データのマイグレーションが正しく完了しました。

- i. カスタム・テーブルまたは名前付きマテリアライズド・ビューが登録されると、マイグレーション終了後に警告が表示されます。マテリアライズド・ビューは自動的にドロップされて再作成されますが、カスタム・テーブルはすべて手動でドロップして再作成する必要があります。

- 4. DB2 for Linux、UNIX、Windows、または z/OS を使用している場合、マイグレーション終了後に、ステップ 1 (49 ページ) で削除したカスタム・オブジェクトを再作成します。

## タスクの結果

これで、Business Process Choreographer ランタイム・データが新しいスキーマにマイグレーションされました。DB2 データベースを使用している場合、このデータベースが使用するテーブル・スペースのページが大きくなります。

### Business Process Choreographer データのマイグレーション・スクリプト:

migrateDB.py スクリプトを使用して、Business Process Choreographer データベースのランタイム・データを新しいスキーマにマイグレーションします。これにより、ビジネス・プロセスとヒューマン・タスクの照会パフォーマンスが改善されます。

### 目的

このスクリプト・ファイルにより、作業項目に関する情報が格納された 2 つのテーブルをマージします。また、Linux、UNIX、Windows、および z/OS 上の DB2 データベースについて、より大きなページ・サイズ (8 k) を持つテーブル・スペースへのマイグレーションも実行します。このデータ・マイグレーションは一回だけ実行します。このマイグレーションにより、パフォーマンスが向上します。

### ロケーション

このスクリプトは、以下に示す管理スクリプト用の Business Process Choreographer サブディレクトリーに格納されています。

Linux、UNIX、および i5/OS プラットフォームの場合:

`install_root/ProcessChoreographer/admin/migrateDB.py`

Windows プラットフォームの場合:

`install_root%ProcessChoreographer%admin% migrateDB.py`

### 制約事項

- このデータ・マイグレーションは、必ず実行する必要があります。このマイグレーションは、Business Process Choreographer が構成されている バージョン 6.2 サーバーまたはクラスターを再起動する前に完了する必要があります。
- このスクリプトは失敗してもロールバックされないため、スクリプトを実行する前にデータベースをバックアップする必要があります。ただし、スクリプトを再開すると、このスクリプトはデータのマイグレーションを継続しようとします。
- そのため、このスクリプトは切断モードで実行する必要があります。切断モードの場合、マイグレーション中は、Business Process Choreographer 構成をホストしているサーバーまたはクラスターを停止する必要があります。
- クラスターを指定した場合は、このスクリプトはデプロイメント・マネージャーではなく、クラスター・メンバーのノード上で実行する必要があります。
- このスクリプトはセマフォを使用し、スクリプトのインスタンスが同時に複数実行されることを防止します。最初のインスタンスが完了する前にスクリプトを再開すると、2 番目のインスタンスは、`com.ibm.bpe.api.DatabaseDataMigrationAlreadyRunningException` 例外と「CWWB0654E: データのマイグレーションは既に開始されています」というメッセージを出力して失敗します。
- データ量とデータベース・サーバーの能力によっては、マイグレーション・プロセスに数時間かかることがあります。データ・マイグレーションの詳細については、『技術情報 21327385』を参照してください。
- i5/OS の場合、qshell を使用してスクリプトを実行する必要があります。

### スクリプトの呼び出し

サーバーを指定した場合、サーバー・ノード上でスクリプトを実行する必要があります。クラスターを指定した場合、このスクリプトはデプロイメント・マネージャーではなく、クラスター・メンバーのノード上で実行する必要があります。wsadmin によってトレース・ファイルが上書きされるため、`-tracefile` オプションを使用して、データ・マイグレーション用のトレース・ファイルの名前と場所を指定します。

スクリプトを呼び出すには、プロファイル『`profile`』を使用してトレース情報をファイル `migration_trace_file` に送信し、次のいずれかのコマンドを入力します。

#### Linux および UNIX プラットフォームの場合:

```
wsadmin.sh -conntype NONE -profileName profile
           -tracefile migration_trace_file
           -f install_root/ProcessChoreographer/admin/migrateDB.py
           parameters
```

#### i5/OS プラットフォームの場合:

```
wsadmin -conntype NONE -profileName profile
        -tracefile migration_trace_file
        -f install_root/ProcessChoreographer/admin/migrateDB.py
        parameters
```

## Windows プラットフォームの場合:

```
wsadmin -conntype NONE -profileName profile
        -tracefile migration_trace_file
        -f install_root¥ProcessChoreographer¥admin¥migrateDB.py
        parameters
```

## パラメーター

スクリプトが使用するパラメーターは次のとおりです。

```
(-server serverName) | (-cluster clusterName)
[[-dbUser userID] -dbPassword password]
[-dbSchema schema]
[-slice slice]
[-force]
```

各部の意味は、次のとおりです。

### **-server** serverName | **-cluster** clusterName

データベースがマイグレーションされる Business Process Choreographer の構成を指定します。

### **-dbUser** userID

データベースを認証するためのオプションのユーザー ID です。-dbUser が指定されていない場合は、データ・ソースの認証別名のユーザーがデフォルトのユーザーになります。認証別名が設定されていない場合は、データベースに接続する際にユーザー修飾子は使用されません。データベースのマイグレーション処理に必要な権限がデフォルトのユーザー ID に設定されていない場合、このオプションを使用して、必要な権限を持つユーザー ID を指定する必要があります。例えば、DB2 で -dbUser を指定し、別のユーザー ID を基にした暗黙のスキーマを使用する場合は、-dbSchema パラメーターを指定する必要があります。

### **-dbPassword** password

Derby データベースを使用する場合を除き、パスワードが必要です。

### **-dbSchema** schema

明示的なスキーマを指定した場合は、このパラメーターは必要ありません。構成されているスキーマ修飾子をオーバーライドする必要がある場合は、このパラメーターを使用します。-dbSchema が指定されていない場合は、データ・ソースのスキーマ名が使用されます。データ・ソースにスキーマが明示的に指定されていない場合は、暗黙 (デフォルト) のスキーマが使用されます。

- DB2 の場合、データベースの接続に使用されるユーザー ID として暗黙のスキーマが指定されます。このため、-dbUser パラメーターを指定する場合は、暗黙のスキーマとして間違ったユーザー ID が使用されないようにするため、-dbSchema パラメーターを指定する必要があります。
- Microsoft SQL Server の場合、暗黙のスキーマは 『dbo』 になります。
- Derby の場合、暗黙のスキーマは 『APP』 になります。

### **-slice** slice

このオプションは、トランザクションのサイズを指定するパラメーターです。10 から 50000 の範囲で指定します。デフォルト値は 500 です。最適な値は、データベース・オブジェクトのサイズやトランザクション・ログのサイズなど、多くの要因によって異なります。一般的に、値を小さくするほど、マイグレーションにかかる時間は長くなります。

## **-force TRUE**

**重要:** スクリプトを再実行する必要がある、マイグレーション・スクリプトの他のインスタンスが実行されていない場合に限り、このオプション・パラメータを使用します。同時に複数のインスタンスが実行されている場合は、データベースの不整合が発生する可能性があるため、バックアップから復元する必要があります。

このオプションを使用する必要があるのは、データベース・サーバーへの接続が失われたなどの理由で、マイグレーションが完了する前にマイグレーション・スクリプトが停止した場合です。このオプションを使用する以外、スクリプトを再開することはできません。スクリプトを再開すると、データのマイグレーションが停止した位置から処理が継続されます。次の

`com.ibm.bpe.api.DatabaseDataMigrationAlreadyRunningException` 例外を回避する場合も、このオプションを使用する以外の方法はありません。

CWWBB0654E: データのマイグレーションは既に開始されています。

## **例**

トランザクション・スライスのサイズが 5,000、データベース・ユーザーが『Sam』、パスワードが『secret』の設定で、クラスター『hamlet』の Business Process Choreographer の構成が使用するデータベースのランタイム・データを変換するスクリプトを呼び出し、トレース情報をファイル `migration_trace_file` に送信するには、次のいずれかのコマンドを入力します。

### **Linux および UNIX プラットフォームの場合:**

```
wsadmin.sh -conntype NONE
            -tracefile migration_trace_file
            -f install_root/ProcessChoreographer/admin/migrateDB.py
            -cluster hamlet -dbUser Sam -dbPassword secret -slice 5000
```

### **i5/OS プラットフォームの場合:**

```
wsadmin -conntype NONE
         -tracefile migration_trace_file
         -f install_root/ProcessChoreographer/admin/migrateDB.py
         -cluster hamlet -dbUser Sam -dbPassword secret -slice 5000
```

### **Windows プラットフォームの場合:**

```
wsadmin -conntype NONE
         -tracefile migration_trace_file
         -f install_root¥ProcessChoreographer¥admin¥migrateDB.py
         -cluster hamlet -dbUser Sam -dbPassword secret -slice 5000
```

## **問題発生後のリカバリー**

スクリプトが `com.ibm.bpe.api.DatabaseDataMigrationAlreadyRunningException` 例外でマイグレーションに失敗した場合は、`-force` オプションでスクリプトを再開することができます。その際は、必ずマイグレーション・スクリプトの他のインスタンスが実行されていないことを確認してから、スクリプトを再開してください。

## **手動による ビジネス・スペース データベースのアップグレード**

サーバーをバージョン 6.1.2 からマイグレーションした後は、バージョン 6.2 のサーバーを始動する前に、ビジネス・スペース・データベースを新しいデータベース・スキーマに手動でアップグレードする必要があります。

## 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

WebSphere Process Server をバージョン 6.1.2 からバージョン 6.2 にマイグレーションした後も、ビジネス・スペース・データベースは引き続きビジネス・スペース 6.1.2 で使用するように構成された状態になっています。ビジネス・スペース・データベースは、ビジネス・スペース バージョン 6.2 がアクセスする前に、必ずアップグレードしてください。この手順では、2 つのデータベース・スクリプトを実行します。最初のスクリプトでデータベース・スキーマをアップグレードします。2 番目のスクリプトで、ビジネス・スペース・データベースのデータを変更し、ビジネス・スペース バージョン 6.2 が使用できるように準備します。

**注:** スクリプトを実行する前には、スクリプトを編集して、@SCHEMA@ というストリングの出現箇所をすべてビジネス・スペース・データベースのスキーマ名で置き換える必要があります。デフォルトのスキーマ名は `IBMBUSSP` です。スクリプトを実行するとき、データベース・ユーザーは、データベースを変更するための十分な権限を持っている必要があります。

### 手順

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%\BusinessSpace\%database_type`

- **Linux** **UNIX** Linux および UNIX プラットフォームの場合:  
`install_root/dbscripts/BusinessSpace /%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 6.2 がインストールされているルート・ディレクトリーです。

`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 on iSeries	DB2iSeries

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 バージョン 8.x	DB2zOSV8  データベースの初期構成で DB2 z/OS V8 以降 (長い表名を使用) を使用した場合は、このディレクトリーのスクリプトを使用します。 <b>注:</b> 初期のデータベース構成中に DB2 z/OS v7 を使用した場合、WebSphere Process Server サポート・サイトにある関連技術情報を参照してください。
DB2 for z/OS バージョン 9.x	DB2zOSV9  データベースの初期構成で DB2 z/OS v9 および DB2 z/OS v9 キーを使用した場合は、このディレクトリーのスクリプトを使用します。 <b>注:</b> 初期のデータベース構成中に DB2 z/OS v7 を使用した場合、WebSphere Process Server サポート・サイトにある関連技術情報を参照してください。
Derby	Derby <b>注:</b> 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。150 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 6.2 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。
Oracle	Oracle <b>注:</b> ビジネス・スペース バージョン 6.1.2 に IFix001 を適用していない場合は、 upgradeSchema612_BusinessSpace.sql の以下の行のコメントを外してからスクリプトを実行してください。  -- ALTER TABLE @SCHEMA@.PAGE -- RENAME COLUMN RESTRICTEDD TO RESTRICTED;

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、データベース構成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。バージョン 6.1.2 の場合、*schema\_version* の値は 612 になります。例えば、DB2 用のビジネス・スペース・データベース・スキーマを WebSphere Process Server バージョン 6.1.2 からバージョン 6.2 にアップグレードするには、*install\_root/dbscripts/BusinessSpace/DB2/upgradeSchema612\_BusinessSpace.sql* が必要です。

**注:** *upgradeData612\_BusinessSpace.sql* スクリプトも必要です。

5. 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。
6. 要件を満たすように SQL スクリプト `upgradeSchema612_BusinessSpace.sql` および `upgradeData612_BusinessSpace.sql` を変更します。例えば、データベースのスキーマ名の変更が必要です。また、ユーザー名、パスワード、ファイル・パスなど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。
  - a. `upgradeSchema612_BusinessSpace.sql` を実行してデータベース・スキーマをアップグレードします。
  - b. `upgradeData612_BusinessSpace.sql` を実行してビジネス・スペース・データベースのデータを変更します。
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。

## タスクの結果

データベースが更新され、ビジネス・スペース バージョン 6.2 で使用するための準備ができました。スクリプトを実行してデータベース・スキーマをアップグレードした後は、このデータベースをビジネス・スペース バージョン 6.1.2 で使用できなくなります。

### 関連タスク

149 ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』

WebSphere Process Server をバージョン 6.1.2 からバージョン 6.2 にマイグレーションした後は、サーバーまたはクラスターを始動する前に、いくつかの追加の作業を行う必要があります。

---

## 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 4 ページの『WebSphere Process Server に関する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、適切なサブトピックを選択してください。

**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、165 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

**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 **i5/OS プラットフォーム** または **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。

**注:** マイグレーション・ウィザードは WebSphere Process Server プロファイルのみをサポートします。WebSphere Application Server プロファイルがある場合、マイグレーション・コマンドを使用する必要があります。

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。

- 現在のシステムが、WebSphere Process Server バージョン 6.2 のインストールに必要なハードウェア要件とソフトウェア要件をすべて満たしていること、およびインストールに必要な十分なスペース（一時スペースも含む）があることを確認してください。詳しくは、WebSphere Process Server detailed system requirements を参照してください。
- 古いバージョンの WebSphere Process Server が存在するコンピューター・システムと物理的に同じコンピューター・システムでマイグレーションする場合、新しいバージョンの WebSphere Process Server は、古いバージョンと並列して同じシステムにインストールされます。

**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、67 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、74 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。

- マイグレーション対象の古い WebSphere Process Server 内のプロファイルは、スタンドアロンです（セルに統合されていません）。
- 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、4 ページの『WebSphere Process Server に関する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 6.0.2.x コンポーネントまたは バージョン 6.1.x 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。

## 手順

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 6.1.x またはバージョン 6.0.2.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。次の構文を使用します。

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

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

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。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。

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

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

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:** マイグレーション・プロセスを開始する前に、マイグレーション元のサーバーを停止する必要があります。構成をマイグレーションする場合、マイグレーション元のサーバーを稼働させる必要はありません。マイグレーション・ツールにより、サーバーが停止している間に、すべての構成データが取得されます。

3. オプション: マイグレーション・ウィザードで **Create new profile** オプションを使用して、新規のバージョン 6.2 プロファイルをマイグレーション・ターゲットとして作成します。ターゲット・プロファイルの作成について詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。
4. 以下にリストしたマイグレーションに必要な既存情報を、あらかじめ確認しておいてください。

### 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、WBIPreUpgrade コマンド行ユーティリティ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、WBIPreUpgrade コマンド行ユーティリティを参照してください。

**管理セキュリティー・ユーザー名 (管理セキュリティーが構成されている場合に必須)** -username パラメーターについては、WBIPostUpgrade コマンド行ユーティリティを参照してください。

**管理セキュリティー・パスワード (管理セキュリティーが構成されている場合に必須)** -password パラメーターについては、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティを参照してください。

**注:** これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

## 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。

– **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:

`wbi_migration.sh`

– **Windows** **Windows** プラットフォーム: `wbi_migration.bat`

**注:** マイグレーション・ウィザードを起動するとき、デフォルトのトレース設定 (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) をオプションで変更できます。デフォルトのトレース設定では、特定のクラスでのみトレースが有効になりますが、完全トレースを有効にするか、またはすべてのトレースを無効にするようにデフォルトを変更できます。

- 完全トレースを有効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
wbi\_migration.sh -W -migrationPanel.traceString="\*=all=enabled"
- **Windows** **Windows** プラットフォーム: wbi\_migration.bat -W  
-migrationPanel.traceString="\*=all=enabled"
- すべてのトレースを無効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。
  - **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
wbi\_migration.sh -W -migrationPanel.traceString="\*=all=disabled"
  - **Windows** **Windows** プラットフォーム: wbi\_migration.bat -W  
-migrationPanel.traceString="\*=all=disabled"

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、21 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. 23 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。新しいバージョンの WebSphere Process Server に必要なデータベース変更の一部は、自動的に行われます。ただし、Business Process Choreographer または ビジネス・スペース が既に構成されている場合、またはデータベース・スキーマにアクセスするための十分な権限がサーバーに設定されていない場合、あるいはその他のデータベース固有の要件を満たしていない場合は、データベースを手動で更新する必要があります。詳しくは、39 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。

## タスクの結果

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、145 ページの『Business Process Choreographer の事後マイグレーション・タスク』を参照してください。サーバーでビジネス・スペース が構成されている場合は、149 ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』を参照してください。最後に、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

4 ページの『WebSphere Process Server に関する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。

21 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

23 ページの『マイグレーション・ウィザードの実行』

AIX、HP-UX、Linux、Solaris、または Windows システムでマイグレーション・ウィザードを実行して、WebSphere Process Server をマイグレーションします。

135 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。

165 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』

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 6.2 サーバーを構成しなければならない場合があります。

39 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。

67 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。)

74 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

#### WBIPostUpgrade コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された *backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

#### WBIPreUpgrade コマンド行ユーティリティー

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。

#### 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

**注:** コマンド行ツールを使用してマイグレーションする場合、 WebSphere Process Server のプロファイルまたは WebSphere Application Server のプロファイルを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・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。

- 現在のシステムが、WebSphere Process Server バージョン 6.2 のインストールに必要なハードウェア要件とソフトウェア要件をすべて満たしていること、およびインストールに必要な十分なスペース (一時スペースも含む) があることを確認してください。詳しくは、WebSphere Process Server detailed system requirements を参照してください。
- 古いバージョンの WebSphere Process Server が存在するコンピューター・システムと物理的に同じコンピューター・システムでマイグレーションする場合、新しいバージョンの WebSphere Process Server は、古いバージョンと並列して同じシステムにインストールされます。

**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、67 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、74 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。

- マイグレーション対象の古い WebSphere Process Server 内のプロファイルは、スタンドアロンです (セルに統合されていません)。

- 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、4ページの『WebSphere Process Server に関する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 6.0.2.x コンポーネントまたは バージョン 6.1.x 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。

## 手順

1. ご使用のオペレーティング・システムに応じて、次のいずれかの手順でログオンします。
  - **i5/OS** **i5/OS プラットフォームの場合:** \*SECOFR ユーザー・クラスまたは \*ALLOBJ 特殊権限を持つ i5/OS ユーザー・プロファイルを使用してログオンします。
  - **Linux** **UNIX** **Linux および UNIX プラットフォームの場合:** root ユーザーとしてログオンします。
  - **Windows** **Windows プラットフォームの場合:** 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 6.1.x および 6.0.2.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。次の構文を使用します。

