WebSphere®

IBM WebSphere Process Server for Multiplatforms

バージョン 7.0.0

WebSphere Process Server のマイグレーション

IBM®

WebSphere®

IBM WebSphere Process Server for Multiplatforms

バージョン 7.0.0

WebSphere Process Server のマイグレーション

IBM®

本書は、WebSphere Process Server for Multiplatforms バージョン 7、リリース 0、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。

本書についてのご意見は、doc-comments@us.ibm.com へ E メールでお寄せください。皆様の率直なご意見をお待ちしています。

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

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 7.0.0
Migrating WebSphere Process Server

第1刷 2010.4

目次

第 1 章 マイグレーション: バージョン間

バージョン間マイグレーションの場合、アプリケーションを製品の新規バージョンにマイグレーションするには、アプリケーションを再デプロイするという方法、オーサリング・ツールを使用してアプリケーションを更新し、再デプロイするという方法、またはランタイム・マイグレーション・ツールを使用して、すべての構成情報を維持し、自動的にアプリケーションを再デプロイするという方法があります。

マイグレーションの概要

以前のバージョンの WebSphere® Process Server から、より新しいバージョンの WebSphere Process Server にアプリケーション、構成、およびデータベースを移動するプロセスを、バージョン間マイグレーション、または単にマイグレーションと呼びます。

バージョン間のマイグレーションとは

バージョン間のマイグレーションとは、旧バージョンの WebSphere Process Server と関連付けられたプロファイル、アプリケーション、データを、新しくインストールされたバージョンの WebSphere Process Server に移動することを指します。

バージョン間マイグレーションの概要

バージョン間のマイグレーション (または単に「マイグレーション」) とは、旧リリースの WebSphere Process Server で開発されたアプリケーションをバージョン 7.0 に移動するプロセスを指します。マイグレーションの実行には、WebSphere Integration Developer または WebSphere Business Modeler に付属の一連のマイグレーション機能を使用するか (アプリケーションをマイグレーションする場合)、実稼働環境で一連のランタイム・マイグレーション手順およびツールを使用することができます (実動構成全体、アプリケーション、データベースをマイグレーションする場合)。

WebSphere Integration Developer および WebSphere Business Modeler では、以前のバージョンを使用して開発されたアプリケーションおよびワークスペースをバージョン 7.0 にインポートし、マイグレーションすることができます。アプリケーションがバージョン 7.0 にマイグレーションされたら、アプリケーションをランタイムでバージョン 7.0 に直接デプロイするか、バージョン 7.0 の新機能を活用できるようにアプリケーションを拡張してからデプロイすることができます。このマイグレーション・スタイルを成果物マイグレーションと呼びます。

実稼働環境にデプロイされたアプリケーションのマイグレーションは、アプリケーションを新バージョンに再デプロイするだけでは実現できません。実動トポロジーの構成、製品データベース、およびデータベース内の製品データはすべて、一貫した一連の BPM 手順とツールを使用することでバージョン 7.0 にマイグレーションされます。実動構成、アプリケーション、およびデータベースをマイグレーションするための一連の手順とツールに関連するプロセスをランタイム・マイグレーションと呼びます。

共通のランタイム・マイグレーション手順およびツールを使用する BPM 製品には、以下のようなものがあります。

- WebSphere Dynamic Process Edition
- WebSphere Business Services Fabric
- WebSphere Process Server
- WebSphere Enterprise Service Bus
- WebSphere Business Monitor
- WebSphere Business Compass

マイグレーションのソースである実稼働環境から、並列するターゲットの実稼働環境にアプリケーションを手動で再デプロイすることもできます。このマイグレーション・スタイルを手動マイグレーションと呼びます。

複数の製品のマイグレーション

ランタイム・マイグレーション方式では、ソース環境で一緒にインストールおよび構成されている複数の BPM 製品をマイグレーションすることができます。例えば、マイグレーションのソース・インストール・ディレクトリーに WebSphere Process Server と WebSphere Business Monitor、および両方の製品によって拡張された一連のプロファイルが含まれている場合、ランタイム・マイグレーション方式を使用すると、このソース環境を、両方の製品が同じターゲット・インストール・ディレクトリーにインストールされたターゲット環境にマイグレーションすることができます。

製品の更新

バージョン間のマイグレーション・プロセスは、暫定修正や更新を開発環境と実稼働環境に適用するプロセスとは異なります。暫定修正、フィックスパック、リフレッシュ・パックの各形式での更新については、ご使用の BPM 製品の「更新」に関するトピックを参照してください。

継承製品のマイグレーション

バージョン間のマイグレーション・プロセスは、継承製品を WebSphere Process Server にマイグレーションするプロセスとは異なります。継承製品のマイグレーションについて詳しくは、 163 ページの『第 2 章 マイグレーション: 継承製品』を参照してください。

BPM マイグレーション・ロードマップ

WebSphere Process Server マイグレーション・ロードマップは、バージョン間マイグレーションに必要な作業の概要を示します。

以下のフロー・ダイアグラムおよびマイグレーション作業の概要説明を参照して、バージョン間マイグレーションに必要な作業について学んでください。

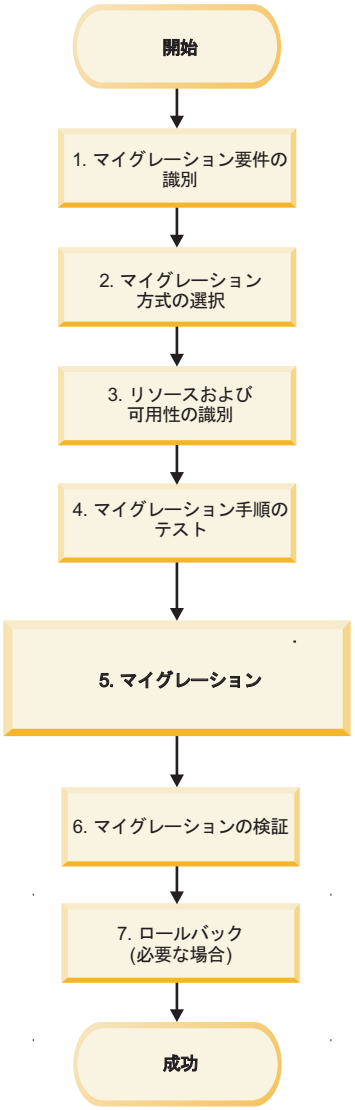

図 1. バージョン間マイグレーションの WebSphere Process Server マイグレーション・ロードマップ

1. マイグレーション要件の識別

マイグレーションを計画する際の最初のステップは、マイグレーション要件を識別することです。

マイグレーション・プロセスに関連した一連の考慮事項のリストについては、『マイグレーション・メソッドの比較』のトピックを参照してください。

マイグレーションの最終目標がバージョン 7.0 で提供される新機能を利用することである場合は、『このリリースの新機能』で説明されている WebSphere Process Server バージョン 7.0 の新機能を検討してください。

2. マイグレーション・メソッドの選択

マイグレーションを行うときは、以下の 3 つのマイグレーション・メソッドから選択できます。

- ランタイム・マイグレーション
- 手動マイグレーション
- 成果物マイグレーション

各マイグレーション・メソッドを検討し、どのマイグレーション・メソッドが要件に適合するかを判別するには、『マイグレーション・メソッド』のトピックを参照してください。

3. リソースおよび可用性の識別

マイグレーションを計画する際には、マイグレーションに必要なすべてのリソースが使用可能であるかどうかを識別することが重要です。これらのリソースには、以下が含まれます。

- 人材: 何人必要で、必要なスキル・レベルは何か。その人材を必要とする時間フレームはどれくらいか。
- ハードウェアおよびソフトウェア・リソース: マイグレーションを確実に成功させるために確保する必要があるハードウェアまたはソフトウェアは何か。

WebSphere Process Server バージョン 7.0 のハードウェアおよびソフトウェア要件について詳しくは、『ハードウェアおよびソフトウェア要件』を参照してください。

4. マイグレーション手順のテスト

マイグレーションを行う前に、マイグレーション手順を詳細にテストしてください。

- 新規環境でのアプリケーションのテスト
- ステージング環境でのマイグレーション手順のテスト
- テスト・システムでのロールバック計画の実習

マイグレーションに最適なテストを判別する際は、テストを正常に実行するために必要なリソースを念頭に置いてください。

5. マイグレーション

ご使用の環境をマイグレーションするために選択したマイグレーション・メソッドに関連するマイグレーション手順を使用します。

6. マイグレーションの検証

マイグレーションを行った後は、使用したマイグレーション・メソッドに応じて、以下のメソッドのいずれかを使用してマイグレーションが成功したかどうかを検証します。

- ランタイム・マイグレーション・メソッドを使用した場合は、『マイグレーションの検査』を参照してください。
- 手動マイグレーション・メソッドを使用した場合は、アプリケーションが正常に機能することを確認します。
- 成果物マイグレーション・メソッドを使用した場合は、アプリケーションが正常に機能することを確認します。

7. ロールバック (必要な場合)

マイグレーションが正常に完了しなかった場合は、環境をロールバックし、マイグレーションを再び実行しなければならないことがあります。使用したマイグレーション・メソッドに応じて、以下のロールバック・メソッドのいずれかを使用します。

- ランタイム・マイグレーション・メソッドを使用した場合は、『環境のロールバック』を参照します。
- 手動マイグレーション・メソッド使用した場合は、アプリケーションのアンインストールと再インストールが必要になることがあります。
- 成果物マイグレーション・メソッドを使用した場合は、アンインストールした後に、WebSphere Integration Developer または WebSphere Business Modeler を使用して、アプリケーションとソース成果物の再インポートと再マイグレーションが必要になることもあります。

マイグレーション方式

新しいバージョンの WebSphere Process Server に移動することを検討する場合、選択できるバージョン間のマイグレーション方式には、ランタイム・マイグレーション、手動マイグレーション、成果物マイグレーションの 3 種類があります。

- 『ランタイム・マイグレーション (実稼働環境)』
- 6 ページの『手動マイグレーション (並列の実稼働環境)』
- 7 ページの『成果物マイグレーション (開発ツールのマイグレーションを使用した並列の実稼働環境)』

ランタイム・マイグレーション (実稼働環境)

実稼働環境では、ランタイム・マイグレーション手順およびツールを使用して、トポロジー構成、アプリケーション、データベースを新しいバージョンの WebSphere Process Server にマイグレーションすることができます。ランタイム・マイグレーション手順およびツールは、スタンドアロン環境とネットワーク・デプロイメント環境の両方のマイグレーションに対応します。さらに、リモート・システムへのマイグレーションや (スタンドアロン環境のみ)、オペレーティング・システムをサポート対象バージョンにアップグレードしている間のマイグレーション (スタンドアロン環境のみ) などのタイプや、フル・ダウン時間枠でのマイグレーションおよび最小限のダウン時間枠でのマイグレーションをサポートするネットワーク・デプロイメント・タイプにも対応します。ランタイム・マイグレーション・プロセスでは、ソースの実動構成がターゲット環境に複製されます。マイグレーション・プロセス中には、ターゲットの実稼働環境がソースの実稼働環境に置き換わるため、2 つの環境が並列で運用されることはありません。

以下のような場合には、ランタイム・マイグレーション手順およびツールを使用してください。

- 開発ツールおよび開発環境に依存せずに、アプリケーションを新しいバージョンに移行する場合。
- ソースの実稼働環境の構成およびアプリケーションが、ターゲットの実稼働環境で自動的に複製されるようにする場合。

- ソース環境内で開始した長期実行プロセスかヒューマン・タスクのインスタンスまたはその両方があり、ターゲット環境で完了する必要がある場合。
- ソース環境のキュー内に製品データがあるか、ソース環境で作成された製品データベース内に失敗したイベントがあり、それらをマイグレーション後もターゲットの実稼働環境で管理する必要がある場合。
- 実稼働環境でマイグレーションを実行するためのダウン時間枠を設けられる場合。

ランタイム・マイグレーションで必要となるタスクの概要は以下のとおりです。

1. 新しい製品バージョンをインストールします。
2. 実動プロファイルおよびデータベースをすべてバックアップします。
3. ソース環境の各プロファイルをターゲット環境にマイグレーションします。
4. 製品データベースをマイグレーションまたはアップグレードします。
5. 製品データベースのデータをマイグレーションします。

ランタイム・マイグレーション手順およびツールについて詳しくは、1 ページの『マイグレーションの概要』トピックを参照してください。

手動マイグレーション (並列の実稼働環境)

マイグレーション手順およびツールを使用する代わりに、バージョン間の手動マイグレーション・プロセスを使用することができます。手動マイグレーション・プロセスを使用すると、ソースの実稼働環境とは異なるように最初から構成される、並列するターゲットの実稼働環境を作成することができます。その後、アプリケーションを選択してソースの実稼働環境からターゲットの実稼働環境に再デプロイすることができます。再デプロイされたアプリケーションは、並列の実稼働環境内に専用のデータベース表とアプリケーション・データを作成するため、ソースの実稼働環境用に構成されたデータベース内に格納されたアプリケーション・データにはアクセスできません。

以下のような場合には、手動マイグレーション・プロセスを使用してください。

- 開発ツールおよび開発環境に依存せずに、アプリケーションを新しいバージョンに移行する場合。
- 新しいバージョンの WebSphere Process Server にマイグレーションするプロセスの一部として、トポロジーを再構成する場合。
- 長期実行プロセス・インスタンスおよびヒューマン・タスクがない場合。または、新しいインスタンスがターゲット実稼働環境で開始されたときに、ソース環境内でプロセス・インスタンスおよびヒューマン・タスクをすべて完了させるまでの間、並列実稼働環境を稼働できる場合。
- ソース環境のキューにアプリケーション・データがあるか、ソース環境で作成された製品データベース内に失敗したイベントがあり、それらをソースの実稼働環境で完了まで管理できる一方、新しいメッセージとイベントはターゲットの実稼働環境に並列で経路指定される場合。
- 実稼働環境でダウン時間を設けることができず、ソースの実稼働環境とターゲットの実稼働環境を並列で同時に管理できる場合。
- ソースの実稼働環境からターゲットの実稼働環境に、アプリケーションを選択して再デプロイする場合。

手動マイグレーションで必要となるタスクの概要は以下のとおりです。

1. 新しい製品バージョンをインストールします。
2. 必要な並列の実稼働環境を構成します。
3. ソース環境からターゲットの実稼働環境にアプリケーションを手動でデプロイします。
4. オプション: 両方の環境を並行して実行し、進行中のビジネス・プロセス・インスタンスおよびヒューマン・タスク・インスタンスはソース環境で完了させ、新規インスタンスはターゲット環境で開始するようにします。

成果物マイグレーション (開発ツールのマイグレーションを使用した並列の実稼働環境)

成果物マイグレーション・プロセスは、並列するターゲットの実稼働環境の構成の点では手動マイグレーション・プロセスと似ていますが、アプリケーションを手動でソース環境からターゲットの実稼働環境に直接再デプロイするのではなく、アプリケーションが開発環境にインポートされ、開発ツールによってマイグレーションされます。この結果、アプリケーション成果物が新しいバージョンにマイグレーションされ、バージョン 7.0 の新機能を活用できるように各アプリケーションを変更することができます。この後、アプリケーションをテストして、並列するターゲットの実稼働環境にデプロイすることができます。手動マイグレーション・プロセスと同様に、アプリケーションがターゲットの実稼働環境にデプロイされると、新しいデータベース表セットが作成されます。このため、これらのアプリケーションは、ソースの実稼働環境用に構成されたデータベース内に格納されたアプリケーション・データにはアクセスできません。

以下のような場合には、成果物マイグレーションを使用してください。

- 開発ツールと開発環境を利用して、アプリケーション成果物を新しいバージョンにマイグレーションし、アプリケーションの互換性を検証する場合。
- 開発ツールを利用して、バージョン 7.0 の新機能を活用できるようにアプリケーションを更新する場合。
- 新しいバージョンの WebSphere Process Server にマイグレーションするプロセスの一部としてトポロジーを再構成する場合、またはソースの実稼働環境の構成を並列の実稼働環境内で手動で複製できる場合。
- 長期実行プロセス・インスタンスおよびヒューマン・タスクがない場合。または、新しいインスタンスがターゲット実稼働環境で開始されたときに、ソース環境内でプロセス・インスタンスおよびヒューマン・タスクをすべて完了させるまでの間、並列実稼働環境を稼働できる場合。
- ソース環境のキューにアプリケーション・データがあるか、ソース環境で作成された製品データベース内に失敗したイベントがあり、それらをソースの実稼働環境で完了まで管理できる一方、新しいメッセージとイベントはターゲットの実稼働環境に並列で経路指定される場合。
- 実稼働環境でダウン時間を設けることができず、ソースの実稼働環境とターゲットの実稼働環境を並列で同時に管理できる場合。
- 開発ツールを使用してソースの実稼働環境からバージョン 7.0 にアプリケーションを選択してマイグレーションし、これらのアプリケーションを選択してターゲットの実稼働環境にデプロイする場合。

成果物マイグレーションで必要となるタスクの概要は以下のとおりです。

1. 新しい製品バージョンをインストールします。
2. 必要な並列の実稼働環境を構成します。
3. ソースの実稼働環境から開発ツールにアプリケーションをインポートし、開発ツールのマイグレーション手順に従ってアプリケーションをマイグレーションします。
4. オプション: バージョン 7.0 の新機能を活用できるようにマイグレーション済みのアプリケーションを更新します。
5. マイグレーション済みのアプリケーションを開発ツールからターゲットの実稼働環境に手動でデプロイします。
6. オプション: 両方の環境を並行して実行し、進行中のビジネス・プロセス・インスタンスおよびヒューマン・タスク・インスタンスはソース環境で完了させ、新規インスタンスはターゲット環境で開始するようにします。

成果物マイグレーションについて詳しくは、 WebSphere Integration Developer および WebSphere Business Modeler バージョン 7.0 のインフォメーション・センターでマイグレーションに関するセクションを参照してください。

マイグレーション方式の比較

WebSphere Process Serverをバージョン 7.0 にマイグレーションする際に最適なマイグレーション方式を判断するには、その環境におけるステートフル・データの量、システムが対応できるダウン時間の長さ、および以前の構成を保存するかどうかを分析する必要があります。

マイグレーション方式に関する考慮事項

バージョン 7.0 にマイグレーションする際の適切なマイグレーション方式を判断する場合、考慮すべき問題がいくつかあります。以下のセクションには、マイグレーション要件に最適な方式を判断する場合に考慮すべき一連の項目を列挙します。

- 実動データ
- ダウン時間
- 長期実行プロセスおよびヒューマン・タスク
- アプリケーションの拡張
- ターゲット環境構成
- リスクの軽減
- 選択的または段階的なアプリケーション・マイグレーション

実動データ

ランタイム・マイグレーション方式の場合、ソースの実稼働環境がターゲットの実稼働環境に置き換えられます。アプリケーション・データに与える影響として、ソース環境によってデータベース内に作成されたデータは、マイグレーション後のターゲット環境で使用できるようになります。これにより、重要なシナリオが実現されます。例えば、ソース環境で開始したプロセスおよびヒューマン・タスクを、マイグレーション後のターゲット環境で終了することができます。キューに入っているメッセージや、ソース環境内に存在する失敗したイベントをマイグレーション後

のターゲットで管理することができます。この機能が提供されるのはランタイム・マイグレーション方式のみです。手動マイグレーション方式と成果物マイグレーション方式はどちらも並列の実稼働環境を作成します。このような環境では、ソース環境のアプリケーションがターゲット環境にデプロイされる場合でも、ソース環境から独立した完全に別個の専用データベースが別途構成されます。

ダウン時間

手動マイグレーション・プロセスと成果物マイグレーション・プロセスは作成される並列のターゲット環境に依存しますが、ランタイム・マイグレーション方式の場合はソース環境がターゲット環境に置き換えられます。この影響として、ランタイム・マイグレーション方式の場合は、データベースをアップグレードしてソース・バージョンからターゲット・バージョンにマイグレーションする際に、マイグレーションされたサーバーを始動するまでのダウン時間枠が必要となります。ランタイム・マイグレーション手順に記載される最小限のダウン時間の手順は一部のケースで使用できますが、それでもダウン時間の必要がなくなるわけではありません。

手動マイグレーション方式と成果物マイグレーション方式はどちらも、ソース環境と同時に実動で使用できる並列環境を作成する必要があります。ソース環境を停止できる状態になるまで、ソース環境とターゲット環境を並列で実行することができます。異なるバージョンで 2 つの環境を同時に実行できるため、運用が多少複雑になり、また一般に容量も余分に必要となります。

長期実行プロセスおよびヒューマン・タスク

プロセスおよびヒューマン・タスクについては、以下の数種類のシナリオとオプションを考慮する必要があります。

- プロセスとタスクの実行時間が短く、マイグレーションのダウン時間枠が開始する前にソース環境で完了できる場合

  マイグレーション・プロセスのためにダウン時間を設けることができ、ダウン時間枠の前にプロセスとタスクを完了できる場合は、3 つのマイグレーション方式がすべて選択可能なオプションとなります。したがって、その他のマイグレーション要件に応じて、使用するオプションを決定できます。
- これらのプロセスとタスクが実行に長時間を要し、マイグレーションでマイグレーションでダウン時間が生じる可能性がある場合。

  このシナリオでは 3 つのオプションのすべてが実行可能ですが、重要なトレードオフを考慮する必要があります。手動マイグレーション方式と成果物マイグレーション方式を使用する場合は、ソース環境で開始したプロセスがそこで完了するまでの間、並列の実稼働環境を同時に実行する必要があります。ダウン時間枠が決定的要因でなければ、ランタイム・マイグレーション・オプションの方が、ソース環境で開始したプロセスとタスクをマイグレーション後のターゲット環境で完了できるため、このシナリオには適しています。
- マイグレーションのためにダウン時間を設けることができない場合

  ダウン時間を設けることができない場合はランタイム・マイグレーション方式が除外されるため、手動または成果物マイグレーション方式を使用して、アプリケーションを再デプロイできる並列のターゲット環境を作成する必要があります。これらの方式では、2 つの異なるプロセスおよびタスク・データベースを持つ並

列環境が作成されるため、新しいプロセスとタスクはターゲット環境で開始するのが理想的です。また、ソース環境のプロセスとインスタンスが完了するまでは、これら 2 つの環境を並列で実行する必要があります。

アプリケーションの拡張

成果物マイグレーションと開発ツールを使用することの利点は、アプリケーションをバージョン 7.0 の成果物レベルに更新した後、バージョン 7.0 で提供されるフィーチャーを使用してアプリケーションを機能拡張できることです。

ターゲット環境構成

ターゲット環境の構成をソース環境と同じにする必要がある場合は一般に、ランタイム・マイグレーション方式の方が適しています。この理由は、ソース環境のトポロジー構成がターゲット環境に自動的に複製されるためです。ただし、正当な理由により、ソース環境とはまったく異なるようにターゲット環境の構成を変更する必要がある場合は、バージョン間のマイグレーションの前または後に独立した作業として構成変更を行うか、バージョン間のマイグレーションと同時に構成変更する場合は、手動または成果物マイグレーション方式を使用する必要があります。

リスクの軽減

手動および成果物マイグレーション方式で並列環境を提供すると、ターゲットの実稼働環境をソース環境から完全に独立させることができ、既存の利用者への対応はソース環境が行うため、ターゲット環境を実動設定で使用する前に厳密にテストすることができます。さらに、成果物マイグレーションでは、開発ツールを利用して、マイグレーションされるアプリケーションに問題 (後方互換性の問題) がないことを確認できるため、リスクを低減することができます。ランタイムまたは手動マイグレーション方式を利用するシナリオであっても、マイグレーション作業の初期段階としてアプリケーションの互換性を確認するために、開発ツールを使用した成果物マイグレーションの検証が行われることもよくあります。

選択的または段階的なアプリケーション・マイグレーション

1 回のダウン時間枠ですべてのアプリケーションをターゲット・バージョンにマイグレーションしたくない状況がある場合は、手動または成果物マイグレーション・アプローチを使用する必要があります。これらのアプローチでは、ソースとターゲットの 2 つの並列環境のサポートが実現され、マイグレーションされたアプリケーションをターゲット環境に選択的または段階的にデプロイすることができます。これに対し、ランタイム・マイグレーション方式では、すべてのアプリケーションがソース環境からターゲット環境にマイグレーションされます。

マイグレーション方式の比較

以下の表を使用して、 3 つのマイグレーション方式の利点、コスト、リスクを比較してください。

表 1. バージョン間マイグレーション・メソッド: 比較

マイグレーション・メソッド	利点	コスト	リスク
ランタイム・マイグレーション	• 開発ツールに依存しない • ソース環境の構成がターゲット環境で複製される • ソース環境のアプリケーションがターゲット環境にマイグレーションされる • 既存のデータベース表を使用して、ソース環境のアプリケーション・データが移動される • プロセスおよびヒューマン・タスクをソース環境で開始し、ターゲット環境で完了できる • キューに入っているアプリケーション・インスタンス・データおよびソース環境で失敗したイベントをマイグレーション後にターゲット環境で処理できる • 別の実稼働環境を管理するための追加ハードウェア/ソフトウェア・リソースが不要	• ターゲットの実稼働環境がソースの実稼働環境のロールを継承する場合はダウン時間が必要 • ノード上のすべてのアプリケーションを同時にマイグレーション可能な状態にする必要がある • 新しいフィーチャーが自動的に有効にならず、成果物マイグレーションを使用してアプリケーション成果物をマイグレーションしないと使用できない場合がある • 平行実稼働環境をセットアップできない • テストの焦点: – マイグレーション・プロセスを検証するためのエンドツーエンドのテスト – リグレッション・テストとパフォーマンス調整	• 考えられるマイグレーション失敗に対処するロールバック計画を準備しておく必要がある。詳しくは、『環境のロールバック』を参照してください。 • 既存のユーザー・アプリケーションは、新規ランタイムでも、旧ランタイムでの機能レベルと同じレベルで実行しなければならない。ただし、アプリケーションが依存するコードの変更 (JDK の変更など) があり、それによって未変更のアプリケーションに悪影響が及ぶ場合もあります。

表 1. バージョン間マイグレーション・メソッド: 比較 (続き)

マイグレーション・メソッド	利点	コスト	リスク
手動マイグレーション	• 開発ツールに依存しない • ソースからターゲットに構成が自動的にマイグレーションされないため、ターゲットの実稼働環境をソースの実稼働環境とは異なる構成にすることができる • 平行実稼働環境がサポートされる – アプリケーションを選択してマイグレーションできる – ダウン時間がない • 実稼働環境にマイグレーションする前に、広範なテストを実行できる (ただし、通常はリグレッション・テストで十分です) • マイグレーション・ツールに依存しない	• 既存のデータは移されず、新規データベース表が作成される • 新しいフィーチャーが自動的に有効にならず、成果物マイグレーションを使用してアプリケーション成果物をマイグレーションしないと使用できない場合がある • 手動 (スクリプト) でアプリケーションをデプロイする必要がある • クライアント・アプリケーションの更新が必要となる • 並列実行の際に追加ライセンスが必要かどうかについて、ハードウェアおよびソフトウェア・ライセンスを評価しなければならない場合がある	• 既存のユーザー・アプリケーションは、新規ランタイムでも、旧ランタイムでの機能レベルと同じレベルで実行しなければならない。ただし、アプリケーションが依存するコードの変更 (JDK の変更など) があり、それによって未変更のアプリケーションに悪影響が及ぶ場合もあります。

表 1. バージョン間マイグレーション・メソッド: 比較 (続き)

マイグレーション・メソッド	利点	コスト	リスク
成果物マイグレーション	• 新規機能を利用できる • 平行実稼働環境がサポートされる – アプリケーションを選択してマイグレーションできる – ダウン時間がない • 実稼働環境にマイグレーションする前に、広範なテストを実行できる • マイグレーション・ツールに依存しない	• 新しい開発環境が必要になる • 既存のデータは移されず、新規データベース表が使用される • 手動 (スクリプト) でアプリケーションをデプロイする必要がある • クライアント・アプリケーションの更新が必要となる • 並列実行の際に追加ライセンスが必要かどうかについて、ハードウェアおよびソフトウェア・ライセンスを評価しなければならない場合がある • アプリケーションの更新に対してテストを追加してカバーする必要がある	• アプリケーションの更新に、一定のレベルのテストが必要となる場合がある