**注:** **i5/OS プラットフォームの場合**は、`QSHELL` でスクリプトを実行する必要があります。`QSHELL` セッションを開始するには、`CL` コマンド・プロンプトを開いて `QSH` と入力します。

- **i5/OS** **i5/OS プラットフォームの場合:** `profile_root/bin/stopServer server_name`
- **Linux** **UNIX** **Linux および UNIX プラットフォームの場合:** `profile_root/bin/stopServer.sh server_name`
- **Windows** **Windows プラットフォームの場合:** `profile_root¥bin¥stopServer.bat server_name`

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。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。

- **i5/OS** **i5/OS プラットフォームの場合:** `profile_root/bin/stopServer server_name -username user_ID -password password`

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

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:** マイグレーション・プロセスを開始する前に、サーバーを停止します。デフォルトでは、ノード上のすべてのサーバーが、マイグレーション・プロセスの開始前に停止されます。

- 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、WBIPreUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティーを参照してください。

**重要:** WBIPostUpgrade を呼び出す際に、`-createTargetProfile` パラメーターを使用します。このオプションにより、マイグレーションに一致する必要な新規ターゲット・プロファイルが作成されます。ターゲット・プロファイルについては、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。

**注:** i5/OS **i5/OS プラットフォーム**でマイグレーションを実行する場合、ターゲット・プロファイル名は、マイグレーション対象のソース・プロファイルのプロファイル名と一致する必要があります。

- 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。新しいバージョンの WebSphere Process Server に必要なデータベース変更の一部は、自動的に行われます。ただし、Business Process Choreographer または ビジネス・スペース が既に構成されている場合、またはデータベース・スキーマにアクセスするための十分な権限がサーバーに設定されていない場合、あるいはその他のデータベース固有の要件を満たしていない場合は、データベースを手動で更新する必要があります。詳しくは、39 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。

## タスクの結果

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、145 ページの『Business Process Choreographer の事後マイグレーション・タスク』を参照してください。サーバーで ビジネス・スペース が構成されている場合は、149 ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』を参照してください。最後に、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

4 ページの『WebSphere Process Server に関する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。

21 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

135 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。

165 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』

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 6.2 サーバーを構成しなければならない場合があります。

39 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。

#### 関連資料

##### `WBIPostUpgrade` コマンド行ユーティリティー

WebSphere Process Server の `WBIPostUpgrade` コマンドを使用して、指定された `backupDirectory` に `WBIPreUpgrade` コマンドによって保存されたプロファイル構成を取得します。

##### `WBIPreUpgrade` コマンド行ユーティリティー

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。

#### 関連情報

##### `backupConfig` コマンド

##### `stopServer` コマンド

## 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。

- ターゲット・システムが、新しいバージョンの WebSphere Process Server のすべてのハードウェア要件とソフトウェア要件を満たしている。
- 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、4 ページの『WebSphere Process Server に関する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 6.0.2.x コンポーネントまたは バージョン 6.1.x 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。

1 ページの『マイグレーションの概要』および 4 ページの『WebSphere Process Server に関する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、165 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。

### 手順

1. **WBIPreUpgrade** ユーティリティーとその関連ファイルを WebSphere Process Server バージョン 6.2 からソース・システムにコピーします。この場合、以下の手順のいずれかを実行します。
  - a. ターゲット・システムで、ソース・システムにコピーするインストール・ファイルの .zip ファイルまたは .tar.gz ファイルを作成します。この場合、以下の手順を実行します。

注: この手順を実行するには、WebSphere Process Server バージョン 6.2 がターゲット・システムに既にインストールされている必要があります。

- 1) ターゲット・システム上の WPS62\_HOME/util/migration ディレクトリーに移動します。
- 2) 使用しているオペレーティング・システムに応じて、次のいずれかのコマンドを実行します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:  
createRemoteMigrationImage.sh  
full\_path\_name\_of\_the\_new\_.tar.gz\_file

例: createRemoteMigrationImage.sh /tmp/migrationImage.tar.gz

- **Windows** **Windows** プラットフォームの場合:  
createRemoteMigrationImage.bat  
full\_path\_name\_of\_the\_new\_.zip\_file

例: createRemoteMigrationImage.bat C:%migrationImage.zip

- 3) マイグレーション・イメージ .zip ファイルを、ターゲット・システムからソース・システム (WebSphere Process Server のバージョン 6.1.x および 6.0.2.x がインストールされているシステム) にコピーします。
- 4) マイグレーション・イメージ・ファイルを migration\_copy という新規ディレクトリーに unzip します。

注: このディレクトリー名は、任意の名前を指定してかまいません。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。

- b. WebSphere Process Server バージョン 6.2 DVD の JDK および migration ディレクトリーからファイルをコピーします。この場合、以下の手順を実行します。WPS/ESB Version 6.2 DVD から migration ディレクトリーと JDK ディレクトリーを新規ディレクトリーにコピーします。ここでは、便宜上 migration\_copy というディレクトリーを使用します。

- 1) ソース・システムに migration\_copy という新規ディレクトリーを作成します。

注: このディレクトリー名は、任意の名前を指定してかまいません。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。

- 2) migration ディレクトリーと JDK ディレクトリーを、この新規ディレクトリーにコピーします。

これで、使用しているソース・システムのディレクトリー構造は以下のようになります。

```
migration_copy/  
  migration/  
    JDK/
```

2. migration\_copy/migration/bin ディレクトリーにナビゲートします。
3. ソース・システムで migration\_copy/bin ディレクトリーの **WBIPreUpgrade** スクリプトを使用して、現在の構成を保管します。マイグレーション固有のバックアップ・ディレクトリー内の構成を、古いバージョンをホストするシステム (ソース・システム) に保管します。詳しくは、WBIPreUpgrade コマンド行ユーティリティーを参照してください。使用しているオペレーティング・システムに応じて、次のいずれかのスクリプトを使用します。

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

```
./WBIPreUpgrade.sh /filepath/migration_specific_backup  
currentWebSphereDirectory
```

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

```
C:¥filepath¥migration_specific_backup currentWebSphereDirectory
```

**WBIPreUpgrade** コマンドは、画面に状況を表示し、migration\_specific\_backup/logs ディレクトリーのログ・ファイルに状況を出力します。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・スタンプがファイル名の一部として指定されます。

4. migration\_specific\_backup ディレクトリーをソース・システムからターゲット・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。

5. ターゲット・システムで **WBIPostUpgrade** コマンドを使用して、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4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、WBIPostUpgrade コマンド行ユーティリティを参照してください。使用しているオペレーティング・システムに応じて、次のいずれかのスクリプトを使用します。

**重要:** WBIPostUpgrade を呼び出す際に、`-createTargetProfile` パラメーターを使用します。このオプションにより、マイグレーションに一致する必要な新規ターゲット・プロファイルが作成されます。ターゲット・プロファイルについては、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。

**注:** 次のスクリプトで、`-profileName` はターゲット・プロファイルを指します。

- **Linux** **UNIX** **Linux/UNIX プラットフォームの場合:**  
`./WBIPostUpgrade.sh /filepath/migration_specific_backup/-profileNameprofileName`
- **Windows** **Windows プラットフォームの場合: WBIPostUpgrade**  
`C:\filepath\migration_specific_backup/-profileNameprofileName`

**WBIPostUpgrade** コマンドは、画面に状況を表示し、`migration_specific_backup/logs` ディレクトリーのログ・ファイルに状況を出力します。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・スタンプがファイル名の一部として指定されます。

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。新しいバージョンの WebSphere Process Server に必要なデータベース変更の一部は、自動的に行われます。ただし、Business Process Choreographer または ビジネス・スペース が既に構成されている場合、またはデータベース・スキーマにアクセスするための十分な権限がサーバーに設定されていない場合、あるいはその他のデータベース固有の要件を満たしていない場合は、データベースを手動で更新する必要があります。詳しくは、39 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。
7. WebSphere Process Server をサポートするデータベースがオリジナルのセットアップではローカル・システムに存在していた場合、このデータベースを別のシステムに移動すると、WebSphere Process Server に対してリモートのデータベースになります。そのため、タイプ 2 の JDBC ドライバーを以前に使用していた場合、タイプ 4 のドライバーに切り替えなければならないことがあります。
8. 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なことがあります。

- 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、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。

## タスクの結果

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、145 ページの『Business Process Choreographer の事後マイグレーション・タスク』を参照してください。サーバーでビジネス・スペース が構成されている場合は、149 ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』を参照してください。最後に、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

4 ページの『WebSphere Process Server に関する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

135 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。

165 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』

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 6.2 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。

74 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

 **WBIPreUpgrade コマンド行ユーティリティ**

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。

 **WBIPostUpgrade コマンド行ユーティリティ**

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された *backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

#### 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。

## 32 ビット Windows プラットフォームから 64 ビット Windows プラットフォームへのマイグレーション

32 ビット版 Windows オペレーティング・システムに WebSphere Process Server バージョン 6.1.x および 6.0.2.x をインストールしてある場合は、WebSphere Process Server バージョン 6.2 へのマイグレーションを実行する前に、64 ビット版 Windows オペレーティング・システムにインストールされている 64 ビット・バージョンの WebSphere Process Server バージョン 6.2 に既存のスタンドアロン・プロファイルをマイグレーションできます。

### 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。

- ターゲット・システムが、新しいバージョンの WebSphere Process Server のすべてのハードウェア要件とソフトウェア要件を満たしている。
- 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、4 ページの『WebSphere Process Server に関する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 6.0.2.x コンポーネントまたは バージョン 6.1.x 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。

1 ページの『マイグレーションの概要』および 4 ページの『WebSphere Process Server に関する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、165 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。

### 手順

1. WBIPreUpgrade ユーティリティーおよび関連ファイルを 32 ビット・バージョンの WebSphere Process Server バージョン 6.2 からソース (32 ビット) システムにコピーします。これを行うには、以下のいずれかの手順を使用します。
  - a. ターゲット・システムで、インストール・ファイルの .zip ファイルまたは .tar.gz ファイルを作成し、ソース・システムにコピーします。これを行うには、以下の手順を使用します。

**注:** この手順を使用するには、事前に 32 ビット・バージョンの WebSphere Process Server バージョン 6.2 をターゲット・システムにインストールしておく必要があります。

- 1) ターゲット・システムの WPS62\_HOME/util/migration ディレクトリーにナビゲートします。
- 2) 使用しているオペレーティング・システムに応じて、以下のいずれかのコマンドを実行します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:  
`createRemoteMigrationImage.sh`  
`full_path_name_of_the_new_.tar.gz_file`

**例:** `createRemoteMigrationImage.sh /tmp/migrationImage.tar.gz`

- **Windows** **Windows** プラットフォームの場合:  
`createRemoteMigrationImage.bat`  
`full_path_name_of_the_new_.zip_file`

例: createRemoteMigrationImage.bat C:%migrationImage.zip

- 3) マイグレーション・イメージの .zip ファイルをターゲット・システムからソース・システム (バージョン 6.1.x および 6.0.2.x の WebSphere Process Server がインストールされているシステム) にコピーします。
- 4) migration\_copy という新規ディレクトリーにマイグレーション・イメージ・ファイルを unzip します。

注: 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。

- b. 32 ビット・バージョンの WebSphere Process Server バージョン 6.2 DVD の JDK および migration ディレクトリーからファイルをコピーします。これを行うには、以下の手順を使用します。WPS/ESB バージョン 6.2 DVD の migration および JDK ディレクトリーを新規ディレクトリーにコピーします。例として、ここでは説明のために migration\_copy を使用します。
  - 1) migration\_copy という新規ディレクトリーをソース・システムに作成します。

注: 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。

- 2) migration および JDK ディレクトリーをこの新規ディレクトリーにコピーします。

これで、ソース・システムに以下のディレクトリー構造が作成されます。

```
migration_copy/  
  migration/  
    JDK/
```

2. WebSphere Process Server バージョン 6.1.x および 6.0.2.x インストール環境のバックアップ・ディレクトリーを別のマシンにコピーします。
3. マイグレーションを実行するマシンに 64 ビット版 Windows 2003 オペレーティング・システムをインストールします。
4. 新しい 64 ビット版 Windows 2003 オペレーティング・システムがあるマシンに WebSphere Process Server バージョン 6.2 をインストールします。
5. バックアップ・ディレクトリーに対して WBIPostUpgrade コマンドを実行し、マイグレーションを完了します。

## タスクの結果

32 ビット版 Windows オペレーティング・システムにインストールされたバージョン 6.1.x および 6.0.2.x インストール環境から WebSphere Process Server バージョン 6.2 にマイグレーションしました。

## 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。

注: コマンド行ツールを使用してマイグレーションする場合、WebSphere Process Server のプロファイルまたは WebSphere Application Server のプロファイルを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・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、4 ページの『WebSphere Process Server に関する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 6.0.2.x コンポーネントまたは バージョン 6.1.x 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。

1 ページの『マイグレーションの概要』および 4 ページの『WebSphere Process Server に関する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、165 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。

### 手順

1. WBIPreUpgrade ユーティリティおよび関連ファイルを WebSphere Process Server バージョン 6.2 からソース・システムにコピーします。これを行うには、以下のいずれかの手順を使用します。
  - a. ターゲット・システムで、インストール・ファイルの .zip ファイルまたは .tar.gz ファイルを作成し、ソース・システムにコピーします。これを行うには、以下の手順を使用します。

注: この手順を使用するには、事前に WebSphere Process Server バージョン 6.2 をターゲット・システムにインストールしておく必要があります。

- 1) ターゲット・システムの WPS62\_HOME/util/migration ディレクトリーにナビゲートします。
- 2) 使用しているオペレーティング・システムに応じて、以下のいずれかのコマンドを実行します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合:  
createRemoteMigrationImage.sh  
*full\_path\_name\_of\_the\_new\_.tar.gz\_file*

例: createRemoteMigrationImage.sh /tmp/migrationImage.tar.gz

- **Windows** Windows プラットフォームの場合:  
`createRemoteMigrationImage.bat`  
`full_path_name_of_the_new_.zip_file`

例: `createRemoteMigrationImage.bat C:%migrationImage.zip`

- 3) マイグレーション・イメージの .zip ファイルをターゲット・システムからソース・システム (バージョン 6.1.x および 6.0.2.x の WebSphere Process Server がインストールされているシステム) にコピーします。
- 4) `migration_copy` という新規ディレクトリーにマイグレーション・イメージ・ファイルを `unzip` します。

注: 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。

- b. WebSphere Process Server バージョン 6.2 DVD の JDK および `migration` ディレクトリーからファイルをコピーします。これを行うには、以下の手順を使用します。WPS/ESB バージョン 6.2 DVD の `migration` および JDK ディレクトリーを新規ディレクトリーにコピーします。例として、ここでは説明のために `migration_copy` を使用します。

- 1) `migration_copy` という新規ディレクトリーをソース・システムに作成します。

注: 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。

- 2) `migration` および JDK ディレクトリーをこの新規ディレクトリーにコピーします。

これで、ソース・システムに以下のディレクトリー構造が作成されます。

```
migration_copy/  
  migration/  
    JDK/
```

2. `migration_copy/migration/bin` ディレクトリーにナビゲートします。
3. `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。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。詳しくは、**WBIPreUpgrade** コマンド行ユーティリティーを参照してください。使用しているオペレーティング・システムに応じて、次のいずれかのスクリプトを使用します。

- **i5/OS** i5/OS プラットフォームの場合: `./WBIPreUpgrade`  
`/filepath/migration_specific_backup currentWebSphereDirectory`

注: i5/OS プラットフォームの場合、このユーザー・プロファイルには \*ALLOBJ 権限が必要です。さらに、i5/OS スクリプトを QSHELL で実行する必要があります。QSHELL セッションを開始するには、CL コマンド・プロンプトを開いて QSH と入力します。

- **Linux** **UNIX** Linux/UNIX プラットフォームの場合:  
`./WBIPreUpgrade.sh /filepath/migration_specific_backup`  
`currentWebSphereDirectory`

- **Windows** **Windows プラットフォームの場合:** `WBIPreUpgrade`  
`C:\filepath\migration_specific_backup currentWebSphereDirectory`

**WBIPreUpgrade** コマンドは、画面に状況を表示し、`migration_specific_backup/logs` ディレクトリーのログ・ファイルに状況を出力します。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・スタンプがファイル名の一部として指定されます。

4. 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。
5. 使用しているオペレーティング・システムに応じて、以下の手順を使用してシステムをバックアップします。

- **i5/OS** **i5/OS プラットフォームの場合:** 以下のバックアップ作業を実行して `i5/OS V5R3` から `i5/OS V5R4` にマイグレーションします。

- `WBIPreUpgrade` コマンドを使用して、`i5/OS V5R3` で構成されたそれぞれの WebSphere Process Server または WebSphere Application Server スタンドアロン・プロファイルをバックアップします。
- データベース・コレクションに対して `i5/OS` ネイティブの `save` コマンドを使用することによって、各プロファイルに関連付けられたデータベース・スキーマをすべてバックアップします。例えば、`crtsavf` と `savlib` です。
- これらのプロファイルでカスタマイズした特別な設定をすべてバックアップします。
- バージョン 6.1.x の各プロファイルの下の `configuration directories/files` に、`i5/OS` ユーザー・プロファイルに割り当てた可能性があるすべての特殊権限を記録しておきます。ターゲット・プロファイルの下の専用権限は、WebSphere Process Server バージョン 6.2 にマイグレーションした後に手動でリストアできます。

- **Linux** **UNIX** **Windows** **Linux、UNIX、および Windows プラットフォームの場合:** (`TAR` や `ZIP` などのユーティリティーを使用して) バックアップ・ディレクトリーを圧縮し、`FTP` を使用してそのファイルを別のシステムにコピーします。

6. 使用しているオペレーティング・システムに応じて、以下の手順を使用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。

- **i5/OS** **i5/OS プラットフォームの場合:** 以下の手順を使用して、新しい `i5/OS V5R4` または `i5/OS V6R1` オペレーティング・システムをインストールします。

- a. 以下のように、インストールするオペレーティング・システムに応じたユーザーへのメモに記載されている説明に従って `i5/OS V5R4` または `i5/OS V6R1` を `i5/OS V5R3` の上にスリップ・インストールします。
  - `i5/OS V5R4` をインストールする場合に使用する手順については、  
[http://www-912.ibm.com/s\\_dir/sline003.nsf/3a8f58452f9800bc862562900059e09e/bdb2077acff30ff28625710f005ca12f?OpenDocument](http://www-912.ibm.com/s_dir/sline003.nsf/3a8f58452f9800bc862562900059e09e/bdb2077acff30ff28625710f005ca12f?OpenDocument) を参照してください。

- i5/OS V6R1 をインストールする場合に使用する手順については、  
[http://www-912.ibm.com/s\\_dir/sline003.nsf/PSP Number View/SF98026](http://www-912.ibm.com/s_dir/sline003.nsf/PSP Number View/SF98026) を参照してください。また、リンク <http://www-03.ibm.com/systems/i/software/websphere/news/sitenews.html> で、『What's New』のセクションの『July 08, 2008』の下にある説明も参照してください。このセクションでは、V6R1 での WebSphere Application Server バージョン 6.1 に関する変更点 (特に HTTP Server プラグインの変更点) について説明しています。
  - b. スリップ・インストールしたオペレーティング・システムに応じた最新の PTF (プログラム一時修正) を適用します。
  - c. サポート製品 (DB2 for i5/OS、Java、IBM Toolbox for Java、IBM HTTP Server など) の最新の PTF グループを適用します。
- Linux UNIX Windows **Linux、UNIX、および Windows プラットフォームの場合:** ホスト名を変えずに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。
7. 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。
  8. 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。