サポートされているソース・マイグレーション・パス

以下の製品とバージョンの組み合わせは、WebSphere Process Server バージョン 7.0 へのバージョン間マイグレーションのソースとしてサポートされています。

- WebSphere Process Server バージョン 6.2.0.x
- WebSphere Process Server バージョン 6.1.2.x
- WebSphere Process Server バージョン 6.1.0.x
- WebSphere Process Server バージョン 6.0.2.x

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、最初に手動マイグレーション・メソッドを使用して、サポートされているマイグレーション・ソースのいずれかのバージョンへマイグレーションします。その後、ランタイム・マイグレーション・メソッドを使用して、そのバージョンからバージョン 7.0 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・タイプ

ランタイム・マイグレーションは、スタンドアロン環境と Network Deployment 環境のマイグレーションをサポートしています。

スタンドアロン・マイグレーション

ランタイム・マイグレーション手順およびツールは、以下の 3 つのタイプのスタンドアロン・マイグレーション・バリアントをサポートしています。

- **並列マイグレーション**: マイグレーションのソースとターゲットが同じシステム上にある場合
- **リモート・マイグレーション**: マイグレーションのソースとターゲットが異なるシステム上にある場合
- **オペレーティング・システム・アップグレード・マイグレーション**: ソース・システム上のオペレーティング・システムがマイグレーション手順のとき、WebSphere Process Server バージョン 7.0 でサポートされている新バージョンにアップグレードされる場合

以下のセクションでは、これらのタイプのスタンドアロン環境マイグレーション・バリアントのそれぞれについて、さらに詳しく説明します。

スタンドアロン並列マイグレーション

スタンドアロン並列マイグレーション・プロセスは最も単純なランタイム・マイグレーション・シナリオで、ターゲット製品がソース製品と同じシステム上にインストールされ、構成、アプリケーション、および製品データベースを含んでいるスタンドアロン・プロファイルが、ランタイム・マイグレーション手順およびツールを使用してターゲット環境へマイグレーションされます。

スタンドアロン・リモート・マイグレーション

スタンドアロン・リモート・マイグレーション・プロセスを使用すると、1 つのシステムから別のシステムへの構成とアプリケーションのマイグレーションをサポートするために、WebSphere Process Server バージョン 7.0 をマイグレーションのソースと異なるシステム上にインストールできます。スタンドアロン・リモート・マイグレーション・プロセスは、以下のような各種のシナリオをサポートするために使用できます。

- 同じタイプのハードウェア、オペレーティング・システム、およびマイグレーションのソースと同じオペレーティング・システム・バージョンを備えたリモート・システムへのマイグレーション
- 異なるタイプのハードウェア (例えば、64 ビットなど)、異なるオペレーティング・システム、または異なるオペレーティング・システム・バージョンを備えたリモート・システムへのマイグレーション

このプロセスでは、ターゲット・システム上のマイグレーション・コマンドを、ソース・プロファイルのコピーの作成にそれらのコマンドを使用するソース・システムへコピーする必要があります。その後、スナップショット・ディレクトリーはターゲット・システムへコピーされ、プロファイル・マイグレーションのソースとして使用されます。

スタンドアロン・オペレーティング・システム・アップグレード・マイグレーション

スタンドアロン・オペレーティング・システム・アップグレード・マイグレーション・プロセスを使用すると、マイグレーションのソースが入っているシステム上のオペレーティング・システムを、マイグレーション・プロセスのときにアップグレードすることができます。一般に、それが必要となるのは、ソース製品バージョンが入っているオペレーティング・システムのバージョンが、もはや WebSphere Process Server バージョン 7.0 でサポートされていない場合です。

このプロセスでは、前のバージョンのオペレーティング・システム上にある各ソース・プロファイルのコピー、コピーされたソース・プロファイルのリモート・ロケーションへのバックアップ、オペレーティング・システムの新バージョンへの再インストール、ターゲット製品のインストール、更新されたオペレーティング・システムを備えたマイグレーション・システムへのコピー済みソース・プロファイルのリストアが必要で、その後、スナップショット・ディレクトリーがプロファイル・マイグレーションのソースとして使用されます。

Network Deployment マイグレーション

Network Deployment 環境マイグレーションはスタンドアロン環境マイグレーションより複雑で、デプロイメント・マネージャー、クラスター、ノード、およびさまざまな有効範囲を持つ製品データベースを適切な順序でマイグレーションする必要があります。すべての Network Deployment マイグレーションで、WebSphere Process Server バージョン 7.0 をマイグレーションのソース製品と並列にインストールする必要があります。マイグレーションのソースが追加の BPM 製品によって拡張されている場合は、それらの製品を WebSphere Process Server バージョン 7.0 と同じインストール・ディレクトリーにインストールする必要があります。

注: スタンドアロン・プロファイルのシナリオでサポートされているリモート・マイグレーションおよびオペレーティング・システム・アップグレード・マイグレーションのバリアントは、Network Deployment マイグレーションではサポートされません。

Network Deployment マイグレーション手順では、2 つの異なるタイプのノードについて言及されます。**クラスター化ノード**と**非クラスター化管理対象ノード**です。クラスター化ノードには、クラスターのメンバーであるサーバーが少なくとも 1 つ格納されています。非クラスター化管理対象ノードには、クラスターのメンバーであるサーバーは格納されていません。

ランタイム・マイグレーション・ツール

スタンドアロン環境および Network Deployment 環境のマイグレーションでは、実稼働環境を管理 (デプロイメント・マネージャー、サーバー、およびノードを開始および停止) し、構成プロファイルをマイグレーションし、製品データベースをアップグレードし、アプリケーション・データをマイグレーションする必要があります。ランタイム・マイグレーション手順は、このプロセスの手引きを行い、必要なステップを実行するためにランタイム・マイグレーション・ツールが使用されます。

以下の 3 つのツール・セットがランタイム・マイグレーション手順をサポートします。

- 『プロファイル・マイグレーション・ツール』
- 17 ページの『データベース・アップグレードおよびマイグレーション・ツール』
- 18 ページの『WebSphere Application Server 管理ツール』

以下のセクションでは、これらのツール・グループのそれぞれについて要約します。

プロファイル・マイグレーション・ツール

プロファイル・マイグレーション・ツールは、マイグレーションされるセル、クラスター、非クラスター化管理対象ノード、またはスタンドアロン・サーバーに寄与するプロファイルのマイグレーションに使用されます。

プロファイル・マイグレーション・ツールは、各プロファイルについて、以下の 3 ステップのプロセスをサポートします。

1. マイグレーションされるソース・プロファイルから構成ファイルのスナップショットを作成する
2. ソース・プロファイルからのスナップショット構成を使用して、ターゲット・インストール内にターゲット・プロファイルを作成する
3. 構成スナップショットをターゲット・プロファイルにマイグレーションする

各プロファイルをマイグレーションするために必要な 3 ステップ・プロセスは、BPMMigrate コマンド行ユーティリティー を介して起動できる BPM プロファイル・マイグレーション・ウィザード、または以下の一連のプロファイル・マイグレーション・コマンド行ツールによってサポートされます。

- BPMSnapshotSourceProfile コマンド行ユーティリティー
- BPMCreateTargetProfile コマンド行ユーティリティー
- BPMMigrateProfile コマンド行ユーティリティー

注: WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合、『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順を使用する必要があります。

注: BPM プロファイル・マイグレーション・ウィザードは、以下のプラットフォームでサポートされます。

- Windows x86 (32 ビット)
- Windows x64 (64 ビット)
- Linux x86 (32 ビット)
- Linux x86-64 (64 ビット)
- Linux PPC (32 ビットのみ)
- AIX PPC (32 ビットのみ)
- Solaris SPARC (32 ビットのみ)
- HP-Unix IA64 (64 ビット)

他のプラットフォームでは、BPM プロファイル・マイグレーション・ウィザードの代わりにコマンド行ツールを使用して、3 ステップ・プロファイル・マイグレーション・プロセスを実行する必要があります。

プロファイル・マイグレーション用の 3 ステップ・プロセスに加えて、以下のコマンド行ユーティリティーがプロファイル・マイグレーションで重要な役割を演じます。

- BPMCreateRemoteMigrationUtilities コマンド行ユーティリティーは、スタンドアロン・プロファイルのリモート・マイグレーションをサポートするために、ソース・マイグレーション・システムへコピーできるアーカイブを作成します。
- BPMMigrateCluster コマンド行ユーティリティーは、Network Deployment 環境内のクラスター・プロファイル構成情報をマイグレーションするために、プロファイル・マイグレーション・ツールのほかに必要です。
- BPMMigrateProfile コマンド行ユーティリティーを使用すると、Network Deployment 環境内のビジネス・ルール・マネージャーのマイグレーション (オプション) が可能になります。ビジネス・ルール・マネージャーについて詳しくは、installBRManager コマンド行ユーティリティーを参照してください。

プロファイル・マイグレーション・コマンドの要約については、128 ページの『ランタイム・マイグレーション・ツールのリファレンス』のトピックを参照してください。

データベース・アップグレードおよびマイグレーション・ツール

WebSphere Process Server バージョン 7.0 は以下の製品データベースを使用し、これらのデータベースは、環境のマイグレーション時に自動または手動でアップグレードされるかマイグレーションされます。

- Business Process Choreographer データベース
- Business Space データベース
- 共通データベース
- Common Event Infrastructure データベース
- メッセージング・エンジン・データベース

Common Event Infrastructure データベースとメッセージング・エンジン・データベースは、どちらもプロファイル・マイグレーション・プロセスで必要に応じて自動的にマイグレーションされます。その他のデータベースは、マイグレーション手順内の詳細なステップに応じて、自動または手動でアップグレードされるかマイグレーションされます。製品データベースを手動で更新する場合は、サポートされている各データベース用のコマンドおよびスクリプトが、データベース・システム上で十分な特権を持つユーザーによって起動されるか、そのデータベース・システムにネットワークで接続した、データベース・クライアント・ユーティリティーを持つシステムによって起動される必要があります。ランタイム・マイグレーション手順は、ご使用のデータベース・タイプに合ったコマンドとスクリプト、およびマイグレーションのソース・リリースをデータベース・システムにコピーする方法を記述しています。

データベース・マイグレーション・コマンドの要約については、 128 ページの『ランタイム・マイグレーション・ツールのリファレンス』のトピックを参照してください。

WebSphere Application Server 管理ツール

マイグレーション手順のとき、デプロイメント・マネージャー、ノード、およびサーバーをさまざまなステップで停止および開始する必要があります。さらに、マイグレーション手順全体を通じて使用される、その他の WebSphere Application Server コマンドがいくつかあります。

マイグレーション手順で必要となる WebSphere Application Server 管理ツールの要約については、 128 ページの『ランタイム・マイグレーション・ツールのリファレンス』のトピックを参照してください。

プロファイル

ランタイム・マイグレーション・ツールを使用すると、 WebSphere Process Server、WebSphere Enterprise Service Bus、および WebSphere Application Server のソース・プロファイルをマイグレーション・ターゲット上の同じプロファイル・タイプにマイグレーションすることができます。

WebSphere Process Server プロファイル

WebSphere Process Server プロファイルは、WebSphere Application Server が「default.wbiserver」、「dmgr.wbiserve」、「managed.wbiserver」のいずれかのプロファイル・テンプレートを使用して作成したプロファイルです。プロファイル管理ツール (PMT) の使用時には、「環境の選択」ページで「 **WebSphere Process Server**」を選択することを意味します。

WebSphere Enterprise Service Bus プロファイル

WebSphere Enterprise Service Bus プロファイルは、 WebSphere Application Server が「default.esbserver」、「dmgr.esbserver」、「managed.esbserver」のいずれかのプロファイル・テンプレートを使用して作成したプロファイルです。プロファイル管理ツール (PMT) の使用時には、「環境の選択」ページで「 **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、 WebSphere Application Server が「default」、「dmgr」、「managed」のいずれかのプロファイル・テンプレートを使用して作成したプロファイルです。プロファイル管理ツール (PMT) の使用時には、「環境の選択」ページで「 **WebSphere Application Server**」を選択することを意味します。

重要: 上記の定義では、マイグレーションされるソース・プロファイルの作成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、 6.0.2 からマイグレーションされる WebSphere Enterprise Service Bus プロファイルを除き、プロファイル管理ツールまたは manageprofiles コマンド行ユーティリティーを使用してマイグレーションのターゲットとなるプロファイルを作成することはできません。

ランタイム・マイグレーション手順では、BPM プロファイル・マイグレーション・ウィザードまたはBPMCreateTargetProfile コマンド行ユーティリティーを使用して、マイグレーション・ターゲット・プロファイルを作成する必要があります。例えば、システムに default.esbserver テンプレートから作成された WebSphere Enterprise Service Bus スタンドアロン・プロファイルを含む WebSphere Enterprise Service Bus 製品インストールがあり、同じシステムに default テンプレートから作成された WebSphere Application Server スタンドアロン・プロファイル、default.esbserver テンプレートから作成された WebSphere Enterprise Service Bus スタンドアロン・プロファイル、および default.wbiserver テンプレートから作成された WebSphere Process Server スタンドアロン・プロファイルを含む WebSphere Process Server 製品インストールがある場合は、4 つのスタンドアロン・プロファイルがすべて WebSphere Process Server インストールへのマイグレーションに有効なソースとなります。

デプロイメント・マネージャーのプロファイル

WebSphere Process Server ネットワーク・デプロイメント環境では、 WebSphere Process Server デプロイメント・マネージャー・プロファイルを使用してデプロイメント・マネージャーを作成する必要があります。

製品プロファイルの拡張

ランタイム・マイグレーション・ツールでは、以下の 1 つ以上の BPM 製品によって拡張されたソース・プロファイルをマイグレーションすることができます。

- WebSphere Dynamic Process Edition
- WebSphere Business Services Fabric
- WebSphere Process Server
- WebSphere Enterprise Service Bus
- WebSphere Business Monitor
- WebSphere Business Compass

  注: WebSphere Business Modeler Publishing Server は、バージョン 7.0 で WebSphere Business Compass に変更されました。

拡張されたソース・プロファイルは、同じ製品プロファイルで拡張されたターゲット・プロファイルにマイグレーションされるため、ターゲット・インストールには、少なくともソースと同じプロファイル機能が必要です。

例えば、ソースのインストールに含まれる管理対象プロファイルが WebSphere Process Server および WebSphere Business Monitor によって拡張されている場合、ターゲットのインストール・ディレクトリーには WebSphere Process Server と WebSphere Business Monitor の両方が含まれている必要があります。このシナリオでは、BPM プロファイル・マイグレーション・ウィザードまたは BPMCreateTargetProfile コマンド行ユーティリティー は WebSphere Process Server と WebSphere Business Monitor によって拡張されたターゲット・プロファイルを作成します。

複数製品拡張環境では、セルにさまざまな拡張レベルでプロファイル内にクラスターとノードが存在する可能性があり、デプロイメント・マネージャー・プロファイ

ルは、それらのクラスターまたはノードのプロファイルの最高の拡張レベルと同じ拡張レベルで拡張されている必要があります。

混合バージョン環境

Network Deployment ベースの実稼働環境のバージョン間マイグレーションを実行すると、多くの場合、Network Deployment 環境でさまざまなバージョンの WebSphere Process Server が実行されている期間が生じます。この概念は、**混合バージョン**と呼ばれます。

理論的には、複数セル、混合バージョン・セル (複数のクラスター、または単一セル内の非クラスター管理対象ノード)、混合バージョン・クラスター (単一クラスター内の管理対象ノード) のいずれも、製品の混合バージョンと見なすことができます。これらのタイプの混合バージョンのうち、WebSphere Process Server でサポートされるのは、**複数セル**と**混合バージョン・セル**の 2 つのみです。

複数セル

初めにバージョン 6.2.0 のセルが 2 つある場合、その一方を、他方のセルに管理上またはデータベースへの影響を及ぼすことなく、バージョン 7.0 にアップグレードすることができます。これは、異なるバージョンの WebSphere Process Server 上で頻繁に実行するアプリケーションを管理する最も簡単な方法です。

混合バージョン・セル

バージョンが異なる複数のセルを持つことができるほかに、単一セル内の複数のクラスターおよび複数の非クラスター化管理対象ノードが、異なるバージョンであってもかまいません。例えば、1 つのセル内に、バージョン 6.2.0 のクラスターと、バージョン 6.2.0 からバージョン 7.0 にマイグレーションしたクラスターがあってもかまいません。混合バージョン・セル環境では、セルを有効範囲とする共通データベースは、異なるバージョンの WebSphere Process Server を実行しているすべてのクラスターおよび非クラスター化管理対象ノードによって共有されています。

注: バージョン 6.2.0 とバージョン 7.0 の両方が、ビジネス・カレンダー機能を利用するサポート・アプリケーションである場合、混合バージョン・セルのシナリオはこれらのバージョンの間でサポートされません。

WebSphere Process Server から バージョン 7.0 へのマイグレーション中に、新しいレベルのノードとマイグレーション前のレベルのノードをセルが同時に実行しており、デプロイメント・マネージャーが最新バージョンにマイグレーション済みである場合には、セル内の、マイグレーション前のレベルのノードで、以下のアクションを実行できません。

- Business Process Choreographer の構成
- ビジネス・プロセスまたはヒューマン・タスク、あるいはその両方を含む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

混合バージョン・クラスター

WebSphere Process Server は、異なるバージョンの WebSphere Process Server 上で稼働している単一クラスター内のノードをサポートしません。この概念は、混合バージョン・クラスターと呼ばれます。異なるバージョンを実行する複数のサーバー

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 7.0 のクラスター・メンバーを開始する前に、古いバージョンのWebSphere Process Server 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、バージョン 7.0 のクラスターのメンバーを始動した後に、バージョン 7.0 よりも前のレベルで構成されたクラスターのメンバーを始動してはなりません。

WebSphere Process Server から バージョン 7.0 へのマイグレーション中に、新しいレベルのノードとマイグレーション前のレベルのノードをセルが同時に実行しており、最新バージョンのクラスターに Business Process Choreographer が構成されている場合には、マイグレーション前のレベルのノードに新規クラスター・メンバーを作成してはなりません。

データベース

WebSphere Process Server は、実動時にいくつかの製品データベースを利用します。これらのデータベースは自動でマイグレーションされるか、ランタイム・マイグレーション手順の一部として手動でマイグレーションする必要があります。

データベース・スコープ

WebSphere Process Server 製品データベースには、セル・スコープ型とクラスター・スコープ型があります。

共通データベースはセル・スコープ型のため、セル内の任意のクラスターまたはクラスター化されていない管理対象ノードをバージョン 7.0 にマイグレーションする際には必ず、共通データベースをマイグレーションする必要があります。混合バージョンのセル環境では、これにより、バージョン 7.0 より前のクラスターおよびクラスター化されていない管理対象ノードが、バージョン 7.0 のクラスターおよびクラスター化されていない管理対象ノードと同じ共通データベース・インスタンスを利用するようになる場合があります。

Business Process Choreographer データベース、Business Space データベース、Common Event Infrastructure データベース、メッセージング・エンジン・データベースはすべてクラスター・スコープ型です。混合バージョンのセル環境では、各クラスターまたはクラスター化されていない各管理対象ノードがこれらのデータベースの固有インスタンスを持ち (構成されている場合)、各インスタンスはその製品バージョンに固有のスキーマとデータを使用します。各クラスターまたはクラスター化されていない各管理対象ノードをマイグレーションすると、そのクラスター・スコープ型のデータベースもランタイム・マイグレーション手順の一部としてマイグレーションされます。

バックアップ

マイグレーション手順には、製品データベースをバックアップするためのステップが含まれており、これにより、スキーマ・マイグレーションまたはデータ・マイグレーションが失敗した場合でも製品データベースをリストアできます。

自動および手動マイグレーション

Common Event Infrastructure データベースおよびメッセージング・エンジン・データベースは、プロファイルのマイグレーション時にランタイム・マイグレーション

手順によって自動的にマイグレーションされます。共通データベースは、ランタイム・マイグレーション手順の一環として自動的にマイグレーションされる場合と、手動マイグレーションが必要な場合があります。Business Process Choreographer データベースおよび Business Space データベースは、すべての環境において手動マイグレーションが必要です。要約すると、以下の環境では WebSphere Process Server に用意されたスクリプトを使用して、手動でデータベースを更新する必要があります。

- サーバー・プロセスに十分な権限がない場合 (つまり、共通データベースおよび Business Process Choreographer データベースに対して十分な権限を持つユーザー ID を使用してサーバー・プロセスが構成されていない場合)
- デフォルトでないテーブル・スペースを使用した場合
- マイグレーション・ソースが Business Space で構成されている場合

製品データベースを手動でマイグレーションすべき場合とその条件についての詳細は、ランタイム・マイグレーション手順に直接記載されています。

許可

各データベース・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。

- **Business Process Choreographer データベース・スクリプトの場合:**

  Linux®、UNIX®、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

  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。

  バージョン 6.0.2、6.1.0、または 6.1.2 からアップグレードする場合、migrateDB.py スクリプトを実行するためには以下のアクセス権が必要です。

  – 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。
  – i5/1 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。

- **共通データベース・スクリプトの場合:**

  以下の許可は必須です。

  CREATE TABLE

  ALTER TABLE

  DROP INDEX

  CREATE INDEX

  CREATE VIEW

  DROP VIEW

  CREATE SEQUENCE

- **Business Space データベース・スクリプトの場合:**

  以下の権限は、どのデータベース・タイプでも必須です。

  ALTER TABLE

  CREATE TABLE

  INSERT

  CREATE INDEX

  すべてのデータベースに該当する権限に加え、特定のデータベースに該当する特定の権限は以下のとおりです。

  Linux、Unix および Windows の DB2 の場合:

  CREATE BUFFERPOOL

  CREATE TABLESPACE

  DB2iSeries の場合:

  CREATE COLLECTION

  DB2zOSV8 および DB2zOSV9 の場合:

  CREATE TABLESPACE

  Oracle の場合:

CREATE TABLESPACE

ALTER SESSION

CREATE USER

ALTER USER

GRANT

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)。

DB2® for z/OS® および OS/390® バージョン 7

DB2® for z/OS® および OS/390® バージョン 7 を使用していて、データベースをまだ DB2 for z/OS バージョン 8 または DB2 9 for z/OS にアップグレードしていない場合は、ランタイム・マイグレーション手順の一部としてアップグレードを行うよう求められます。

Oracle 9i および Oracle JDBC ドライバー

Oracle 9i を使用していて、データベースをまだ 10g または 11g にアップグレードしていない場合は、ランタイム・マイグレーション手順の一部としてアップグレードを行うよう求められます。

Oracle ojdbc14.jar または ojdbc5.jar JDBC ドライバーを使用している場合は、ランタイム・マイグレーション手順の一部として ojdbc6.jar JDBC ドライバーをインストールおよび構成するよう求められます。

データ・マイグレーション後: データベースの再調整およびカスタム・ビューの再作成

追加の索引およびカスタム・ビューがあれば、データ・マイグレーション中に失われるため、それらを再作成する必要があります。

カスタム索引を作成することは、複雑なデータベース照会を行うヒューマン・ワークフロー・アプリケーションのパフォーマンスにとって特に重要です。

Cloudscape から Derby へのマイグレーション

バージョン 6.1.0 の WebSphere Process Server では、 Cloudscape データベースが Derby に置き換わりました。大半の環境では、プロファイル・マイグレーション・ツールによって自動的に Cloudscape データベースが Derby にマイグレーションされます。例外については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』のトピックに説明があります。

ダウン時間要件

スタンドアロン・マイグレーションと Network Deployment マイグレーションでは、どちらの場合もアプリケーションが利用不可になる期間が必要です。

ランタイム・マイグレーション

すべてのランタイム・マイグレーション・メソッド手順で、ダウン時間の期間が必要です。

ダウン時間がマイグレーションのオプションでない場合は、手動または成果物マイグレーション・メソッドを考慮してください。詳しくは、『マイグレーション・メソッド』トピックを参照してください。

スタンドアロン環境

スタンドアロン・マイグレーション手順の 3 つのバリアントでは、いずれの場合でも手順の実行中、結果としてスタンドアロン・サーバーが利用不可になります。

Network Deployment 環境

Network Deployment マイグレーションは、以下のフル・ダウン時間手順または最小限のダウン時間手順によって行うことができます。

ネットワーク・フル・ダウン時間手順はマイグレーション・ダウン時間ウィンドウを前提としており、そのウィンドウ内で Network Deployment 環境が静止し、すべてのプロファイルがマイグレーションされ、データベースがアップグレードされ、マイグレーションされたバージョンの環境が始動します。最小限のダウン時間手順では、クラスター内の半分のノードがマイグレーションされる間、残る半分がコンシューマー要求に対するサービスを行い、ダウン時間は、旧バージョンを実行しているノードがシャットダウンされ、データベースがアップグレードされ、マイグレーションされたノードが始動される期間だけに最小化されます。フル・ダウン時間手順はマイグレーション用にスケジュールされたダウン時間ウィンドウ内でマイグレーションを完了できる場合に使用し、それ以外の場合は、最小限のダウン時間手順を使用してください。

マイグレーションされるもの

BPM ランタイム・マイグレーション手順を使用して WebSphere Process Server バージョン 7.0 にマイグレーションする場合は、以下の項目がマイグレーションされます。ユーザー・アプリケーション、アダプター、プロファイル構成データ、データ・ソースとプロバイダー、および長期実行プロセス。

ユーザー・アプリケーション

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) は、サポートされているマイグレーション・シナリオではバイナリー互換です。すべてのユーザー・アプリケーションは、新しいターゲット・バージョン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。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。

- **システム・アプリケーション**: すべてのシステム・アプリケーション (install_root /systemApps ディレクトリーに存在するアプリケーション) には、新バージョンがインストールされます。

  すべてのサポート・アプリケーション (ビジネス・ルール・マネージャーや 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**: サンプル・アプリケーションは別の方法で処理されます。スタンドアロン・プロファイルの場合、マイグレーション・プロセスではサンプル・アプリケーションが何もインストールされません。 スタンドアロン・プロファイルでサンプル・アプリケーションを使用可能にする場合は、新しいバージョンの WebSphere Process Server 用インストール・ウィザードを使用してそれらのサンプル・アプリケーションを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。

ビジネス・ルール・マネージャー

すべてのバージョンのビジネス・ルール・マネージャーは、同じバージョンかそれ以降のバージョンのビジネス・ルールが入っているアプリケーションを (ほとんどの場合) 管理できますが、以前のバージョンで作成およびデプロイされたビジネス・ルールを含むアプリケーションを管理することはサポートしていません。ビジネス・ルール・マネージャーはセルを有効範囲とし、あるセル内にデプロイされたすべてのビジネス・ルールを管理します。また、セルにはバージョンを混用でき、例えばバージョン 6.2.0 のクラスターとバージョン 7.0 のクラスターが入っている場合があるので、一般に、すべてのビジネス・ルール・アプリケーションのマイグレーションが完了するまで、ビジネス・ルール・マネージャーのマイグレーションを遅らせることが賢明です。この概念をサポートするために、ビジネス・ルール・マネージャー・アプリケーションは、クラスター化されていない最後の管理対象ノードまたはセル内の最後のクラスターがマイグレーションされるまで、自動ではマイグレーションされません。

注: 最後にマイグレーションしたノードが WebSphere Process Server プロファイルではない場合、ビジネス・ルール・リソースおよびビジネス・ルール・マネージャー・マイグレーション・スクリプトは、使用できません。そのため、ビジネス・ルール・マネージャーは、マイグレーション・プロセス中に自動的にマイグレーションされません。このシナリオでは、システム全体のマイグレーション後に、WebSphere Process Server カスタム・ノードでビジネス・ルール・マネージャー・マイグレーション・スクリプトを手動で実行する必要があります。詳しくは、『installBRManager コマンド行ユーティリティー』を参照してください。