9. **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 7 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、WBIPostUpgrade コマンド行ユーティリティー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
```

**重要:** WBIPostUpgrade を呼び出すときには、`-createTargetProfile` パラメーターを使用します。このオプションにより、マイグレーションに必要な対応する新しいターゲット・プロファイルが作成されます。ターゲット・プロファイルについて詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。

**注:** i5/OS **i5/OS プラットフォーム** でマイグレーションする場合、ターゲット・プロファイル名は、マイグレーション対象のソース・プロファイルのプロファイル名に一致していなければなりません。

10. 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。新しいバージョンの WebSphere Process Server に必要なデータベース変更の一部は、自動的に行われます。ただし、Business Process Choreographer または ビジネス・スペース が既に構成されている場合、またはデータベース・スキーマにアクセスするための十分な権限がサーバーに設定さ

れていない場合、あるいはその他のデータベース固有の要件を満たしていない場合は、データベースを手動で更新する必要があります。詳しくは、39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。

## タスクの結果

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、145ページの『Business Process Choreographer の事後マイグレーション・タスク』を参照してください。サーバーでビジネス・スペースが構成されている場合は、149ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』を参照してください。最後に、143ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

13ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

4ページの『WebSphere Process Server に関する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

135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。

165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。

143ページの『WebSphere Process Server の事後マイグレーション・タスク』

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 6.2 サーバーを構成しなければならない場合があります。

67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

#### WBIPreUpgrade コマンド行ユーティリティー

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。

#### WBIPostUpgrade コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された `backupDirectory` に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

#### 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。

---

## 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 4 ページの『WebSphere Process Server に関する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 6.2 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。クラスター環境内にはない管理対象ノードについては、93 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、108 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。

**注:** マイグレーション前に、ビジネス・ルール・マネージャーがデプロイメント・ターゲット (サーバーまたはクラスター) で稼働中の場合、マイグレーションを開始する前に 133 ページの『ネットワーク・デプロイメント環境でのビジネス・ルール・マネージャーのマイグレーション』を確認する必要があります。

**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、165 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4 ページの『WebSphere Process Server に関する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。(WebSphere Process Server バージョン 6.1 デプロイメント・マネージャーから WebSphere Process Server バージョン 6.2 デプロイメント・マネージャーへのマイグレーションなど。)

- 82 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』

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。

- 88 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、165 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

108 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:** 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 **i5/OS プラットフォーム** または **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。

**注:** マイグレーション・ウィザードは WebSphere Process Server プロファイルのみをサポートします。WebSphere Application Server プロファイルがある場合、マイグレーション・コマンドを使用する必要があります。

**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 6.1.x および 6.0.2.x デプロイメント・マネージャーをバージョン 6.2 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 6.2 デプロイメント・マネージャーにより、バージョン 6.1.x および バージョン 6.2 の管理対象ノードを管理することができます。バージョン 6.1.x からのマイグレーションの場合は、これによって、セルを一度に 1 ノードずつ新規リリースにアップグレードできるため、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ることができます。バージョン 6.0.2.x からのマイグレーションで、ダウン時間を最小限に抑えながらセルを稼働させるには、バージョン 6.2 へのマイグレーションを 2 段階に分けて行います。最初に バージョン 6.0.2.x から バージョン 6.1.x へマイグレーションし、次に バージョン 6.1.x から バージョン 6.2 へマイグレーションします。あるいは、セル全体をシャットダウンし、1 回のステップでバージョン 6.0.2.x からバージョン 6.2 に直接マイグレーションすることもできます。

クラスターのメンバーで、WebSphere Process Server の異なるバージョン (6.0.2.x、6.1.x、6.2) を実行することはできません。異なるバージョンを実行するサーバーが含まれているクラスターを構成した場合は、最初のバージョン 6.2 のクラスター・メンバーを開始する前に、WebSphere Process Server の以前のバージョンを実行するすべてのメンバーを停止する必要があります。また、一度バージョン 6.2 のクラスター・メンバーを開始した後は、そのクラスター内で バージョン 6.1.x および 6.0.2.x のクラスター・メンバーを開始しないでください。

リリースが混在したセルの使用には制限があります。詳しくは、ほかの WebSphere 製品のインストール済み環境との共存のサブトピックにある『リリースの混在したセルの使用に関する制限』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。

- 古いバージョンが存在するシステムに WebSphere Process Server の新規バージョンがインストールされて共存している。
- 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は、4 ページの『WebSphere Process Server に関する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 6.0.2.x コンポーネントまたは バージョン 6.1.x 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。

### 手順

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、WBIPreUpgrade コマンド行ユーティリティー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、WBIPreUpgrade コマンド行ユーティリティーを参照してください。

**管理セキュリティ・ユーザー名 (管理セキュリティが構成されている場合に必須)** -username パラメーターについては、WBIPostUpgrade コマンド行ユーティリティーを参照してください。

**管理セキュリティ・パスワード (管理セキュリティが構成されている場合に必須)** -password パラメーターについては、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティを参照してください。

注: これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

3. オプション: マイグレーション・ウィザードで **Create new profile** オプションを使用して、新規のバージョン 6.2 プロファイルをマイグレーション・ターゲットとして作成します。ターゲット・プロファイルの作成について詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』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。

次の構文を使用します。

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

セキュリティ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。 `stopManager` コマンドについて詳しくは、『`stopManager`コマンド』を参照してください。

注: マイグレーション・プロセスを開始する前に、必ず旧バージョンのデプロイメント・マネージャーを停止してください。停止しない場合、マイグレーション・プロセス中に行った構成変更がターゲット・プロファイルにマイグレーションされません。

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。
  - **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
`wbi_migration.sh`
  - **Windows** **Windows** プラットフォーム: `wbi_migration.bat`

注: マイグレーション・ウィザードを起動するとき、デフォルトのトレース設定 (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) をオプションで変更できます。デフォルトのトレース設定では、特定のクラスでのみトレースが有効になりますが、完全トレースを有効にするか、またはすべてのトレースを無効にするようにデフォルトを変更できます。

- 完全トレースを有効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

- **Linux** **UNIX** **Linux および UNIX プラットフォーム:**  
wbi\_migration.sh -W -migrationPanel.traceString="\*=all=enabled"

- **Windows** **Windows プラットフォーム:** wbi\_migration.bat -W  
-migrationPanel.traceString="\*=all=enabled"

- すべてのトレースを無効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

- **Linux** **UNIX** **Linux および UNIX プラットフォーム:**  
wbi\_migration.sh -W -migrationPanel.traceString="\*=all=disabled"

- **Windows** **Windows プラットフォーム:** wbi\_migration.bat -W  
-migrationPanel.traceString="\*=all=disabled"

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、21 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. 23 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、40 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、データベース表が新規スキーマ・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 6.2 デプロイメント・マネージャーを始動します。

**注意:**

バージョン 6.2 デプロイメント・マネージャーを開始すると、統合ノードはマイグレーション済みのデプロイメント・マネージャーと同期を取ります。この同期により、アプリケーションは再インストールされます。アクティブ・サーバーで実行中のアプリケーションが存在する場合はこれらのアプリケーションが再始動されるため、わずかな時間ですがアプリケーションが使用できなくなる時間が発生します。

デプロイメント・マネージャーを開始するには、`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 6.1.x および 6.0.2.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。

このステップは、削除予定のデプロイメント・マネージャーの構成が正しくマイグレーションされたことを確認してから実行してください。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。

## タスクの結果

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。このデプロイメント・マネージャーによって管理されているサーバーまたはクラスターに Business Process Choreographer が構成されている場合、13 ページの『Business Process Choreographer に関する事前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、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

4 ページの『WebSphere Process Server に関する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。

21 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

23 ページの『マイグレーション・ウィザードの実行』

AIX、HP-UX、Linux、Solaris、または Windows システムでマイグレーション・ウィザードを実行して、WebSphere Process Server をマイグレーションします。

88 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』

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。

93 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。

135 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。

165 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』  
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 6.2 サーバーを構成しなければならない場合があります。

39 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。

137 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 6.2 デプロイメント・セルを、バージョン 6.1.x および 6.0.2.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

##### **WBIPostUpgrade** コマンド行ユーティリティー

WebSphere Process Server の **WBIPostUpgrade** コマンドを使用して、指定された *backupDirectory* に **WBIPreUpgrade** コマンドによって保存されたプロファイル構成を取得します。

##### **WBIPreUpgrade** コマンド行ユーティリティー

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。

#### 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。

 ほかの 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 6.2 のインストール済み環境は、同一システム上で任意のバージョンの WebSphere Enterprise Service Bus、WebSphere Process Server、および精選された WebSphere 製品と共存できます。

## 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:** WebSphere Process Server Network Deployment セルでは、必ずデプロイメント・マネージャーが WebSphere Process Server プロファイル内になければなりません。

**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 6.1.x および 6.0.2.x デプロイメント・マネージャーをバージョン 6.2 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 6.2 デプロイメント・マネージャーにより、バージョン 6.1.x および バージョン 6.2 の管理対象ノードを管理することができます。バージョン 6.1.x からのマイグレーションの場合は、これによって、セルを一度に 1 ノードずつ新規リリースにアップグレードできるため、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ることができます。バージョン 6.0.2.x からのマイグレーションで、ダウン時間を最小限に抑えながらセルを稼働させるには、バージョン 6.2 へのマイグレーションを 2 段階に分けて行います。最初に バージョン 6.0.2.x から バージョン 6.1.x へマイグレーションし、次に バージョン 6.1.x から バージョン 6.2へマイグレーションします。あるいは、セル全体をシャットダウンし、1 回のステップでバージョン 6.0.2.x からバージョン 6.2 に直接マイグレーションすることもできます。

クラスタのメンバーで、WebSphere Process Server の異なるバージョン (6.0.2.x、6.1.x、6.2) を実行することはできません。異なるバージョンを実行するサーバーが含まれているクラスタを構成した場合は、最初のバージョン 6.2 のクラスタ・メンバーを開始する前に、WebSphere Process Server の以前のバージョンを実行するすべてのメンバーを停止する必要があります。また、一度バージョン 6.2 のクラスタ・メンバーを開始した後は、そのクラスタ内で バージョン 6.1.x および 6.0.2.x のクラスタ・メンバーを開始しないでください。

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。
- 古いバージョンが存在するシステムに WebSphere Process Server の新規バージョンがインストールされて共存している。
- 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は、4 ページの『WebSphere Process Server に関する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 6.0.2.x コンポーネントまたは バージョン 6.1.x 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。

## 手順

1. ご使用のオペレーティング・システムに応じて、次のいずれかの手順でログオンします。
  - **i5/OS** **i5/OS プラットフォームの場合:** \*SECOFR ユーザー・クラスまたは \*ALLOBJ 特殊権限を持つ i5/OS ユーザー・プロファイルを使用してログオンします。
  - **Linux** **UNIX** **Linux および UNIX プラットフォームの場合:** root ユーザーとしてログオンします。
  - **Windows** **Windows プラットフォームの場合:** 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、WBIPreUpgrade コマンド行ユーティリティー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、WBIPreUpgrade コマンド行ユーティリティーを参照してください。

**管理セキュリティ・ユーザー名 (管理セキュリティが構成されている場合に必須)** -username パラメーターについては、WBIPostUpgrade コマンド行ユーティリティーを参照してください。

**管理セキュリティ・パスワード (管理セキュリティが構成されている場合に必須)** -password パラメーターについては、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティを参照してください。

注: これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

3. 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。

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。次の構文を使用します。

注: i5/OS プラットフォームの場合は、QSHHELL でスクリプトを実行する必要があります。QSHHELL セッションを開始するには、CL コマンド・プロンプトを開いて QSH と入力します。

- **i5/OS** i5/OS プラットフォームの場合: *profile\_root/bin/stopManager*
- **Linux** **UNIX** Linux および UNIX プラットフォームの場合:  
*profile\_root/bin/stopManager.sh*
- **Windows** Windows プラットフォームの場合:  
*profile\_root%bin%stopManager.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。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。

- **i5/OS** i5/OS プラットフォームの場合: *profile\_root/bin/stopManager -username user\_ID -password password*
- **Linux** **UNIX** Linux および UNIX プラットフォームの場合:  
*profile\_root/bin/stopManager.sh -username user\_ID -password password*
- **Windows** Windows プラットフォームの場合:  
*profile\_root%bin%stopManager.bat -username user\_ID -password password*

注: マイグレーション・プロセスを開始する前に、必ず旧バージョンのデプロイメント・マネージャーを停止してください。停止しない場合、マイグレーション・プロセス中に行った構成変更がターゲット・プロファイルにマイグレーションされません。

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、WBIPreUpgrade コマンド行ユーティリティ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。 バックアップ・ディレクトリーにバックアップされた構成が、WBIPostUpgrade ツールによって、新しい WebSphere Process Server デプロイメント・マネージャー・プロファイルにリストアされます。詳しくは、WBIPostUpgrade コマンド行ユーティリティーを参照してください。

**重要:** WBIPostUpgrade を呼び出すときには、`-createTargetProfile` パラメーターを使用します。このオプションにより、マイグレーションに必要な対応する新しいターゲット・プロファイルが作成されます。ターゲット・プロファイルについて詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。

**注:** i5/OS **i5/OS プラットフォーム** でマイグレーションする場合、ターゲット・プロファイル名は、マイグレーション対象のソース・プロファイルのプロファイル名に一致していなければなりません。

6. 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、40 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、データベース表が新規スキーマ・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。

7. バージョン 6.2 デプロイメント・マネージャーを始動します。

**注意:**

バージョン 6.2 デプロイメント・マネージャーを開始すると、統合ノードはマイグレーション済みのデプロイメント・マネージャーと同期を取ります。この同期により、アプリケーションは再インストールされます。アクティブ・サーバーで実行中のアプリケーションが存在する場合はこれらのアプリケーションが再始動されるため、わずかな時間ですがアプリケーションが使用できなくなる時間が発生します。

デプロイメント・マネージャーを開始するには、`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。

8. オプション: バージョン 6.1.x および 6.0.2.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。

このステップは、削除予定のデプロイメント・マネージャーの構成が正しくマイグレーションされたことを確認してから実行してください。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。

## タスクの結果

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。このデプロイメント・マネージャーによって管理されているサーバーまたはクラスターに Business Process Choreographer が構成されている場合、13 ページの『Business Process Choreographer に関する事前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、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

4 ページの『WebSphere Process Server に関する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。

21 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

82 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。

93 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。

135 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。

165 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』

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 6.2 サーバーを構成しなければならない場合があります。

39 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。

137 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 6.2 デプロイメント・セルを、バージョン  
6.1.x および 6.0.2.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

#### WBIPostUpgrade コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された  
*backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構  
成を取得します。

#### WBIPreUpgrade コマンド行ユーティリティー

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。

#### 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。

#### ほかの 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 6.2 のインストール済み環境は、同一シス  
テム上で任意のバージョンの WebSphere Enterprise Service Bus、WebSphere  
Process Server、および精選された WebSphere 製品と共存できます。

## 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 4 ページの『WebSphere Process Server に関する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、108 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、165 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

108 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。

121 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。

82 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。

88 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』

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

**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 **i5/OS プラットフォーム** または **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。

注: マイグレーション・ウィザードは WebSphere Process Server プロファイルのみをサポートします。 WebSphere Application Server プロファイルがある場合、マイグレーション・コマンドを使用する必要があります。

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。
- 古いバージョンが存在するシステムに WebSphere Process Server の新規バージョンがインストールされて共存している。
- 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、4 ページの『WebSphere Process Server に関する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 6.1.x および 6.0.2.x 管理対象ノードをバージョン 6.2 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 6.1.x および 6.0.2.x デプロイメント・マネージャーをバージョン 6.2 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、81 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 6.1.x および 6.0.2.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。

## 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後は、バージョン 6.2 デプロイメント・マネージャーがバージョン 6.1.x とバージョン 6.2 のノードの両方を管理できます。以前のバージョン 6.1.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 6.1.x 管理対象ノードとしてバージョン 6.2 デプロイメント・マネージャー内で実行されるようになります。

バージョン 6.1.x WebSphere Process Server 管理対象ノード (バージョン 6.2 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 6.2 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。バージョン 6.1.x のすべての管理対象ノードのマイグレーションが終了したら、**convertScriptCompatibility**

スクリプトを使用して、デプロイメント・マネージャーの後方互換性サポートを、バージョン 6.1.x の管理スクリプトの後方互換性サポートから、バージョン 6.1.x およびバージョン 6.2 の管理スクリプト限定の後方互換性サポートに変更します。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、165 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。

## 手順

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 6.1.x またはバージョン 6.0.2.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。次の構文を使用します。

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

• **Windows** Windows プラットフォームの場合:  
`profile_root%bin%stopServer.bat server_name`

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。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。

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

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

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: マイグレーション・プロセスを開始する前に、マイグレーション元のサーバーを停止する必要があります。構成をマイグレーションする場合、マイグレ

ーション元のサーバーを稼働させる必要はありません。マイグレーション・ツールにより、サーバーが停止している間に、すべての構成データが取得されます。

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
```

4. 以下にリストしたマイグレーションに必要な既存情報を、あらかじめ確認しておいてください。

#### 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、WBIPreUpgrade コマンド行ユーティリティー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、WBIPreUpgrade コマンド行ユーティリティーを参照してください。

#### 管理セキュリティー・ユーザー名 (管理セキュリティーが構成されている場合に必須)

*-username* パラメーターについては、WBIPostUpgrade コマンド行ユーティリティーを参照してください。

#### 管理セキュリティー・パスワード (管理セキュリティーが構成されている場合に必須)

*-password* パラメーターについては、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティーを参照してください。

注: これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

- バージョン 6.2 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。
- 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。

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

`wbi_migration.sh`

- **Windows** **Windows プラットフォーム: wbi\_migration.bat**

注: マイグレーション・ウィザードを起動するとき、デフォルトのトレース設定 (`*=all=enabled:com.ibm.ws.migration.common.*=all=disabled`) をオプションで変更できます。デフォルトのトレース設定では、特定のクラスでのみトレースが有効になりますが、完全トレースを有効にするか、またはすべてのトレースを無効にするようにデフォルトを変更できます。

- 完全トレースを有効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

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

`wbi_migration.sh -W -migrationPanel.traceString="*=all=enabled"`

- **Windows** **Windows プラットフォーム: wbi\_migration.bat -W**

`-migrationPanel.traceString="*=all=enabled"`

- すべてのトレースを無効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

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

`wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`

- **Windows** **Windows プラットフォーム: wbi\_migration.bat -W**

`-migrationPanel.traceString="*=all=disabled"`

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 6.1.x および 6.0.2.x 管理対象ノードからバージョン 6.2 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 6.2 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。

8. Business Process Choreographer またはビジネス・スペースが構成されているサーバーをマイグレーションする場合は、この時点で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ります。43 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』または 54 ページの『手動による ビジネス・スペース データベースのアップグレード』を参照してください。
9. 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`
10. 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。
11. 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 10 を繰り返します。
12. 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 6.2 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 6.2 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。

注: これは、バージョン 6.0.2.x からマイグレーションする場合にのみ該当します。

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。

## タスクの結果

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、145 ページの『Business Process Choreographer の事後マイグレーション・タスク』を参照してください。サーバーでビジネス・スペース が構成されている場合は、149 ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』を参照してください。最後に、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

## 関連概念

4 ページの『WebSphere Process Server に関する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。

21 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

23 ページの『マイグレーション・ウィザードの実行』

AIX、HP-UX、Linux、Solaris、または Windows システムでマイグレーション・ウィザードを実行して、WebSphere Process Server をマイグレーションします。

101 ページの『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。

### サーバーの停止

サーバーの構成を変更する場合、またはサーバーにデプロイされているモジュールを変更する場合は、サーバー・プロセスを停止する必要があります。単一サーバー (クラスター・メンバーを含む) を停止できます。

43 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 が構成されたサーバーまたはクラスターのマイグレーション後、Business Process Choreographer データベースのスキーマを手動でアップグレードし、サーバーまたはクラスター・メンバーを始動する前にデータをマイグレーションする必要があります。

39 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。

140 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 6.2 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』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 6.2 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

#### WBIPostUpgrade コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された *backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

#### WBIPreUpgrade コマンド行ユーティリティー

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。

### 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

**注:** コマンド行ツールを使用してマイグレーションする場合、WebSphere Process Server のプロファイルまたは WebSphere Application Server のプロファイルを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・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。
- 古いバージョンが存在するシステムに WebSphere Process Server の新規バージョンがインストールされて共存している。
- 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、4ページの『WebSphere Process Server に関する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 6.1.x および 6.0.2.x 管理対象ノードをバージョン 6.2 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 6.1.x および 6.0.2.x デプロイメント・マネージャーをバージョン 6.2 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、81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 6.1.x および 6.0.2.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。

## このタスクについて

バージョン 6.0.2.x 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、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後は、バージョン 6.2 デプロイメント・マネージャーがバージョン 6.0.2.x とバージョン 6.2 のノードの両方を管理できます。つまり、バージョン 6.0.2.x 管理対象ノードはバージョン 6.2 のデプロイメント・マネージャーで実行できます。バージョン 6.0.2.x WebSphere Process Server 管理対象ノード (バージョン 6.2 のデプロイメント・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 6.2 管理対象ノードにそれぞれ順次マイグレーションできます。すべてのバージョン 6.0.2.x 管理対象ノードをマイグレーションしたら、convertScriptCompatibility スクリプトを使用して、バージョン 6.0.2.x 管理スクリプトの後方互換性をサポートするモードから、完全にバージョン 6.2 の構成モデルとなっているモードに構成を変換します。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。

## 手順

1. ご使用のオペレーティング・システムに応じて、次のいずれかの手順でログオンします。
  - **i5/OS** **i5/OS プラットフォームの場合:** \*SEC0FR ユーザー・クラスまたは \*ALLOBJ 特殊権限を持つ i5/OS ユーザー・プロファイルを使用してログオンします。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合: root ユーザーとしてログオンします。
  - **Windows** **Windows** プラットフォームの場合: 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 6.1.x および 6.0.2.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。次の構文を使用します。

**注: i5/OS プラットフォームの場合は、QShell でスクリプトを実行する必要があります。QShell セッションを開始するには、CL コマンド・プロンプトを開いて QSH と入力します。**

- **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/stopServer server_name`
- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合: `profile_root/bin/stopServer.sh server_name`
- **Windows** **Windows** プラットフォームの場合: `profile_root%bin%stopServer.bat server_name`

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。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。

- **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/stopServer server_name -username user_ID -password password`
- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合: `profile_root/bin/stopServer.sh server_name -username user_ID -password password`
- **Windows** **Windows** プラットフォームの場合: `profile_root%bin%stopServer.bat server_name -username user_ID -password password`

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る場合は、`-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:** マイグレーション・プロセスを開始する前に、サーバーを停止します。デフォルトでは、ノード上のすべてのサーバーが、マイグレーション・プロセスの開始前に停止されます。

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)。

- **i5/OS** **i5/OS** プラットフォームの場合: *profile\_root/bin/stopNode*

- **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*

セキュリティが有効な場合は、代わりに以下のコマンドのいずれかを使用します。

- **i5/OS** **i5/OS** プラットフォームの場合: *profile\_root/bin/stopNode -username user\_ID -password password*

- **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: マイグレーション・プロセスを開始する前に、必ず古いノードを停止してください。サーバー構成をマイグレーションするため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。

4. 以下にリストしたマイグレーションに必要な既存情報を、あらかじめ確認しておいてください。

#### 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、WBIPreUpgrade コマンド行ユーティリティー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、WBIPreUpgrade コマンド行ユーティリティーを参照してください。

#### 管理セキュリティ・ユーザー名 (管理セキュリティが構成されている場合に必須)

*-username* パラメーターについては、WBIPostUpgrade コマンド行ユーティリティーを参照してください。

#### 管理セキュリティ・パスワード (管理セキュリティが構成されている場合に必須)

*-password* パラメーターについては、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティー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、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティを参照してください。

**注:** これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

- バージョン 6.2 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。
- 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 ツールによって、既存のプロファイルの構成ファイルが指定のバックアップ・ディレクトリーに保存されます。
- 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。バックアップ・ディレクトリーにバックアップされた構成が、WBIPostUpgrade ツールによって、新しい WebSphere Process Server デプロイメント・マネージャー・プロファイルにリストアされません。

**重要:** WBIPostUpgrade を呼び出すときには、-createTargetProfile パラメーターを使用します。このオプションにより、マイグレーションに必要な対応する新しいターゲット・プロファイルが作成されます。ターゲット・プロファイルについて詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。

**注:** **i5/OS** **i5/OS** プラットフォーム でマイグレーションする場合、ターゲット・プロファイル名は、マイグレーション対象のソース・プロファイルのプロファイル名に一致していなければなりません。

- 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 またはビジネス・スペースが構成されているサーバーをマイグレーションする場合は、この時点で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ります。43 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』または 54 ページの『手動による ビジネス・スペース 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)。

• **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/startNode`

• **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管理対象ノードごとに、ステップ 1 から 11 を繰り返します。

注: 再度ステップ 6 を実行 (WBIPreUpgrade を実行) する必要があるのは、バージョン 6.1.x からマイグレーションしている場合か、バージョン 6.0.2.x からマイグレーションしており、かつ最初に WBIPreUpgrade を実行した後にバージョン 6.0.2.x システムが再構成された場合のみです。

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 6.2 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 6.2 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。

注: これは、バージョン 6.0.2.x からマイグレーションする場合にのみ該当します。

使用しているオペレーティング・システムに応じて、次のいずれかの `convertScriptCompatibility` コマンドを `bin` ディレクトリーから発行します。

- **i5/OS** i5/OS プラットフォームの場合: `install_root/bin/convertScriptCompatibility`
- **Linux** **UNIX** Linux/UNIX プラットフォームの場合: `install_root/bin/convertScriptCompatibility.sh`
- **Windows** Windows プラットフォームの場合: `install_root%bin%convertScriptCompatibility.bat`

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

## タスクの結果

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、145 ページの『Business Process Choreographer の事後マイグレーション・タスク』を参照してください。サーバーでビジネス・スペース が構成されている場合は、149 ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』を参照してください。最後に、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

4 ページの『WebSphere Process Server に関する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。

21 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

94 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。

#### サーバーの停止

サーバーの構成を変更する場合、またはサーバーにデプロイされているモジュールを変更する場合は、サーバー・プロセスを停止する必要があります。単一サーバー (クラスター・メンバーを含む) を停止できます。

43 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 が構成されたサーバーまたはクラスターのマイグレーション後、Business Process Choreographer データベースのスキーマを手動でアップグレードし、サーバーまたはクラスター・メンバーを始動する前にデータをマイグレーションする必要があります。

39 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。

140 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 6.2 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』  
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 6.2 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

 [WBIPostUpgrade コマンド行ユーティリティー](#)

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された  
*backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構  
成を取得します。

 [WBIPreUpgrade コマンド行ユーティリティー](#)

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。

#### 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。

クラスター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、クラスター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、クラスターごとに 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

81 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

新しいバージョン (例えば、バージョン 6.2) 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 6.1.x および 6.0.2.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。

**重要:** クラスター内で、バージョン 6.1.x および 6.0.2.x メンバーとバージョン 6.2 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 6.2 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 6.1.x および 6.0.2.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 6.2 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 6.1.x および 6.0.2.x クラスター・メンバーを開始しないでください。

## 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。

### 手順

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。 81 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. クラスター・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、111 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 116 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 6.2 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. ステップ3b (109 ページ)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、`WBIPProfileUpgrade` スクリプトを参照してください。
8. Business Process Choreographer またはビジネス・スペースが構成されているクラスターをマイグレーションする場合は、この時点で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ります。43 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』または 54 ページの『手動によるビジネス・スペース 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。

## タスクの結果

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。クラスターで Business Process Choreographer が構成されている場合は、145 ページの『Business Process Choreographer の事後マイグレーション・タスク』を参照してください。サーバーでビジネス・スペースが構成されている場合は、149 ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』を参照してください。最後に、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

### 関連タスク

81 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。

93 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。

121 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。

43 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 が構成されたサーバーまたはクラスターのマイグレーション後、Business Process Choreographer データベースのスキーマを手動でアップグレードし、サーバーまたはクラスター・メンバーを始動する前にデータをマイグレーションする必要があります。

135 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。

165 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』

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 6.2 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

 **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。

**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:** 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 **i5/OS プラットフォーム** または **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。

**注:** マイグレーション・ウィザードは WebSphere Process Server プロファイルのみをサポートします。WebSphere Application Server プロファイルがある場合、マイグレーション・コマンドを使用する必要があります。

**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、108 ページの『クラスターのマイグレーション』または 121 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』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・システムで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旧バージョンと共存させている。
- 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、4 ページの『WebSphere Process Server に関する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 6.1.x および 6.0.2.x 管理対象ノードをバージョン 6.2 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 6.1.x および 6.0.2.x デプロイメント・マネージャーをバージョン 6.2 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、81 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 6.1.x および 6.0.2.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。

### このタスクについて

バージョン 6.0.2.x 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、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後は、バージョン 6.2 デプロイメント・マネージャーがバージョン 6.0.2.x とバージョン 6.2 のノードの両方を管理できます。つまり、バージョン 6.0.2.x 管理対象ノードはバージョン 6.2 のデプロイメント・マネージャーで実行できます。バージョン 6.0.2.x WebSphere Process Server 管理対象ノード (バージョン 6.2 のデプロイメント・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 6.2 管理対象ノードにそれぞれ順次マイグレーションできます。すべてのバージョン 6.0.2.x 管理対象ノードをマイグレーションしたら、convertScriptCompatibility スクリプトを使用して、バージョン 6.0.2.x 管理スクリプトの後方互換性をサポートするモードから、完全にバージョン 6.2 の構成モデルとなっているモードに構成を変換します。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。

## 手順

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 6.1.x またはバージョン 6.0.2.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。次の構文を使用します。

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

```
profile_root/bin/stopServer.sh server_name
```

- **Windows** Windows プラットフォームの場合:

```
profile_root%bin%stopServer.bat server_name
```

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。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。

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

```
profile_root/bin/stopServer.sh server_name -username user_ID -password password
```

- **Windows** Windows プラットフォームの場合:

```
profile_root%bin%stopServer.bat server_name -username user_ID -password password
```

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: マイグレーション・プロセスを開始する前に、マイグレーション元のサーバーを停止する必要があります。構成をマイグレーションする場合、マイグレーション元のサーバーを稼働させる必要はありません。マイグレーション・ツールにより、サーバーが停止している間に、すべての構成データが取得されます。

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
```

4. 以下にリストしたマイグレーションに必要な既存情報を、あらかじめ確認しておいてください。

#### 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、`WBIPreUpgrade` コマンド行ユーティリティー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、`WBIPreUpgrade` コマンド行ユーティリティーを参照してください。

**管理セキュリティー・ユーザー名** (管理セキュリティーが構成されている場合に必須) `-username` パラメーターについては、`WBIPostUpgrade` コマンド行ユーティリティーを参照してください。

**管理セキュリティー・パスワード** (管理セキュリティーが構成されている場合に必須) `-password` パラメーターについては、`WBIPostUpgrade` コマンド行ユーティリティー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、`WBIPostUpgrade` コマンド行ユーティリティー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、`WBIPostUpgrade` コマンド行ユーティリティー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、`WBIPostUpgrade` コマンド行ユーティリティーを参照してください。

**注:** これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

5. バージョン 6.2 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。
6. 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。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
wbi\_migration.sh

- **Windows** **Windows** プラットフォーム: wbi\_migration.bat

注: マイグレーション・ウィザードを起動するとき、デフォルトのトレース設定 (\*=**all=enabled**:com.ibm.ws.migration.common.\*=**all=disabled**) をオプションで変更できます。デフォルトのトレース設定では、特定のクラスでのみトレースが有効になりますが、完全トレースを有効にするか、またはすべてのトレースを無効にするようにデフォルトを変更できます。

- 完全トレースを有効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
wbi\_migration.sh -W -migrationPanel.traceString="\*=**all=enabled**"

- **Windows** **Windows** プラットフォーム: wbi\_migration.bat -W  
-migrationPanel.traceString="\*=**all=enabled**"

- すべてのトレースを無効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
wbi\_migration.sh -W -migrationPanel.traceString="\*=**all=disabled**"

- **Windows** **Windows** プラットフォーム: wbi\_migration.bat -W  
-migrationPanel.traceString="\*=**all=disabled**"

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 6.1.x および 6.0.2.x 管理対象ノードからバージョン 6.2 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 6.2 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。

7. 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 6 (98 ページ) を繰り返します。
8. 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 6.2 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 6.2 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。

注: これは、バージョン 6.0.2.x からマイグレーションする場合にのみ該当します。

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。

## タスクの結果

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。

## 次のタスク

108 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 121 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、108 ページの『クラスターのマイグレーション』または 121 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。

**注:** コマンド行ツールを使用してマイグレーションする場合、WebSphere Process Server のプロファイルまたは WebSphere Application Server のプロファイルを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・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・システムで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旧バージョンと共存させている。
- 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、4 ページの『WebSphere Process Server に関する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 6.1.x および 6.0.2.x 管理対象ノードをバージョン 6.2 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 6.1.x および 6.0.2.x デプロイメント・マネージャーをバージョン 6.2 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、81 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 6.1.x および 6.0.2.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。

### このタスクについて

バージョン 6.0.2.x 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、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後は、バージョン 6.2 デプロイメント・マネージャーがバージョン 6.0.2.x とバージョン 6.2 のノードの両方を管理できます。つまり、バージョン 6.0.2.x 管理対象ノードはバージョン 6.2 のデプロイメント・マネージャーで実行できます。バージョン 6.0.2.x WebSphere Process Server 管理対象ノード (バージョン 6.2 のデプロイメント・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 6.2 管理対象ノードにそれぞれ順次マイグレーションできます。すべてのバージョン 6.0.2.x 管理対象ノードをマイグレーションしたら、convertScriptCompatibility スクリプトを使用して、バージョン 6.0.2.x 管理スクリプトの後方互換性をサポートするモードから、完全にバージョン 6.2 の構成モデルとなっているモードに構成を変換します。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。

### 手順

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 6.1.x および 6.0.2.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。次の構文を使用します。

**注:** i5/OS プラットフォームの場合は、QSHHELL でスクリプトを実行する必要があります。QSHHELL セッションを開始するには、CL コマンド・プロンプトを開いて QSH と入力します。

- **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/stopServer server_name`
- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合: `profile_root/bin/stopServer.sh server_name`
- **Windows** **Windows** プラットフォームの場合: `profile_root%bin%stopServer.bat server_name`

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。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。

- **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/stopServer server_name -username user_ID -password password`
- **Linux** **UNIX** **Linux** および **UNIX** プラットフォームの場合: `profile_root/bin/stopServer.sh server_name -username user_ID -password password`
- **Windows** **Windows** プラットフォームの場合: `profile_root%bin%stopServer.bat server_name -username user_ID -password password`

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:** マイグレーション・プロセスを開始する前に、サーバーを停止します。デフォルトでは、ノード上のすべてのサーバーが、マイグレーション・プロセスの開始前に停止されます。

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)。

- **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/stopNode`
- **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`

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

- **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/stopNode -username user_ID -password password`
- **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: マイグレーション・プロセスを開始する前に、必ず古いノードを停止してください。サーバー構成をマイグレーションするため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。

4. 以下にリストしたマイグレーションに必要な既存情報を、あらかじめ確認しておいてください。

**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、WBIPreUpgrade コマンド行ユーティリティ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、WBIPreUpgrade コマンド行ユーティリティを参照してください。

**管理セキュリティ・ユーザー名 (管理セキュリティが構成されている場合に必須)** -username パラメーターについては、WBIPostUpgrade コマンド行ユーティリティを参照してください。

**管理セキュリティ・パスワード (管理セキュリティが構成されている場合に必須)** -password パラメーターについては、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティを参照してください。

注: これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

5. バージョン 6.2 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。
6. 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 ツールによって、既存のプロファイルの構成ファイルが指定のバックアップ・ディレクトリーに保存されます。
7. 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。バックアップ・ディレクトリーにバックアップされた構成が、WBIPostUpgrade ツールによって、新しい WebSphere Process Server デプロイメント・マネージャー・プロファイルにリストアされます。

**重要:** `WBIPostUpgrade` を呼び出すときには、`-createTargetProfile` パラメーターを使用します。このオプションにより、マイグレーションに必要な対応する新しいターゲット・プロファイルが作成されます。ターゲット・プロファイルについて詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。

**注:** **i5/OS** **i5/OS** プラットフォーム でマイグレーションする場合、ターゲット・プロファイル名は、マイグレーション対象のソース・プロファイルのプロファイル名に一致していなければなりません。

8. ステップ 1 から 7 までを繰り返します (ステップ 6 は実行しない場合があります)。

**注:** 再度ステップ 6 を実行 (`WBIPreUpgrade` を実行) する必要があるのは、バージョン 6.1.x からマイグレーションしている場合か、バージョン 6.0.2.x からマイグレーションしており、かつ最初に `WBIPreUpgrade` を実行した後にバージョン 6.0.2.x システムが再構成された場合のみです。同一の `WebSphere Process Server` インストール済み環境にある追加の管理対象プロファイルをマイグレーションする場合はステップ 7 をスキップすることができますが、その際はステップ 1 もステップできる場合があります。

9. **Linux** **UNIX** **Windows** 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 6.2 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 6.2 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。

**注:** このステップは、バージョン 6.0.2.x からマイグレーションする場合のみ実行してください。

**注:** このステップは、i5/OS プラットフォームには適用できません。

`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。

## タスクの結果

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。

## 次のタスク

108 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 121 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

111 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、残りの半分の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

新しいバージョン (例えば、バージョン 6.2) 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 6.1.x および 6.0.2.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。

**重要:** クラスター内で、バージョン 6.1.x および 6.0.2.x メンバーとバージョン 6.2 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 6.2 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 6.1.x および 6.0.2.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 6.2 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 6.0.2.x クラスター・メンバーを開始しないでください。

### このタスクについて

以下のステップを実行すると、ダウン時間を最小限に抑えたまま、新しいバージョンの WebSphere Process Server でクラスター機能を使用できるようになります。

**制約事項:** 以下の手順は、バージョン 6.1.x からバージョン 6.2 にマイグレーションする場合にのみサポートされます。バージョン 6.0.2.x からマイグレーションする場合にクラスターのマイグレーション中のダウン時間を最小限に抑えるには、最初にバージョン 6.1.x へマイグレーションし、次にバージョン 6.2 へマイグレーションする必要があります。

### 手順

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。 81 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. クラスター・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 (121 ページ) 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 (121 ページ) 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。
- ステップ 3 (109 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、クラスター内のすべてのサーバーが停止する時間を削減できます。
  - 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。
  - 最初のプロファイルのセットをそれぞれ 1 つずつマイグレーションしますが、その際、新規ノード・エージェントまたはサーバーを**始動しないでください**。111 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 116 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。
  - 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、クラスター・サービスが利用不可になります。
  - WebSphere Process Server バージョン 6.2 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。
  - 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ーについて、`WBIPProfileUpgrade` スクリプトを実行します。(つまり、ステップ3 (109 ページ)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、`WBIPProfileUpgrade` スクリプトを参照してください。
  - `Business Process Choreographer` またはビジネス・スペースが構成されているクラスターをマイグレーションする場合は、この時点で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ります。43 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』または 54 ページの『手動による `ビジネス・スペース` データベースのアップグレード』を参照してください。
  - 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。
  - 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。最初のプロファイルと同様に、マイグレーションするには、111 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 116 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。

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。

## タスクの結果

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。クラスターで Business Process Choreographer が構成されている場合は、145 ページの『Business Process Choreographer の事後マイグレーション・タスク』を参照してください。クラスターでビジネス・スペースが構成されている場合は、149 ページの『WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク』を参照してください。最後に、143 ページの『WebSphere Process Server の事後マイグレーション・タスク』で説明されている検査を実行します。

### 関連概念

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

### 関連タスク

81 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。

93 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。

108 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。

43 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 が構成されたサーバーまたはクラスターのマイグレーション後、Business Process Choreographer データベースのスキーマを手動でアップグレードし、サーバーまたはクラスター・メンバーを始動する前にデータをマイグレーションする必要があります。

135 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。

165 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。

143 ページの『WebSphere Process Server の事後マイグレーション・タスク』  
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 6.2 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

#### 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。

### 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:** 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 **i5/OS プラットフォーム** または **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。

**注:** マイグレーション・ウィザードは WebSphere Process Server プロファイルのみ  
をサポートします。WebSphere Application Server プロファイルがある場合、マイ  
グレーション・コマンドを使用する必要があります。

**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、108 ページ  
の『クラスターのマイグレーション』または 121 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・システムで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旧バージョンと共存させている。
- 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、4 ペ  
ージの『WebSphere Process Server に関する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 6.1.x および 6.0.2.x 管理対象ノードをバージョン 6.2 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 6.1.x および 6.0.2.x デプロイメント・マネージャーをバージョン 6.2 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、81 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 6.1.x および 6.0.2.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。

### このタスクについて

バージョン 6.0.2.x 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、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後は、バージョン 6.2 デプロイメント・マネージャーがバージョン 6.0.2.x とバージョン 6.2 のノードの両方を管理できます。つまり、バージョン 6.0.2.x 管理対象ノードはバージョン 6.2 のデプロイメント・マネージャーで実行できます。バージョン 6.0.2.x WebSphere Process Server 管理対象ノード (バージョン 6.2 のデプロイメント・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 6.2 管理対象ノードにそれぞれ順次マイグレーションできます。すべてのバージョン 6.0.2.x 管理対象ノードをマイグレーションしたら、convertScriptCompatibility スクリプトを使用して、バージョン 6.0.2.x 管理スクリプトの後方互換性をサポートするモードから、完全にバージョン 6.2 の構成モデルとなっているモードに構成を変換します。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。

### 手順

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 6.1.x またはバージョン 6.0.2.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。次の構文を使用します。

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

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