例えば、あるセルにクラスター 1、クラスター 2、クラスター 3、クラスター 4 という 4 つのクラスターが含まれており、それぞれがバージョン 6.2.0 を実行し、ビジネス・ルール・マネージャーが cluster1 にデプロイされており、クラスター 1 か

ら始めてクラスター 2、3、4 と順次にマイグレーションしたいというシナリオを考えてみます。クラスター 1 を最初にバージョン 7.0 にマイグレーションした場合でも、クラスター 1 にデプロイされたビジネス・ルール・マネージャーはバージョン 6.2.0 のままであり、クラスター 2、3、および 4 にデプロイされたビジネス・ルール・アプリケーションを引き続き管理できます。ビジネス・ルール・マネージャーは、クラスター 2 と 3 がマイグレーションされる間、引き続き 6.2.0 のバージョンで実行されますが、その後、クラスター4 がマイグレーションされるときに、自動的にバージョン 7.0 にマイグレーションされます。

セル内の最後のノードがマイグレーションされるまで待つ代わりに、もっと早い段階でビジネス・ルール・マネージャーを手動でマイグレーションする方が意味がある場合もあります。例えば、前記のシナリオを少し変更して、ビジネス・ルール・マネージャーがクラスター 1 にデプロイされており、クラスター 2 のみにビジネス・ルール・アプリケーションが含まれているとします。最初のシナリオと同様に、クラスター 1 がバージョン 7.0 にマイグレーションされるとき、ビジネス・ルール・マネージャーはバージョン 6.2.0 のままであり、クラスター 2 にデプロイされたビジネス・ルールを管理できます。クラスター 2 がバージョン 7.0 にマイグレーションされるときは、ビジネス・ルール・マネージャーをマイグレーションすることに意味があります。クラスター 3 と 4 にはビジネス・ルールが含まれておらず、この時点でセル内のルールはバージョン 7.0 だけになっているからです。このシナリオをサポートするために、ビジネス・ルールのマイグレーション・プロセスでは、マイグレーション・プロセスのさまざまな段階で手動で呼び出すことができる installBRManager コマンド行ユーティリティーが用意されています。詳しくは、『installBRManager コマンド行ユーティリティー』を参照してください。

注: スタンドアロンのマイグレーション・シナリオでは、ビジネス・ルール・マネージャーは常に、スタンドアロン・プロファイルがマイグレーションされるときに自動的にマイグレーションされます。

アダプター

WebSphere Adapter バージョン 6.1.0、6.1.2、および 6.2.0 の場合は、バージョン 7.0.0.1 以降の対応するアダプターをターゲット環境で適用する必要があります。

マイグレーション後、スタンドアロンであるか、またはアプリケーション組み込みであるかに関わらず、すべての WebSphere Adapter は、対応する新しいアダプターにアップグレードされ、ターゲット環境の *WPS_HOME*/installableApps/ フォルダーに配置されます。また、WebSphere Adapter を参照するアプリケーションは、新しいアダプターを参照するように更新されます。

プロファイル構成データ

バージョン間マイグレーション・ツール (ウィザードまたはコマンド) 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。

JDBC プロバイダーおよびデータ・ソース

プロファイル・マイグレーションにより、既存のデータ・ソースおよびプロバイダーごとに JDBC プロバイダー定義とデータ・ソース定義が自動的にマイグレーションされます。

長期実行プロセス

長期実行ビジネス・プロセス・インスタンス、およびヒューマン・タスク・インスタンスは、バージョン間マイグレーション中に、それらのインスタンスを保管しているデータベースが引き継がれるときに処理されます。マイグレーション中に、データベース・スキーマがアップグレードされ、データは新しいスキーマに変換されます。マイグレーション後、それらのインスタンスは、マイグレーションされた環境で実行を継続します。

注: 前にインストールした事前定義ヒューマン・タスク・アプリケーションは、実行中のインスタンスを依然として所有している場合があるので、これらのアプリケーションはマイグレーション時にアンインストールされません。つまり、マイグレーション後は、事前定義ヒューマン・タスク・アプリケーションの新しいバージョンと以前のバージョンの両方がシステムにインストールされています。バージョン番号は、アプリケーションがいつ最終更新されたのかを示しています。以前のバージョンのアプリケーションを安全にアンインストールできるタイミングについては、118 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。

マイグレーションされないもの

特定の成果物は、自動的にはマイグレーションされません。それらの成果物の大部分は、ユーザーが作成したものであり、WebSphere Process Server によって認識されないものです。それらは、認識されないので、マイグレーションされません。

- **参照による共用 (共用ライブラリー) 成果物**

  SCA ライブラリーの共用のために参照による共用パターンを使用している場合は、`lib/ext` および `config` ディレクトリーに存在する成果物 (Java の .jar ライブラリーなど) は、マイグレーション・ターゲットへマイグレーションされません。参照による共用ライブラリーの WebSphere 構成設定は、プロファイル・マイグレーションのときに転送されますが、実際のライブラリー .jar 成果物は、マイグレーション後に手動でコピーしてください。

- **WebSphere Process Server インストール・ディレクトリーまたはプロファイル・ディレクトリー構造に追加される大部分のカスタム・プロファイルまたは成果物**

  大部分の非製品ファイル (カスタム Jython スクリプトなど) は、マイグレーションの一部として転送されません。

  注: ただし、Business Process Choreographer のカスタム XSL 変換ファイルに限っては、自動的にマイグレーションされます。これらのファイルは、*install_root*`/ProcessChoreographer/Staff` ディレクトリーにあります。これらのファイルについて詳しくは、118 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。

  同様に、WebSphere 固有のスクリプトを変更してある場合は、それらの変更をマイグレーション後にマイグレーション・ターゲットに手動で再適用する必要があります。

重要: ユーザーが変更したスクリプトを誤って削除することがないよう、カスタム・スクリプトや変更した製品スクリプトは、すべてインストール・ディレクトリーの外部に保持してください。

互換性に関する既知の問題

以下の項目は、WebSphere Process Server バージョン 7.0 にマイグレーションする場合の互換性に関する既知の問題です。

データの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

注: Business Process Choreographer が構成済みの場合のみ、これが適用されます。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る可能性があります。例えば、WORK_ITEM ビュー内の異なる WIID のカウント数について、実際よりも多い数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。

SCA ワイヤリング

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、モジュールを更新して bindingType を正しく設定し、呼び出しを行うときの Web サービスの URL を示すようにするか (MFC または POJO コンポーネントの場合)、モジュールの更新を行わない場合は、WebSphere 変数の SCA_USE_WS_FOR_DYNAMIC_INVOCATION を設定して、モジュール名をセミコロンで区切ってリストとして指定する必要があります (例えば sca/myModule1;sca/myModule2)。

ランタイム・マイグレーション前のチェックリスト

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、このチェックリストに示す各項目を確認してください。

- 30 ページの『ハードウェア、オペレーティング・システム、およびデータベースの前提条件』
- 30 ページの『WebSphere Process Server インストール・イメージ』
- 30 ページの『DB2 for z/OS バージョン 8 または 9 のインストール・イメージ』
- 30 ページの『Oracle データベースおよび JDBC ドライバーのアップグレード』
- 31 ページの『WebSphere Application Server にバンドルされた Data Direct ドライバー』
- 31 ページの『WebSphere Adapter バージョン 7.0.0.1 以降の適用』

- 31 ページの『ソース・プロファイルのバックアップ・ディレクトリー・ストレージ』
- 31 ページの『ソース・データベースのバックアップ・ストレージ』
- 32 ページの『ソース・プロファイルのスナップショット・ディレクトリー・ストレージ』
- 32 ページの『ターゲット・プロファイルのディレクトリー・ストレージ』
- 33 ページの『Business Process Choreograher データ・マイグレーション: 実体化ビュー』
- 33 ページの『ulimit 設定』
- 34 ページの『データベース許可』
- 34 ページの『適切な手順および手順差異の判別』
- 34 ページの『root 構成から非 root へのマイグレーション』
- 34 ページの『非 root 構成から root へのマイグレーション』

ハードウェア、オペレーティング・システム、およびデータベースの前提条件

ターゲット・マイグレーション環境が WebSphere Process Server バージョン 7.0 でサポートされている操作環境であることを確認してください。これには、ハードウェア・プラットフォーム、オペレーティング・システム、およびデータベースが含まれます。WebSphere Process Server バージョン 7.0 でサポートされている操作環境については、 WebSphere Process Server をインストールするための前提条件を参照してください。

WebSphere Process Server インストール・イメージ

WebSphere Process Server インストール・イメージと最新のフィックスパックをダウンロードし、マイグレーション対象の各システムにインストールできるように準備しておきます。システムに WebSphere Process Server とフィックスパックをインストールするのに十分なストレージがあることを確認してください。

DB2 for z/OS バージョン 8 または 9 のインストール・イメージ

データベース・サーバー上で DB2 バージョン 7 を使用している場合は、DB2 for z/OS バージョン 8 またはバージョン 9 のインストール・イメージをダウンロードし、手順内の 1 つのステップとしてそれらをインストールできるように準備しておきます。

Oracle データベースおよび JDBC ドライバーのアップグレード

Oracle 9i を使用しており、データベースを 10g または 11g にまだアップグレードしていない場合は、Oracle 10g または 11g のインストール・イメージをダウンロードし、手順内の 1 つのステップとして新しいデータベース・バージョンにアップグレードできるように準備しておきます。

Oracle ojdbc14.jar または ojdbc5.jar JDBC ドライバーを使用している場合は、新しい ojdbc6.jar JDBC ドライバーをダウンロードし、手順内の 1 つのステップとしてそのインストールと構成を行うことができるように準備しておきます。

WebSphere Application Server にバンドルされた Data Direct ドライバー

WebSphere Application Server にバンドルされた組み込み Data Direct ドライバーは、WebSphere Process Server バージョン 7.0 ではサポートされません。既存の組み込み Data Direct ドライバーのライセンスを購入するか、または MSSQL Server 用の Microsoft JDBC ドライバー (Microsoft の Web サイトからダウンロード可能) をダウンロードする必要があります。

ソース・バージョンが 6.1.2 または 6.2.0 であるか、Data Direct ドライバーの購入を計画している場合は、組み込み Data Direct ドライバーを使用する既存のデータ・ソースを、ソース環境で新規 JDBC ドライバーを使用するように更新してください。そのためには、以下のステップを実行します。

1. 正しい JDBC プロバイダー・タイプの新規データ・ソースを作成し、プロパティーを設定します。設定するプロパティーは、 JNDI 名、statementCacheSize、releationalResourceAdapter、 authMechanismPreference、 authDataAlias、databaseName、 serverName、 portNumber、および既存のデータ・ソースに一致する URL です。
2. 組み込みドライバーを使用する既存のデータ・ソースを削除します。
3. データ・ソースの接続をテストします。
4. ソース環境ですべてのアプリケーションが継続して機能することをテストします。

ソース・バージョン 6.0.2 または 6.1.0 を使用しており、Microsoft JDBC ドライバーの使用を予定している場合、ダウンロードしたドライバー JAR ファイルを、以前のドライバー JAR ファイルがあるロケーションにコピーします。

マイグレーション手順の実行中に、データ・ソース構成の更新を実行します。

WebSphere Adapter バージョン 7.0.0.1 以降の適用

ソース環境内のいずれかのアプリケーションが、バージョン 6.1.0 またはバージョン 6.2.0 の WebSphere Adapter を埋め込む場合、あるいはノード・レベルまたはクラスター・レベルで構成された WebSphere Adapter バージョン 6.1.0 または 6.2.0 を使用する場合は、マイグレーション手順を開始する前に、バージョン 7.0.0.1 以降の対応するアダプターをターゲット環境に適用する必要があります。これを行うには、バージョン 7.0.0.1 以降の対応するアダプターを、ターゲット環境内の *WPS_HOME*/installableApps/ フォルダーにコピーします。

ソース・プロファイルのバックアップ・ディレクトリー・ストレージ

マイグレーション時に、後の時点でロールバックが必要になる場合に備えて、マイグレーションされるプロファイルがバックアップされます。プロファイルのバックアップ・ディレクトリー用に使用可能なスペースは、少なくともソース・プロファイルの構成ディレクトリーとアプリケーションのサイズであることが必要です。

ソース・データベースのバックアップ・ストレージ

マイグレーション手順では、ソースの製品データベースをマイグレーションの前にバックアップしておくことを強くお勧めします。それらのデータベースをバックア

ップするための十分なスペースが存在することを確認してください。バックアップに必要なサイズは、実動データベースのサイズとデータベース・バックアップ戦略の特性によって異なります。

ソース・プロファイルのスナップショット・ディレクトリー・ストレージ

マイグレーションされるプロファイル内の構成ファイルは、マイグレーション手順のときにスナップショット・ディレクトリーにコピーされ、その後、そのディレクトリーはプロファイル・マイグレーションのソースになります。このディレクトリーは、BPMSnapshotSourceProfile コマンドのオプションのパラメーターであるか、BPM プロファイル・マイグレーション・ウィザードで構成可能な値であり、デフォルトでは MigrationSnapshots です。

マイグレーションの前に、スナップショット・ディレクトリー用に十分なストレージが存在することを確認してください。ストレージ要件は、以下の量を合計することによって見積もることができます。

- マイグレーションするプロファイル構成情報のサイズ:
  - *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
- マイグレーションする共用ライブラリーのサイズ:
  - libraries.xml 構成ファイルで参照される共用ライブラリー
- マイグレーションするリソース・アダプター・アーカイブのサイズ:
  - 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
- トレースを使用可能にする場合は、スナップショット・ディレクトリーに書き込まれるトレース・ファイル用に 200 MB (構成のサイズと複雑さによって異なる) を追加で割り振ります。

ターゲット・プロファイルのディレクトリー・ストレージ

マイグレーション時に、ターゲット・プロファイルが BPMCreateTargetProfile コマンドまたは BPM プロファイル・マイグレーション・ウィザードを使用して作成され、ソース・プロファイルは、ターゲット・インストールから参照されるターゲット・プロファイルにマイグレーションされます。

マイグレーションの前に、ターゲット・プロファイル・ディレクトリー用に十分なストレージが存在することを確認してください。ストレージ要件は、以下の量を合計することによって見積もることができます。

- マイグレーションするプロファイル構成情報のサイズ:
  - *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

- マイグレーションする共用ライブラリーのサイズ:
  - libraries.xml 構成ファイルで参照される共用ライブラリー
- マイグレーションするリソース・アダプター・アーカイブのサイズ:
  - 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
- トレースを使用可能にする場合は、スナップショット・ディレクトリーに書き込まれるトレース・ファイル用に 200 MB (構成のサイズと複雑さによって異なる) を追加で割り振ります。

Business Process Choreograher データ・マイグレーション: 実体化ビュー

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 で名前付き実体化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 が稼働中であることを確認してください。
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・プロパティー**」をクリックします。
3. ビジネス・コンテナーのカスタム・プロパティーのリストで、**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。

ulimit 設定

UNIX システムでは、プロファイルのマイグレーション時にオープン・ファイルの数が多すぎてエラーが起きるのを避けるために、プロファイル・マイグレーション・プロセスを実行するシステム上の ulimit 設定を大きくします。

データベース許可

単一ユーザー ID を使用してすべてのデータベース・スクリプトを実行できるか、それともデータベース管理者がそれらのすべて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。

製品データベースに必要な許可の詳細については、『データベース』のトピックを参照してください。

適切な手順および手順差異の判別

スタンドアロン・プロファイルをマイグレーションする場合は、並列マイグレーションを行うのか、リモート・システムへのマイグレーションを行うのか、それともソース・システム上のオペレーティング・システムをマイグレーション・プロセス時にアップグレードする必要があるマイグレーションを行うのかを決定します。Network Deployment 環境をマイグレーションする場合は、フル・ダウン時間手順と最小限のダウン時間手順を入念に分析して、どの手順がお客様の要件に最も適合するかを判別してください。

root 構成から非 root へのマイグレーション

root ユーザー権限を持つ以前のバージョンの環境を、非 root ユーザー権限を持つバージョン 7.0 にマイグレーションする場合は、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『root 構成から非 root へのマイグレーション (Migrating root configurations to non-root)』トピックのステップを実行してから、マイグレーション手順を試行してください。

注: 『root 構成から非 root へのマイグレーション (Migrating root configurations to non-root)』の説明に記載されている *USER_HOME* の参照は、*USER_INSTALL_ROOT* またはソース・プロファイルのルート・ディレクトリーを指します。

非 root 構成から root へのマイグレーション

非 root ユーザー権限を持つ以前のバージョンの環境を、root ユーザー権限を持つバージョン 7.0 にマイグレーションする場合は、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『root 以外の構成から root へのマイグレーション』トピックのステップを実行してから、マイグレーション手順を試行してください。

ランタイム・マイグレーション手順

バージョン間マイグレーションを行うには、ランタイム・マイグレーション手順を使用します。

ランタイム・マイグレーション手順について

ランタイム・マイグレーション・ツールおよび資料では、スタンドアロン環境のマイグレーション、フル・ダウン時間でのネットワーク・デプロイメント環境のマイグレーション、最小限のダウン時間でのネットワーク・デプロイメント環境のマイグレーションの 3 つのマイグレーション手順をサポートしています。

3 つのランタイム・マイグレーション手順のそれぞれには、一連のステップとサブプロシージャーが含まれています。各手順の仕組みを理解することに加え、選択したマイグレーション手順のテスト方法を検討することも同様に重要です。以下の各セクションでは、各手順の概要と、マイグレーションのテストについて検討するための情報を記載します。

- 『スタンドアロン環境のマイグレーション』
- 『フル・ダウン時間でのネットワーク・デプロイメント環境のマイグレーション』
- 『最小限のダウン時間でのネットワーク・デプロイメント環境のマイグレーション』
- 36 ページの『マイグレーションのテスト』

スタンドアロン環境のマイグレーション

スタンドアロン環境のマイグレーション手順では、環境のバックアップ、スタンドアロン・プロファイルのマイグレーション、およびこのプロファイル用に構成された製品データベースのアップグレードの各ステップについて説明します。この手順には、並列マイグレーション、リモート・マイグレーション、オペレーティング・システムのアップグレード・マイグレーションなど、スタンドアロン環境のマイグレーションでサポートされるさまざまなメカニズムのタイプが含まれています。スタンドアロン環境をマイグレーションする前に、これらのタイプのうちどれが要件に最も適しているかを判断してください。

この手順の使用については、 36 ページの『スタンドアロン環境のマイグレーション』を参照してください。

フル・ダウン時間でのネットワーク・デプロイメント環境のマイグレーション

ネットワーク・デプロイメント環境をマイグレーションする手順は 2 種類あります。これらの違いは、マイグレーションが行われるダウン時間枠の長さにあります。フル・ダウン時間の手順は最も単純な手順です。マイグレーションのダウン時間枠を許容できる場合は、この手順をお勧めします。マイグレーションの所要時間は、ソースのバージョン、セル数、クラスター数、ノード数、アプリケーション数、およびデータベース内のデータ量などのさまざまな要因によって決まります。マイグレーションの所要時間を判断するには、ステージング環境で完全なマイグレーション・プロセスを使用します。ネットワーク・デプロイメント環境を正しくマイグレーションするには、ネットワーク・デプロイメント手順の各ステップを記載される順序どおりに慎重に実行することが重要です。

この手順の使用については、 43 ページの『フル・ダウン時間での Network Deployment 環境のマイグレーション』を参照してください。

最小限のダウン時間でのネットワーク・デプロイメント環境のマイグレーション

フル・ダウン時間の手順を使用したマイグレーションを実行するだけのマイグレーション期間は許容できないが、最小限のダウン時間の手順に必要なダウン時間は許容できる場合や、マイグレーションに必要なダウン時間の長さがビジネスに直接的

な影響を与える場合には、最小限のダウン時間の手順を使用してください。最小限のダウン時間の手順はフル・ダウン時間の手順よりも複雑なため、ダウン時間の長さが重要となる場合にのみ使用するようにしてください。最小限のダウン時間を許容できない場合は、ランタイム・マイグレーション方式ではなく手動または成果物マイグレーション方式を検討する必要があります。最小限のダウン時間の手順では、マイグレーションを 2 つのグループに分割し、一方のグループが実行を続けている間にもう一方のグループをマイグレーションすることで、クラスターのダウン時間を最小限に抑えます。最小限のダウン時間は、データベース・スキーマと製品データを更新するために、マイグレーション済みのノード・グループをオンラインにする直前に発生します。

注: ビジネス・カレンダーまたはメディエーション・フロー・コンポーネントを利用するアプリケーションがソース・バージョンに含まれている場合は、これらのアプリケーションでダウン時間が許容されていない限り、最小限のダウン時間の手順を使用することはできません。ビジネス・カレンダーまたはメディエーション・フロー・コンポーネントを利用するアプリケーションを実行しているサーバーがノードに含まれている場合、そのノードはバージョン 7.0 にマイグレーションされるまで停止したままになります。

この手順の使用については、58 ページの『最小限のダウン時間での Network Deployment 環境のマイグレーション』を参照してください。

マイグレーションのテスト

実動マイグレーションを実動環境で試す前に、ステージング環境で徹底的にテストすることが重要です。さらに、構成データまたはアプリケーションがターゲット環境に正しくマイグレーションできなかった場合にロールバックできるように、手順内のバックアップ・ステップを慎重に実行することが重要です。また、標準アプリケーションまたはすべてのアプリケーションを問題なくバージョン 7.0 の環境にデプロイできること、あるいは開発ツールを使用してアプリケーションを正しくマイグレーションできることを確認するために、ランタイム・マイグレーションと共に手動または成果物マイグレーション方式を使用することもよくあります。これにより、アプリケーションの後方互換性が確実に維持されるようになります。ネットワーク・デプロイメント環境をマイグレーションする場合は、より複雑なネットワーク・デプロイメントのフル・ダウン時間の手順または最小限のダウン時間の手順を使用する前に、最初にステージング環境でスタンドアロン環境のマイグレーションを行って、ツールの使用法やランタイム・マイグレーション・プロセスの本質を学ぶことも有用です。

スタンドアロン環境のマイグレーション

この手順を使用して、スタンドアロン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『 BPM ランタイム・マイグレーション前のチェックリスト』トピックを確認してください。

このタスクについて

スタンドアロン環境をマイグレーションするには、以下の手順を使用します。

手順

1. **マイグレーション・ターゲット製品をインストールします。**
   - 並列マイグレーションの場合、ターゲット製品および最新のフィックスパックを、マイグレーションのソース製品と同じシステムにインストールします。
   - リモート・マイグレーションの場合、ターゲット製品および最新のフィックスパックを、マイグレーションのターゲットとしてのサービスを提供するシステムにインストールします。
   - オペレーティング・システムのアップグレードのマイグレーションの場合、オペレーティング・システムの更新後まで、インストールを据え置いてください。

   注: ソース・バージョンのインストールに使用したのと同じユーザー ID でターゲット・バージョンをインストールするか、ソース・インストール済み環境の構成およびデータにアクセスする権限を持っている必要があります。

   注: 複数の製品によって拡張されたソース・プロファイルからマイグレーションするには、それらの製品の新バージョンを同じターゲット・インストール・ディレクトリーにインストールする必要があります。例えば、ソース・プロファイルが WebSphere Process Server および WebSphere Business Monitor によって拡張されている場合、これら両方の製品が同じターゲット・インストール・ディレクトリーにインストールされる必要があります。
2. **DB2® for z/OS® および OS/390® Version 7 をアップグレードします。**

   DB2® for z/OS® および OS/390® Version 7 を使用しており、データベースを DB2 for z/OS Version 8 または DB2 9 for z/OS にアップグレードしていない場合、DB2 for z/OS のドキュメンテーションの説明を参照して今すぐにアップグレードを実行してください。
3. **Oracle 9i および Oracle JDBC ドライバーをアップグレードします。**

   重要: Oracle データベースにアクセスするすべての WebSphere Process Server インストール済み環境で、このステップを実行する必要があります。
   a. Oracle 9i を使用しており、データベースを 10g または 11g にアップグレードしていない場合、Oracle の資料の説明を参照して、今すぐにアップグレードを実行してください。
   b. `ojdbc14.jar` または `ojdbc5.jar` ドライバーを使用している場合、*ORACLE_JDBC_DRIVER_PATH* WebSphere 変数によって示されているディレクトリーに新しい `ojdbc6.jar` ドライバーをインストールする必要があります。これを行うには、以下の手順を使用します。
      1) 以前の環境の *ORACLE_JDBC_DRIVER_PATH* 変数の値を確認します。これを行うには、以下の方法のいずれかを使用します。
         - 管理コンソールで、「**環境**」 → 「**WebSphere 変数**」を選択し、ソース・プロファイルのノードと一致するスコープを選択します。
         - 以下のディレクトリーにある `variables.xml` ファイルにナビゲートします。
           source_profile_root¥config¥cells¥*cell_name*¥nodes¥*node_name*¥.