```
profile_root%bin%stopServer.bat server_name
```

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。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。

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

```
profile_root/bin/stopServer.sh server_name -username user_ID -password password
```

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

```
profile_root%bin%stopServer.bat server_name -username user_ID -password password
```

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:** マイグレーション・プロセスを開始する前に、マイグレーション元のサーバーを停止する必要があります。構成をマイグレーションする場合、マイグレーション元のサーバーを稼働させる必要はありません。マイグレーション・ツールにより、サーバーが停止している間に、すべての構成データが取得されます。

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
```

4. 以下にリストしたマイグレーションに必要な既存情報を、あらかじめ確認しておいてください。

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、`WBIPreUpgrade` コマンド行ユーティリティー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、WBIPreUpgrade コマンド行ユーティリティを参照してください。

**管理セキュリティ・ユーザー名 (管理セキュリティが構成されている場合に必須)** -username パラメーターについては、WBIPostUpgrade コマンド行ユーティリティを参照してください。

**管理セキュリティ・パスワード (管理セキュリティが構成されている場合に必須)** -password パラメーターについては、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティ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、WBIPostUpgrade コマンド行ユーティリティを参照してください。

**注:** これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

- バージョン 6.2 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。
- 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。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
wbi\_migration.sh

- **Windows** **Windows** プラットフォーム: wbi\_migration.bat

**注:** マイグレーション・ウィザードを起動するとき、デフォルトのトレース設定 (\*=all=enabled:com.ibm.ws.migration.common.\*=all=disabled) をオプションで変更できます。デフォルトのトレース設定では、特定のクラスでのみトレースが有効になりますが、完全トレースを有効にするか、またはすべてのトレースを無効にするようにデフォルトを変更できます。

- 完全トレースを有効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。

- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム:  
wbi\_migration.sh -W -migrationPanel.traceString="\*=all=enabled"

- **Windows** **Windows** プラットフォーム: `wbi_migration.bat -W -migrationPanel.traceString="*=all=enabled"`
- すべてのトレースを無効にするには、ご使用のオペレーティング・システムに応じて以下のスクリプトのいずれかを実行し、マイグレーション・ウィザードを呼び出します。
- **Linux** **UNIX** **Linux** および **UNIX** プラットフォーム: `wbi_migration.sh -W -migrationPanel.traceString="*=all=disabled"`
- **Windows** **Windows** プラットフォーム: `wbi_migration.bat -W -migrationPanel.traceString="*=all=disabled"`

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 6.1.x および 6.0.2.x 管理対象ノードからバージョン 6.2 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 6.2 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。

7. 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 6 (98 ページ) を繰り返します。
8. 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 6.2 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 6.2 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。

**注:** これは、バージョン 6.0.2.x からマイグレーションする場合にのみ該当します。

`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。

## タスクの結果

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。

## 次のタスク

108 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 121 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

116 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、108 ページの『クラスターのマイグレーション』または 121 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』の説明に従ってください。

**注:** コマンド行ツールを使用してマイグレーションする場合、WebSphere Process Server のプロファイルまたは WebSphere Application Server のプロファイルを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・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・システムで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旧バージョンと共存させている。
- 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、4 ページの『WebSphere Process Server に関する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 6.1.x および 6.0.2.x 管理対象ノードをバージョン 6.2 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 6.1.x および 6.0.2.x デプロイメント・マネージャーをバージョン 6.2 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、81 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 6.1.x および 6.0.2.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。

### このタスクについて

バージョン 6.0.2.x 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、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後は、バージョン 6.2 デプロイメント・マネージャーがバージョン 6.0.2.x とバージョン 6.2 のノードの両方を管理できます。つまり、バージョン 6.0.2.x 管理対象ノードはバージョン 6.2 のデプロイメント・マネージャーで実行できます。バージョン 6.0.2.x WebSphere Process Server 管理対象ノード (バージョン 6.2 のデプロイメント・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 6.2 管理対象ノードにそれぞれ順次マイグレーションできます。すべてのバージョン 6.0.2.x 管理対象ノードをマイグレーションしたら、convertScriptCompatibility スクリプトを使用して、バージョン 6.0.2.x 管理スクリプトの後方互換性をサポートするモードから、完全にバージョン 6.2 の構成モデルとなっているモードに構成を変換します。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。

### 手順

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 6.1.x および 6.0.2.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。次の構文を使用します。

**注:** i5/OS プラットフォームの場合は、QSHELL でスクリプトを実行する必要があります。QSHELL セッションを開始するには、CL コマンド・プロンプトを開いて QSH と入力します。

- **i5/OS** i5/OS プラットフォームの場合: `profile_root/bin/stopServer server_name`
- **Linux** **UNIX** Linux および UNIX プラットフォームの場合: `profile_root/bin/stopServer.sh server_name`
- **Windows** Windows プラットフォームの場合: `profile_root¥bin¥stopServer.bat server_name`

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。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。

- **i5/OS** i5/OS プラットフォームの場合: `profile_root/bin/stopServer server_name -username user_ID -password password`
- **Linux** **UNIX** Linux および UNIX プラットフォームの場合: `profile_root/bin/stopServer.sh server_name -username user_ID -password password`

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

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:** マイグレーション・プロセスを開始する前に、サーバーを停止します。デフォルトでは、ノード上のすべてのサーバーが、マイグレーション・プロセスの開始前に停止されます。

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)。

- **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/stopNode`

- **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`

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

- **i5/OS** **i5/OS** プラットフォームの場合: `profile_root/bin/stopNode  
 -username user_ID -password password`

- **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:** マイグレーション・プロセスを開始する前に、必ず古いノードを停止してください。サーバー構成をマイグレーションするため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。

4. 以下にリストしたマイグレーションに必要な既存情報を、あらかじめ確認しておいてください。

#### 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、`WBIPreUpgrade` コマンド行ユーティリティー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、`WBIPreUpgrade` コマンド行ユーティリティーを参照してください。

**管理セキュリティ・ユーザー名** (管理セキュリティが構成されている場合に必須) `-username` パラメーターについては、`WBIPostUpgrade` コマンド行ユーティリティを参照してください。

**管理セキュリティ・パスワード** (管理セキュリティが構成されている場合に必須) `-password` パラメーターについては、`WBIPostUpgrade` コマンド行ユーティリティ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、`WBIPostUpgrade` コマンド行ユーティリティ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、`WBIPostUpgrade` コマンド行ユーティリティ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、`WBIPostUpgrade` コマンド行ユーティリティを参照してください。

**注:** これが適用されるのは、バージョン 6.0.2.x からバージョン 6.2 にマイグレーションするときのみです。

- バージョン 6.2 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。
- 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` ツールによって、既存のプロファイルの構成ファイルが指定のバックアップ・ディレクトリーに保存されます。
- 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。 バックアップ・ディレクトリーにバックアップされた構成が、`WBIPostUpgrade` ツールによって、新しい `WebSphere Process Server` デプロイメント・マネージャー・プロファイルにリストアされます。

**重要:** `WBIPostUpgrade` を呼び出すときには、`-createTargetProfile` パラメーターを使用します。このオプションにより、マイグレーションに必要な対応する新しいターゲット・プロファイルが作成されます。ターゲット・プロファイルについて詳しくは、10 ページの『ターゲット・プロファイルの考慮事項』を参照してください。

**注:** i5/OS **i5/OS プラットフォーム** でマイグレーションする場合、ターゲット・プロファイル名は、マイグレーション対象のソース・プロファイルのプロファイル名に一致していなければなりません。

- ステップ 1 から 7 までを繰り返します (ステップ 6 は実行しない場合があります)。

**注:** 再度ステップ 6 を実行 (`WBIPreUpgrade` を実行) する必要があるのは、バージョン 6.1.x からマイグレーションしている場合か、バージョン 6.0.2.x からマイグレーションしており、かつ最初に `WBIPreUpgrade` を実行した後にバージョン 6.0.2.x システムが再構成された場合のみです。同一の `WebSphere Process Server` インストール済み環境にある追加の管理対象プロファイルをマイグレーション

ョンする場合はステップ 7 をスキップすることができますが、その際はステップ 1 もステップできる場合があります。

9. **Linux** **UNIX** **Windows** 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 6.2 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 6.2 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。

**注:** このステップは、バージョン 6.0.2.x からマイグレーションする場合のみ実行してください。

**注:** このステップは、i5/OS プラットフォームには適用できません。

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。

### タスクの結果

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。

### 次のタスク

108 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 121 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

111 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。

## ネットワーク・デプロイメント環境でのビジネス・ルール・マネージャーのマイグレーション

ネットワーク・デプロイメント環境のマイグレーションの一部として、セル内のサーバーまたはクラスターにデプロイ済みのビジネス・ルール・マネージャー・アプリケーションは、セルの最後のノードのマイグレーションまで自動的にマイグレーションされません。その結果、最後のノードがマイグレーションされないサーバーまたはクラスターで稼働するビジネス・ルール・マネージャーは手動でマイグレーションする必要があります。

WebSphere Process Server の前のバージョンのビジネス・ルール・マネージャーは、WebSphere Process Server 6.2 との互換性がありません。また、デプロイメント・マネージャーまたはノードのマイグレーションの一部として、セル内のサーバーまたはクラスターにデプロイ済みのビジネス・ルール・マネージャー・アプリケーションのあらゆるインスタンスは、セルの最後のノードのマイグレーションまで

自動的にマイグレーションされません。環境にとって最も適切なタイミングでマイグレーションするために、それまでは自動的にマイグレーションは実行されません。ビジネス・ルール・マネージャーがデプロイされているデプロイメント・ターゲット (サーバーまたはクラスター) が即時マイグレーションされない環境は、既存の (前バージョンの) ビジネス・ルール・マネージャーによってセル内のビジネス・ルールを管理し続けることができるため利点があります。

ただし、ビジネス・ルール・マネージャーのデプロイメント・ターゲットを含むノードがバージョン 6.2 にマイグレーションされ、そのデプロイメント・ターゲットで稼働するビジネス・ルール・マネージャーがバージョン 6.2 にアップグレードされていない場合は、エラーが発生します。エラーの発生を防ぐために、デプロイメント・ターゲットの通常マイグレーションを実行する前に、ビジネス・ルール・マネージャーを手動でマイグレーションできます。

混合モードで稼働しているセルに WebSphere Process Server 6.2 ノードと前バージョンのノードが含まれている場合、WebSphere Integration Developer 6.2 で作成されたビジネス・ルールを持つアプリケーションがそのセルにインストールされるまで、前バージョンのビジネス・ルール・マネージャーを使用したビジネス・ルールを続けて管理することができます。このバージョンの WebSphere Integration Developer によるビジネス・ルールには、前バージョンのビジネス・ルール・マネージャーではサポートされない機能が含まれています。そのため、これらのルールを変更するとビジネス・ルールが欠落したり、正常に機能しない可能性があります。

**注:** 前バージョンの WebSphere Process Server のスタンドアロン・プロファイルをバージョン 6.2 にマイグレーションする場合、マイグレーション・プロセスの一部としてビジネス・ルール・マネージャー・アプリケーションが自動的にマイグレーションされます。ビジネス・ルール・マネージャーがプロファイルにインストールされていれば、マイグレーションされます。これ以上の構成は必要ありません。

## デプロイメント・ターゲットへのビジネス・ルール・マネージャーのマイグレーション

ビジネス・ルール・マネージャーをデプロイメント・ターゲットにマイグレーションするには、wsadmin コマンドを使用します。

### 手順

デプロイメント・ターゲットのタイプによって、wsadmin コマンドを以下のように実行します。

デプロイメント・ターゲットのタイプ	実行するコマンド
サーバー	<code>install_root/bin/wsadmin -f installBRManager.jacl -s server_name -n node_name</code>
クラスター	<code>install_root/bin/wsadmin -f installBRManager.jacl -cl cluster_name</code>

デプロイメント・ターゲットのタイプ	実行するコマンド
複数のターゲット (現在のビジネス・ルール・マネージャーが複数のターゲットにマップされている場合)	<pre>install_root/bin/wsadmin -f installBRManager.jacl -m "{{{target1} {target2} ... {targetn}}}"</pre> <p>ここで、各 {targetn} は、{ -s server_name -n node_name} または { -cl cluster_name}</p>

## 次のタスク

WebSphere Process Server ソフトウェアの残りの部分をデプロイメント・ターゲットにマイグレーションするには、80 ページの『Network Deployment 環境のマイグレーション』で説明されている適切なステップに従ってください。

## 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。

#### 手順

1. WBIPostUpgrade コマンドおよび WBIProfileUpgrade.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. ファイル `backupDirectory/logs/WBIPostUpgrade.profileName.timestamp.log` に、以下のメッセージが記載されているかどうかを確認します。
    - 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。

**注:** `backupDirectory` は、マイグレーションされたデータが最初に保管されるディレクトリーです。このデータは、マイグレーション・ウィザード、WBIPreUpgrade コマンドまたは WBIPostUpgrade コマンドの指定に従い、マイグレーション中に取り出されます。

**注:** `profileName` は、WebSphere Process Server のバージョン 6.2 で作成した新規プロファイルの名前です。

- b. ファイル `backupDirectory/logs/WBIProfileUpgrade.ant.profile_name.timestamp.log` に、メッセージ「BUILD SUCCESSFUL」が記載されているかどうかを確認します。

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・ファイルに、プロファイルの作成や拡張に関する致命的エラーが記録されていないかどうかを調べます。プロファイルのログ・ファイルは、`install_root/logs/manageprofiles/profile_name` ディレクトリーに格納されます。

3. サーバーのログ・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。
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。

注: 「テスト接続」は、ME データ・ソースでは動作しません。ME データ・ソースの接続を検証するには、サーバーの始動後、ログにエラーがないことを確認します。
  - 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、165 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

137 ページの『環境のロールバック』

WebSphere Process Server バージョン 6.2 環境へのマイグレーション後に、バージョン 6.1.x および 6.0.2.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。

165 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

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。

---

## 環境のロールバック

WebSphere Process Server バージョン 6.2 環境へのマイグレーション後に、バージョン 6.1.x および 6.0.2.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

通常は、マイグレーションを行っても旧リリースの構成が変更されることはありませんが、デプロイメント・マネージャーや管理対象ノードなど、最小限の変更が行われる場合もあります。これらの変更は元に戻すことができます。

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

4 ページの『WebSphere Process Server に関する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

135 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 6.2 デプロイメント・セルを、バージョン 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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. (オプション) **backupConfig** コマンドまたは必要なバックアップ・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 6.1.x および 6.0.2.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ーを実行して、バージョン 6.1.x および 6.0.2.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。

#### 手順

1. WebSphere Process Server バージョン 6.2 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 6.2 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・ユーティリティーを使用して、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ーを実行して、デプロイメント・マネージャーのバージョン 6.1.x および 6.0.2.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 6.2 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 6.1.x および 6.0.2.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・ユーティリティーを使用して、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ーを実行して、管理対象ノードのバージョン 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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 6.2 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 6.2 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 6.2 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 6.2 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 6.1.x および 6.0.2.x 環境で開始します。

## タスクの結果

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

『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 6.2 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。

82 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。

88 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』

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 6.2 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 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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 6.2 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、137 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。

### 手順

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 6.2 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を実行して、バージョン 6.2 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 6.1.x および 6.0.2.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を実行して、バージョン 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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 6.2 デプロイメント・マネージャーを始動します。
5. 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。

6. バージョン 6.2 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 6.2 のア

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 6.2 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 6.2 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。

## タスクの結果

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

137 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 6.2 デプロイメント・セルを、バージョン 6.1.x および 6.0.2.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。

94 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。

101 ページの『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。

### 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](#)

---

## WebSphere Process Server の事後マイグレーション・タスク

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 6.2 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