注: セル名とノード名がソース・プロファイルの情報と一致している必要があります。

2) 新しい ojdbc6.jar ドライバーを、*ORACLE_JDBC_DRIVER_PATH* WebSphere 変数で示されているディレクトリーにインストールします。変数で示されているロケーションに応じて、以下のいずれかのステップを使用します。

- この変数が示しているディレクトリーが、WebSphere Process Server インストール済み環境の外部にある場合は、odbc6.jar ファイルを ojdbc14.jar または ojdbc5.jar ファイルと同じフォルダーにコピーします。
- この変数が示しているディレクトリーが、WebSphere Process Server インストール済み環境内にある場合は、WebSphere Process Server バージョン 7.0 インストール済み環境内に同じディレクトリー構造を作成して、そのディレクトリーに odbc6.jar ファイルをコピーします。

4. **マイグレーション・ソース・サーバーを停止します。**

マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopServer コマンドを使用して、マイグレーション・ソース・サーバーを停止します。以下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:** stopServer.sh *server_name* -username *user_name* -password *password*
- Windows **Windows® プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

注:

- プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。
- 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、これらのパラメーターは、Windows サービスがサーバー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。
- プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

5. **マイグレーション・ソース・プロファイルをバックアップします。**

backupConfig コマンドを使用して、マイグレーション・ソース・サーバー上のプロファイル構成をバックアップします。

以下の構文を使用して、profile1 という名前のプロファイルを /ProfileBackups/profile1.zip にバックアップします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:**
  `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
- Windows **Windows プラットフォームの場合:** `backupConfig.bat c:¥ProfileBackups¥profile1.zip -profileName profile1`

`backupConfig` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

6. **.nifRegistry ファイルをバックアップします。**

   .nifRegistry ファイルは、インストール済みのすべての WebSphere Process Server 製品のインストール・ルートを示します。また、インストール済みのすべての WebSphere Application Server 製品のインストール・ルートも示します。場所は以下のとおりです。

   - Linux UNIX **Linux または UNIX プラットフォームの場合**:
     `/opt/.ibm/.nif/.nifregistry`
   - Windows **Windows プラットフォームの場合**:
     - 製品をインストールしたユーザー ID が管理特権を持っていた場合のファイルの場所は Windows ルート・ディレクトリー (ほとんどの Windows システムでは `C:¥Windows` または `C:¥WINNT`) です。
     - 製品をインストールしたユーザー ID が管理特権を持っていなかった場合のファイルの場所は、そのユーザー ID のホーム・ディレクトリーです。

7. **マイグレーション・ソースの製品データベースをバックアップします。**

   データベースの資料に従って、スタンドアロン・プロファイルによって構成された以下のデータベースをバックアップします。

   - Business Process Choreographer データベース
   - Business Space データベース
   - 共通データベース
   - Common Event Infrastructure データベース
   - メッセージング・エンジン・データベース

8. **スタンドアロン・サーバー・プロファイルをマイグレーションします。**
   - 並列マイグレーションの場合、BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、ソース・プロファイルをマイグレーションできます。しかし、WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合、『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順を使用する必要があります。
     - BPM プロファイル・マイグレーション・ウィザードを使用するには、ソース・プロファイルを含むシステムで、 78 ページの『BPM プロファイル・マイグレーション・ウィザードを使用したプロファイルのマイグレーション』の手順に従います。

– BPM マイグレーション・コマンド行ユーティリティーを使用するには、ソース・プロファイルを含むシステムで『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順に従います。

- リモート・マイグレーションを行う場合は、スタンドアロン・プロファイルのリモート・システムへのマイグレーション (Migrating a stand-alone profile to a remote system)の手順を実行します。ソース・オペレーティング・システムが i5/OS の場合、スタンドアロン・プロファイルをサポートされるオペレーティング・システムにマイグレーションするには、リモート・マイグレーションの手順が必要です。
- オペレーティング・システムのアップグレードのマイグレーションの場合、『オペレーティング・システムのアップグレード時にスタンドアロン・サーバーをマイグレーション』の手順を参照してください。

9. **Cloudscape または Derby データベースを更新します。**

   Cloudscape または Derby データベースを使用している場合、サポートされるバージョンを使用していることを確認する必要があります。 Cloudscape を Derby にマイグレーションする方法については、 IBM Cloudscape または Apache Derby データベースのマイグレーションを参照してください。

10. **データ・ソース構成を更新します。** 組み込み DataDirect ドライバーを使用するデータ・ソースがあり、ライセンス交付を受けた DataDirect JDBC ドライバーまたは Microsoft JDBC ドライバーを使用するようにそれらのデータ・ソースをソース環境で更新していない場合は、データ・ソース構成を更新します。これを行うには、以下の手順を使用します。

    重要: 一部のコンポーネントがデータベースへの接続を確立できなかったため、 `SystemOut.log` ファイルにエラーが記録されている可能性があります。

    a. **マイグレーション・サーバーを始動します。**

       マイグレーション・ターゲット・サーバーの *profile_root*`/bin` ディレクトリーから、またはターゲット・プロファイルのファースト・ステップ・コンソールから、`startServer` コマンドを使用して、マイグレーション・ターゲット・サーバーを始動します。以下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:** `startServer.sh` *server_name*
       - Windows **Windows プラットフォームの場合:** `startServer.bat` *server_name*

       `startServer` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

    b. 管理コンソールにログインします。

    c. 以下のステップを実行して、データ・ソース構成を更新します。

       1) 正しい JDBC プロバイダー・タイプの新規データ・ソースを作成し、プロパティーを設定します。設定するプロパティーは、 JNDI 名、 statementCacheSize、 releationalResourceAdapter、 authMechanismPreference、 authDataAlias、 databaseName、 serverName、 portNumber、および既存のデータ・ソースに一致する URL です。

2) 組み込みドライバーを使用する既存のデータ・ソースを削除します。
3) 「テスト接続」オプションを使用して、データ・ソース構成が機能するかどうかを確認します。
4) マイグレーション・ソース・サーバーを停止します。

マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopServer コマンドを使用して、マイグレーション・ソース・サーバーを停止します。以下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:** stopServer.sh *server_name* -username *user_name* -password *password*
- Windows **Windows プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

注:

- プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。
- 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、これらのパラメーターは、Windows サービスがサーバー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。
- プロファイルでセキュリティーが有効になっていない場合、-username および -password パラメーターは必要ありません。

stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

11. **データベース・マイグレーションおよびアップグレード・スクリプトをデータベース・システムにコピーします。**

ターゲット・マイグレーション・システムで、ご使用のデータベース・タイプにあわせてカスタマイズされた、データベース・マイグレーションおよびアップグレードのコマンドとスクリプトを見つけて、データベース・システムにコピーします。 コマンドおよびスクリプトは、以下のディレクトリーにあります。

- **共通データベース**: *install_root*/dbscripts/CommonDB/*database_type*
- **Business Space データベース**:
  - **スタンドアロン・サーバー**: *profile_root*/dbscripts/BusinessSpace/*node_name_server_name*/*database_product_name*/*database_name*
  - **クラスター**: *profile_root*/dbscripts/BusinessSpace/*cluster_name*/*database_product_name*/*database_name*

注: Business Process Choreographer データベース・コマンドとスクリプトは、後のプロセスで DBDesignGenerator コマンドを使用してコピーされます。詳し

くは、 95 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』を参照してください。
次の表を使用して、ご使用の特定のデータベース・タイプに対応したディレクトリー名を判別します。

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 バージョン 6.1.0 の WebSphere Process Server では、Cloudscape データベースが Derby に置き換わりました。大半の環境では、プロファイル・マイグレーション・ツールによって自動的に Cloudscape データベースが Derby にマイグレーションされます。例外については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』のトピックに説明があります。
Informix®	Informix
Oracle	Oracle
Microsoft® SQL Server	SQLServer

12. **製品データベースをマイグレーションします。**

    以下の手順を使用して、スタンドアロン・サーバー上に構成された各製品データベースをマイグレーションします。

    a. 共通データベースのデータ・ソースに対して定義されたデータベース・ユーザーに十分な権限がない場合は、『共通データベース・スキーマのアップグレード』の手順を使用して共通データベース・スキーマを手動でアップグレードします。

    b. 『Business Process Choreographer データベース・スキーマのアップグレード』の手順に従って、Business Process Choreographer データベース・スキーマを手動でアップグレードします。

c. マイグレーション元のソース・バージョンが 6.0.2、6.1.0、または 6.1.2 の場合は、『Business Process Choreographer データベース・データのマイグレーション』の手順に従って、Business Process Choreographer データベース・データをマイグレーションします。

d. 『Business Space データベース・スキーマのマイグレーション』の手順を使用して Business Space データベース・スキーマをマイグレーションします。

e. 『Business Space データベース・データのマイグレーション』の手順を使用して Business Space データベース・データをマイグレーションします。

f. オプション: ご使用の環境に合わせて、必要に応じてメッセージング・エンジン・データベースをマイグレーションします。 メッセージング・エンジンのマイグレーションが必要な場合とその方法について詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターで『データ・ストアに基づくメッセージング・エンジンのマイグレーション』を参照してください。

13. **マイグレーション・サーバーを始動します。**

マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはターゲット・プロファイルのファースト・ステップ・コンソールから、startServer コマンドを使用して、マイグレーション・ターゲット・サーバーを始動します。以下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:** startServer.sh *server_name*
- Windows **Windows プラットフォームの場合:** startServer.bat *server_name*

startServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

タスクの結果

スタンドアロン環境はターゲット・バージョンにマイグレーションされます。

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。手順については、 109 ページの『マイグレーションの検査』を参照してください。

フル・ダウン時間での Network Deployment 環境のマイグレーション

この手順を使用して、フル・ダウン時間を発生させて 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『BPM ランタイム・マイグレーション前のチェックリスト』トピックを確認してください。

このタスクについて

このステップに従って、フル・ダウン時間を発生させて Network Deployment 環境をマイグレーションします。

手順

1. **マイグレーション・ターゲット製品をインストールします。**

   ターゲット製品および最新のフィックスパックを、マイグレーションのソース製品と同じシステムにインストールします。

   注: ソース・バージョンのインストールに使用したのと同じユーザー ID でターゲット・バージョンをインストールするか、ソース・インストール済み環境の構成およびデータにアクセスする権限を持っている必要があります。

   注: 複数の製品によって拡張されたソース・プロファイルからマイグレーションするには、それらの製品の新バージョンを同じターゲット・インストール・ディレクトリーにインストールする必要があります。例えば、ソース・プロファイルが WebSphere Process Server および WebSphere Business Monitor によって拡張されている場合、これら両方の製品が同じターゲット・インストール・ディレクトリーにインストールされる必要があります。

2. **DB2® for z/OS® および OS/390® Version 7 をアップグレードします。**

   DB2® for z/OS® および OS/390® Version 7 を使用しており、データベースを DB2 for z/OS Version 8 または DB2 9 for z/OS にアップグレードしていない場合、DB2 for z/OS のドキュメンテーションの説明を参照して今すぐにアップグレードを実行してください。

3. **Oracle 9i および Oracle JDBC ドライバーをアップグレードします。**

   重要: Oracle データベースにアクセスするすべての WebSphere Process Server インストール済み環境で、このステップを実行する必要があります。

   a. Oracle 9i を使用しており、データベースを 10g または 11g にアップグレードしていない場合、Oracle の資料の説明を参照して、今すぐにアップグレードを実行してください。

   b. `ojdbc14.jar` または `ojdbc5.jar` ドライバーを使用している場合、*`ORACLE_JDBC_DRIVER_PATH`* WebSphere 変数によって示されているディレクトリーに新しい `ojdbc6.jar` ドライバーをインストールする必要があります。これを行うには、以下の手順を使用します。

      1) 以前の環境の *`ORACLE_JDBC_DRIVER_PATH`* 変数の値を確認します。これを行うには、以下の方法のいずれかを使用します。

         - 管理コンソールで、「**環境**」 → 「**WebSphere 変数**」を選択し、ソース・プロファイルのノードと一致するスコープを選択します。
         - 以下のディレクトリーにある `variables.xml` ファイルにナビゲートします。
           `source_profile_root``¥config¥cells¥`*`cell_name`*`¥nodes¥`*`node_name`*`¥`.

           注: セル名とノード名がソース・プロファイルの情報と一致している必要があります。

2) 新しい ojdbc6.jar ドライバーを、*ORACLE_JDBC_DRIVER_PATH* WebSphere 変数で示されているディレクトリーにインストールします。変数で示されているロケーションに応じて、以下のいずれかのステップを使用します。
   - この変数が示しているディレクトリーが、WebSphere Process Server インストール済み環境の外部にある場合は、odbc6.jar ファイルを ojdbc14.jar または ojdbc5.jar ファイルと同じフォルダーにコピーします。
   - この変数が示しているディレクトリーが、WebSphere Process Server インストール済み環境内にある場合は、WebSphere Process Server バージョン 7.0 インストール済み環境内に同じディレクトリー構造を作成して、そのディレクトリーに odbc6.jar ファイルをコピーします。

4. **マイグレーション対象のクラスター、クラスター管理対象ノード、および非クラスター管理対象ノードを識別します。**

   セル全体をマイグレーションする場合、以下をマイグレーションします。
   - デプロイメント・マネージャー。
   - セル内のクラスターのメンバーであるアプリケーション・サーバーを保有しないすべてのノード (非クラスター管理対象ノード)。
   - これらのクラスターのメンバーであるアプリケーション・サーバーを保有するすべてのクラスターおよびすべてのノード (クラスター管理対象ノード)。

   セル全体をマイグレーション**しない**場合、クラスターをマイグレーション**しない**場合、およびセル内のクラスターのメンバーであるアプリケーション・サーバーを保有しない 1 つ以上のノード (非クラスター管理対象ノード) をマイグレーション**する**場合、以下をマイグレーションします。
   - デプロイメント・マネージャー。
   - マイグレーションしたい各非クラスター管理対象ノード。

   セル全体をマイグレーション**しない**場合で、セル内の 1 つ以上のクラスターをマイグレーションし、ゼロ個以上の非クラスター管理対象ノードをマイグレーションする場合、以下をマイグレーションします。
   - デプロイメント・マネージャー。
   - マイグレーションしたい各非クラスター管理対象ノード。
   - 明示的なマイグレーション対象の各クラスター、およびクラスターのメンバーであるアプリケーション・サーバーを保有するすべてのノード (クラスター管理対象ノード)。
   - あらゆるクラスター、およびマイグレーション対象のクラスターから暗黙的な影響を受けるクラスターのクラスター管理対象ノード。影響を受けるすべてのクラスターおよびそのクラスター管理対象ノードの推移的閉包を識別するには、以下の手順を使用します。
     – マイグレーション対象の各クラスターについて、クラスターに対応するアプリケーション・サーバーを保有するクラスター管理対象ノードを識別します。

- 各クラスター管理対象ノードについて、メンバーになっている他のクラスター、および存在する場合は、そのノードで実行中のアプリケーション・サーバーを識別します。
- これらの各クラスターについてプロセスを繰り返し、この手順の一部としてマイグレーションが必要なクラスターおよびクラスター管理対象ノードの完全なセットを決定します。

5. **デプロイメント・マネージャーを停止します。**

   マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopManager コマンドを使用して、ソース・デプロイメント・マネージャーを停止します。

   以下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:** stopManager.sh -username *user_name* -password *password*
   - Windows **Windows プラットフォームの場合:** stopManager.bat -username *user_name* -password *password*

   プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

   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、これらのパラメーターは、Windows サービスがサーバー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。

   プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

   stopManager コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopManager コマンド』のトピックを参照してください。

6. **非クラスター管理対象ノードのマイグレーション・ソース・サーバーを停止します。**

   マイグレーションする非クラスター管理対象ノードに関連付けられた各サーバーについて、このステップを繰り返します。

   マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopServer コマンドを使用して、マイグレーション・ソース・サーバーを停止します。

   以下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:** stopServer.sh *server_name* -username *user_name* -password *password*

- Windows **Windows プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

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、これらのパラメーターは、Windows サービスがサーバー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。

プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

7. **非クラスター管理対象ノードのマイグレーション・ソース・ノード・エージェントを停止します。**

マイグレーションする非クラスター管理対象ノードに関連付けられた各ノード・エージェントについて、このステップを繰り返します。

マイグレーション・ソース・システムの *profile_root*/bin から、 stopNode コマンドを使用して、マイグレーション・ソースのノード・エージェントを停止します。

以下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:** stopNode.sh -username *user_name* -password *password*
- Windows **Windows プラットフォームの場合:** stopNode.bat -username *user_name* -password *password*

プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

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、これらのパラメーターは、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。

プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

stopNode コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopNode コマンド』のトピックを参照してください。

8. **クラスター管理対象ノードのマイグレーション・ソース・サーバーを停止します。**

   マイグレーションするクラスター管理対象ノードに関連付けられた各サーバーについて、このステップを繰り返します。

   マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopServer コマンドを使用して、マイグレーション・ソース・サーバーを停止します。

   以下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:** stopServer.sh *server_name* -username *user_name* -password *password*
   - Windows **Windows プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

   プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

   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、これらのパラメーターは、Windows サービスがサーバー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。

   プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

   stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

9. **クラスター管理対象ノードのマイグレーション・ソース・ノード・エージェントを停止します。**

   マイグレーションするクラスター管理対象ノードに関連付けられた各ノード・エージェントについて、このステップを繰り返します。

   マイグレーションによって影響を受ける各ノード・エージェントについてこのステップを繰り返します。

   マイグレーション・ソース・システムの *profile_root*/bin から、 stopNode コマンドを使用して、マイグレーション・ソースのノード・エージェントを停止します。

   以下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:** stopNode.sh -username *user_name* -password *password*
   - Windows **Windows プラットフォームの場合:** stopNode.bat -username *user_name* -password *password*

プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

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、これらのパラメーターは、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。

プロファイルでセキュリティーが有効になっていない場合、 `-username` および `-password` パラメーターは必要ありません。

`stopNode` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopNode コマンド』のトピックを参照してください。

10. **マイグレーション・ソース・プロファイルをバックアップします。**

    マイグレーション対象の各プロファイルについてこのステップを繰り返します (デプロイメント・マネージャー、各非クラスター管理対象ノード、および各管理対象ノードなど)。

    `backupConfig` コマンドを使用して、マイグレーション・ソース・サーバー上のプロファイル構成をバックアップします。

    以下の構文を使用して、profile1 という名前のプロファイルを /ProfileBackups/profile1.zip にバックアップします。

    - Linux UNIX **Linux および UNIX プラットフォームの場合:**
      ```
      backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1
      ```
    - Windows **Windows プラットフォームの場合:**
      ```
      backupConfig.bat c:¥ProfileBackups¥profile1.zip -profileName profile1
      ```

    `backupConfig` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

11. **.nifRegistry ファイルをバックアップします。**

    .nifRegistry ファイルは、インストール済みのすべての WebSphere Process Server 製品のインストール・ルートを示します。また、インストール済みのすべての WebSphere Application Server 製品のインストール・ルートも示します。場所は以下のとおりです。

    - Linux UNIX **Linux または UNIX プラットフォームの場合**:
      ```
      /opt/.ibm/.nif/.nifregistry
      ```
    - Windows **Windows プラットフォームの場合**:
      - 製品をインストールしたユーザー ID が管理特権を持っていた場合のファイルの場所は Windows ルート・ディレクトリー (ほとんどの Windows システムでは `C:¥Windows` または `C:¥WINNT`) です。
      - 製品をインストールしたユーザー ID が管理特権を持っていなかった場合のファイルの場所は、そのユーザー ID のホーム・ディレクトリーです。

12. **マイグレーション・ソース製品データベースをバックアップします。**

    ご使用のデータベースのドキュメンテーションに応じて、任意のマイグレーション・ソース・プロファイルによって構成された以下のデータベースをバックアップします。

    - Business Process Choreographer データベース
    - Business Space データベース
    - 共通データベース
    - Common Event Infrastructure データベース
    - メッセージング・エンジン・データベース

13. **デプロイメント・マネージャー・プロファイルをマイグレーションします。**

    BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、デプロイメント・マネージャー・ソース・プロファイルをマイグレーションできます。BPM プロファイル・マイグレーション・ウィザードを使用するには、デプロイメント・マネージャー・プロファイルが含まれているシステム上で、 78 ページの『BPM プロファイル・マイグレーション・ウィザードを使用したプロファイルのマイグレーション』の手順を実行します。BPM マイグレーション・コマンド行ユーティリティーを使用するには、デプロイメント・マネージャー・プロファイルを含むシステムで『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』サブプロシージャーに従います。

    注: WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合は、 78 ページの『BPM プロファイル・マイグレーション・ウィザードを使用したプロファイルのマイグレーション』 の手順を使用する必要があります。

14. **Cloudscape または Derby データベースを更新します。**

    Cloudscape または Derby データベースを使用している場合、サポートされるバージョンを使用していることを確認する必要があります。 Cloudscape を Derby にマイグレーションする方法については、 IBM Cloudscape または Apache Derby データベースのマイグレーションを参照してください。

15. **共通データベースをマイグレーションおよびアップグレードするスクリプトを、データベース・システムにコピーします。**

    データベース・タイプに合わせてカスタマイズされた、共通データベースをマイグレーションおよびアップグレードするコマンドとスクリプトを、ターゲット・マイグレーション・システムで見つけてデータベース・システムにコピーします。コマンドおよびスクリプトは、以下のディレクトリーにあります。
    install_root/dbscripts/CommonDB/*database_type*

    注: Business Process Choreographer データベースのコマンドとスクリプトは、後で DBDesignGenerator コマンドを使用してコピーします。詳しくは、 95 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』を参照してください。
    次の表を使用して、ご使用の特定のデータベース・タイプに対応したディレクトリー名を判別します。

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 バージョン 6.1.0 の WebSphere Process Server では、Cloudscape データベースが Derby に置き換わりました。大半の環境では、プロファイル・マイグレーション・ツールによって自動的に Cloudscape データベースが Derby にマイグレーションされます。例外については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』のトピックに説明があります。
Informix	Informix
Oracle	Oracle
Microsoft SQL Server	SQLServer

16. **セル・スコープの共通データベースをアップグレードします。**

    共通データベースのデータ・ソースに対して定義されたデータベース・ユーザーに十分な権限がない場合は、『共通データベース・スキーマのアップグレード』の手順を使用して共通データベース・スキーマを手動でアップグレードします。

17. **ターゲット・デプロイメント・マネージャーを始動します。**

    デプロイメント・マネージャー・システムの *profile_root*/bin ディレクトリーから、またはデプロイメント・マネージャー・プロファイルのファースト・ステップ・コンソールから、startManager コマンドを使用して、ターゲット・デプロイメント・マネージャーを始動します。

    以下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:** startManager.sh
    - Windows **Windows プラットフォームの場合:** startManager.bat

startManager コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startManager コマンド』のトピックを参照してください。

18. **データ・ソース構成を更新します。** 組み込み DataDirect ドライバーを使用するデータ・ソースがあり、ライセンス交付を受けた DataDirect JDBC ドライバーまたは Microsoft JDBC ドライバーを使用するようにそれらのデータ・ソースをソース環境で更新していない場合は、データ・ソース構成を更新します。これを行うには、以下の手順を使用します。

    重要: 一部のコンポーネントがデータベースへの接続を確立できなかったため、 SystemOut.log ファイルにエラーが記録されている可能性があります。

    a. 管理コンソールにログインします。
    b. 正しい JDBC プロバイダー・タイプの新規データ・ソースを作成し、プロパティーを設定します。設定するプロパティーは、 JNDI 名、statementCacheSize、 releationalResourceAdapter、 authMechanismPreference、authDataAlias、 databaseName、serverName、 portNumber、および既存のデータ・ソースに一致する URL です。
    c. 組み込みドライバーを使用する既存のデータ・ソースを削除します。
    d. 「テスト接続」オプションを使用して、データ・ソース構成が機能するかどうかを確認します。
    e. デプロイメント・マネージャーを再始動します。

19. **非クラスター管理対象ノードをマイグレーションします。**

    このステップは、マイグレーション対象の非クラスター管理対象ノードごとに繰り返す必要があります。

    BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、非クラスター管理対象ノード・ソース・プロファイルをマイグレーションできます。BPM プロファイル・マイグレーション・ウィザードを使用するには、クラスター化されていない管理対象ノードのプロファイルが含まれているシステム上で、 78 ページの『BPM プロファイル・マイグレーション・ウィザードを使用したプロファイルのマイグレーション』の手順を実行します。BPM マイグレーション・コマンド行ユーティリティーを使用するには、非クラスター管理対象ノード・プロファイルを含むシステムで『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』サブプロシージャーに従います。

    注: WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合、『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順を使用する必要があります。

20. **非クラスター管理対象ノード製品データベースをマイグレーションします。**

    このステップは、マイグレーション対象の非クラスター管理対象ノードごとに繰り返す必要があります。

    以下の手順を使用して、非クラスター管理対象ノード上に構成された各製品データベースをマイグレーションします。

a. 以下の条件のいずれかが該当する場合は、『Business Process Choreographer データベース・スキーマのアップグレード』の手順を使用して Business Process Choreographer データベース・スキーマを手動でアップグレードします。
   - Business Process Choreographer データベースにデフォルトのテーブル・スペースを使用しなかった。Business Process Choreographer のサンプル構成を使用したか、サンプル SQL スクリプトで指定されたデフォルト・テーブル・スペース内のすべてのデータベース・オブジェクトを作成済みの場合、データベースではデフォルト・テーブル・スペースを使用します。通常、テスト環境がこれに該当します。
   - BPEDB データ・ソース用に構成されたデータベース・ユーザーが、テーブルの作成と変更、および索引とビューの作成と除去の操作のすべてを実行することを許可されているわけではなく、テーブル SCHEMA_VERSION の場合は、照会、更新、削除、および挿入のすべての実行を許可されているわけではない。

b. マイグレーション元のソース・バージョンが 6.0.2、6.1.0、または 6.1.2 の場合は、『Business Process Choreographer データベース・データのマイグレーション』の手順に従って、Business Process Choreographer データベース・データをマイグレーションします。

c. 『Business Space データベース・スキーマのマイグレーション』の手順を使用して Business Space データベース・スキーマをマイグレーションします。

d. 『Business Space データベース・データのマイグレーション』の手順を使用して Business Space データベース・データをマイグレーションします。

e. オプション: ご使用の環境に合わせて、必要に応じてメッセージング・エンジン・データベースをマイグレーションします。 メッセージング・エンジンのマイグレーションが必要な場合とその方法について詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターで『データ・ストアに基づくメッセージング・エンジンのマイグレーション』を参照してください。

21. **オプション: ビジネス・ルール・マネージャーをマイグレーションします。**

このステップは、マイグレーション対象の非クラスター管理対象ノードごとに繰り返す必要があります。

ビジネス・ルール・マネージャーは、セル内の最後のノードがマイグレーションされるときに自動的にマイグレーションされますが、マイグレーションされた非クラスター管理対象ノードにビジネス・ルール・マネージャーが含まれている場合、手動でマイグレーションできます。

サーバー server1 および非クラスター管理対象ノード node1 のビジネス・ルール・マネージャーを手動でマイグレーションするには、以下のコマンドを使用します。

```
wsadmin -f installBRManager.jacl -s server1 -n node1
```

installBRManager コマンドについて詳しくは、『installBRManager コマンド』のトピックを参照してください。

22. **クラスターをマイグレーションします。**

マイグレーションする必要のあるネットワーク・デプロイメント環境内の各クラスターに対して、以下の手順を繰り返します。

a. **管理対象ノードをマイグレーションします。**

このステップは、クラスター内の各管理対象ノードについて繰り返す必要があります。

BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、管理対象ノード・ソース・プロファイルをマイグレーションできます。BPM プロファイル・マイグレーション・ウィザードを使用するには、ソース・プロファイルを含むシステムで 78 ページの『BPM プロファイル・マイグレーション・ウィザードを使用したプロファイルのマイグレーション』の手順に従います。BPM マイグレーション・コマンド行ユーティリティーを使用するには、ソース・プロファイルを含むシステムで『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』サブプロシージャーに従います。

注: WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合、『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順を使用する必要があります。

b. **クラスター・スコープ・プロファイルをマイグレーションします。**

デプロイメント・マネージャーを含むシステムで、 *install_root*/bin ディレクトリーから BPMMigrateCluster コマンドを使用して、クラスター・スコープ・プロファイルをマイグレーションします。

以下の構文を使用して、/MigrationSnapshots/ProcServer620 ディレクトリーにコピーされた dmgrProfile という名前のデプロイメント・マネージャー・プロファイルと共に、applicationCluster1 という名前のクラスターをマイグレーションします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:**
```
BPMMigrateCluster.sh /MigrationSnapshots/ProcServer620
applicationCluster1 dmgrProfile
```
- Windows **Windows プラットフォームの場合:**
```
BPMMigrateCluster.bat
c:¥MigrationSnapshots¥ProcServer620 applicationCluster1
dmgrProfile
```

BPMMigrateCluster コマンドについて詳しくは、『BPMMigrateCluster コマンド』のトピックを参照してください。

c. **管理対象プロファイルをバックアップします。**

管理対象ノード内の各プロファイルに対して、このステップを繰り返します。このバックアップは、次のステップで syncNode コマンドの実行が失敗した場合に必要となります。syncNode に関する問題が解決したら、バックアップをリストアしてから syncNode コマンドを再度実行します。

backupConfig コマンドを使用して、非クラスター管理対象ノード上のプロファイル構成をバックアップします。

以下の構文を使用して、profile1 という名前のプロファイルを /ProfileBackups/profile1.zip にバックアップします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:**
  backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1
- Windows **Windows プラットフォームの場合:** backupConfig.bat c:¥ProfileBackups¥profile1.zip -profileName profile1

backupConfig コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

d. **管理対象ノードを同期します。**

このステップは、クラスター内の各管理対象ノードについて繰り返す必要があります。

マイグレーション・ターゲット・プロファイルの *profile_root*/bin ディレクトリーから、またはターゲット・プロファイルのファースト・ステップ・コンソールから、 syncNode コマンドを使用して、ターゲット・デプロイメント・マネージャーとノードを同期します。

以下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:**
  syncNode.sh *deployment_manager_machine_name_or_ip_address deployment_manager_port_no*
- Windows **Windows プラットフォームの場合:** syncNode.bat *deployment_manager_machine_name_or_ip_address deployment_manager_port_no*

syncNode コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『syncNode コマンド』のトピックを参照してください。

e. **クラスター・スコープ製品データベースをマイグレーションします。**

以下の手順を使用して、クラスターについて構成された各製品データベースをマイグレーションします。

1) 以下の条件のいずれかが該当する場合は、『Business Process Choreographer データベース・スキーマのアップグレード』の手順を使用して Business Process Choreographer データベース・スキーマを手動でアップグレードします。
   - Business Process Choreographer データベースにデフォルトのテーブル・スペースを使用しなかった。Business Process Choreographer のサンプル構成を使用したか、サンプル SQL スクリプトで指定されたデフォルト・テーブル・スペース内のすべてのデータベース・オブジェ