現在の環境で可能な場合は、以下の検査を実行します。

- バージョン 6.1.x および 6.0.2.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 6.2 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.profile_name.timestamp.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 6.2 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、150 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。
- バージョン 6.1.x からバージョン 6.2 にマイグレーションした後は、WebSphere Adapter のプロパティーを調べて、新しいインストール・ロケーション用に正しく構成されていることを確認します。一部のアダプター・プロパティーは、マイグレーション中に、自動マイグレーションでは対応していない方法で変更する必要がある場合があります。
- バージョン 6.2 にマイグレーションした後は、ポートが正しくマップされていることを確認して、グローバル・セキュリティーがオンになったときにリモート成果物ローダーがアプリケーション・クラスターのセキュリティー・ポートにアクセスできることを確認する必要があります。ポートが正しく構成されていることを確認するには、以下の手順を使用します。
  1. 管理コンソールで、「環境」 → 「仮想ホスト」にナビゲートします。
  2. 「default\_host」 → 「ホスト別名」を選択します。
  3. アプリケーション・クラスター・セキュリティー・ポートが「\*」（すべてのホストを意味します）にマップされているかどうかを確認します。そのような場合、**新規**をクリックしてから「ホスト名」フィールドに「\*」を入力し、アプリケーション・クラスターのポート番号を「ポート」フィールドに入力することで、「\*」に変更します。
  4. **適用**または**OK**をクリックし、「保管」を選択して変更内容を保存します。

## Business Process Choreographer の事後マイグレーション・タスク

サーバーまたはクラスターで Business Process Choreographer を実行する場合、追加のタスクをいくつか実行してからサーバーやクラスターを起動する必要があります。

### 始める前に

Business Process Choreographer データベース・スキーマが正常にアップグレードされ、ランタイム・データが正常にマイグレーションされていることを確認してください。また、サーバーとクラスターが正常にマイグレーションされていることを確認してください。

### このタスクについて

ご使用の環境によっては、実動環境で WebSphere Process Serverバージョン 6.2 を使用する前に、以下のタスクを実行する必要があります。

#### 手順

1. *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, VMMTransformation.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 6.2 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 6.1.x および 6.0.2.x スタッフ・プラグイン構成 (WebSphere Process Serverバージョン 6.2 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。
2. Business Flow Manager と Human Task Manager の REST API エンドポイントを構成し、すべての参照を更新します。マイグレーション前のバージョンに応じて、次のいずれかの処理を実行します。
  - バージョン 6.1.2 からのマイグレーションの場合、エンドポイントは WCCM 内に自動的に作成されます。このため、bpcEndpoints.xml ファイルは必要ありません。ただし、カスタマイズの内容は失われ、Web サーバーの代わりに、いずれかのクラスター・メンバーまたはスタンドアロン・サーバーがビジネス・スペースによって使用されます。マイグレーション前に REST Web モジュールが Web サーバーにマップされていた場合、これらのモジュールは Web サーバーにマップされたままですが、次の処理を実行してビジネス・スペース内の参照を変更し、再び Web サーバーを参照するように設定する必要があります。
    - a. Business Flow Manager のエンドポイントを変更するには、「サーバー」→「アプリケーション・サーバー」→「*server\_name*」または「サーバー」→「クラスター」→「*cluster\_name*」をクリックし、「ビジネス・インテグレーション」で「**Business Process Choreographer**」を展開して「**Business Flow Manager**」をクリックし、「追加プロパティ」で「**REST サービスのエンドポイント**」をクリックします。

- b. Human Task Manager のエンドポイントを変更するには、「サーバー」 → 「アプリケーション・サーバー」 → 「*server\_name*」または「サーバー」 → 「クラスター」 → 「*cluster\_name*」をクリックし、「ビジネス・インテグレーション」で「**Business Process Choreographer**」を展開して「**Human Task Manager**」をクリックし、「追加プロパティ」で「**REST サービスのエンドポイント**」をクリックします。
- バージョン 6.0.2 または 6.1 からマイグレーションした場合は、マイグレーション中に REST API が構成されています。Web モジュールを Web サーバーにマップして REST API Web モジュールのコンテキスト・ルートを変更する場合は、Business Process Choreographer Explorer と Business Space の REST API の参照も更新する必要があります。

コンテキスト・ルートを変更するには、以下の手順を実行します。

- 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」 → 「**BPEContainer\_suffix**」 → 「Web モジュールのコンテキスト・ルート」をクリックします。この *suffix* は、Business Process Choreographer が構成される *node\_name\_server\_name* または *cluster\_name* のいずれかを表します。
- b. Web モジュール BFMRESTAPI のコンテキスト・ルートに、正しい固有の値が設定されていることを確認します。
- c. 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」 → 「**TaskContainer\_suffix**」 → 「Web モジュールのコンテキスト・ルート」をクリックします。
- d. Web モジュール HTMRESTAPI のコンテキスト・ルートに、正しい固有の値が設定されていることを確認します。

Business Process Choreographer Explorer のエンドポイント参照を変更するには、「サーバー」 → 「アプリケーション・サーバー」 → 「*server\_name*」または「サーバー」 → 「クラスター」 → 「*cluster\_name*」をクリックし、「ビジネス・インテグレーション」で「**Business Process Choreographer**」を展開して「**Business Process Choreographer Explorer**」をクリックします。

ビジネス・スペースのエンドポイント参照を変更するには、次の操作を実行します。

- a. Business Flow Manager のエンドポイントを変更するには、「サーバー」 → 「アプリケーション・サーバー」 → 「*server\_name*」または「サーバー」 → 「クラスター」 → 「*cluster\_name*」をクリックし、「ビジネス・インテグレーション」で「**Business Process Choreographer**」を展開して「**Business Flow Manager**」をクリックし、「追加プロパティ」で「**REST サービスのエンドポイント**」をクリックします。
- b. Human Task Manager のエンドポイントを変更するには、「サーバー」 → 「アプリケーション・サーバー」 → 「*server\_name*」または「サーバー」 → 「クラスター」 → 「*cluster\_name*」をクリックし、「ビジネス・インテグレーション」で「**Business Process Choreographer**」を展開して「**Human Task Manager**」をクリックし、「追加プロパティ」で「**REST サービスのエンドポイント**」をクリックします。

3. 事前に定義されたヒューマン・タスクの最新バージョンが Network Deployment 環境にインストールされていることを確認します。インストールされていない場合は、手動でインストールする必要があります。
  - a. 事前に定義されたヒューマン・タスクの最新バージョンがインストールされているかどうかを確認するには、「アプリケーション」 → 「エンタープライズ・アプリケーション」をクリックし、HTM\_PredefinedTasks\_Vnnn\_scope と HTM\_PredefinedTasksMsg\_Vnnn\_scope という名前のアプリケーションを探します。バージョン番号 *nnn* が *install\_root/installableApps/HTM\_PredefinedTasks\_Vnnn.ear* と *install\_root/installableApps/HTM\_PredefinedTasksMsg\_Vnnn.ear* に格納されているバージョン番号と同じであれば、最新バージョンがインストールされていることになります。この場合は、ステップ 4 まで処理をスキップすることができます。それ以外の場合は、以下の処理を実行してアプリケーションをインストールします。
  - b. 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」をクリックします。
  - c. 「インストール」の次に「参照」をクリックし、*install\_root/installableApps* ディレクトリーにナビゲートします。
  - d. HTM\_PredefinedTasks\_Vnnn.ear の最新バージョンを探して選択し、「開く」の次に「次へ」をクリックします。
  - e. 「インストール・オプションの選択」ページで、アプリケーション名を HTM\_PredefinedTasks\_VnnnApp から HTM\_PredefinedTasks\_Vnnn\_scope に変更します。この *scope* は、*nodeName\_serverName* または *clusterName* のいずれかを表します (事前に定義されたタスクを単一サーバーにインストールしたかクラスターにインストールしたかによって異なります)。次に、「次へ」をクリックします。
  - f. 「サーバーにモジュールをマップ」ページで、事前に定義されたタスク・アプリケーションのデプロイメント・ターゲットが、Business Process Choreographer が構成されたデプロイメント・ターゲットになっていることを確認します。
  - g. 「次へ」をクリックしてから「終了」をクリックします。
  - h. インストールが正常に終了したら、「保管」をクリックして変更内容を保存します。
  - i. HTM\_PredefinedTaskMsg\_Vnnn.ear という名前のエンタープライズ・アプリケーションをインストールします。HTM\_PredefinedTaskMsg\_Vnnn.ear に対して、ステップ 3c から 3h を繰り返します。
4. 事前に定義されたヒューマン・タスクの旧バージョンのインスタンスが実行されていないことを確認し、次の手順で旧バージョンを削除します。
  - a. すべてのインスタンスが削除されていることを確認します。
  - b. 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」をクリックし、次の 2 つのアプリケーションを選択して「アンインストール」をクリックします。
    - HTM\_PredefinedTasks\_Vnnn\_scope.ear
    - HTM\_PredefinedTaskMsg\_Vnnn\_scope.ear

この *nnn* は、旧バージョン番号 (612 など) を表します。*scope* は、*nodeName\_serverName* または *clusterName* のいずれかを表します (事前に定義されたタスクを単一サーバーにインストールしたかクラスターにインストールしたかによって異なります)。

5. オプション: 作業項目のデータ・マイグレーションで使用した余分なストレージ・スペースを解放するため、データベースからテーブル `WI_ASSOC_OID_T` を削除します。
6. オプション: DB2 for Linux、UNIX、Windows、または z/OS を使用している場合は、次の旧テーブルをデータベースから削除して、テーブル・スペースのマイグレーションで使用した余分なストレージ・スペースを解放します。
  - `PROCESS_TEMPLATE_B_O`
  - `ACTIVITY_TEMPLATE_B_O`
  - `SCOPED_VARIABLE_INSTANCE_B_O`
  - `CORRELATION_SET_INSTANCE_B_O`
  - `STAFF_QUERY_INSTANCE_O`
  - `TASK_TEMPLATE_O`
  - `TASK_INSTANCE_O`

**重要:** 新しいテーブルを削除しないように注意してください。名前は似ていますが、新しいテーブルにはサフィックス 『`_T`』 が付いています。

7. オプション: データベースの再調整を行います。この処理は、後から実行してもかまいません。例えば、DB2 データベースの場合は、`REORG` と `RUNSTATS` を実行します。
8. Business Process Choreographer Observer の構成が以前に存在していた場合は、「マイグレーション後に Business Process Choreographer Explorer レポート機能を有効にする (Enabling the Business Process Choreographer Explorer reporting function after migration)」を実行して、新しいレポート機能に切り替えます。
9. バージョン 6.0.2 からマイグレーションし、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記述した場合は、API を使用する前にクライアントを変更してログインを実行します。J2EE ロールの `BPEAPIUser` と `TaskAPIUser` は、マイグレーション後に値 `Everyone` に設定されます。これにより、アプリケーション・セキュリティが有効な場合も 6.0.2 にログインする必要がなくなり、後方互換性を保つことができます。ただし、値 `Everyone` の使用は推奨しません。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。

- a. 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。
- b. 右のパネルで、名前 `BPEContainer_scope` を選択します。この *scope* は、*nodeName\_serverName* または *clusterName* のいずれかを表します (ユーザーが Business Process Choreographer をサーバー上に構成したかクラスター上に構成したかによって異なります)。

- c. 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役割」を選択します。
  - d. J2EE BPEAPIUser ロールのマッピングを「Everyone」から「All authenticated」に変更します。
  - e. 「OK」を選択します。
  - f. 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。
  - g. 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。
10. バージョン 6.2 にアップグレードする前に、faces-config-beans.xml 構成ファイルを変更してバージョン 6.0.2 または 6.1.x の Business Process Choreographer Explorer の照会のしきい値を指定した場合、この変更内容を再適用する必要があります。詳しくは、技術情報 の『Business Process Choreographer Explorer のカスタマイズとチューニングのオプション (Business Process Choreographer Explorer - Customization and Tuning Options)』を参照してください。

注: バージョン 6.1 以降の場合、faces-config-beans.xml ファイルの設定によって影響を受けるのは、事前に定義されたビューだけです。カスタム・ビューのしきい値は、それぞれの定義の一部として指定されます。

#### 関連情報

 管理コンソールを使用した、ビジネス・プロセスおよびヒューマン・タスク・アプリケーションのアンインストール

 管理コマンドを使用した、ビジネス・プロセスおよびヒューマン・タスク・アプリケーションのアンインストール

---

## WebSphere が提供するビジネス・スペースの事後マイグレーション・タスク

WebSphere Process Server をバージョン 6.1.2 からバージョン 6.2 にマイグレーションした後は、サーバーまたはクラスターを始動する前に、いくつかの追加の作業を行う必要があります。

### 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

WebSphere Process Server バージョン 6.1.2 からマイグレーションしており、ビジネス・スペースが既に構成されている場合は、マイグレーション後に 2 つの手動ステップを実行しなければビジネス・スペースを使用することはできません。

### 手順

1. ビジネス・スペース・データベースをアップグレードします。これを行うには、2 つのスクリプトを実行します。詳しくは、『手動によるビジネス・スペース・データベースのアップグレード (Upgrading the Business Space database manually)』を参照してください。
2. ビジネス・スペースでウィジェットが使用するエンドポイントを構成します。管理コンソールの「REST サービスのエンドポイント」ページを使用して、ウィジェットのエンドポイントがビジネス・スペースに表示されるように更新します。『管理コンソールでビジネス・スペース・ウィジェットのエンドポイントを使用可能にする』を参照してください。

## タスクの結果

ビジネス・スペース バージョン 6.2 を使用できます。

**注:** ビジネス・スペース バージョン 6.1.2 を使用していた場合は、ビジネス・スペース バージョン 6.2 を使用する前に、必ずブラウザのキャッシュを消去してください。これにより、不注意でビジネス・スペース バージョン 6.1.2 のコードおよびイメージが引き続き使用されてしまうことを防ぐことができます。

### 関連タスク

54 ページの『手動による ビジネス・スペース データベースのアップグレード』

サーバーをバージョン 6.1.2 からマイグレーションした後は、バージョン 6.2 のサーバーを始動する前に、ビジネス・スペース・データベースを新しいデータベース・スキーマに手動でアップグレードする必要があります。

 管理コンソールでビジネス・スペース・ウィジェットのエンドポイントを使用可能にする

 エンドポイント・ファイルでのビジネス・スペース・ウィジェットの使用可能化 (Enabling Business Space widgets in the endpoints files)

---

## 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 6.2 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 4 ページの『WebSphere Process Server に関する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有効になっています。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 6.2 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、165 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。

### 手順

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 6.1.x および 6.0.2.x からバージョン 6.2 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を使用して、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。

153 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 6.2 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、157 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。

162 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

29 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。

4 ページの『WebSphere Process Server に関する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

39 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。

153 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 6.2 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 6.2 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経由で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。

162 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 6.2 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 6.2 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経由で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ています。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。(157 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。) この要件により、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。

### 手順

1. それぞれの新しい WebSphere Process Server バージョン 6.2 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/WebSphere60/AppServer/cloudscape/db2j.properties.
```

```
MIGR0344I: Processing configuration file /opt/WebSphere60/AppServer/config/cells/migr06/applications/MyBankApp.ear/deployments/MyBankApp/deployment.xml.
```

```
DSRA7600E: Cloudscape migration of database instance /opt/WebSphere61/Express/profiles/default/databases/_opt_WebSphere60_AppServer_bin_DefaultDB failed, reason: java.sql.SQLException: Failure creating target db
```

```
MIGR0430W: Cloudscape Database /fvt/temp/60BaseXExpress/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/myFulldbName_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\60BaseExpress\PostUpgrade50BaseFVTest2\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/60BaseExpress/PostUpgrade50BaseFVTest9
/testRun/pre/websphere_backup/bin/DefaultDB>
```

```
connecting to source db <jdbc:db2j:/fvt/temp/60BaseExpress/PostUpgrade50BaseFVTest9
/testRun/pre/websphere_backup/bin/DefaultDB> took 0.26 seconds
```

```
creating target db <jdbc:derby:/opt/WebSphere61/Express/profiles/default/databases
/_opt_WebSphere60_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/myFulldbName_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_WebSphere60_AppServer_bin_DefaultDB already exists.
Aborting migration
at com.ibm.db2j.tools.migration.MigrateFrom60Impl.go(Unknown Source)
at com.ibm.db2j.tools.migration.MigrateFrom60Impl.doMigrate(Unknown Source)
at com.ibm.db2j.tools.MigrateFrom60.doMigrate(Unknown Source)
at com.ibm.ws.adapter.migration.CloudscapeMigrationUtility.migr
```

## タスクの結果

- 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 5. 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"> <li>• 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。</li> <li>• 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。)</li> </ul>
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>

表 5. 新しいクラス情報 (続き)

クラス・タイプ	古い値	新しい値
データ・ソースのヘルパー・クラス	com.ibm.websphere.rsadapter.Cloudscape DataStoreHelper	com.ibm.websphere.rsadapter.Derby DataStoreHelper

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  
す。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め  
られます。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。

## 次のタスク

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、157 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  
と、サーバーはランタイム例外を発行します。

### 関連タスク

157 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 6.2 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。

150 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 6.2 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。

162 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



Apache Derby



IBM Cloudscape インフォメーション・センター

## 手動による Cloudscape のアップグレード

WebSphere Process Server バージョン 6.2 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。153 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。

## 手順

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。
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・スクリプトを検査します。使用しているオペレーティング・システムに応じて、WebSphere Process Server には次のいずれかのマイグレーション・スクリプトが用意されています。
  - **Linux** **UNIX** **Linux および UNIX プラットフォームの場合:** 以下のディレクトリーにある db2jmmigrate.sh スクリプトを使用します:  
`install_root/derby/bin/embedded/...`

- **Windows** **Windows** プラットフォームの場合: 以下のディレクトリーにある db2jmmigrate.bat スクリプトを使用します: `install_root\derby\bin\embedded\...`

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。スクリプトで使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。

- 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。
  - 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。
  - サーバー名を選択します。
  - 「ログ・レベルの詳細の変更 (Change Log Level Details)」をクリックします。
  - 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。
  - 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。
  - 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。

- 「適用」を選択し、次に「OK」を選択します。
- 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>db2jmmigrate.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・ユーティリティー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\my01dDB_migrationLog.log for progress
Migration Completed Successfully
E:%WebSphere\ProcServer\derby\bin\embedded>

```

そうでない場合、ログには以下の例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:%temp%\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:%temp%\myNewDB>
creating target db <jdbc:derby:e:%temp%\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

```

## タスクの結果

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、データベース表およびスキーマを WebSphere Process Server バージョン 6.2 レベルにアップグレードします。詳しくは、39 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

153 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 6.2 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 6.2 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経由で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。

162 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。

39 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。

### 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 6.2 のサーバー上で UDDI レジストリーが実行されていることを確認します。例えば、2 つのノードにまたがるクラスターがあり、UDDI レジストリーを実行しているサーバーが WebSphere Process Server バージョン 6.2 である場合、一方のノードをバージョン 6.2 にアップグレードして、もう一方のノードを以前のレベルのままにしておくことができます。
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。

## タスクの結果

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、153 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

153 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 6.2 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 6.2 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経由で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。

157 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 6.2 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。

150 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 6.2 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

## 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場合、バージョン 6.2 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際に、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ことがあります。