クトを作成済みの場合、データベースではデフォルト・テーブル・スペースを使用します。通常、テスト環境がこれに該当します。

- BPEDB データ・ソース用に構成されたデータベース・ユーザーが、テーブルの作成と変更、および索引とビューの作成と除去の操作のすべてを実行することを許可されているわけではなく、テーブル SCHEMA_VERSION の場合は、照会、更新、削除、および挿入のすべての実行を許可されているわけではない。

2) マイグレーション元のソース・バージョンが 6.0.2、6.1.0、または 6.1.2 の場合は、『Business Process Choreographer データベース・データのマイグレーション』の手順に従って、Business Process Choreographer データベース・データをマイグレーションします。

3) 『Business Space データベース・スキーマのマイグレーション』の手順を使用して Business Space データベース・スキーマをマイグレーションします。

4) 『Business Space データベース・データのマイグレーション』の手順を使用して Business Space データベース・データをマイグレーションします。

5) オプション: ご使用の環境に合わせて、必要に応じてメッセージング・エンジン・データベースをマイグレーションします。メッセージング・エンジンのマイグレーションが必要な場合とその方法について詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターで『データ・ストアに基づくメッセージング・エンジンのマイグレーション』を参照してください。

f. **オプション: ビジネス・ルール・マネージャーをマイグレーションします。**

ビジネス・ルール・マネージャーは、セル内の最後のノードがマイグレーションされるときに自動的にマイグレーションされますが、マイグレーションされたクラスターにビジネス・ルール・マネージャーが含まれている場合、手動でマイグレーションできます。

クラスター cluster1 のビジネス・ルール・マネージャーを手動でマイグレーションするには、以下のコマンドを使用します。

```
wsadmin -f installBRManager.jacl -cl cluster1
```

installBRManager コマンドについて詳しくは、installBRManager コマンド行ユーティリティーのトピックを参照してください。

g. **マイグレーション・ターゲット・ノード・エージェントを始動します。**

マイグレーションされた非クラスター管理対象ノード、およびマイグレーションされた各クラスターのクラスター管理対象ノードごとに、このステップを繰り返します。

マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、startNode コマンドを使用して、マイグレーション・ターゲット・ノード・エージェントを始動します。

以下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:** `startNode.sh`
- Windows **Windows プラットフォームの場合:** `startNode.bat`

`startNode` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startNode コマンド』のトピックを参照してください。

23. **マイグレーション・ターゲット・サーバーを始動します。**

マイグレーション済みのクラスター化されていない各管理対象ノードと、マイグレーション済みのクラスター化された各管理対象ノード用に構成された各サーバーに対して、このステップを繰り返します。

マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、`startServer` コマンドを使用して、マイグレーション・ターゲット・サーバーを始動します。

以下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:** `startServer.sh` *server_name*
- Windows **Windows プラットフォームの場合:** `startServer.bat` *server_name*

`startServer` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

24. **オプション: ソース・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。**

マイグレーションが完了した後は、マイグレーション・ソース・デプロイメント・マネージャーをアンインストールできます。

25. **互換モードを除去します。**

互換性オプション (デフォルト) を選択していて、すべてのノードが完全にターゲット・バージョンにマイグレーションされている場合は、`convertScriptCompatibility` スクリプトをデプロイメント・マネージャーおよび各ノード上の *install_root*/bin ディレクトリーから実行して、互換性を除去します。

以下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:** `convertScriptCompatibility.sh`
- Windows **Windows プラットフォームの場合:** `convertScriptCompatibility.bat`

convertScriptCompatibility コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『convertScriptCompatibility』のトピックを参照してください。

タスクの結果

Network Deployment 環境はターゲット・バージョンにマイグレーションされます。

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。手順については、109 ページの『マイグレーションの検査』を参照してください。

最小限のダウン時間での Network Deployment 環境のマイグレーション

この手順を使用して、最小限のダウン時間を発生させて 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 29 ページの『ランタイム・マイグレーション前のチェックリスト』のトピックを確認します。

注: ビジネス・カレンダーまたはメディエーション・フロー・コンポーネントを利用するアプリケーションがソース・バージョンに含まれている場合は、これらのアプリケーションでダウン時間が許容されていない限り、最小限のダウン時間の手順を使用することはできません。ビジネス・カレンダーまたはメディエーション・フロー・コンポーネントを利用するアプリケーションを実行しているサーバーがノードに含まれている場合、そのノードはバージョン 7.0 にマイグレーションされるまで停止したままになります。

このタスクについて

このステップに従って、最小限のダウン時間を発生させて Network Deployment 環境をマイグレーションします。

手順

1. **マイグレーション・ターゲット製品をインストールします。**

   ターゲット製品および最新のフィックスパックを、マイグレーションのソース製品と同じシステムにインストールします。

   注: ソース・バージョンのインストールに使用したのと同じユーザー ID でターゲット・バージョンをインストールするか、ソース・インストール済み環境の構成およびデータにアクセスする権限を持っている必要があります。

   注: 複数の製品によって拡張されたソース・プロファイルからマイグレーションするには、それらの製品の新バージョンを同じターゲット・インストール・ディレクトリーにインストールする必要があります。例えば、ソース・プロファイルが WebSphere Process Server および WebSphere Business Monitor によ

って拡張されている場合、これら両方の製品が同じターゲット・インストール・ディレクトリーにインストールされる必要があります。

2. **DB2® for z/OS® および OS/390® Version 7 をアップグレードします。**

   DB2® for z/OS® および OS/390® Version 7 を使用しており、データベースを DB2 for z/OS Version 8 または DB2 9 for z/OS にアップグレードしていない場合、DB2 for z/OS のドキュメンテーションの説明を参照して今すぐにアップグレードを実行してください。

3. **Oracle 9i および Oracle JDBC ドライバーをアップグレードします。**

   重要: Oracle データベースにアクセスするすべての WebSphere Process Server インストール済み環境で、このステップを実行する必要があります。

   a. Oracle 9i を使用しており、データベースを 10g または 11g にアップグレードしていない場合、Oracle の資料の説明を参照して、今すぐにアップグレードを実行してください。

   b. `ojdbc14.jar` または `ojdbc5.jar` ドライバーを使用している場合、*ORACLE_JDBC_DRIVER_PATH* WebSphere 変数によって示されているディレクトリーに新しい `ojdbc6.jar` ドライバーをインストールする必要があります。これを行うには、以下の手順を使用します。

      1) 以前の環境の *ORACLE_JDBC_DRIVER_PATH* 変数の値を確認します。これを行うには、以下の方法のいずれかを使用します。
         - 管理コンソールで、「**環境**」 → 「**WebSphere 変数**」を選択し、ソース・プロファイルのノードと一致するスコープを選択します。
         - 以下のディレクトリーにある `variables.xml` ファイルにナビゲートします。
           source_profile_root¥config¥cells¥*cell_name*¥nodes¥*node_name*¥.

           注: セル名とノード名がソース・プロファイルの情報と一致している必要があります。

      2) 新しい `ojdbc6.jar` ドライバーを、*ORACLE_JDBC_DRIVER_PATH* WebSphere 変数で示されているディレクトリーにインストールします。変数で示されているロケーションに応じて、以下のいずれかのステップを使用します。
         - この変数が示しているディレクトリーが、WebSphere Process Server インストール済み環境の外部にある場合は、`odbc6.jar` ファイルを `ojdbc14.jar` または `ojdbc5.jar` ファイルと同じフォルダーにコピーします。
         - この変数が示しているディレクトリーが、WebSphere Process Server インストール済み環境内にある場合は、WebSphere Process Server バージョン 7.0 インストール済み環境内に同じディレクトリー構造を作成して、そのディレクトリーに `odbc6.jar` ファイルをコピーします。

4. **マイグレーション対象のクラスター、クラスター管理対象ノード、および非クラスター管理対象ノードを識別します。**

   セル全体をマイグレーションする場合、以下をマイグレーションします。
   - デプロイメント・マネージャー。

- セル内のクラスターのメンバーであるアプリケーション・サーバーを保有しないすべてのノード (非クラスター管理対象ノード)。
- これらのクラスターのメンバーであるアプリケーション・サーバーを保有するすべてのクラスターおよびすべてのノード (クラスター管理対象ノード)。

セル全体をマイグレーション**しない**場合、クラスターをマイグレーション**しない**場合、およびセル内のクラスターのメンバーであるアプリケーション・サーバーを保有しない 1 つ以上のノード (非クラスター管理対象ノード) をマイグレーション**する**場合、以下をマイグレーションします。

- デプロイメント・マネージャー。
- マイグレーションしたい各非クラスター管理対象ノード。

セル全体をマイグレーション**しない**場合で、セル内の 1 つ以上のクラスターをマイグレーションし、ゼロ個以上の非クラスター管理対象ノードをマイグレーションする場合、以下をマイグレーションします。

- デプロイメント・マネージャー。
- マイグレーションしたい各非クラスター管理対象ノード。
- 明示的なマイグレーション対象の各クラスター、およびクラスターのメンバーであるアプリケーション・サーバーを保有するすべてのノード (クラスター管理対象ノード)。
- あらゆるクラスター、およびマイグレーション対象のクラスターから暗黙的な影響を受けるクラスターのクラスター管理対象ノード。影響を受けるすべてのクラスターおよびそのクラスター管理対象ノードの推移的閉包を識別するには、以下の手順を使用します。
  - マイグレーション対象の各クラスターについて、クラスターに対応するアプリケーション・サーバーを保有するクラスター管理対象ノードを識別します。
  - 各クラスター管理対象ノードについて、メンバーになっている他のクラスター、および存在する場合は、そのノードで実行中のアプリケーション・サーバーを識別します。
  - これらの各クラスターについてプロセスを繰り返し、この手順の一部としてマイグレーションが必要なクラスターおよびクラスター管理対象ノードの完全なセットを決定します。

5. **すべてのノードについて同期を無効にします。**

   ソース・デプロイメント・マネージャーで管理コンソールを使用する、すべての非クラスター管理対象ノードおよびクラスター管理対象ノードについて、同期を無効にします。

   「システム管理」>「ノード・エージェント」に移動します。

   ノードのノード・エージェントをクリックします。

   「ファイル同期化サービス」をクリックします。

   「サーバー始動時にサービスを使用可能にする」、「自動同期 (Automatic Synchronization)」、および「始動同期」の既存の設定をメモして、後の手順でノード同期を再度有効にするときにこれらの設定を復元できるようにします。

「サーバー始動時にサービスを使用可能にする」、「自動マイグレーション」、および「始動同期」のチェック・ボックスのチェック・マークを外します。

「適用」、「OK」、「保存」をクリックして、構成の変更を保存します。

6. **デプロイメント・マネージャーを停止します。**

   マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopManager コマンドを使用して、マイグレーション・ソースのデプロイメント・マネージャーを停止します。

   以下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:** stopManager.sh -username *user_name* -password *password*
   - Windows **Windows プラットフォームの場合:** stopManager.bat -username *user_name* -password *password*

   プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

   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、これらのパラメーターは、Windows サービスがサーバー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。

   プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

   stopManager コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopManager コマンド』のトピックを参照してください。

7. **ソース・デプロイメント・マネージャー・プロファイルをバックアップします。**

   backupConfig コマンドを使用して、ソース・デプロイメント・マネージャー・システム上のデプロイメント・マネージャー・プロファイル構成をバックアップします。

   以下の構文を使用して、dmgrProfile という名前のプロファイルを /ProfileBackups/dmgrProfile.zip にバックアップします。

   - Linux UNIX **Linux および UNIX プラットフォームの場合:** backupConfig.sh /ProfileBackups/dmgrProfile.zip -profileName dmgrProfile
   - Windows **Windows プラットフォームの場合:** backupConfig.bat c:¥ProfileBackups¥profile1.zip -profileName dmgrProfile

backupConfig コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

8. **.nifRegistry ファイルをバックアップします。**

   .nifRegistry ファイルは、インストール済みのすべての WebSphere Process Server 製品のインストール・ルートを示します。また、インストール済みのすべての WebSphere Application Server 製品のインストール・ルートも示します。場所は以下のとおりです。

   - Linux UNIX **Linux または UNIX プラットフォームの場合**: /opt/.ibm/.nif/.nifregistry
   - Windows **Windows プラットフォームの場合**:
     - 製品をインストールしたユーザー ID が管理特権を持っていた場合のファイルの場所は Windows ルート・ディレクトリー (ほとんどの Windows システムでは C:¥Windows または C:¥WINNT) です。
     - 製品をインストールしたユーザー ID が管理特権を持っていなかった場合のファイルの場所は、そのユーザー ID のホーム・ディレクトリーです。

9. **セル・スコープの共通データベースをバックアップします。**

   ご使用のデータベース・サーバーのドキュメンテーションを使用して、セル・スコープの共通データベースをバックアップします。

10. **デプロイメント・マネージャー・プロファイルをマイグレーションします。**

    BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、デプロイメント・マネージャー・ソース・プロファイルをマイグレーションできます。BPM プロファイル・マイグレーション・ウィザードを使用するには、デプロイメント・マネージャー・プロファイルを含むシステムで『BPM マイグレーション・ウィザードを使用したプロファイルのマイグレーション』サブプロシージャーに従います。BPM マイグレーション・コマンド行ユーティリティーを使用するには、デプロイメント・マネージャー・プロファイルを含むシステムで『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』サブプロシージャーに従います。

    注: WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合、 『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順を使用する必要があります。

11. **Cloudscape または Derby セル・スコープ・データベースを更新します。**

    Cloudscape または Derby データベースを共通データベースに使用している場合、サポートされるバージョンを使用していることを確認する必要があります。 Cloudscape を Derby にマイグレーションする方法については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』を参照してください。

12. **共通データベースをマイグレーションおよびアップグレードするスクリプトを、データベース・システムにコピーします。**

データベース・タイプに合わせてカスタマイズされた、共通データベースをマイグレーションおよびアップグレードするコマンドとスクリプトを、ターゲット・マイグレーション・システムで見つけてデータベース・システムにコピーします。コマンドおよびスクリプトは、以下のディレクトリーにあります。
install_root/dbscripts/CommonDB/*database_type*

注: Business Process Choreographer データベース・コマンドとスクリプトは、後のプロセスで DBDesignGenerator コマンドを使用してコピーされます。詳しくは、95 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』を参照してください。
次の表を使用して、ご使用の特定のデータベース・タイプに対応したディレクトリー名を判別します。

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 バージョン 6.1.0 の WebSphere Process Server では、Cloudscape データベースが Derby に置き換わりました。大半の環境では、プロファイル・マイグレーション・ツールによって自動的に Cloudscape データベースが Derby にマイグレーションされます。例外については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』のトピックに説明があります。
Informix	Informix
Oracle	Oracle
Microsoft SQL Server	SQLServer

13. **セル・スコープの共通データベースをアップグレードします。**

    共通データベースのデータ・ソースに対して定義されたデータベース・ユーザーに十分な権限がない場合は、『共通データベース・スキーマのアップグレード』の手順を使用して共通データベース・スキーマを手動でアップグレードします。

14. **ターゲット・デプロイメント・マネージャーを始動します。**

デプロイメント・マネージャー・システムの *profile_root*/bin ディレクトリーから、またはデプロイメント・マネージャー・プロファイルのファースト・ステップ・コンソールから、startManager コマンドを使用して、ターゲット・デプロイメント・マネージャーを始動します。

以下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:** startManager.sh
- Windows **Windows プラットフォームの場合:** startManager.bat

startManager コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startManager コマンド』のトピックを参照してください。

15. **データ・ソース構成を更新します。** 組み込み DataDirect ドライバーを使用するデータ・ソースがあり、ライセンス交付を受けた DataDirect JDBC ドライバーまたは Microsoft JDBC ドライバーを使用するようにそれらのデータ・ソースをソース環境で更新していない場合は、データ・ソース構成を更新します。これを行うには、以下の手順を使用します。

    重要: 一部のコンポーネントがデータベースへの接続を確立できなかったため、 SystemOut.log ファイルにエラーが記録されている可能性があります。

    a. 管理コンソールにログインします。
    b. 正しい JDBC プロバイダー・タイプの新規データ・ソースを作成し、プロパティーを設定します。設定するプロパティーは、 JNDI 名、statementCacheSize、 releationalResourceAdapter、 authMechanismPreference、authDataAlias、 databaseName、serverName、 portNumber、および既存のデータ・ソースに一致する URL です。
    c. 組み込みドライバーを使用する既存のデータ・ソースを削除します。
    d. 「テスト接続」オプションを使用して、データ・ソース構成が機能するかどうかを確認します。
    e. デプロイメント・マネージャーを再始動します。

16. **非クラスター管理対象ノードをマイグレーションします。**

    マイグレーションのソースである、各非クラスター管理対象ノードについて、ステップ 15 から 25 を繰り返します。

17. **非クラスター管理対象ノードのマイグレーション・ソース・サーバーを停止します。**

    マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopServer コマンドを使用して、マイグレーション・ソース・サーバーを停止します。

    以下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:** stopServer.sh *server_name* -username *user_name* -password *password*

- Windows **Windows プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

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、これらのパラメーターは、Windows サービスがサーバー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。

プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

18. **非クラスター管理対象ノードのマイグレーション・ソース・ノード・エージェントを停止します。**

マイグレーション・ソース・システムの *profile_root*/bin から、 stopNode コマンドを使用して、マイグレーション・ソースのノード・エージェントを停止します。

以下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:** stopNode.sh -username *user_name* -password *password*
- Windows **Windows プラットフォームの場合:** stopNode.bat -username *user_name* -password *password*

プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

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、これらのパラメーターは、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。

プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

stopNode コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopNode コマンド』のトピックを参照してください。

19. **非クラスター管理対象ノードのマイグレーション・ソース・プロファイルをバックアップします。**

backupConfig コマンドを使用して、非クラスター管理対象ノード上のプロファイル構成をバックアップします。

以下の構文を使用して、profile1 という名前のプロファイルを /ProfileBackups/profile1.zip にバックアップします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1
- Windows **Windows プラットフォームの場合:** backupConfig.bat c:¥ProfileBackups¥profile1.zip -profileName profile1

backupConfig コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

20. **非クラスター管理対象ノード用に構成された、サーバー・スコープ製品データベースをバックアップします。**

ご使用のデータベースのドキュメンテーションに応じて、非クラスター管理対象ノード用に構成された以下の製品データベースをバックアップします。

- Business Process Choreographer データベース
- Business Space データベース
- Common Event Infrastructure データベース
- メッセージング・エンジン・データベース

21. **非クラスター管理対象ノードをマイグレーションします。**

BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、非クラスター管理対象ノード・ソース・プロファイルをマイグレーションできます。BPM プロファイル・マイグレーション・ウィザードを使用するには、非クラスター管理対象ノード・プロファイルを含むシステムで『BPM マイグレーション・ウィザードを使用したプロファイルのマイグレーション』サブプロシージャーに従います。BPM マイグレーション・コマンド行ユーティリティーを使用するには、非クラスター管理対象ノード・プロファイルを含むシステムで『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順に従います。

注: WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合、『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順を使用する必要があります。

22. **Cloudscape または Derby 非クラスター管理対象ノード・スコープ・データベースを更新します。**

非クラスター管理対象ノード・スコープ・データベース用に構成された Cloudscape または Derby データベースを使用している場合、サポートされるバージョンを使用していることを確認する必要があります。 Cloudscape を Derby にマイグレーションする方法については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』を参照してください。

23. **非クラスター管理対象ノード製品データベースをマイグレーションします。**

以下の手順を使用して、非クラスター管理対象ノード上に構成された各製品データベースをマイグレーションします。

a. 以下の条件のいずれかが該当する場合は、『Business Process Choreographer データベース・スキーマのアップグレード』の手順を使用して Business Process Choreographer データベース・スキーマを手動でアップグレードします。
   - Business Process Choreographer データベースにデフォルトのテーブル・スペースを使用しなかった。Business Process Choreographer のサンプル構成を使用したか、サンプル SQL スクリプトで指定されたデフォルト・テーブル・スペース内のすべてのデータベース・オブジェクトを作成済みの場合、データベースではデフォルト・テーブル・スペースを使用します。通常、テスト環境がこれに該当します。
   - BPEDB データ・ソース用に構成されたデータベース・ユーザーが、テーブルの作成と変更、および索引とビューの作成と除去の操作のすべてを実行することを許可されているわけではなく、テーブル SCHEMA_VERSION の場合は、照会、更新、削除、および挿入のすべての実行を許可されているわけではない。

b. マイグレーション元のソース・バージョンが 6.0.2、6.1.0、または 6.1.2 の場合は、『Business Process Choreographer データベース・データのマイグレーション』の手順に従って、Business Process Choreographer データベース・データをマイグレーションします。

c. 『Business Space データベース・スキーマのマイグレーション』の手順を使用して Business Space データベース・スキーマをマイグレーションします。

d. 『Business Space データベース・データのマイグレーション』の手順を使用して Business Space データベース・データをマイグレーションします。

e. オプション: ご使用の環境に合わせて、必要に応じてメッセージング・エンジン・データベースをマイグレーションします。 メッセージング・エンジンのマイグレーションが必要な場合とその方法について詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターで『データ・ストアに基づくメッセージング・エンジンのマイグレーション』を参照してください。

24. **オプション: ビジネス・ルール・マネージャーをマイグレーションします。**

ビジネス・ルール・マネージャーは、セル内の最後のノードがマイグレーションされるときに自動的にマイグレーションされますが、マイグレーションされた非クラスター管理対象ノードにビジネス・ルール・マネージャーが含まれている場合、手動でマイグレーションできます。

サーバー server1 および非クラスター管理対象ノード node1 のビジネス・ルール・マネージャーを手動でマイグレーションするには、以下のコマンドを使用します。

```
wsadmin -f installBRManager.jacl -s server1 -n node1
```

installBRManager コマンドについて詳しくは、『installBRManager コマンド』のトピックを参照してください。

25. **非クラスター管理対象ノードの同期を有効にします。**

    ターゲット・デプロイメント・マネージャーで管理コンソールを使用してマイグレーションされた、非クラスター管理対象ノードの同期を有効にします。

    「システム管理」>「ノード・エージェント」に移動します。

    ノードのノード・エージェントをクリックします。

    「ファイル同期化サービス」をクリックします。

    「サーバー始動時にサービスを使用可能にする」、「自動マイグレーション」、および「始動同期」の設定を復元します。

    「適用」、「OK」、「保存」をクリックして、構成の変更を保存します。

26. **マイグレーション・ターゲットの非クラスター管理対象ノード・エージェントを始動します。**

    マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、startNode コマンドを使用して、マイグレーション・ターゲットの非クラスター管理対象ノードのノード・エージェントを始動します。

    以下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:** startNode.sh
    - Windows **Windows プラットフォームの場合:** startNode.bat

    startNode コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startNode コマンド』のトピックを参照してください。

27. **マイグレーション・ターゲットの非クラスター管理対象ノード・サーバーを始動します。**

    マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、startServer コマンドを使用して、マイグレーション・ターゲットの非クラスター管理対象ノードのターゲット・サーバーを始動します。

    以下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:** startServer.sh *server_name*
    - Windows **Windows プラットフォームの場合:** startServer.bat *server_name*

    startServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

28. **クラスターをマイグレーションします。**

マイグレーションする必要がある Network Deployment 環境の各クラスターについて、ステップ 27 から 45 を繰り返します。

クラスターに対応するサーバーを含むノードを、グループ A およびグループ B の 2 つのおおよそ等しいサイズのグループに分割します。グループ B ノードは、消費者の要求へのサービスを続行し、グループ A ノードはオフラインになりマイグレーションされます。グループ A ノードがマイグレーションされると、すべてのノードが停止し、そのクラスター用に構成されたデータベースがマイグレーションされます。そして、マイグレーション済みのグループ A ノードが始動されて、消費者の要求へのサービスを開始できます。次にグループ B ノードがマイグレーションされ始動されます。 マイグレーションを 2 つのグループのノードに分割することで、製品データベースをマイグレーションするために、クラスターがダウンする必要のある時間が最小化されます。

29. **グループ A のクラスター管理対象ノードのマイグレーション・ソース・サーバーを停止します。**

    グループ A の一部としてマイグレーションするクラスター管理対象ノードに関連付けられた各サーバーについて、このステップを繰り返します。

    マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopServer コマンドを使用して、マイグレーション・ソース・サーバーを停止します。

    以下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:** stopServer.sh *server_name* -username *user_name* -password *password*
    - Windows **Windows プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

    プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

    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、これらのパラメーターは、Windows サービスがサーバー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。

    プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

    stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

30. **グループ A のクラスター管理対象ノードのマイグレーション・ソース・ノード・エージェントを停止します。**

グループ A の一部としてマイグレーションするクラスター管理対象ノードに関連付けられた各ノード・エージェントについて、このステップを繰り返します。

マイグレーションによって影響を受ける各ノード・エージェントについてこのステップを繰り返します。

マイグレーション・ソース・システムの *profile_root*/bin から、 stopNode コマンドを使用して、マイグレーション・ソースのノード・エージェントを停止します。

以下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:** stopNode.sh -username *user_name* -password *password*
- Windows **Windows プラットフォームの場合:** stopNode.bat -username *user_name* -password *password*

プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

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、これらのパラメーターは、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。

プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

stopNode コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopNode コマンド』のトピックを参照してください。

31. **グループ A のマイグレーション・ソース・プロファイルをバックアップします。**

グループ A でマイグレーションされる各プロファイルについて、このステップを繰り返します。

backupConfig コマンドを使用して、非クラスター管理対象ノード上のプロファイル構成をバックアップします。

以下の構文を使用して、profile1 という名前のプロファイルを /ProfileBackups/profile1.zip にバックアップします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1
- Windows **Windows プラットフォームの場合:** backupConfig.bat c:¥ProfileBackups¥profile1.zip -profileName profile1

backupConfig コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

32. **グループ A の管理対象ノードをマイグレーションします。**

このステップは、クラスター内のグループ A の各管理対象ノードについて繰り返す必要があります。

BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、クラスター管理対象ノード・ソース・プロファイルをマイグレーションできます。BPM プロファイル・マイグレーション・ウィザードを使用するには、クラスター管理対象ノード・プロファイルを含むシステムで『BPM マイグレーション・ウィザードを使用したプロファイルのマイグレーション』サブプロシージャーに従います。BPM マイグレーション・コマンド行ユーティリティーを使用するには、クラスター管理対象ノード・プロファイルを含むシステムで『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』サブプロシージャーに従います。

注: WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合、『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順を使用する必要があります。

33. **グループ B のクラスター管理対象ノードのマイグレーション・ソース・サーバーを停止します。**

グループ B の一部としてマイグレーションするクラスター管理対象ノードに関連付けられた各サーバーについて、このステップを繰り返します。

マイグレーション・ソース・システムの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから stopServer コマンドを使用して、マイグレーション・ソース・サーバーを停止します。

以下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:** stopServer.sh *server_name* -username *user_name* -password *password*
- Windows **Windows プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

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、これらのパラメーターは、Windows サービスがサーバー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。

プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

34. **グループ B のクラスター管理対象ノードのマイグレーション・ソース・ノード・エージェントを停止します。**

    グループ B の一部としてマイグレーションするクラスター管理対象ノードに関連付けられた各ノード・エージェントについて、このステップを繰り返します。

    マイグレーションによって影響を受ける各ノード・エージェントについてこのステップを繰り返します。

    マイグレーション・ソース・システムの *profile_root*/bin から、 stopNode コマンドを使用して、マイグレーション・ソースのノード・エージェントを停止します。