```
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。

- プロファイル作成の一部として設定された WebSphere Application Server 変数は、変更しないでください。

古いプロファイルに設定されているこれらの変数を変更した場合、以下のよう  
に、データベース接続やロードなどのクラス欠落例外が発生する可能性があります。

```
10/25/08 13:22:39:650 GMT+08:00] 0000002e J2CUtilityCla E J2CA0036E: An  
exception occurred while invoking method setDataSourceProperties on  
com.ibm.ws.rsadapter.spi.WSManagedConnectionFactoryImpl used by resource  
jdbc/com.ibm.ws.sib/ewps6101.Messaging-BPC.cwfpcCell101.Bus :  
com.ibm.ws.exception.WsException: DSRA0023E: The DataSource  
implementation class "com.ibm.db2.jcc.DB2XADataSource" could not be  
found.DB2,
```

Derby ドライバーと SQL Embedded JDBC ドライバーは、WebSphere Process Server の製品インストールにバンドルされています。これらのドライバーを上位バージョンに変更する必要がある場合、以下に示すとおり、製品インストールと同じ場所にドライバーをコピーする必要があります。

- **Derby:** `%was.install.root%\derby\lib`
- **DB2:** `%was.install.root%\universalDriver_wbi/lib`
- **SQL:** `%was.install.root%\lib`

新規の JDBC プロバイダーとデータ・ソースがアプリケーションに対して必要な場合、有効な `jdbcclasspath` を選択して WebSphere Application Server 変数を設定することにより、これらのリソースを作成することができます。例えば、前のインストールでは存在しなかった DB2 をセル・レベルで設定する必要がある場合、以下の手順を実行します。

1. 管理コンソールで、「リソース」 → 「JDBC」 → 「JDBC プロバイダー」 → 「DB2 Universal JDBC ドライバー・プロバイダー (XA)」にナビゲートします。

2. 「クラスパス」ボックスで、以下のパスを設定します。

- DB2UNIVERSAL\_JDBC\_DRIVER\_PATH =%was.install.root%/universalDriver\_wbi/lib
- DB2UNIVERSAL\_JDBC\_DRIVER\_NATIVEPATH=""

独自のドライバーが必要な場合は、パスとして

DB2UNIVERSAL\_JDBC\_DRIVER\_PATH=%myDriverLocation% を設定します。

- WebSphere Process Server の旧バージョンからバージョン 6.2 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・ファイルを探し、それらを表示して、手掛かりがないかを調べます。

バージョン 6.1.x からのマイグレーションの場合

- migration\_backup/logs/backupConfig.profile\_name.timestamp.log
- migration\_backup/logs/restoreConfig.profile\_name.timestamp.log
- migration\_backup/logs/WBIMigration.profile\_name.timestamp.log
- migration\_backup/logs/WBIPostUpgrade.profile\_name.timestamp.log
- migration\_backup/logs/WBIPreUpgrade.timestamp.log
- migration\_backup/logs/WBIProfileUpgrade.ant.profile\_name.timestamp.log

バージョン 6.0.2.x からのマイグレーションの場合

- profile\_root/logs/WASPostUpgrade.timestamp.log
- migration\_backup/WASPreUpgrade.timestamp.log
- migration\_backup/logs/WASPostUpgrade.profile\_name.timestamp.trace
- migration\_backup/logs/WASPreUpgrade.timestamp.trace
- migration\_backup/logs/WBIPostUpgrade.profile\_name.timestamp.log
- migration\_backup/logs/WBIPreUpgrade.timestamp.log
- migration\_backup/logs/WBIProfileUpgrade.ant.profile\_name.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。

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 6.2 の `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、発生することがあります。この場合は、バージョン 6.2 の `install_root/bin` ディレクトリーに WBIPreUpgrade スクリプトが存在することを確認してから、このスクリプトをこのディレクトリーから起動します。リモート・システムにマイグレーションする場合、WBIPreUpgrade は `migration_copy/bin` ディレクトリーから起動する必要があります。詳しくは、67 ページの『リモート・システムへのマイグレーション』を参照してください。

この問題は、WBIPreUpgrade コマンドを WebSphere Process Server バージョン 6.2 `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 6.2 `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。リモート・システムにマイグレーションする場合、WBIPreUpgrade は `migration_copy/bin` ディレクトリーから起動する必要があります。詳しくは、67 ページの『リモート・システムへのマイグレーション』を参照してください。

- 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) `jdb-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 コマンド行ユーティリティー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 6.2 *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 6.2 *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 6.2 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 6.2 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。その際、選択したタイムアウトの少なくとも 3 倍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、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をすべて合計した時間です。

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/profile_name/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` プロパティの値を見つけます。
6. `com.ibm.SOAP.requestTimeout` の値を バージョン 6.2 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 コマンド行ユーティリティー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 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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 6.2 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 6.2 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` ノードから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 6.2 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 6.2 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 コマンド行ユーティリティー』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 6.1.x および 6.0.2.x から バージョン 6.2 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 6.1.x および 6.0.2.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 6.2 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 6.1.x および 6.0.2.x ノードを停止すると、バージョン 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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. 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 6.2 のプロファイルを削除します。以下の手順を実行します。
  - a. コマンド・プロンプトを開き、使用するオペレーティング・システムに基づいて以下のいずれかのコマンドを実行します。

```
- i5/OS i5/OS プラットフォーム: manageprofiles -delete  
-profileName profile_name
```

```
- Linux UNIX Linux および UNIX プラットフォーム:  
manageprofiles.sh -delete -profileName profile_name
```

```
- Windows Windows プラットフォーム: manageprofiles.bat -delete  
-profileName profile_name
```

変数 `profile_name` は削除するプロファイルの名前を示します。

- b. 以下のログ・ファイルを調べて、プロファイルの削除が完了したことを確認します。

```
- i5/OS i5/OS プラットフォーム: user_data_root/profileRegistry/logs/  
manageprofiles/profile_name_delete.log
```

```
- Linux UNIX Linux および UNIX プラットフォーム:  
install_root/logs/manageprofiles/profile_name_delete.log
```

```
- Windows Windows プラットフォーム:  
install_root¥logs¥manageprofiles¥profile_name_delete.log
```
  6. 前のステップで削除したバージョン 6.2 のプロファイルの `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 6.1.x および 6.0.2.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 6.2 構成にインストール

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 6.1.x および 6.0.2.x 構成内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。

マイグレーション中にアプリケーション・インストールがこの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 6.1.x および 6.0.2.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 6.2 構成に手動でインストールできます。

- 管理対象ノードをバージョン 6.2 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。
- アプリケーションから CBE イベントを発行する場合、マイグレーション後にサーバーを起動すると、以下の例外が表示されることがあります。

NameNotFoundException for event bus

```
[9/17/08 20:00:30:718 CST] 0000003c B0Core          | No schema found with
namespace "http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/
db2admincustomer" and location Db2admincustomer.xsd
```

```
[9/17/08 20:00:30:734 CST] 00000037 EventBusSender E
com.ibm.events.emitter.impl.EventBusSender initialize CEIEM0020E The
emitter failed to initialize because the JNDI lookup on the event bus
home name failed.
```

JNDI name: ejb/com/ibm/events/bus/EventBus

Context: Node01Cell/nodes/Node01/servers/server1

Exception (if any): javax.naming.NameNotFoundException

この問題は、CEI 関連アプリケーションを開始する前に JCAOutbound アプリケーションから CBE イベントを発行しようとした場合に発生します。この問題を修正するには、CEI が開始されてからアプリケーションが起動するように、アプリケーションの開始ウェイトを増やします。

- ネットワーク・デプロイメント環境で、マイグレーション後にビジネス・ルール・マネージャーにアクセスしたときに、エラー SRVE0026E: [Servlet Error]-[com/ibm/wbiservers/brules/BusinessRuleManager]: java.lang.NoClassDefFoundError が発生した場合は、そのノードの通常マイグレーションを続行する前に、デプロイメント・ターゲットにビジネス・ルール・マネージャー・アプリケーションを手動でインストールする必要があります。詳しくは、133 ページの『ネットワーク・デプロイメント環境でのビジネス・ルール・マネージャーのマイグレーション』を参照してください。
- 管理対象ノードのバージョン 6.2 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 6.2 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/62AppServer/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 6.2 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 6.2 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

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

### 関連タスク

135 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

 **WBIPreUpgrade** コマンド行ユーティリティー

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。

 **WBIPostUpgrade** コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された *backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

### 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 その他の 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、178 ページの『WebSphere InterChange Server または WebSphere Business Integration Server Express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、234 ページの『WebSphere Studio Application Developer Integration Edition から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、235 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 6.2 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 6.2 にデプロイできます。

別の製品から 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 または WebSphere Business Integration Server Expressからのマイグレーション

WebSphere Integration Developer ウィザードまたは WebSphere Process Server の reposMigrate コマンドを使用して、WebSphere InterChange Server バージョン 4.3 以降または WebSphere Business Integration Server Express バージョン 4.4 以降から WebSphere Process Server 6.2 にマイグレーションします。

### このタスクについて

このバージョンの WebSphere InterChange Server または WebSphere Business Integration Server Express の場合...	実行内容
WebSphere InterChange Server バージョン 4.3 以降、または WebSphere Business Integration Server Express バージョン 4.4 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 4.3 より前のバージョン、または WebSphere Business Integration Server Express 4.4 より前のバージョン	まず WebSphere InterChange Server version 4.3 以降または WebSphere Business Integration Server Express バージョン 4.4 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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

231 ページの『WebSphere InterChange Server または WebSphere Business Integration Server Express 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

189 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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: 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。

### 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 または WebSphere Business Integration Server Express 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ください。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 または Business Object Designer Express™ 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 または WebSphere Business Integration Server Express 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・プロパティーを使用して、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 または WebSphere Business Integration Server Express 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: 共通コード・ユーティリティー

IBM では、WebSphere InterChange Server または WebSphere Business Integration Server Express 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。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 または WebSphere Business Integration Server Express 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ください。

### 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 または WebSphere Business Integration Server Express 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 または

WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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ことを避けてください。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: 一般的な開発**

WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express の開発ガイドに記載されています。公開済みの WebSphere API 用に、互換性 API が WebSphere InterChange Server または WebSphere Business Integration Server Express Process Server で提供されています。WebSphere InterChange Server および WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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ります。

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。

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

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 または WebSphere Business Integration Server Express 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: マップ

WebSphere InterChange Server または WebSphere Business Integration Server Express 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 ツールまたは Map Designer Express 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 (特にエント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 または WebSphere Business Integration Server Express 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: 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 秒の間隔を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: 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 または WebSphere Business Integration Server Express 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 ツールまたは Relationship Designer Express ツールのみを使用してください。また、WebSphere InterChange Server または WebSphere Business Integration Server Express 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、必ずリレーションシップ・マネージャー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。

## **reposMigrate コマンドを使用した WebSphere InterChange Server または WebSphere Business Integration Server Express の成果物のマイグレーション**

WebSphere InterChange Server または WebSphere Business Integration Server Express 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` または `WebSphere Business Integration Server Express` 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必要があります。

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` または `WebSphere Business Integration Server Express` `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` または `WebSphere Business Integration Server Express 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` または `WebSphere Business Integration Server Express` 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` または `WebSphere Business Integration Server Express` の成果物に対応する成果物が `WebSphere Process Server` にない場合、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` または `WebSphere Business Integration Server Express` 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。

## 手順

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 または WebSphere Business Integration Server Express 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、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: WebSphere Process Server ランタイムでのマイグレーション済み WebSphere InterChange Server 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に、**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 WBI Scheduler エントリー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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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。

## 次のタスク

**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。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

『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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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

 reposMigrate コマンド

WebSphere InterChange Server v4.3 の資料

 WebSphere Integration Developer インフォメーション・センター

## 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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。

190 ページの『セキュリティー』

190 ページの『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)』

191 ページの『既存の WebSphere InterChange Server または WebSphere Business Integration Server Express データベース接続プールの処理』

192 ページの『既存の WebSphere InterChange Server または WebSphere Business Integration Server Express リレーションシップ・データベースの使用』

- 192 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』
- 193 ページの『Access Enterprise JavaBean (EJB) サポート』
- 194 ページの『DynamicSend API 構成』
- 194 ページの『BaseCollaboration.dynamicSend メソッド呼び出しの使用可能化』
- 196 ページの『イベント順序付けマイグレーション』
- 197 ページの『失敗したイベント』
- 197 ページの『マップのマイグレーション』
- 197 ページの『コラボレーションのマイグレーション』
- 198 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』
- 199 ページの『WebSphere Process Server での logError API E メール通知の使用可能化』
- 199 ページの『WebSphere Process Server での非同期呼び出しの処理』
- 200 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』
- 200 ページの『WebSphere Process Server での相関値の処理』
- 201 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 または WebSphere Business Integration Server Express 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、203 ページの『WebSphere InterChange Server または WebSphere Business Integration Server Express をマイグレーションした後のグローバル・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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 プロバイダーで使用するために、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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express  
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 または WebSphere Business  
Integration Server Express 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 または WebSphere Business  
Integration Server Express 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 または WebSphere Business Integration Server Express リポジ  
トリーを、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 または WebSphere Business Integration Server Express 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、193 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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であるコラボレーションまたはコネクタを格納する WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express の共用ライブラリーを 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 プロジェクトのセレクター・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。「プロパティー」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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express のマイグレーションにより、WebSphere InterChange Server または WebSphere Business Integration Server Express 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・マップの構造はすべて同一です。相違点としては、名前、呼び出すサブマップの名前、または verb 属性での ASI 情報などがあります。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 または WebSphere Business Integration Server Express マップをマイグレーションしたものです。WebSphere InterChange Server または WebSphere Business Integration Server Express 入力マップに、サポートされる WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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。

**コラボレーション・オブジェクト:** WebSphere InterChange Server または WebSphere Business Integration Server Express 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、関連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、関連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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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、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

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 または WebSphere Business Integration Server Express の logError API が、  
WebSphere InterChange Server または WebSphere Business Integration Server Express  
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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express  
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うと失敗します。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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express の場合とは異なります。WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express API を使用したか、WebSphere InterChange Server または WebSphere Business Integration Server Express 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無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

230 ページの『WebSphere InterChange Server または WebSphere Business Integration Server Express からマイグレーションする場合の制限事項』  
WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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する必要がある場合があります。

231 ページの『WebSphere InterChange Server または WebSphere Business Integration Server Express 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

186 ページの『reposMigrate コマンドを使用した WebSphere InterChange Server または WebSphere Business Integration Server Express の成果物のマイグレーション』  
WebSphere InterChange Server または WebSphere Business Integration Server Express 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

178 ページの『事前マイグレーションの考慮事項』  
WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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

-  [serviceDeploy コマンド](#)
-  [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](#)

## WebSphere InterChange Server または WebSphere Business Integration Server Express をマイグレーションした後のグローバル・セキュリティの構成

WebSphere InterChange Server または WebSphere Business Integration Server Express 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、以下に示す追加のセキュリティ構成ステップを実行します。

### 始める前に

最初に、WebSphere Process Server のセキュリティ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
- セキュリティ役割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: セキュリティ役割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』を参照してください。

### 手順

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」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. セキュリティー役割をユーザー・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」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役割」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役割」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」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 (204 ページ) 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。

### 関連情報

 Wsadmin ツール

WebSphere InterChange Server v4.3 の資料

 アセンブリ・ツールを使用した RunAs ロールへのユーザーのマッピング

 アダプターの保護

 アプリケーションとその環境の保護

 セキュア・アプリケーションのデプロイ (インストール)

 エンドツーエンド・セキュリティの構築

## 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express API』  
WebSphere Process Server および WebSphere Integration Developer で提供される WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express API

WebSphere Process Server および WebSphere Integration Developer で提供される WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express API を以下に示します。これらの API は、WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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):BusinessObjectInterface
- createBusinessObject(String, Locale):BusinessObjectInterface
- createBusinessObject(String, String):BusinessObjectInterface
- 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(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)
- setConfigMOnName(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

## WebSphere InterChange Server または WebSphere Business Integration Server Express からマイグレーションする場合の制限事項

WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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する必要がある場合があります。

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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express アプリケーションのイベント要約と変更内容の要約

約に、予期される情報が含まれていません。 **原因:** WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express API は WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express の API を使用するため、コードを手動で更新して WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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

189 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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

230 ページの『WebSphere InterChange Server または WebSphere Business Integration Server Express からマイグレーションする場合の制限事項』  
WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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する必要がある場合があります。

#### 関連資料

189 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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。

178 ページの『事前マイグレーションの考慮事項』

WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express API に対するロギングとトレースの有効化

サポートされる WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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。

#### 手順

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 または WebSphere Business Integration Server Express 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、サポートされる WebSphere InterChange Server API 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。

#### 関連情報

 サポートされる WebSphere InterChange Server API

### 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express の断片コードで使用されるほかの変数は直列化され、*CollabTemplateName\_var* という名前の BPEL 変数に、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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 からのマイグレーション

WebSphere Studio Application Developer Integration Edition から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・ワークスペースをアクティブな 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 6.2、バージョン 6.1.x、およびバージョン 6.0.2.x 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。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。

- 『WebSphere Process Server バージョン 6.2 で使用すべきでないフィーチャー』
- 241 ページの『WebSphere Process Server バージョン 6.1.2 で使用すべきでないフィーチャー』
- 241 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』
- 246 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』
- 248 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』
- 248 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』

以下の情報は、非推奨の項目をバージョンとリリースごとにまとめたものです。各セクションに、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 6.2 で使用すべきでないフィーチャー

#### BOCopy Service の 2 つのメソッド: copyInto() と copyIntoShallow()

BOCopy Service 内の copyInto() メソッドと copyIntoShallow() メソッドは非推奨です。

これは、コピーと設定を同時に行うと、コピーや設定で発生する問題がマスクされてしまうためです。この場合は、単純にコピーと設定を別々に実行します。こうすれば、特に複雑な回避策は必要ありません。API を組み合わせて使用するのではなく、copy() メソッドの次に set() メソッドを使用してください。

#### 推奨されるマイグレーション・アクション

copyInto() と copyIntoShallow() の代わりに、次のメソッドを使用します。

- copyInto() の代わりに copy() を使用し、次に set() を使用
- copyIntoShallow() の代わりに copyShallow() を使用し、次に set() を使用

## スタンドアロン・プロファイルに使用する CEI パラメーター (CommonDB では通常パラメーター)

スタンドアロン・プロファイルに使用する CEI パラメーター (CommonDB では通常パラメーター) の大半は非推奨です。

### 推奨されるマイグレーション・アクション

バージョン 6.1.x および 6.0.2.x で manageprofiles コマンドを使用していて、このコマンドをバージョン 6.2 でも使用する場合は、コマンドを編集して新しいパラメーター・セットを使用する必要があります。

注: プロファイル管理ツールを使用している場合は、GUI (グラフィカル・ユーザー・インターフェース) によって正しいパラメーターが渡されます。次の表に非推奨の CEI パラメーターを示します。バージョン 6.2 以降は、CommonDB 用と同じパラメーターを、対応する CEI パラメーターに対して使用します。CEI パラメーターの変更方法の例を、この表の下に示します。