    以下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:** stopNode.sh -username *user_name* -password *password*
    - Windows **Windows プラットフォームの場合:** stopNode.bat -username *user_name* -password *password*

    プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

    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、これらのパラメーターは、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。

    プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

    stopNode コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopNode コマンド』のトピックを参照してください。

35. **クラスターをマイグレーションします。**

    デプロイメント・マネージャーを含むシステムで、 *install_root*/bin ディレクトリーから BPMMigrateCluster コマンドを使用して、クラスター・スコープ・プロファイルをマイグレーションします。

    以下の構文を使用して、/MigrationSnapshots/ProcServer620 ディレクトリーにコピーされた dmgrProfile という名前のデプロイメント・マネージャー・プロファイルと共に、applicationCluster1 という名前のクラスターをマイグレーションします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:**
  `BPMMigrateCluster.sh /MigrationSnapshots/ProcServer620 applicationCluster1 dmgrProfile`
- Windows **Windows プラットフォームの場合:** `BPMMigrateCluster.bat c:¥MigrationSnapshots¥ProcServer620 applicationCluster1 dmgrProfile`

`BPMMigrateCluster` コマンドについて詳しくは、『BPMMigrateCluster コマンド』のトピックを参照してください。

36. **すべてのクラスター・ノードについて同期を有効にします。**

    ターゲット・デプロイメント・マネージャーで管理コンソールを使用して、クラスター内のすべてのノード (グループ A およびグループ B の両方) について同期を有効にします。これを行うには、以下の手順を使用します。

    a. WebSphere Application Server 管理コンソールから、「**システム管理**」 → 「**ノード・エージェント**」を選択します。
    b. ノードのノード・エージェントをクリックします。
    c. 「**ファイル同期化サービス**」をクリックします。
    d. 「**サーバー始動時にサービスを使用可能にする**」、「**自動同期**」、および「**始動同期**」を選択します。
    e. 「**適用**」をクリックし、次に「**OK**」をクリックして構成変更を保存します。

37. **グループ A のマイグレーション・ソース・プロファイルをバックアップします。**

    グループ A でマイグレーションする各プロファイルに対して、このステップを繰り返します。このバックアップは、次のステップで syncNode コマンドの実行が失敗した場合に必要となります。 syncNode に関する問題が解決したら、バックアップをリストアしてから syncNode コマンドを再度実行します。

    `backupConfig` コマンドを使用して、非クラスター管理対象ノード上のプロファイル構成をバックアップします。

    以下の構文を使用して、profile1 という名前のプロファイルを /ProfileBackups/profile1.zip にバックアップします。

    - Linux UNIX **Linux および UNIX プラットフォームの場合:**
      `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
    - Windows **Windows プラットフォームの場合:** `backupConfig.bat c:¥ProfileBackups¥profile1.zip -profileName profile1`

    `backupConfig` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

38. **すべてのグループ A のノードを同期します。**

    このステップは、クラスター内のグループ A の各クラスター管理対象ノードについて繰り返す必要があります。

マイグレーション・ターゲット・プロファイルの *profile_root*/bin ディレクトリーから、またはターゲット・プロファイルのファースト・ステップ・コンソールから、 syncNode コマンドを使用して、ターゲット・デプロイメント・マネージャーとノードを同期します。

以下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:** syncNode.sh *deployment_manager_machine_name_or_ip_address deployment_manager_port_no*
- Windows **Windows プラットフォームの場合:** syncNode.bat *deployment_manager_machine_name_or_ip_address deployment_manager_port_no*

syncNode コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『syncNode コマンド』のトピックを参照してください。

39. **クラスター用に構成されたクラスター・スコープ製品データベースをバックアップします。**

ご使用のデータベースのドキュメンテーションに応じて、クラスター用に構成された以下の製品データベースをバックアップします。

- Business Process Choreographer データベース
- Business Space データベース
- Common Event Infrastructure データベース
- メッセージング・エンジン・データベース

40. **Cloudscape または Derby クラスター・スコープ・データベースを更新します。**

クラスター用に構成された Cloudscape または Derby データベースを使用している場合、サポートされるバージョンを使用していることを確認する必要があります。 Cloudscape を Derby にマイグレーションする方法については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』を参照してください。

41. **クラスター・スコープ製品データベースをマイグレーションします。**

以下の手順を使用して、クラスターについて構成された各製品データベースをマイグレーションします。

a. 以下の条件のいずれかが該当する場合は、『Business Process Choreographer データベース・スキーマのアップグレード』の手順を使用して Business Process Choreographer データベース・スキーマを手動でアップグレードします。

   - Business Process Choreographer データベースにデフォルトのテーブル・スペースを使用しなかった。Business Process Choreographer のサンプル構成を使用したか、サンプル SQL スクリプトで指定されたデフォルト・テーブル・スペース内のすべてのデータベース・オブジェクトを作成済みの場

合、データベースではデフォルト・テーブル・スペースを使用します。通常、テスト環境がこれに該当します。

- BPEDB データ・ソース用に構成されたデータベース・ユーザーが、テーブルの作成と変更、および索引とビューの作成と除去の操作のすべてを実行することを許可されているわけではなく、テーブル SCHEMA_VERSION の場合は、照会、更新、削除、および挿入のすべての実行を許可されているわけではない。

b. マイグレーション元のソース・バージョンが 6.0.2、6.1.0、または 6.1.2 の場合は、『Business Process Choreographer データベース・データのマイグレーション』の手順に従って、Business Process Choreographer データベース・データをマイグレーションします。

c. 『Business Space データベース・スキーマのマイグレーション』の手順を使用して Business Space データベース・スキーマをマイグレーションします。

d. 『Business Space データベース・データのマイグレーション』の手順を使用して Business Space データベース・データをマイグレーションします。

e. オプション: ご使用の環境に合わせて、必要に応じてメッセージング・エンジン・データベースをマイグレーションします。 メッセージング・エンジンのマイグレーションが必要な場合とその方法について詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターで『データ・ストアに基づくメッセージング・エンジンのマイグレーション』を参照してください。

42. **オプション: ビジネス・ルール・マネージャーをマイグレーションします。**

ビジネス・ルール・マネージャーは、セル内の最後のノードがマイグレーションされるときに自動的にマイグレーションされますが、マイグレーションされたクラスターにビジネス・ルール・マネージャーが含まれている場合、手動でマイグレーションできます。

クラスター cluster1 のビジネス・ルール・マネージャーを手動でマイグレーションするには、以下のコマンドを使用します。

```
wsadmin -f installBRManager.jacl -cl cluster1
```

installBRManager コマンドについて詳しくは、『installBRManager コマンド』のトピックを参照してください。

43. **グループ A のマイグレーション・ターゲット・ノード・エージェントを始動します。**

このステップは、クラスター内のグループ A の各クラスター管理対象ノードについて繰り返す必要があります。

マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、startNode コマンドを使用して、マイグレーション・ターゲット・ノード・エージェントを始動します。

以下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:** `startNode.sh`
- Windows **Windows プラットフォームの場合:** `startNode.bat`

`startNode` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startNode コマンド』のトピックを参照してください。

44. **グループ A のマイグレーション・ターゲット・サーバーを始動します。**

このステップは、クラスター内のグループ A のクラスター管理対象ノードに関連付けられた各サーバーついて繰り返す必要があります。

マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、`startServer` コマンドを使用して、マイグレーション・ターゲット・サーバーを始動します。

以下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:** `startServer.sh` *server_name*
- Windows **Windows プラットフォームの場合:** `startServer.bat` *server_name*

`startServer` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

45. **グループ B のマイグレーション・ソース・プロファイルをバックアップします。**

グループ B でマイグレーションされる各プロファイルについて、このステップを繰り返します。

`backupConfig` コマンドを使用して、非クラスター管理対象ノード上のプロファイル構成をバックアップします。

以下の構文を使用して、profile1 という名前のプロファイルを /ProfileBackups/profile1.zip にバックアップします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** `backupConfig.sh /ProfileBackups/profile1.zip -profileName profile1`
- Windows **Windows プラットフォームの場合:** `backupConfig.bat c:¥ProfileBackups¥profile1.zip -profileName profile1`

`backupConfig` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

46. **グループ B の管理対象ノードをマイグレーションします。**

このステップは、クラスター内のグループ B の各管理対象ノードについて繰り返す必要があります。

BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、管理対象ノード・ソース・プロファイルをマイグレーションできます。

BPM プロファイル・マイグレーション・ウィザードまたはBPM マイグレーション・コマンド行ユーティリティーを使用して、クラスター管理対象ノード・ソース・プロファイルをマイグレーションできます。BPM プロファイル・マイグレーション・ウィザードを使用するには、クラスター管理対象ノード・プロファイルを含むシステムで『BPM マイグレーション・ウィザードを使用したプロファイルのマイグレーション』サブプロシージャーに従います。BPM マイグレーション・コマンド行ユーティリティーを使用するには、クラスター管理対象ノード・プロファイルを含むシステムで『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』サブプロシージャーに従います。

注: WebSphere Enterprise Service Bus バージョン 6.0.2 からマイグレーションする場合、『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』の手順を使用する必要があります。

47. **グループ B のマイグレーション・ターゲット・ノード・エージェントを始動します。**

    このステップは、クラスター内のグループ B の各クラスター管理対象ノードについて繰り返す必要があります。

    マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、startNode コマンドを使用して、マイグレーション・ターゲット・ノード・エージェントを始動します。

    以下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:** startNode.sh
    - Windows **Windows プラットフォームの場合:** startNode.bat

    startNode コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startNode コマンド』のトピックを参照してください。

48. **グループ B のマイグレーション・ターゲット・サーバーを始動します。**

    このステップは、クラスター内のグループ B のクラスター管理対象ノードに関連付けられた各サーバーついて繰り返す必要があります。

    マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、startServer コマンドを使用して、マイグレーション・ターゲット・サーバーを始動します。

以下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:** `startServer.sh server_name`
- Windows **Windows プラットフォームの場合:** `startServer.bat server_name`

`startServer` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

49. **オプション: ソース・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。**

マイグレーションが完了した後は、マイグレーション・ソース・デプロイメント・マネージャーをアンインストールできます。

50. **互換モードを除去します。**

互換性オプション (デフォルト) を選択していて、すべてのノードがターゲット・バージョンに完全にマイグレーションされたら、デプロイメント上で *install_root*/bin ディレクトリーから `convertScriptCompatibility` スクリプトを実行して、互換性を除去します。

以下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:** `convertScriptCompatibility.sh`
- Windows **Windows プラットフォームの場合:** `convertScriptCompatibility.bat`

`convertScriptCompatibility` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『convertScriptCompatibility』のトピックを参照してください。

タスクの結果

Network Deployment 環境はターゲット・バージョンにマイグレーションされます。

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。手順については、109 ページの『マイグレーションの検査』を参照してください。

ランタイム・マイグレーションのサブ手順

バージョン間マイグレーションを行うプロセスの一部として、ランタイム・マイグレーションのサブ手順を使用します。

BPM プロファイル・マイグレーション・ウィザードを使用したプロファイルのマイグレーション

BPM プロファイル・マイグレーション・ウィザードは、プロファイルをマイグレーションするプロセスの手引きを行うグラフィカル・ユーザー・インターフェース

(GUI) です。プロファイルのマイグレーションは、スタンドアロン環境またはネットワーク・デプロイメント環境のマイグレーションに必要な一連のステップの一部です。

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。この手順は、スタンドアロン環境とネットワーク・デプロイメント環境のどちらをマイグレーションするかによって異なります。詳細については、『スタンドアロン環境のマイグレーション』、『フル・ダウン時間での Network Deployment 環境のマイグレーション』、『最小限のダウン時間での Network Deployment 環境のマイグレーション』の各トピックを参照してください。

このタスクについて

ここでは、BPM プロファイル・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するための手順を説明します。

手順

1. **マイグレーション・ウィザードを起動します。**

   `BPMMigrate` コマンドを実行して、*target_install_root*`/bin` ディレクトリーからマイグレーション・ウィザードを呼び出します。

   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:** `BPMMigrate.sh`
   - Windows **Windows プラットフォームの場合:** `BPMMigrate.bat`

   `BPMMigrate` コマンドについて詳しくは、BPMMigrate コマンド行ユーティリティーのトピックを参照してください。

2. **ウェルカム画面の説明を確認します。**

   Business Process Management プロファイル・マイグレーション・ウィザードのウェルカム画面に表示されている情報を確認し、マイグレーション・プロセスについて理解したら「**次へ**」をクリックします。

3. **ウィザード・マイグレーションのタイプ (「標準」または「カスタム」) を選択します。**

   「標準マイグレーションまたはカスタム・マイグレーションの選択」画面で、標準またはカスタムのウィザード・マイグレーションのどちらかを選択して「**次へ**」をクリックします。

   - 「**標準**」を選択した場合、デフォルトの構成設定で BPM プロファイルがマイグレーションされます。
   - 「**カスタム**」を選択した場合、構成設定をカスタマイズすることができます。

   デフォルトの構成設定は以下のとおりです。

- **スナップショット・ディレクトリー**:
  - Linux UNIX /MigrationSnapshots/*source_install_root*
  - Windows C:¥MigrationSnapshots¥*source_install_root*
- **ターゲット・プロファイル名**: ターゲット・プロファイル名のデフォルトは、ソース・プロファイル名です。
- **ターゲット・プロファイル・ディレクトリー**: ターゲット・プロファイル・ディレクトリーのデフォルトは、*target_install_directory*/profiles/*source_profile_name* です。この *source_profile_name* は、ソース・プロファイル名です。
- **ポート値割り当て**: ソース・プロファイルのポート割り当てと同じ値です。
- **スクリプト互換性 (デプロイメント・マネージャー・プロファイルのみ)**: これを True に設定すると、ソース・プロファイルのスクリプトをマイグレーション後も使用することができます。
- **アプリケーション・ディレクトリーの設定 (デプロイメント・マネージャー・プロファイルのみ)**: ターゲット・プロファイルのデフォルト・ターゲット・インストール・ディレクトリーです。

4. **ソース・インストールを選択します。**

   「マイグレーションのソースとして使用するインストールの選択」画面で、検出された BPM 製品のリストからソース・インストール・ディレクトリーを選択するか、または「**参照**」を選択して、検出されなかった BPM 製品のインストール・ディレクトリーを選択して「**次へ**」をクリックします。

   制約事項: WebSphere ESB バージョン 6.0.2.x からマイグレーションする場合は、83 ページの『BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション』 の手順を使用する必要があります。

5. **ソース・プロファイルを選択します。**

   「マイグレーションのソースとして使用するソース・プロファイルの選択」画面で、リストからソース・プロファイルを選択します。プロファイルのセキュリティー設定が有効になっている場合はユーザー名とパスワードを入力して、「**次へ**」をクリックします。

6. **カスタム・マイグレーションの場合はカスタム設定を定義し、標準マイグレーションの場合は検証ステップに進みます。**

   注: ステップ 3 (79 ページ) で「**標準**」を選択した場合は、ステップ 7 (82 ページ) に進みます。

   ステップ 3 (79 ページ) で 「**カスタム**」を選択した場合は、以下のステップを実行します。

   a. **スナップショット・ディレクトリーを選択します。**

      「ソース・プロファイルに使用するスナップショット・ディレクトリーを入力または参照」画面で、デフォルトのスナップショット・ディレクトリーのままにするか、「**参照**」をクリックして新しいスナップショット・ディレクトリーにナビゲートし、「**次へ**」をクリックします。

b. **ターゲット・プロファイル名とターゲット・プロファイル・ディレクトリーを指定します。**

「ターゲット・プロファイル名およびディレクトリーの選択」 画面で、 デフォルトのターゲット・プロファイル名とディレクトリーをそのまま使用するか、「**ターゲット・プロファイル名**」フィールドと「**ターゲット・プロファイル・ディレクトリー**」フィールドで新しいターゲット・プロファイル名とディレクトリーを入力して「**次へ**」をクリックします。

c. **アプリケーションのマイグレーション設定を選択します。**

注: この画面は、デプロイメント・マネージャー・プロファイルをマイグレーションする場合のみ表示されます。

「アプリケーション・マイグレーション設定の選択」画面で、マイグレーションしたアプリケーションの格納先を指定して「**次へ**」をクリックします。デフォルトの場合、「**ターゲット・インストールのデフォルト・ディレクトリーにアプリケーションをインストール**」が選択されます。

- **ターゲット・インストール環境のデフォルト・ディレクトリーにアプリケーションをインストールします。**
- **現在のアプリケーション・インストール・ディレクトリーを保持します。**

  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用されます。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 コマンドを呼び出すときに、以下のサポートが使用できません。
    - JSP のプリコンパイル
    - バイナリー構成の使用
    - EJB のデプロイ
  – 後になって、以前のインストールを管理する (例えば、アンインストール) ときにこれらのロケーションからアプリケーションを削除すると、マイグレーションされたアプリケーションを意図せずに失ってしまう危険性があります。

d. **ポートのマイグレーション設定を選択します。**

注: この画面は、スタンドアロン・プロファイルをマイグレーションする場合のみ表示されます。

「ポート・マイグレーション設定の選択」画面で、ターゲット・プロファイル・ポート値の割り当てに以下のいずれかのオプションを選択して「**次へ**」をクリックします。

- **同じポート割り当てをソース・プロファイルとして使用します。**
- **ターゲット・プロファイルを使用して作成されたポートをオーバーライドしません。**

- **ターゲット・プロファイルに対して、以下のポート番号で始まる使用可能なポートを割り当てます。**

  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。

注: デフォルトの場合、「**同じポート割り当てをソース・プロファイルとして使用します**」が選択されます。

e. **スクリプトの互換設定を選択します。**

注: この画面は、デプロイメント・マネージャー・プロファイルをマイグレーションする場合のみ表示されます。

「スクリプト互換性設定の選択」画面で、「**ターゲット・インストールで使用するためにソース・プロファイル管理スクリプトを使用可能にします**」ボックスを選択またはクリアして「**次へ**」をクリックします。 このオプションを選択すると、任意指定の WebSphere Application Server `-scriptCompatibility` パラメーターが True に設定されます。このパラメーターを true に設定すると、次のような WebSphere Application Server バージョン 6.x 構成定義が作成されるマイグレーションが可能になります。

- トランスポート
- ProcessDef
- バージョン 6.x SSL

これらは以下の WebSphere Application Server バージョン 7.0 構成定義の代替となります。

- チャネル
- ProcessDefs
- バージョン 7.0 SSL

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、サード・パーティーの構成 API を使用してバージョン 6.x の構成定義を作成または変更する既存の **wsadmin** スクリプトまたはプログラムがある場合は、このオプションを選択します。

注: これは、環境内のノードすべてが WebSphere Application Server バージョン 7.0 レベルになるまで一時的な遷移を提供することを意味します。すべてのノードが バージョン 7.0 になったら、以下の操作を実行する必要があります。

1) バージョン 7.0 の設定すべてを使用するように管理スクリプトを変更します。
2) convertScriptCompatibility コマンドを使用して構成を変換し、バージョン 7.0 すべてに一致させます。

   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 BPMMigrateProfile コマンドを使用してください。

7. **マイグレーション・ウィザードの選択内容を確認します。**

「プロファイル・マイグレーションの要約」画面で、ウィザード上で選択したマイグレーション内容を確認し、「**次へ**」をクリックしてマイグレーションを開始します。

8. **マイグレーションの状況をモニターします。**

   「マイグレーション実行」画面に、プロファイルのマイグレーション状況が表示されます。この画面で、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。

9. **失敗した場合は、マイグレーションをやり直します。**

   ソース・プロファイルのコピー中、ターゲット・プロファイルの作成中、またはソース・プロファイルをターゲット・プロファイルにマイグレーションする際にプロファイルのマイグレーションが失敗した場合は、以下の手順を実行してマイグレーションをやり直してください。

   a. 失敗の根本原因を修正します。
   b. マイグレーションの失敗によって生成された以下の成果物を削除します。
      - スナップショット・ディレクトリー
      - ターゲット・プロファイル (manageprofiles コマンド行ユーティリティーを使用)。

      注: デプロイメント・マネージャー・プロファイルがマイグレーションされ、ソース・デプロイメント・マネージャーが無効になった場合は、`migrationDisablementReversal` コマンドを実行してソース・デプロイメント・マネージャーをもう一度有効に設定し、マイグレーションをロールバックする必要があります。ただし、プロファイルをもう一度マイグレーションする場合は、デプロイメント・マネージャーを再度有効にする必要はありません。
   c. 「戻る」ボタンを押すかウィザードを再開して、マイグレーションを実行します。

10. マイグレーションが正常に完了した場合は「**次へ**」をクリックし、次に「**完了**」をクリックしてウィザードを終了します。

タスクの結果

プロファイルは、BPM 製品の以前のバージョンから WebSphere Process Server バージョン 7.0 に移行されました。

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。手順については、109 ページの『マイグレーションの検査』を参照してください。

BPM コマンド行ユーティリティーを使用したプロファイルのマイグレーション

コマンド行ユーティリティーを使用してプロファイルをマイグレーションするには、このサブ手順を使用します。

始める前に

『スタンドアロン環境のマイグレーション』、『完全ダウンタイムでのネットワーク・デプロイメント環境のマイグレーション』、『最小ダウンタイムでのネットワーク・デプロイメント環境のマイグレーション』の各トピックを参照してください。

このタスクについて

コマンド行ユーティリティーを使用してプロファイルをマイグレーションするには、以下の手順を実行します。

手順

1. **ソース・プロファイルのコピーを作成します。**

   install_root/bin ディレクトリーから BPMSnapshotSourceProfile コマンドを使用して、ターゲット・プロファイルにマイグレーションするソース・プロファイル内の構成ファイルのコピーを作成します。ユーザー指定のスナップショット・ディレクトリーは、ソースまたはターゲット製品のインストール・ディレクトリー内には置かないでください。これらのディレクトリーを、スナップショット・ディレクトリー内の構成ファイルに影響を与えずに、後で必要に応じて除去できるようにするためです。

   以下の構文を使用して、ProcServer620 インストール・ルート・ディレクトリーにあるソース・プロファイル sourceProfile1 を /MigrationSnapshots/ProcServer620 snapshot ディレクトリーにコピーします。

   - Linux UNIX **Linux および UNIX プラットフォームの場合:**
     ```
     BPMSnapshotSourceProfile.sh /opt/ibm/WebSphere/ProcServer620
     sourceProfile1 /MigrationSnapshots/ProcServer620
     ```
   - Windows **Windows プラットフォームの場合:**
     ```
     BPMSnapshotSourceProfile.bat
     "C:¥Program Files¥IBM¥WebSphere¥ProcServer620" sourceProfile1
     c:¥MigrationSnapshots¥ProcServer620
     ```

   BPMSnapshotSourceProfile コマンドについて詳しくは、『BPMSnapshotSourceProfile コマンド』のトピックを参照してください。

2. **ターゲット・プロファイルを作成します。**

   WebSphere ESB バージョン 6.0.2 **以外の**製品とソースの組み合わせからプロファイルをマイグレーションする場合は、BPMCreateTargetProfile コマンドを *install_root*/bin ディレクトリーから使用してターゲット・プロファイルを作成します。このプロファイルは、BPMMigrateProfile コマンドを使用してソース・プロファイルを新しいターゲット・プロファイルにマイグレーションしないと使用可能になりません。

   以下の構文を使用して、/MigrationSnapshots/ProcServer620 スナップショット・ディレクトリーにコピーされたソース・プロファイル sourceProfile1 を使用してマイグレーション用のターゲット・プロファイルを作成します。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** `BPMCreateTargetProfile.sh /MigrationSnapshots/ProcServer620 sourceProfile1`
- Windows **Windows プラットフォームの場合:** `BPMCreateTargetProfile.bat "C:¥MigrationSnapshots¥ProcServer620" sourceProfile1`

`BPMCreateTargetProfile` コマンドについて詳しくは、『BPMCreateTargetProfile コマンド』のトピックを参照してください。

WebSphere ESB プロファイルを 6.0.2 からマイグレーションする場合は、プロファイル管理ツールまたは manageprofiles コマンド行ユーティリティーを使用して、ターゲット・マイグレーション・プロファイルを作成する必要があります。詳しくは、『プロファイルの作成』を参照してください。

3. **ソース・プロファイルをターゲット・プロファイルにマイグレーションします。**

`BPMMigrateProfile` コマンドを使用して、ソース・プロファイルをターゲット・プロファイルにマイグレーションします。このコマンドは、`BPMSnapshotSourceProfile` コマンドで指定されたスナップショット・ディレクトリーから構成情報を読み取って、それをターゲット・プロファイルにマイグレーションします。

以下の構文を使用して、`/MigrationSnapshots/ProcServer620` ディレクトリーにコピーされたソース・プロファイル sourceProfile1 をターゲット・プロファイルにマイグレーションします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** `BPMMigrateProfile.sh /MigrationSnapshots/ProcServer620 sourceProfile1`
- Windows **Windows プラットフォームの場合:** `BPMMigrateProfile.bat C:¥MigrationSnapshots¥ProcServer620 sourceProfile1`

ソース・プロファイルでセキュリティーが有効になっている場合は、 `-username` パラメーターと `-password` パラメーターが必須であり、指定するユーザー名がオペレーターまたは管理者ロールのメンバーである必要があります。

Windows Windows オペレーティング・システムの場合、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` パラメーターと `-password` パラメーターを指定する必要はありません。この場合、これらのパラメーターは、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。

`BPMMigrateProfile` コマンドについて詳しくは、『BPMMigrateProfile コマンド』のトピックを参照してください。

4. **マイグレーション状況を確認します。**

`BPMMigrationStatus` コマンドを使用して、マイグレーションの現在の状態を検証します。

以下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:** BPMMigrationStatus.sh
- Windows **Windows プラットフォームの場合:** BPMMigrationStatus.bat

BPMMigrationStatus コマンドについて詳しくは、『BPMMigrationStatus コマンド』のトピックを参照してください。

タスクの結果

旧バージョンの WebSphere Process Server から WebSphere Process Server バージョン 7.0 にプロファイルがマイグレーションされます。

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。手順については、109 ページの『マイグレーションの検査』を参照してください。

リモート・システムへのスタンドアロン・プロファイルのマイグレーション

プロファイルをリモート・システムにマイグレーションするための、スタンドアロン・サーバー・マイグレーションのサブプロシージャーです。

始める前に

スタンドアロン・サーバー・プロファイルのマイグレーションに関するトピックを参照してください。

このタスクについて

この手順の以下のステップに従って、プロファイルをリモート・システムにマイグレーションします。

手順

1. **ターゲット・システムでデフォルト・プロファイルを作成します。** マイグレーション・ターゲット・システムで、デフォルト・プロファイルを作成します。トピック『プロファイルの作成』の説明に従って、デフォルトのターゲット・プロファイルを作成します。
2. **リモート・マイグレーション・ユーティリティー・イメージを作成します。**

   マイグレーション・ターゲット・システム、またはバージョン 7.0 がインストール済みのあらゆるシステムで、*install_root*/bin ディレクトリーから BPMCreateRemoteMigrationUtilities コマンドを使用して、リモート・マイグレーション・イメージを作成します。

   以下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:** BPMCreateRemoteMigrationUtilities.sh remoteMigrationUtilities.gzip
   - Windows **Windows プラットフォームの場合:** BPMCreateRemoteMigrationUtilities.bat remoteMigrationUtilities.zip

BPMCreateRemoteMigrationUtilities コマンドについて詳しくは、『BPMCreateRemoteMigrationUtilities コマンド』のトピックを参照してください。

3. **リモート・マイグレーション・ユーティリティーをソース・システムにコピーします。**

   FTP、RCP、またはその他のメカニズムを使用して、リモート・マイグレーション・ユーティリティーを、ターゲット・システムからソース・システムにコピーし、ソース・システム上のリモート・マイグレーション・ユーティリティーを独自の固有ディレクトリーに unzip します。

4. **マイグレーション・ソース・プロファイルをスナップショットします。**

   マイグレーション・ソース・システムで、リモート・マイグレーション・ユーティリティー bin ディレクトリーから BPMSnapshotSourceProfile コマンドを使用して、マイグレーションされる構成ファイルを含むスナップショット・ディレクトリーを作成します。

   以下の構文を使用して、ProcServer620 インストール・ルート・ディレクトリーにある sourceProfile1 という名前のソース・プロファイルのスナップショットを、 /MigrationSnapshots/ProcServer620 スナップショット・ディレクトリーに作成します。

   - Linux UNIX **Linux および UNIX プラットフォームの場合:**
     ```
     BPMSnapshotSourceProfile.sh /opt/ibm/WebSphere/ProcServer620
     sourceProfile1 /MigrationSnapshots/ProcServer620
     ```
   - Windows **Windows プラットフォームの場合:**
     ```
     BPMSnapshotSourceProfile.bat
     "C:¥Program Files¥IBM¥WebSphere¥ProcServer620" sourceProfile1
     c:¥MigrationSnapshots¥ProcServer620
     ```

   BPMSnapshotSourceProfile コマンドについて詳しくは、『BPMSnapshotSourceProfile コマンド』のトピックを参照してください。

5. **マイグレーション・ソース・スナップショット・ディレクトリーをマイグレーション・ターゲット・システムにコピーします。**

   ソース・スナップショット・ディレクトリーの zip を作成し、ターゲット・システム上の同じディレクトリーにコピーしてから、unzip します。

6. **ターゲット・プロファイルを作成します。**

   WebSphere ESB バージョン 6.0.2 **以外の**あらゆる製品およびソースの組み合わせからプロファイルをマイグレーションする場合、 BPMCreateTargetProfile コマンドを使用してターゲット・プロファイルを作成します。このプロファイルは、BPMMigrateProfile コマンドを使用してソース・プロファイルを新しいターゲット・プロファイルにマイグレーションしないと使用可能になりません。

   以下の構文を使用して、/MigrationSnapshots/ProcServer620 スナップショット・ディレクトリーにコピーされた sourceProfile1 という名前のソース・プロファイルを使用し、マイグレーション用のターゲット・プロファイルを作成します。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** `BPMCreateTargetProfile.sh /MigrationSnapshots/ProcServer620 sourceProfile1`
- Windows **Windows プラットフォームの場合:** `BPMCreateTargetProfile.bat -remoteMigration true C:¥MigrationSnapshots¥ProcServer620 sourceProfile1`

`BPMCreateTargetProfile` コマンドについて詳しくは、『BPMCreateTargetProfile コマンド』のトピックを参照してください。

WebSphere ESB プロファイルを 6.0.2 からマイグレーションする場合は、プロファイル管理ツールまたは manageprofiles コマンド行ユーティリティーを使用して、ターゲット・マイグレーション・プロファイルを作成する必要があります。

7. **ソース・プロファイルをターゲット・プロファイルにマイグレーションします。**

`BPMMigrateProfile` コマンドを使用して、ソース・プロファイルをターゲット・プロファイルにマイグレーションします。 このコマンドは、`BPMSnapshotSourceProfile` コマンドによって指定されたスナップショット・ディレクトリーから構成情報を読み取り、その情報をターゲット・システムにコピーして、ターゲット・プロファイルにマイグレーションします。

以下の構文を使用して、`/MigrationSnapshots/ProcServer620` ディレクトリーにコピーされた sourceProfile1 という名前のソース・プロファイルをターゲット・プロファイルにマイグレーションします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** `BPMMigrateProfile.sh /MigrationSnapshots/ProcServer620 sourceProfile1`
- Windows **Windows プラットフォームの場合:** `BPMMigrateProfile.bat C:¥MigrationSnapshots¥ProcServer620 sourceProfile1`

ソース・プロファイルでセキュリティーが有効になっていない場合、ユーザー名およびパスワードのパラメーターは必要ありません。有効な場合、入力されたユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

Windows オペレーティング・システムの場合、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 パラメーターと -password パラメーターを指定する必要はありません。この場合、これらのパラメーターは、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。

`BPMMigrateProfile` コマンドについて詳しくは、BPMMigrateProfile コマンド行ユーティリティーのトピックを参照してください。

8. **マイグレーション状況を確認します。**

`BPMMigrationStatus` コマンドを使用して、マイグレーションの現在の状態を検証します。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** `BPMMigrationStatus.sh`
- Windows **Windows プラットフォームの場合:** `BPMMigrationStatus.bat`

`BPMMigrationStatus` コマンドについて詳しくは、『BPMMigrationStatus コマンド』のトピックを参照してください。

9. ファイル・システムのプロファイル・ディレクトリーの下で、古いホスト名の値が使用されている構成をスキャンします。古いホスト名が使用されている構成を分析し、古いホスト名のマシンにデータベースがまだ存在しているなどの理由で古いホスト名が必要でない限り、新しいホスト名に置き換えます。

タスクの結果

旧バージョンの BPM 製品からリモート・システム上の WebSphere Process Server バージョン 7.0 にプロファイルがマイグレーションされます。

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。手順については、109 ページの『マイグレーションの検査』を参照してください。

オペレーティング・システムのアップグレード時にスタンドアロン・サーバーをマイグレーション

オペレーティング・システムをアップグレード中のシステムで、プロファイルをマイグレーションするための、スタンドアロン・サーバー・マイグレーションのサブプロシージャーです。

始める前に

スタンドアロン・サーバー・プロファイルのマイグレーションに関するトピックを参照してください。

このタスクについて

この手順のステップに従って、オペレーティング・システムをアップグレード中のシステム上のプロファイルをマイグレーションします。

手順

1. **リモート・マイグレーション・ユーティリティー・イメージを作成します。**

   バージョン 7.0 がインストール済みのあらゆるシステムの場合、*install_root*/bin ディレクトリーから `BPMCreateRemoteMigrationUtilities` コマンドを使用して、リモート・マイグレーション・ユーティリティー・イメージを作成します。

   以下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:** `BPMCreateRemoteMigrationUtilities.sh remoteMigrationUtilities.gzip`

- Windows **Windows プラットフォームの場合:**
  BPMCreateRemoteMigrationUtilities.bat remoteMigrationUtilities.zip

BPMCreateRemoteMigrationUtilities コマンドについて詳しくは、『BPMCreateRemoteMigrationUtilities コマンド』のトピックを参照してください。

2. **リモート・マイグレーション・ユーティリティーをソース・システムにコピーします。**

   FTP、RCP、またはその他のメカニズムを使用して、リモート・マイグレーション・ユーティリティーを、ターゲット・システムからソース・システムにコピーし、ソース・システム上のリモート・マイグレーション・ユーティリティーを独自の固有ディレクトリーに unzip します。

3. **マイグレーション・ソース・プロファイルをスナップショットします。**

   マイグレーション・ソース・システムで、リモート・マイグレーション・ユーティリティー bin ディレクトリーから BPMSnapshotSourceProfile コマンドを使用して、マイグレーションされる構成ファイルを含むスナップショット・ディレクトリーを作成します。

   以下の構文を使用して、ProcServer620 インストール・ルート・ディレクトリーにある sourceProfile1 という名前のソース・プロファイルのスナップショットを、 /MigrationSnapshots/ProcServer620 スナップショット・ディレクトリーに作成します。

   - Linux UNIX **Linux および UNIX プラットフォームの場合:**
     BPMSnapshotSourceProfile.sh /opt/ibm/WebSphere/ProcServer620 sourceProfile1 /MigrationSnapshots/ProcServer620
   - Windows **Windows プラットフォームの場合:**
     BPMSnapshotSourceProfile.bat "C:¥Program Files¥IBM¥WebSphere¥ProcServer620" sourceProfile1 c:¥MigrationSnapshots¥ProcServer620

   BPMSnapshotSourceProfile コマンドについて詳しくは、『BPMSnapshotSourceProfile コマンド』のトピックを参照してください。

4. **マイグレーション・ソース・スナップショット・ディレクトリーを一時的な場所にコピーします。**

   ソース・スナップショット・ディレクトリーの zip を作成し、ソース・システムのアップグレード中、リモート・システムに一時的にコピーします。

5. **ソース・システムのオペレーティング・システムをアップグレードします。**

   システムのオペレーティング・システムを適切なバージョンにアップグレードします。

6. **マイグレーション・ターゲット製品をインストールします。**

   ターゲット製品および最新のフィックスパックを、マイグレーションのソース製品と同じシステムにインストールします。

注: 複数の製品によって拡張されたソース・プロファイルからマイグレーションするには、それらの製品の新バージョンを同じターゲット・インストール・ディレクトリーにインストールする必要があります。例えば、ソース・プロファイルが WebSphere Process Server および WebSphere Business Monitor によって拡張されている場合、これら両方の製品が同じターゲット・インストール・ディレクトリーにインストールされる必要があります。

7. **マイグレーション・ソース・スナップショット・ディレクトリーを復元します。**

   リモート・システムに一時的に保管された、スナップショット・ディレクトリー zip ファイルを、最新にアップグレードされたマイグレーション・システムにコピーして戻します。

8. **ターゲット・プロファイルを作成します。**

   WebSphere ESB バージョン 6.0.2 **以外の**あらゆる製品およびソースの組み合わせからプロファイルをマイグレーションする場合、 `BPMCreateTargetProfile` コマンドを使用してターゲット・プロファイルを作成します。このプロファイルは、`BPMMigrateProfile` コマンドを使用してソース・プロファイルを新しいターゲット・プロファイルにマイグレーションしないと使用可能になりません。

   以下の構文を使用して、/MigrationSnapshots/ProcServer620 スナップショット・ディレクトリーにコピーされた sourceProfile1 という名前のソース・プロファイルを使用し、マイグレーション用のターゲット・プロファイルを作成します。

   - Linux UNIX **Linux および UNIX プラットフォームの場合:**
     ```
     BPMCreateTargetProfile.sh /MigrationSnapshots/ProcServer620
     sourceProfile1
     ```
   - Windows **Windows プラットフォームの場合:**
     ```
     BPMCreateTargetProfile.bat
     "C:¥MigrationSnapshots¥ProcServer620" sourceProfile1
     ```

   `BPMCreateTargetProfile` コマンドについて詳しくは、『BPMCreateTargetProfile コマンド』のトピックを参照してください。

   WebSphere ESB プロファイルを 6.0.2 からマイグレーションする場合は、プロファイル管理ツールまたは manageprofiles コマンド行ユーティリティーを使用して、ターゲット・マイグレーション・プロファイルを作成する必要があります。

9. **ソース・プロファイルをターゲット・プロファイルにマイグレーションします。**

   `BPMMigrateProfile` コマンドを使用して、ソース・プロファイルをターゲット・プロファイルにマイグレーションします。 このコマンドは、`BPMSnapshotSourceProfile` コマンドによって指定されたスナップショット・ディレクトリーから構成情報を読み取り、その情報をターゲット・プロファイルにマイグレーションします。

以下の構文を使用して、/MigrationSnapshots/ProcServer620 ディレクトリーにコピーされた sourceProfile1 という名前のソース・プロファイルをターゲット・プロファイルにマイグレーションします。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** `BPMMigrateProfile.sh /MigrationSnapshots/ProcServer620 sourceProfile1`
- Windows **Windows プラットフォームの場合:** `BPMMigrateProfile.bat "C:¥MigrationSnapshots¥ProcServer620" sourceProfile1`

ソース・プロファイルでセキュリティーが有効になっていない場合、`-username` および `-password` パラメーターは必要ありません。有効な場合、入力されたユーザー名は、オペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

Windows オペレーティング・システムの場合、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` パラメーターと `-password` パラメーターを指定する必要はありません。この場合、これらのパラメーターは、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。

`BPMMigrateProfile` コマンドについて詳しくは、BPMMigrateProfile コマンド行ユーティリティーのトピックを参照してください。

10. **マイグレーション状況を確認します。**

`BPMMigrationStatus` コマンドを使用して、マイグレーションの現在の状態を検証します。

- Linux UNIX **Linux および UNIX プラットフォームの場合:** `BPMMigrationStatus.sh`
- Windows **Windows プラットフォームの場合:** `BPMMigrationStatus.bat`

`BPMMigrationStatus` コマンドについて詳しくは、『BPMMigrationStatus コマンド』のトピックを参照してください。

タスクの結果

旧バージョンの BPM 製品から WebSphere Process Server バージョン 7.0 にプロファイルがマイグレーションされ、オペレーティング・システムがアップグレードされます。

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。手順については、109 ページの『マイグレーションの検査』を参照してください。

データベースのマイグレーション

サーバーまたはクラスターをマイグレーションしたら、共通データベース、Business Process Choreographer データベース、および Business Space データベース

用のスキーマを手動でアップグレードする必要があります。さらに、サーバーまたは任意のクラスター・メンバーを始動する前に、データ・マイグレーションの実行が必要な場合もあります。

Common Event Infrastructure データベースおよびメッセージング・エンジン・データベースは、プロファイルのマイグレーション時にランタイム・マイグレーション手順によって自動的にマイグレーションされます。詳しくは、21 ページの『データベース』を参照してください。

共通データベース・スキーマのアップグレード:

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

『スタンドアロン環境のマイグレーション』、『完全ダウンタイムでのネットワーク・デプロイメント環境のマイグレーション』、『最小ダウンタイムでのネットワーク・デプロイメント環境のマイグレーション』の各トピックを参照してください。

このタスクについて

この手順は、以下のデータベース・タイプの共通データベースのアップグレードに対応しています。

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` バージョン 6.1.0 の WebSphere Process Server では、Cloudscape データベースが Derby に置き換わりました。大半の環境では、プロファイル・マイグレーション・ツールによって自動的に Cloudscape データベースが Derby にマイグレーションされます。例外については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』のトピックに説明があります。
Informix	`Informix`
Oracle	`Oracle`
Microsoft SQL Server	`SQLServer`

手順

データベース・タイプに応じて、以下のいずれかの手順を使用して、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します。

- **DB2、Derby、Informix、Oracle、および SQLServer**

  DB2、Derby、Informix、Oracle、および SQLServer の場合は、以下の手順を実行します。

  注: Oracle をデータベース・タイプとしてプロファイルをマイグレーションしている場合には、アップグレード・スクリプトを実行する前に、表示特権を持っていることを確認してください。

  1. データベース・システム上で、対話モード (パラメーターの入力を求めるプロンプトが表示される) または非対話モード (コマンド行にパラメーターが指定される) のいずれかで upgradeSchema コマンドを呼び出します。

     対話モードでは、以下の構文を使用して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:** `upgradeSchema.sh`
     - Windows **Windows プラットフォームの場合:** `upgradeSchema.bat`

     共通データベース upgradeSchema コマンドについて詳しくは、共通データベース用の upgradeSchema コマンド行ユーティリティーを参照してください。

- **DB2iSeries**

  DB2iSeries の場合は、以下の手順を実行します。

  1. ターゲット・マイグレーション・システム上の以下のディレクトリーからコピーされたデータベース・システムで、共通データベースの DB2iSeries SQL スクリプトを見つけます。*install_root*`/dbscripts/CommonDB/DB2iSeries`

     これらのスクリプトは、編集した後で呼び出す必要があります。各スクリプトにはアップグレード元のソース製品のバージョン (602、610、612、620 のい

ずれか) を含むファイル名があり、 **upgradeSchema** または **wbiserver_upgradeSchema**で始まります。

2. SQL スクリプトを確認し、必要に応じて自分の要件を満たすように変更します。例えば、ユーザー名、パスワード、またはファイル・パスの変更が必要な場合があります。
3. データベース・システム上のデータベース・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。

   注: データベース固有のツールを使用して .sql スクリプトを実行するようにデータベース・システムを完全にセットアップすることが非常に重要です。例えば、dbType が DB2_Universal である場合、任意の db2 コマンドをコマンド・プロンプトで実行できます。これは、Oracle の場合には sqlplus コマンド、SQL サーバーの場合には osql コマンドに該当します。
4. データベース・システム上の DB2iSeries SQL スクリプトが含まれているディレクトリーから、各 SQL スクリプトを呼び出します。

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。

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。

始める前に

『スタンドアロン環境のマイグレーション』、『完全ダウンタイムでのネットワーク・デプロイメント環境のマイグレーション』、『最小ダウンタイムでのネットワーク・デプロイメント環境のマイグレーション』の各トピックを参照してください。

このタスクについて

Business Process Choreographer データベースをアップグレードするには、以下の手順を実行します。

手順

1. マイグレーション中に、データベース設計ファイルが生成されます。データベース設計ファイルを使用して、スキーマをアップグレードするために必要なスクリプト (1 つまたは複数) を生成するには、データベース設計ファイルをカスタマイズしておく必要があります。
   a. 生成されたデータベース設計ファイルを探索します。
      - Linux UNIX Linux および UNIX プラットフォームの場合: *profile_root*/dbscripts/ProcessChoreographer/*database_type*/*database_name*/*database_schema*/createSchema.properties

- Windows Windows プラットフォームの場合:
  profile_root¥dbscripts¥ProcessChoreographer
  ¥*database_type*¥*database_name*¥*database_schema*¥createSchema.properties

ここで、

profile_root

- Business Process Choreographer がサーバー上に構成されている場合、これは対応するノードのプロファイルになります。
- Business Process Choreographer がクラスター上に構成されている場合、これは BPMMigrateCluster を実行するプロファイルになり (以前の WBIProfileUpgrade.ant)、通常はデプロイメント・マネージャー・プロファイルです。

database_name
データベースの名前です。

database_schema
データベース・スキーマの名前です。これはオプションで、暗黙のスキーマが使用されている場合は設定されません。

database_type
使用しているデータベース・タイプに対応するディレクトリーの名前です。

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
Derby	Derby バージョン 6.1.0 の WebSphere Process Server では、Cloudscape データベースが Derby に置き換わりました。大半の環境では、プロファイル・マイグレーション・ツールによって自動的に Cloudscape データベースが Derby にマイグレーションされます。例外については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』のトピックに説明があります。
Informix	Informix
Oracle	Oracle
Microsoft SQL Server	SQLServer

b. 適切なデータベース設計 (createSchema.properties) ファイルをコピーします。
c. データベース設計ツールを始動し、プロパティー・ファイルに定義されているデータベース構成を編集します。

1)

- Linux UNIX Linux および UNIX プラットフォームの場合は、以下のコマンドを入力します。

```
install_root/util/dbUtils/DbDesignGenerator.sh
       -e copy_of_createSchema.properties_file
```

- Windows Windows プラットフォームの場合は、以下のコマンドを入力します。

```
install_root¥util¥dbUtils¥DbDesignGenerator.bat
       -e copy_of_createSchema.properties_file
```

このツールの使用について詳しくは、データベース設計ツールを使用したデータベース設計ファイルの作成を参照してください。

2) すべての質問に答えるか、または Enter を押してデフォルト値を受け入れます。 特に、マイグレーション・シナリオを選択し、データベース名、データベース・スキーマ修飾子、およびあらゆるテーブル・スペース名が正しいことを確認してください。

3) 入力ファイルを上書きするか、新規ファイルに変更内容を保存するかを選択できます。

d. 変更を加えたデータベース設計ファイルに対してデータベース設計ツールを実行し、アップグレード・スクリプトを生成します。

- Linux UNIX Linux および UNIX プラットフォームの場合は、以下のコマンドを入力します。

```
install_root/util/dbUtils/DbDesignGenerator.sh
     -g copy_of_createSchema.properties_file
    [-d output_directory]
```

- Windows Windows プラットフォームの場合は、以下のコマンドを入力します。

```
install_root/util/dbUtils/DbDesignGenerator.sh
     -g copy_of_createSchema.properties_file
    [-d output_directory]
```

-d オプションの入力による出力ディレクトリーの指定を行わなかった場合は、生成されたファイルが現行ディレクトリーのサブディレクトリーに書き込まれます。

- ツールは、このバージョンにマイグレーション可能なすべてのスキーマ・バージョンについて、 upgradeSchema*schema_version*.sql を生成します。
- データベースがテーブル・スペースを使用している場合、ツールは upgradeTablespaces*schema_version*.sql スクリプトも生成します。
- DB2 を使用していて、なおかつ 6.2 より前のバージョンからマイグレーションする場合は、ツールによって upgradeTablespace*schema_version*.sql スクリプトが作成されます。このスクリプトは 8k のテーブル・スペースを作成します。

例えば、以下のファイルが生成される可能性があります。

```
upgradeSchema602.sql
upgradeSchema610.sql
upgradeSchema612.sql
upgradeSchema620.sql
upgradeTablespace602.sql
upgradeTablespace610.sql
upgradeTablespace612.sql
```

2. 別のシステムでこのスクリプトを実行する場合、データベースがホストされているシステムに、適切な生成済みアップグレード・スクリプトをコピーします。コピーする必要があるのは、マイグレーション元の *schema_version* と一致する 1 つ以上のスクリプトのみです。例えば、バージョン 6.2 からマイグレーションする場合、ファイル `upgradeSchema620.sql` をコピーします。
3. 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(*SYSRPYL)
      ADDRPYLE 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。
4. マイグレーション元のバージョン用に生成された `createTablespace`*schema_version*`.sql` ファイルがある場合は、そのファイルを実行して 8k のテーブル・スペースを作成します。データベースに対する SQL スクリプトの実行方法について詳しくは、ご使用のデータベースの製品ドキュメンテーション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。
5. マイグレーション元のバージョン用に生成された `upgradeTablespaces`*schema_version*`.sql` ファイルが存在する場合は、それを実行してテーブル・スペースをアップグレードします。データベースに対する SQL スクリプトの実行方法について詳しくは、ご使用のデータベースの製品ドキュメンテーション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。
6. マイグレーション元のバージョン用の `upgradeSchema`*schema_version*`.sql` スクリプトを実行します。 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。

   注: スキーマのアップグレード後に最初にサーバーを開始するとき、以下のいずれかのメッセージが SystemOut.log ファイルに書き込まれます。

   ```
   CWWBB0613I: データベース・マイグレーション: 700/1 から 700/0 へ正常に完了しました。
   CWWBB0615E: 700/1 から 700/0 へのデータベース・マイグレーションが失敗しました。
   ```

   「/」文字の後の値は、マイグレーションが正常に行われた後にゼロにリセットされるバイナリー・フラグであり、製品のバージョン番号の一部ではありません。

データベースのマイグレーションが失敗した場合は、ログ・ファイルでその他の障害メッセージを調べ、問題があればすべて修正してから、サーバーの再始動を試行してください。

7. 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。

      ```
      WRKRPLYLE
      ```

   b. ステップ 3c (98 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 データベース・データのマイグレーション:

バージョン 6.1.x または 6.0.2.x からマイグレーションする場合は、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した後、サーバーまたは任意のクラスター・メンバーを始動する前にデータ・マイグレーションを実行する必要があります。バージョン 6.2 からマイグレーションする場合は、このデータ・マイグレーションを実行しないでください。

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 1327385」を参照してください。

手順

1. DB2 for Linux、UNIX、Windows、または z/OS を使用する場合は、次のようにします。
   a. カスタム作成した索引、ビュー、トリガーをすべて除去します。また以下のテーブルはデータ・マイグレーションの影響を受けるため、これらのテーブルを参照する項目も除去します。
      - 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. マイグレーション対象の Business Process Choreographer 構成がクラスター上にある場合、このクラスターについて、 BPMMigrateCluster ツールを手動で実行済みであることを確認してください。
3. データベース・マイグレーション・スクリプトを実行するノード上で、syncNode コマンドを実行して、ノードとデプロイメント・マネージャーを同期します。
4. 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。
5. 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、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: CWWBB0652E: 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 4 (100 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: CWWBB0653E: 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: CWWBB0651I: 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。

6. 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 (99 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 データベース・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 Space データベース・スキーマのマイグレーション:

サーバーをバージョン 6.1.2 または バージョン 6.2.0.x からマイグレーションした後には、バージョン 7.0 のサーバーを始動する前に、Business Space 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

『スタンドアロン環境のマイグレーション』、『完全ダウンタイムでのネットワーク・デプロイメント環境のマイグレーション』、『最小ダウンタイムでのネットワーク・デプロイメント環境のマイグレーション』の各トピックを参照してください。

このタスクについて

この手順は、以下のデータベース・タイプについて、Business Space データベース・スキーマのマイグレーションをサポートします。

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 WebSphere Process Server の バージョン 6.1.0 では、Cloudscape データベースは Derby に置き換わりました、 大半の環境では、プロファイル・マイグレーション・ツールによって自動的に Cloudscape データベースが Derby にマイグレーションされます。例外については、『IBM Cloudscape または Apache Derby データベースのマイグレーション』のトピックに説明があります。
Informix	Informix
Oracle	Oracle
Microsoft SQL Server	SQLServer

手順

1. データベースでの読み取りアクセス権と書き込みアクセス権を持つユーザーとしてデータベース・サーバーにログオンします。
2. データベースに接続します。

3. 最後に構成したプロファイル内の migrateSchema スクリプトを探し、データベースと同じシステム上の任意のロケーションに保存します。

   デフォルトでは、スクリプトは以下のディレクトリーにあります。

   - **スタンドアロン・サーバー**: *profile_root*/dbscripts/BusinessSpace/*node_name_server_name*/*database_product_name*/*database_name*
   - **クラスター**: *profile_root*/dbscripts/BusinessSpace/*cluster_name*/*database_product_name*/*database_name*

   スクリプトは、最後に構成したサーバーまたはクラスターのプロファイル内にあります。

   注: デフォルト値がご使用の環境に一致しない場合、このスクリプトを変更しなければならない場合があります。

   注: また、SQL スクリプト *was_root*/dbscripts/BusinessSpace/*database_product_name*/*database_name* を使用して、Business Space データベース・スキーマをアップグレードすることもできます。これらのスクリプトは、変更して、変数 *DB_NAME* および *DB_USER* を実際の値に置き換える必要があります。また、スキーマ名がデータベースに既に存在する必要があります。
4. データベース・システム上で、以下の構文を使用して、migrateSchema コマンドを起動します。

   - Linux UNIX `migrateSchema.sh`
   - Windows `migrateSchema.bat`
   - **IBM i**: `migrateSchema`

   migrateSchema コマンドについて詳しくは、Business Space データベース用の migrateSchema コマンド行ユーティリティーを参照してください。
5. DB2 および DB2 for z/OS の場合は、次のコマンドを使用してコマンド行インターフェースをBusiness Space のデータベースにバインドします。

   ```
   db2 connect to database_name
   db2 bind DB2_installation_directory¥bnd¥@db2cli.lst blocking all
   grant public
   db2 connect reset
   ```

   各部の意味は、次のとおりです。

   database_name はBusiness Space データベースの名前です

   DB2_installation_directory は、DB2 がインストールされるディレクトリーです
6. サーバーを始動します。

タスクの結果

データベース・スキーマがマイグレーションされ、Business Space バージョン 7.0 で使用するための準備ができました。

次のタスク

- Business Space で使用できるようにしたいウィジェットのエンドポイントを更新します。
- Business Space と、チームが使用しているウィジェット用に、セキュリティーをセットアップします。

Business Space データベース・データのマイグレーション:

Business Space データベース・スキーマをマイグレーションしたら、 Business Space データベース・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

Business Space データベース・スキーマをマイグレーションします。

注: Business Space データのマイグレーション時に、すべてのBusiness Space ユーザーごとにマイグレーションされる個別設定情報は、直近に表示した 10 個のページと、直近に調整した 60 個のウィジェットに制限されます。

手順

1. **ウィジェット定義ファイルをコピーします。**

   プロファイルのマイグレーション中に、バージョン 6.2.0 および バージョン 6.1.2 ウィジェット定義ファイルが、バージョン 7.0 ターゲット・サーバー上の *profile_root*/BusinessSpace/datamigration/widgets ディレクトリーに自動的にコピーされます。ただし、バージョン 7.0 ウィジェット定義ファイルおよびバージョン 6.2.0 または バージョン 6.1.2 カスタム・ウィジェット定義ファイルは、このディレクトリーに手動でコピーする必要があります。