表 6. 非推奨の CEI パラメーター

CEI 変数名	CommonDB 変数名	適用可能データベース
nodeName	nodeName	すべて
ceiServerName	serverName	すべて
ceiDbExecuteScripts	dbDelayConfig	すべて
ceiJdbcClassPath	dbJDBCClasspath	すべて
ceiDbHostName	dbHostName	すべて
ceiDbPort	dbServerPort	すべて
ceiDbUser	dbUserId	MSSQL を除くすべて
ceiDbPassword	dbPassword	MSSQL を除くすべて
ceiOutputScriptDir	dbOutputscriptDir	すべて
ceiStorageGroup	dbStorageGroup	DB2 z/OS
ceiDbAliasName	dbSchemaName	DB2 z/OS
ceiDbSubSystemName	dbConnectionLocation	DB2 z/OS
ceiNativeJdbcClassPath	dbJDBCClasspath	DB2 iSeries® Native
ceiCollection	dbSchemaName	DB2 iSeries Native
ceiToolboxJdbcClassPath	dbJDBCClasspath	DB2 iSeries Toolbox
ceiCollection	dbSchemaName	DB2 iSeries Toolbox
ceiDbInformixDir	dbLocation	Informix
ceiDbServerName	dbInstance	Informix
ceiDbSysUser	dbSysUserId	Oracle
ceiDbSysPassword	dbSysPassword	Oracle

### 例 1: manageprofiles

以前のコマンドと新しいコマンドを示す例です (manageprofiles コマンドを使用している場合)。このバージョンからは、「ceiDBName」以外の CEI パラメーターを渡す必要はなくなりました。

旧

```
612 manageprofiles.bat -create -profileName -templatePath
¥profileTemplates¥default.wbiserver -dbType DB2_Universal -dbDelayConfig
false -dbCreateNew true -dbJDBCClasspath <classpath> -dbHostName localhost
-dbServerPort <port> -dbUserId <userid> -dbPassword <password>
-ceiDbProduct CEI_DB_DB2 -ceiDbExecuteScripts true -ceiJdbcClassPath
<classpath> -ceiDbHostName localhost -ceiDbPort <port> -ceiDbUser <userid>
-ceiDbPassword <password>
```

新

```
62 manageprofiles.bat -create -profileName -templatePath
¥profileTemplates¥default.wbiserver -dbDelayConfig false -dbType
DB2_Universal -dbJDBCClasspath <classpath> -dbHostName -dbServerPort <port>
-dbUserId <userid> -dbPassword <password>
```

## J2EE ロールの BPEAPIUser と TaskAPIUser のマップに使用する値「Everyone」

J2EE ロールの BPEAPIUser と TaskAPIUser のマップに値「Everyone」を使用することは推奨しません。

### 推奨されるマイグレーション・アクション

J2EE ロールの BPEAPIUser と TaskAPIUser のマップに値「Everyone」を使用した場合は、Business Process Choreographer API を使用する前に、ログインして Business Process Choreographer クライアント・アプリケーションを修正してください。

## FailedEventManagerMBean インターフェースと API

以下に示す FailedEventManagerMBean インターフェース、メソッド、および操作は推奨しません。

- com.ibm.wbiserver.manualrecovery.FailedEventWithParameters (クラス全体)
- com.ibm.wbiserver.manualrecovery.FailedEventManager (メソッド)
- FailedEventManagerMBean.xml (操作)

### 推奨されるマイグレーション・アクション

カスタム・コードを使用して FailedEventManagerMBean で失敗したイベントを管理する場合に限り、新しいインターフェースと MBean 操作に切り替えることをお勧めします。推奨する新しいインターフェース、メソッド、および操作を次の表に示します。

表 7. *FailedEventManagerMBean* の新しいインターフェース、メソッド、および操作

非推奨のインターフェース、操作、またはメソッド	新しいインターフェース、操作、またはメソッド
<b>com.ibm.wbiserver.manualrecovery. FailedEventWithParameters</b>	com.ibm.wbiserver.manualrecovery.SCAEvent
<b>com.ibm.wbiserver.manualrecovery. FailedEventManager</b>	
<ul style="list-style-type: none"> <li>リスト <code>getFailedEventsForDestination(String destModuleName, String destComponentName, String destMethodName, int pagesize)</code> により <code>FailedEventReadException</code> をスロー</li> <li>リスト <code>getFailedEventsForTimePeriod(Date begin, Date end, int pagesize)</code> により <code>FailedEventReadException</code> をスロー</li> </ul>	リスト <code>&lt;FailedEvent&gt; queryFailedEvents(QueryFilters queryFilters, int offset, int maxRows)</code> により <code>FailedEventReadException</code> をスロー
<code>FailedEventWithParameters</code> <code>getFailedEventWithParameters(String msgId)</code> により <code>FailedEventDataException</code> をスロー	<code>SCAEvent</code> <code>getEventDetailForSCA(FailedEvent failedEvent)</code> により <code>FailedEventDataException</code> をスロー
<code>void discardFailedEvents(String[] msgIds)</code> により <code>DiscardFailedException</code> をスロー	<code>void discardFailedEvents(List&lt;FailedEvent&gt; failedEvents)</code> が <code>DiscardFailedException</code> によりスロー
<code>void resubmitFailedEvents(String[] msgIds)</code> により <code>ResubmissionFailedException</code> をスロー	<code>void resubmitFailedEvents(List failedEvents)</code> により <code>ResubmissionFailedException</code> をスロー
<b>FailedEventManagerMBean.xml</b>	
<ul style="list-style-type: none"> <li><code>getFailedEventsForDestination</code></li> <li><code>getFailedEventsForTimePeriod</code></li> </ul>	<code>queryFailedEvents</code>
<code>getFailedEventWithParameters</code>	<code>getEventDetailForSCA</code>
<code>discardFailedEvents</code>	次のパラメーターでの <code>discardFailedEvents</code> <ul style="list-style-type: none"> <li><code>name="failedEvents"</code></li> <li><code>description="失敗したイベントのリスト"</code></li> <li><code>type="java.util.List"</code></li> </ul>
<code>resubmitFailedEvents</code>	<code>resubmitFailedEvents</code> <ul style="list-style-type: none"> <li><code>name="failedEvents"</code></li> <li><code>description="失敗したイベントのリスト"</code></li> <li><code>type="java.util.List"</code></li> </ul>

## Microsoft SQL Server 用の WebSphere Connect JDBC ドライバー (DataDirect より)

WebSphere Application Server Supplemental CD に登録されている Microsoft SQL Server 用の WebSphere Connect JDBC ドライバー (DataDirect より) は、WebSphere Application Server、バージョン 7 の CD には同梱されていません。

### 推奨されるマイグレーション・アクション

同梱されている DataDirect ドライバーを使用する Microsoft SQL データベースをマイグレーションして、Microsoft が提供する新しい JDBC ドライバーを代わりに使用する必要があります。新しい JDBC ドライバーは、現時点では WebSphere Process Server と WebSphere Enterprise Service Bus ではサポートされていませんが、今後サポートされる予定です。別のデータベース・タイプ (Microsoft SQL に組

み込まれているドライバーなど)に変更するか、WebSphere Process Server と WebSphere Enterprise Service Bus でサポート予定の新しい JDBC ドライバーがリリースされてからマイグレーションしてください。

## WebSphere Process Server バージョン 6.1.2 で使用すべきでない フィーチャー

WebSphere Process Server バージョン 6.1.2 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 6.1 で使用すべきでない フィーチャー

### Container Manager Persistence over Anything (CMP/A)

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。

- com.ibm.websphere.rsadapter.WSProceduralPushDownHelper
- com.ibm.websphere.rsadapter.WSPushDownHelper
- com.ibm.websphere.rsadapter.WSPushDownHelperFactory
- com.ibm.websphere.rsadapter.WSRelationalPushDownHelper

#### 推奨されるマイグレーション・アクション

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。

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。

## JACL スクリプト (WebSphere Application Server バージョン 6.1 では非推奨)

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。

#### 推奨されるマイグレーション・アクション

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。

1. `<install_root>%ProcessChoreographer%admin%bpcTemplates.jacl`
2. `<install_root>%ProcessChoreographer%config%bpeconfig.jacl`

3. <install\_root>%ProcessChoreographer%config%bpeunconfig.jacl
4. <install\_root>%ProcessChoreographer%config%bpeupgrade.jacl
5. <install\_root>%ProcessChoreographer%config%clientconfig.jacl

## IBM Web Services Client for C++

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せん。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せんが、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。

### 推奨されるマイグレーション・アクション

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。

## Business Process Choreographer

### 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)`

## 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。

### 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せん。

- 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にビジネス・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.createByClass(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

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。

### 推奨されるマイグレーション・アクション

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。

## zOS

`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。

### 推奨されるマイグレーション・アクション

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。

## WebSphere InterChange Server

サポートされる WebSphere InterChange Server API 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) は現在是非推奨ではなくなりました。

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。

### 推奨されるマイグレーション・アクション

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。

## WebSphere Enterprise Service Bus (WESB)

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。

### 推奨されるマイグレーション・アクション

新規プロパティーが 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。

## 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

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。

### 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。

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。

### 推奨されるマイグレーション・アクション

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。

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。

サポートされる WebSphere InterChange Server API 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ので、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 で使用する代替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。

```
getInputMessageType()
getOutputMessageType()
```

### 推奨されるマイグレーション・アクション

代替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。

---

## 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場合、バージョン 6.2 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際に、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ことがあります。

```
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。

- プロファイル作成の一部として設定された WebSphere Application Server 変数は、変更しないでください。

古いプロファイルに設定されているこれらの変数を変更した場合、以下のよう  
に、データベース接続やロードなどのクラス欠落例外が発生する可能性があります。

```
10/25/08 13:22:39:650 GMT+08:00] 0000002e J2CUtilityCla E J2CA0036E: An  
exception occurred while invoking method setDataSourceProperties on  
com.ibm.ws.rsadapter.spi.WSManagedConnectionFactoryImpl used by resource  
jdbc/com.ibm.ws.sib/ewps6101.Messaging-BPC.cwfpcCell01.Bus :  
com.ibm.ws.exception.WsException: DSRA0023E: The DataSource  
implementation class "com.ibm.db2.jcc.DB2XADataSource" could not be  
found.DB2,
```

Derby ドライバーと SQL Embedded JDBC ドライバーは、WebSphere Process Server の製品インストールにバンドルされています。これらのドライバーを上位バージョンに変更する必要がある場合、以下に示すとおり、製品インストールと同じ場所にドライバーをコピーする必要があります。

- **Derby:** `%was.install.root%\%derby%\lib`
- **DB2:** `%was.install.root%\universalDriver_wbi\lib`
- **SQL:** `%was.install.root%\lib`

新規の JDBC プロバイダーとデータ・ソースがアプリケーションに対して必要な場合、有効な `jdbcclasspath` を選択して WebSphere Application Server 変数を設定

することにより、これらのリソースを作成することができます。例えば、前のインストールでは存在しなかった DB2 をセル・レベルで設定する必要がある場合、以下の手順を実行します。

1. 管理コンソールで、「リソース」 → 「JDBC」 → 「JDBC プロバイダー」 → 「DB2 Universal JDBC ドライバー・プロバイダー (XA)」にナビゲートします。

2. 「クラスパス」ボックスで、以下のパスを設定します。

- DB2UNIVERSAL\_JDBC\_DRIVER\_PATH =%was.install.root%/universalDriver\_wbi/lib

- DB2UNIVERSAL\_JDBC\_DRIVER\_NATIVEPATH=""

独自のドライバーが必要な場合は、パスとして

DB2UNIVERSAL\_JDBC\_DRIVER\_PATH=%myDriverLocation% を設定します。

- WebSphere Process Server の旧バージョンからバージョン 6.2 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・ファイルを探し、それらを表示して、手掛かりがないかを調べます。

バージョン 6.1.x からのマイグレーションの場合

- migration\_backup/logs/backupConfig.profile\_name.timestamp.log
- migration\_backup/logs/restoreConfig.profile\_name.timestamp.log
- migration\_backup/logs/WBIMigration.profile\_name.timestamp.log
- migration\_backup/logs/WBIPostUpgrade.profile\_name.timestamp.log
- migration\_backup/logs/WBIPreUpgrade.timestamp.log
- migration\_backup/logs/WBIProfileUpgrade.ant.profile\_name.timestamp.log

バージョン 6.0.2.x からのマイグレーションの場合

- profile\_root/logs/WASPostUpgrade.timestamp.log
- migration\_backup/WASPreUpgrade.timestamp.log
- migration\_backup/logs/WASPostUpgrade.profile\_name.timestamp.trace
- migration\_backup/logs/WASPreUpgrade.timestamp.trace
- migration\_backup/logs/WBIPostUpgrade.profile\_name.timestamp.log
- migration\_backup/logs/WBIPreUpgrade.timestamp.log
- migration\_backup/logs/WBIProfileUpgrade.ant.profile\_name.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。

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 6.2 の `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、発生することがあります。この場合は、バージョン 6.2 の `install_root/bin` ディレクトリーに WBIPreUpgrade スクリプトが存在することを確認してから、このスクリプトをこのディレクトリーから起動します。リモート・システムにマイグレーションする場合、WBIPreUpgrade は `migration_copy/bin` ディレクトリーから起動する必要があります。詳しくは、67 ページの『リモート・システムへのマイグレーション』を参照してください。

この問題は、WBIPreUpgrade コマンドを WebSphere Process Server バージョン 6.2 `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 6.2 `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。リモート・システムにマイグレーションする場合、WBIPreUpgrade は `migration_copy/bin` ディレクトリーから起動する必要があります。詳しくは、67 ページの『リモート・システムへのマイグレーション』を参照してください。

- 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 コマンド行ユーティリティー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 6.2 *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 6.2 *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 6.2 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 6.2 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。その際、選択したタイムアウトの少なくとも 3 倍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

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、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をすべて合計した時間です。

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/profile_name/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 6.2 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` コマンド行ユーティリティー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 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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 6.2 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 6.2 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 ノードから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 6.2 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 6.2 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* コマンド行ユーティリティー』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を WebSphere Process Server バージョン 6.1.x および 6.0.2.x から バージョン 6.2 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 6.1.x および 6.0.2.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 6.2 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 6.1.x および 6.0.2.x ノードを停止すると、バージョン 6.1.x および 6.0.2.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 6.1.x および 6.0.2.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. 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 6.2 のプロファイルを削除します。以下の手順を実行します。
    - a. コマンド・プロンプトを開き、使用するオペレーティング・システムに基づいて以下のいずれかのコマンドを実行します。

```
- i5/OS i5/OS プラットフォーム: manageprofiles -delete  
-profileName profile_name
```

```
- Linux UNIX Linux および UNIX プラットフォーム:  
manageprofiles.sh -delete -profileName profile_name
```

```
- Windows Windows プラットフォーム: manageprofiles.bat -delete  
-profileName profile_name
```

変数 `profile_name` は削除するプロファイルの名前を示します。

- b. 以下のログ・ファイル調べて、プロファイルの削除が完了したことを確認します。

```
- i5/OS i5/OS プラットフォーム: user_data_root/profileRegistry/logs/  
manageprofiles/profile_name_delete.log
```

```
- Linux UNIX Linux および UNIX プラットフォーム:  
install_root/logs/manageprofiles/profile_name_delete.log
```

```
- Windows Windows プラットフォーム:  
install_root¥logs¥manageprofiles¥profile_name_delete.log
```

6. 前のステップで削除したバージョン 6.2 のプロファイルの `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 6.1.x および 6.0.2.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 6.2 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 6.1.x および 6.0.2.x 構成内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。

マイグレーション中にアプリケーション・インストールがこの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 6.1.x および 6.0.2.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 6.2 構成に手動でインストールできます。

- 管理対象ノードをバージョン 6.2 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。
- アプリケーションから CBE イベントを発行する場合、マイグレーション後にサーバーを起動すると、以下の例外が表示されることがあります。

NameNotFoundException for event bus

```
[9/17/08 20:00:30:718 CST] 0000003c B0Core | No schema found with namespace "http://www.ibm.com/xmlns/prod/websphere/j2ca/jdbc/db2admincustomer" and location Db2admincustomer.xsd
```

```
[9/17/08 20:00:30:734 CST] 00000037 EventBusSender E com.ibm.events.emitter.impl.EventBusSender initialize CEIEM0020E The emitter failed to initialize because the JNDI lookup on the event bus home name failed.
```

JNDI name: ejb/com/ibm/events/bus/EventBus

Context: Node01Cell/nodes/Node01/servers/server1

Exception (if any): javax.naming.NameNotFoundException

この問題は、CEI 関連アプリケーションを開始する前に JCAOutbound アプリケーションから CBE イベントを発行しようとした場合に発生します。この問題を修正するには、CEI が開始されてからアプリケーションが起動するように、アプリケーションの開始ウェイトを増やします。

- ネットワーク・デプロイメント環境で、マイグレーション後にビジネス・ルール・マネージャーにアクセスしたときに、エラー SRVE0026E: [Servlet Error]-[com/ibm/wbiservers/brules/BusinessRuleManager]: java.lang.NoClassDefFoundError が発生した場合は、そのノードの通常マイグレーションを続行する前に、デプロイメント・ターゲットにビジネス・ルール・マネージャー・アプリケーションを手動でインストールする必要があります。詳しくは、133 ページの『ネットワーク・デプロイメント環境でのビジネス・ルール・マネージャーのマイグレーション』を参照してください。
- 管理対象ノードのバージョン 6.2 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 6.2 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/62AppServer/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 6.2 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 6.2 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

13 ページの『Business Process Choreographer に関する事前マイグレーションの考慮事項』

サーバーで Business Process Choreographer を稼働する場合は、Business Process Choreographer をマイグレーションする前に、計画と検討が必要な特定の事項に注意する必要があります。

### 関連タスク

135 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

 **WBIPreUpgrade** コマンド行ユーティリティー

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。

 **WBIPostUpgrade** コマンド行ユーティリティー

WebSphere Process Server の WBIPostUpgrade コマンドを使用して、指定された *backupDirectory* に WBIPreUpgrade コマンドによって保存されたプロファイル構成を取得します。

### 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 または WebSphere Business Integration Server Express 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

230 ページの『WebSphere InterChange Server または WebSphere Business Integration Server Express からマイグレーションする場合の制限事項』  
WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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する必要がある場合があります。

### 関連資料

189 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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。

178 ページの『事前マイグレーションの考慮事項』  
WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express API に対するロギングとトレースの有効化

サポートされる WebSphere InterChange Server または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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。

### 手順

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 または WebSphere Business Integration Server Express 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、サポートされる WebSphere InterChange Server API 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。

### 関連情報



サポートされる WebSphere InterChange Server API

## 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express の断片コードで使用されるほかの変数は直列化され、*CollabTemplateName\_var* という名前の BPEL 変数に、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。断片間で変数を共用する (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 または WebSphere Business Integration Server Express 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 または WebSphere Business Integration Server Express 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  
1001 Hillsdale Blvd., Suite 400  
Foster City, CA 94404  
U.S.A.

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

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

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

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

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 ロゴ、および [ibm.com](http://ibm.com) は、International Business Machines Corporation の米国およびその他の国における商標または登録商標です。これらおよび他の IBM 商標に、この情報の最初に現れる個所で商標表示 (® または ™) が付されている場合、これらの表示は、この情報が公開された時点で、米国において、IBM が所有する登録商標またはコモン・ロー上の商標であることを示しています。このような商標は、その他の国においても登録商標またはコモン・ロー上の商標である可能性があります。現時点での IBM の商標リストについては、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) の「Copyright and trademark information」をご覧ください。

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

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

Java は、Sun Microsystems, Inc. の米国およびその他の国における商標です。

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

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。









Printed in Japan