   環境に応じて、以下のいずれかの手順を使用してください。

   - スタンドアロン環境または非クラスター管理対象ノード環境の場合は、ウィジェット・ファイルをターゲット・プロファイルにコピーします。
   - クラスター化された Business Space 環境の場合は、そのクラスターに参加しているすべてのプロファイル上にウィジェット・ファイルをコピーします。

   ウィジェット定義ファイルをコピーするには、以下の手順を実行します。

   a. 非カスタム・Business Space バージョン 7.0 のウィジェット定義ファイルを、すべて *profile_root*/BusinessSpace/datamigration/widgets ディレクトリーにコピーします。 これらのファイルは、バージョン 7.0 の *profile_root*/installedApps ディレクトリー内で *iwidget.xml* または *iWidget.xml* のいずれかを含むファイル名を検索することによって、見つけることができます。

      注: ファイルの上書きに関する警告が表示された場合、それを受け入れます。これは、プロファイルのマイグレーション中に自動的にコピーされた非カスタムの バージョン 6.2.0 または バージョン 6.1.2 ウィジェット定義ファイルを、新規のバージョン 7.0 ウィジェット定義ファイルによって上書きしていることを意味します。

b. バージョン 6.2.0 または バージョン 6.1.2 のカスタム・ウィジェットがある場合、Business Space データをマイグレーションする前に、すべてのカスタム・ウィジェット定義ファイルを Business Space のバージョン 7.0 のインストール済み環境にコピーする必要があります。 これを実行するには、旧バージョンのBusiness Space から、すべてのカスタム・ウィジェット定義ファイルを *profile_root*/BusinessSpace/datamigration/widgets ディレクトリーにコピーします。

2. **ターゲット環境でサーバーを始動します。** 環境に応じて、以下のいずれかの手順を使用してください。

- スタンドアロン環境の場合、ターゲット・サーバーを始動します。

  マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはターゲット・プロファイルのファースト・ステップ・コンソールから、startServer コマンドを使用して、マイグレーション・ターゲット・サーバーを始動します。

  以下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:** startServer.sh *server_name*

  – Windows **Windows プラットフォームの場合:** startServer.bat *server_name*

  startServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

- Network Deployment 環境の場合は、以下の手順を使用します。

  重要: Network Deployment 環境の構成方法に応じて、以下のいずれかの方法で手順を実行してください。

  – Business Space が構成されている非クラスター管理対象ノードに、更新対象の Business Space データベースが属している場合は、このノード上でノード・エージェントとサーバーを始動します。

  – 更新対象の Business Space データベースがクラスター環境に属している場合は、このクラスターに参加しているノードを選択し、そのノード上でノード・エージェントとサーバーを始動します。

    注: Business Space のクラスター環境の場合は、そのクラスターに参加しているノードを 1 つだけ始動する必要があります。

  a. **マイグレーション・ターゲット・ノード・エージェントを始動します。**

     マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、startNode コマンドを使用して、マイグレーション・ターゲット・ノード・エージェントを始動します。

     以下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:** `startNode.sh`

– Windows **Windows プラットフォームの場合:** `startNode.bat`

`startNode` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startNode コマンド』のトピックを参照してください。

b. **マイグレーション・ターゲット・サーバーを始動します。**

マイグレーション・ターゲット・サーバーの *profile_root*/bin ディレクトリーから、またはプロファイルのファースト・ステップ・コンソールから、`startServer` コマンドを使用して、マイグレーション・ターゲット・サーバーを始動します。

以下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:** `startServer.sh` *server_name*

– Windows **Windows プラットフォームの場合:** `startServer.bat` *server_name*

`startServer` コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

3. **Business Space データをマイグレーションします。**

前のステップでターゲット・サーバーを始動したノード上で、`migrateBSpaceData` スクリプトを実行して Business Space バージョン 6.1.2 または バージョン 6.2.0 のデータを Business Space バージョン 7.0 にマイグレーションします。

次のように、ご使用のオペレーティング・システム用のスクリプトを選択します。

- **Windows**: `migrateBSpaceData.bat`
- **AIX、HP-UX、Linux、Solaris**: `migrateBSpaceData.sh`

このスクリプトは、*install_root*/BusinessSpace/scripts/ ディレクトリーにあります。 `migrateBSpaceData` スクリプトについて詳しくは、migrateBSpaceData コマンド行ユーティリティーを参照してください。

4. オプション: **カスタム・ウィジェットのウィジェット・カタログをマイグレーションします。**

カスタム・ウィジェットがあり、Network Deployment 環境をマイグレーションしている場合は、`updateBSpaceWidgets` コマンドをデプロイメント・マネージャー・プロファイルに対して実行して、*profile_root*/BusinessSpace/datamigration/catalog フォルダーに XML 形式で生成された、カスタム・ウィジェットのマイグレーション済みウィジェット・カタログに、データを取り込み

ます。 updateBSpaceWidgets コマンドを、デプロイメント・マネージャー・プロファイルの *profile_root*¥bin ディレクトリーから実行します。

例

```
wsadmin>$AdminTask updateBusinessSpaceWidgets {-clusterName cluster_name
-catalogs profile_root/BusinessSpace/datamigration/catalog }
```

注: カタログ・ファイルが生成されるのは、カスタム・ウィジェットがある場合に限ります。

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

5. **製品ウィジェットとカスタム・ウィジェットの両方のウィジェット・エンドポイントをマイグレーションします。**

   Network Deployment 環境をマイグレーションしている場合は、updateBSpaceWidgets コマンドをデプロイメント・マネージャー・プロファイルに対して実行して、*profile_root*/BusinessSpace/datamigration/endpoints フォルダーに XML 形式で生成された、製品ウィジェットとカスタム・ウィジェットの両方のマイグレーション済みウィジェット・エンドポイントに、データを取り込みます。 updateBSpaceWidgets コマンドを、デプロイメント・マネージャー・プロファイルの *profile_root*¥bin ディレクトリーから実行します。

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

   例

   ```
   wsadmin>$AdminTask updateBusinessSpaceWidgets {-clusterName cluster_name
   -endpoints profile_root/BusinessSpace/datamigration/endpoint }
   ```

6. **ターゲット・サーバーを停止します。** 環境に応じて、以下のいずれかの手順を使用してください。
   - スタンドアロン環境の場合、ターゲット・サーバーを停止します。

     マイグレーション・ターゲット・システムの *profile_root*/bin ディレクトリーから stopServer コマンドを使用して、マイグレーション・ターゲット・サーバーを停止します。

     以下の構文を使用します。
     - **i5/OS プラットフォームの場合:** stopServer *server_name* -username *user_name* -password *password*
     - Linux UNIX **Linux および UNIX プラットフォームの場合:** stopServer.sh *server_name* -username *user_name* -password *password*
     - Windows **Windows プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

     注:

– プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。
– 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、これらのパラメーターは、Windows サービスがサーバー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。
– プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

- Network Deployment 環境の場合は、ターゲット・クラスター内のサーバー (ステップ 2 で始動されたもの) を停止します。

  このステップは、クラスター内の各サーバーについて繰り返す必要があります。

  マイグレーション・ソース・ターゲットの *profile_root*/bin ディレクトリーから stopServer コマンドを使用して、マイグレーション・ターゲット・サーバーを停止します。

  以下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:**
  stopServer.sh *server_name* -username *user_name* -password *password*

  – Windows **Windows プラットフォームの場合:** stopServer.bat *server_name* -username *user_name* -password *password*

  プロファイルのセキュリティーが有効になっている場合、入力されたユーザー名がオペレーター・ロールまたは管理者ロールのメンバーでなければなりません。

  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、これらのパラメーターは、Windows サービスがサーバー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。

  プロファイルでセキュリティーが有効になっていない場合、 -username および -password パラメーターは必要ありません。

  stopServer コマンドについて詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

タスクの結果

Business Space のデータベース・データが、Business Space バージョン 7.0 にマイグレーションされます。

マイグレーションの検査

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

始める前に

マイグレーションされたサーバーが始動していることを確認してください。

手順

1. BPMMigrateProfile コマンドおよび BPMMigrateCluster コマンド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/BPMMigrateProfile.*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、BPMSnapshotSourceProfile または BPMMigrateProfile 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 7.0 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/BPMMigrateCluster.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 ディレクトリーに格納されます。ログ・ファイルにはプロファイル名が含まれています (例: create <プロファイル名>.log)。
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. 共通データベースのアップグレードを検証します。WebSphere Process Server に対して構成したユーザーが必要な権限をすべて持っているため、共通データベー

スのアップグレードを手動で実行しなかった場合は、デプロイメント・マネージャーの始動時にデータベースが正常にアップグレードされたことを確認してください。

a. デプロイメント・マネージャーのプロファイル・ディレクトリーにナビゲートします。 一般に、これは install_root/profiles/<profile name> です。

b. *logs* フォルダーにナビゲートし、 SystemOut.log ファイルを確認します。メッセージ「The Common Database Schema upgrade is started」および「CWLDB0003I: WebSphere Process Server Schema version was updated to "7.0.0.0" successfully.」を見つけてください。

5. 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. 右隅にあるパネルで、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、132 ページの『ランタイム・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。

環境のロールバック

WebSphere Process Server バージョン 7.0 環境へのマイグレーション後に、マイグレーション元のバージョンにロールバックできます (バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 環境でも可能)。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。

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 7.0 デプロイメント・セルを、バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 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 6.2.0、6.1.2、6.1.0、または 6.0.2 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・ユーティリティーを使用して、既存の構成をバックアップします。
   - **backupConfig** コマンド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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 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 7.0 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 7.0 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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 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 7.0 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 6.2.0、6.1.2、6.1.0、または 6.0.2 *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ひととおり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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 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 6.2.0、6.1.2、6.1.0、または 6.0.2 *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一つずつ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 7.0 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 インフォメーション・センターの『syncNode コマンド』を参照してください。

5. バージョン 7.0 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 7.0 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 7.0 プロファイルを削除します。

   WebSphere Application Server 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 6.2.0、6.1.2、6.1.0、または 6.0.2 環境で開始します。

9. 58 ページの『最小限のダウン時間での Network Deployment 環境のマイグレーション』の手順に従った際に、同期が使用不可になっていた場合、すべてのノードについて同期を使用可能にしてください。 これを行うには、以下の手順を使用します。

   a. WebSphere Application Server 管理コンソールから、「**システム管理**」 → 「**ノード・エージェント**」を選択します。

   b. ノードのノード・エージェントをクリックします。

   c. 「**ファイル同期化サービス**」をクリックします。

   d. 「**サーバー始動時にサービスを使用可能にする**、「**自動同期**」および「**始動同期**」を選択します。

   e. 「**適用**」をクリックし、次に「**OK**」をクリックして構成の変更を保存します。

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れば、ここで再開できます。

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 7.0 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・リポジトリー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 6.2.0、6.1.2、6.1.0、または 6.0.2 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・ユーティリティーを使用して、既存の構成をバックアップします。
   - **backupConfig** コマンド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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 7.0 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、 111 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 7.0 環境で現在実行中のサーバーをすべて停止します。
3. 以前の構成をリストアします。
   a. **restoreConfig** コマンドまたは望ましいユーティリティーを実行して、バージョン 7.0 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 6.2.0、6.1.2、6.1.0、または 6.0.2 構成をリストアします。
      - **restoreConfig** コマンドまたは望ましいユーティリティーを実行して、バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 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 6.2.0、6.1.2、6.1.0、または 6.0.2*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 7.0 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 7.0 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 7.0 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。
7. バージョン 7.0 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。
8. バージョン 7.0 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れば、ここで再開できます。

マイグレーション後のタスク

マイグレーション後のタスクとは、バージョン 7.0 に正しくマイグレーションした後で、WebSphere Process Server、Business Process Choreographer、および Business Space 上で実行するタスクを言います。

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したり、バージョン 7.0 サーバーをさらに構成することが必要な場合があります。

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しておきます。

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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 7.0 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 BPMMigrateProfile.*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 7.0 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。
- バージョン 6.2.0.x からバージョン 7.0 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 7.0 へのマイグレーション後、WebSpehere Adapter for SAP バージョン 7.0.0.1 でアプリケーションを実行する前に、以下のステップを実行します。
  1. sapjco3.jar ファイルを *WPS_HOME*/lib フォルダーにコピーします。
  2. *WPS_HOME*/lib フォルダーから、以下のファイルを削除します。
     - JCO 2.1.x jar
     - sapjco.jar
  3. sapjco3 ダイナミック・リンク・ファイルを *WPS_HOME*/bin フォルダーにコピーします。ご使用のオペレーティング・システムに応じて、sapjco3 ダイナミック・リンク・ファイルには次の名前が付けられます。
     - **AIX/Linux**: libsapjco3.so
     - **HP**: libsapjco3.sl
     - **Windows**: sapjco3.dll
  4. JCO 2.1.x の該当のダイナミック・リンク・ファイルを *WPS_HOME*/bin フォルダーから削除します。
  5. WebSphere Integration Development インストール済み環境で、ResourceAdapters/SAP_7.0.0.0/ext/ ディレクトリーにナビゲートし、CWYAP_SAPAdapterExt.jar ファイルを *WPS_HOME*/lib フォルダーにコピーします。

  WebSphere Adapter for SAP について詳しくは、 WebSphere Adapter for SAP Software の資料を参照してください。
- オプション: バージョン 7.0 にマイグレーションした後は、ターゲット重要度プロパティーのデフォルト値が バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 から変更されていることに注意してください。バージョン 7.0 では、デフォルト値は targetSignificance=prefered から targetSignificance=required に変更されています。新規デフォルト値は、WebSphere Process Server 構成に含まれる JMS アクティベーション・スペックおよび接続ファクトリーに設定されます。

  マイグレーション済み環境 (バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2) でターゲット重要度の値を変更するかどうかを決定する必要があります。
- マイグレーション済み バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 環境に Business Process Choreographer Observer アプリケーションが組み込まれていて、事後マイグレーション作業で Business Process Choreographer Explorer を新規デプロイメント・ターゲットに移動する必要がある場合、Business Process

Choreographer Observer アプリケーションは Business Process Choreographer Explorer と一緒に移動されません。このようなシナリオでは、6.2 以前の Business Process Choreographer Observer アプリケーションを、移動後の Business Process Choreographer Explorer 構成にマージしてから、古い Business Process Choreographer Observer アプリケーションを削除する必要があります。

または、マイグレーション済み Observer を、移動する前の Explorer にマージすることもできます。そして、Observer を結合した (Observer レポート作成機能が備わった) Explorer を移動します。

- バージョン 7.0 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。そのようになっていない場合は、「**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。

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、プロセス・サーバー定義の Java 仮想マシン設定に変換します。ほとんどの設定は直接マップされますが、一部の設定は、そのロールが WebSphere Application Server バージョン 7.0 では異なるため、マイグレーションされません。このような場合、構成設定が存在しないか、意味が異なっているか、スコープが異なっている可能性があります。プロセス定義設定または JVM 設定の変更について詳しくは、WebSphere Application Server バージョン 7.0 インフォメーション・センターの以下のトピックを参照してください。

- プロセス定義設定
- Java 仮想マシン設定

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、必要に応じて、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 7.0 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. バージョン 7.0 へのマイグレーション前に担当者割り当てを使用した場合、以下を実行する必要があります。
   a. *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.xsl、LDAPTransformation.xsl、SystemTransformation.xsl、VMMTransformation.xsl、UserRegistryTransformation.xsl) 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 7.0 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。 クラスター環境では、デプロイメント・マネージャー上と、Business Process Choreographer が構成されているクラスターのメンバーをホストする各ノード上で、変換ファイルが使用可能でなければなりません。これらがすべて同じバージョンの変換ファイルを使用していることを確認してください。

      注: *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、手動コピーが必要になることがあります。これは、バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 担当者ディレクトリー構成 (従来のスタッフ・プラグイン構成) で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。
   b. 代替機能を使用し、代替情報が VMM 用に構成されたユーザー・リポジトリーの 1 つに保管されている場合、 substitutionStartDate および substitutionEndDate 用の新規プロパティーをリポジトリーに追加する必要があります。 実行する必要のあるステップは、代替情報を VMM ファイル・レジストリーに保管したか、または VMM プロパティー拡張レジストリーに保管したかによって異なります。

      VMM ファイル・レジストリーの場合:

      1) substitutionStartDate および substitutionEndDate プロパティーを、wimxmlextension.xml ファイルにある PersonAccount エンティティー・タイプの定義に追加します。 Network Deployment 環境では、デプロイメント・マネージャーでファイルを編集します。

         - Linux UNIX Linux および UNIX プラットフォームの場合、このファイルは *profile_root*/config/cells/*cellName*/wim/model にあります。
         - Windows Windows プラットフォームの場合、このファイルは *profile_root*¥config¥cells¥*cellName*¥wim¥model にあります。

         新規プロパティーを含むようにファイルを拡張します。新規プロパティーは太字で強調表示されています。

```
<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
     dataType="STRING" multiValued="false" propertyName="isAbsent">
    <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
     dataType="STRING" multiValued="true" propertyName="substitutes">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
     dataType="STRING" multiValued="false" propertyName="substitutionStartDate">
    <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
</wim:propertySchema>

<wim:propertySchema nsURI="http://www.ibm.com/websphere/wim"
     dataType="STRING" multiValued="false" propertyName="substitutionEndDate">
     <wim:applicableEntityTypeNames>PersonAccount</wim:applicableEntityTypeNames>
</wim:propertySchema>
```

2) 変更は、サーバーを再始動すると有効になります。 Network Deployment 環境では、デプロイメント・マネージャーも再始動する必要があります。

VMM プロパティー拡張レジストリーの場合:

1) 代替プロパティー isAbsent および substitutes がプロパティー拡張リポジトリー用に定義されていることを確認します。 マイグレーション前にこれらが定義されていなかった場合、代替情報は VMM プロパティー拡張リポジトリーに保管されていないため、このマイグレーション・ステップは必要ありません。

   ディレクトリー *install_root*/bin に移動し、ローカル・モード、または接続モードで次のコマンドを入力します。 Network Deployment 環境では、デプロイメント・マネージャーでコマンドを入力します。

   ```
   wsadmin -username admin -password adminPassWord
   $AdminTask listIdMgrPropertyExtensions
   ```

2) 次のコマンドを入力して、新規プロパティー substitutionStartDate および substitutionEndDate をプロパティー拡張リポジトリー構成に追加します。

   ```
   $AdminTask addIdMgrPropertyToEntityTypes
      {-name substitutionStartDate
       -dataType String
       -isMultiValued false
       -entityTypeNames PersonAccount
       -repositoryIds LA}

   $AdminTask addIdMgrPropertyToEntityTypes
      {-name substitutionEndDate
       -dataType String
       -isMultiValued false
       -entityTypeNames PersonAccount
       -repositoryIds LA}
   ```

3) 変更は、サーバーを再始動すると有効になります。 Network Deployment 環境では、デプロイメント・マネージャーも再始動する必要があります。

4) 次のコマンドを入力して、新規プロパティーがプロパティー拡張リポジトリー構成に追加されたことを確認します。

   ```
   $AdminTask listIdMgrPropertyExtensions
   ```

2. Business Flow Manager および Human Task Manager の REST API エンドポイントを構成し、参照をすべて更新して、Web モジュールを Web サーバーにマップします。

a. 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 WebSphere の構成リポジトリー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 Business Space 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 Business Space 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。

   1) Business Flow Manager のエンドポイントを変更するには、「**サーバー**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」または「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_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、「**追加プロパティー**」で「**REST サービスのエンドポイント**」をクリックします。

   2) Human Task Manager のエンドポイントを変更するには、「**サーバー**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」または「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_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、「**追加プロパティー**」で「**REST サービスのエンドポイント**」をクリックします。

b. バージョン 6.2 以降からマイグレーションした場合で、bpcEndpoints.xml ファイルをまだ使用している場合は、エンドポイント構成が自動的にマイグレーションされないため、管理コンソールを使用して Business Space 用の REST API の参照が正しいことを確認する必要があります。なお、バージョン 6.2 以降は、bpcEndpoints.xml ファイルではなく管理コンソールを使用して、Business Space 用の Business Process Choreographer REST API エンドポイントを構成してください。Business Space 用の Business Process Choreographer REST API エンドポイントを確認または変更するには、以下の手順を実行します。

   1) Business Flow Manager のエンドポイントを変更するには、「**サーバー**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」または「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_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、「**追加プロパティー**」で「**REST サービスのエンドポイント**」をクリックします。

   2) Human Task Manager のエンドポイントを変更するには、「**サーバー**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」または「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_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、「**追加プロパティー**」で「**REST サービスのエンドポイント**」をクリックします。

3) Business Space にこれらのエンドポイントを登録するには、「**サーバー**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」または「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_name***」をクリックした後、「**ビジネス・インテグレーション**」で「**Business Space の構成**」を展開し、「**追加のプロパティー**」で「**REST サービス・エンドポイント登録**」をクリックして、Business Flow Manager サービスと Human Task Manager サービスの「**サービス・エンドポイント・ターゲット**」が正しく選択されていることを確認します。

c. REST API はマイグレーション中に構成されました。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。

1) 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**」 → 「**アプリケーション・タイプ**」 → 「**WebSphere エンタープライズ・アプリケーション**」 → 「**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**」 → 「**アプリケーション・タイプ**」 → 「**WebSphere エンタープライズ・アプリケーション**」 → 「**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。

2) 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**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」 または 「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_name***」のいずれかをクリックし、「**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**」をクリックしてから、構成済みの Business Process Choreographer Explorer インスタンスのリストで、編集を行うインスタンスをクリックし、「**Business Flow Manager REST API URL**」および「**Human Task Manager REST API URL**」の値を変更します。必要に応じて、その他のインスタンスでもこれを繰り返します。

3) Business Space 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**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」または「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_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、「**追加プロパティー**」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**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」または「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_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、「**追加プロパティー**」で「**REST サービスのエンドポイント**」をクリックします。

d. JAX Web サービス API はマイグレーション中に構成されました。 Web モジュールを Web サーバーにマップして、JAX Web サービス API の Web モジュール用にコンテキスト・ルートを変更することが必要な場合もあります。

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。

1) 管理コンソールで、「**アプリケーション**」 → 「**アプリケーション・タイプ**」 → 「**WebSphere エンタープライズ・アプリケーション**」 → 「**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。

2) Web モジュール BFMJAXWSAPI 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。

3) 管理コンソールで、「**アプリケーション**」 → 「**アプリケーション・タイプ**」 → 「**WebSphere エンタープライズ・アプリケーション**」 → 「**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。

4) Web モジュール HTMJAXWSAPI 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。

3. クラスターのマイグレーションに「最小限のダウン時間」のシナリオを実行した場合は、bpeupgrade.jacl スクリプトを実行して、事前定義された新規バージョンのヒューマン・タスクをデプロイし、新規の Business Process Choreographer JAX Web サービス API が追加されていることを確認します。

注意:
管理コンソールを使用して、事前定義されたヒューマン・タスク・アプリケーションを更新しようとしないでください。

a. デプロイメント・マネージャーを停止します。

b. デプロイメント・マネージャーで、bpeupgrade.jacl スクリプトがあるディレクトリーに移動し、そのスクリプトを実行します。

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

install_root/ProcessChoreographer/config ディレクトリーに移動し、以下のコマンドを入力します。

```
../../bin/wsadmin.sh -conntype NONE -profileName profileName
                     -f bpeupgrade.jacl -cluster clusterName
```

Windows **Windows プラットフォームの場合:**

install_root¥ProcessChoreographer¥config ディレクトリーに移動し、以下のコマンドを入力します。

```
..¥..¥bin¥wsadmin -conntype NONE -profileName profileName
                  -f bpeupgrade.jacl -cluster clusterName
```

ここで *profileName* はデプロイメント・マネージャーのプロファイルの名前、*clusterName* は Business Process Choreographer が構成されているクラスターの名前です。

c. デプロイメント・マネージャーを始動します。

d. 構成の変更をノードと同期化し、クラスター・メンバーを再始動します。

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。

注: 古い事前定義されたヒューマン・タスク・アプリケーションのインスタンスが実行中の可能性があるため、古い事前定義されたヒューマン・タスク・アプリケーションはマイグレーション中にアンインストールされません。 これは、マイグレーション後に、新旧バージョン両方の事前定義されたヒューマン・タスク・アプリケーションがシステムに存在することを意味します。 バージョン番号付けは、最後にアプリケーションが更新された時期を示すため、現行リリースよりも古く見える可能性がありますが、単に変更されていないことを意味しているだけです。

a. すべての旧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**」 → 「**アプリケーション・タイプ**」 → 「**WebSphere エンタープライズ・アプリケーション**」をクリックします。

c. 次のアプリケーションのいずれかで複数のバージョンが存在する場合は、古い方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_V*nnn_scope*.ear
- HTM_PredefinedTaskMsg_V*nnn_scope*.ear

ここで、

nnn アプリケーションが最後に更新されたときのバージョン番号 (620 など)。 これらのアプリケーションの最新バージョンが現行リリースよりも古く見えますが、単に変更されていないことを意味しているだけです。 重要なことは、2 つのアプリケーションの複数のバージョンが存在する場合、最も古いアプリケーションのみを削除することです。

scope *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: バージョン 6.1.x または 6.0.2.x からマイグレーションした場合は、データベースからテーブル WI_ASSOC_OID_T を削除して、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・スペースを解放できます。
6. オプション: バージョン 6.1.x または 6.0.2 からマイグレーションし、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. バージョン 6.0.2 または 6.1.x からマイグレーションした場合で、なおかつ Business Process Choreographer Observer 構成があった場合は、マイグレーション後の Business Process Choreographer Explorer レポート作成機能の有効化を実行して、新しいレポート作成機能に切り替えます。
9. 最初にユーザーを認証せずに、Business Process Choreographer API を使用するバージョン 6.0.2 のクライアントを書き込んだ場合は、API を使用する前に、ログインを実行するようにクライアントを変更する必要があります。 Java EE 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・セキュリティーが有効な場合もバージョン 6.0.2 にログインする必要がなくなり、以前のバージョンとの互換性が提供されます。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**」 → 「**アプリケーション・タイプ**」 → 「**WebSphere エンタープライズ・アプリケーション**」を選択します。
   b. 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. 右側のパネルの「詳細プロパティー」の下で、**「ユーザー/グループ・マッピングへのセキュリティー役割」**を選択します。

d. Java EE 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. アプリケーションで Business Process Choreographer EJB API を使用しており、なおかつ EJB スタブが含まれた `bpe137650.jar` ファイルと `task137650.jar` ファイルのいずれかまたは両方をアプリケーションと共にパッケージ化した場合は、これらのユーティリティー JAR ファイルを削除します。

11. バージョン 7.0 にマイグレーションする前に、`faces-config-beans.xml` 構成ファイルを変更して 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『Business Process Choreographer Explorer - カスタマイズおよび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。

12. オプション: ビジネス・プロセス・ナビゲーション・モードを新しいデフォルトに変更します。 バージョン 7.0 から、ビジネス・プロセスのデフォルトのナビゲーション・モードは作業マネージャーを使用します。バージョン 7.0 の前は、デフォルトのナビゲーション・モードは JMS メッセージングを使用していました。ナビゲーション・モードは、マイグレーション中に変更されないため、作業マネージャー・ベースのナビゲーションを使用してパフォーマンスを向上したい場合、関連リンクで説明しているように手動で選択する必要があります。

13. オプション: 反復されるインライン・ヒューマン・タスク用にデータベース保存の動作を変更します。 バージョン 7.0 の前は、複数の「while」ループまたは「repeat-until」ループの一部として処理されたインライン・ヒューマン・タスクは、デフォルトでデータベースに保持されていました。バージョン 7.0 からの新規のデフォルトの動作では、「while」ループまたは「repeat-until」ループが複数回反復された場合、前の反復で処理されたインライン・ヒューマン・タスクはデータベースから削除されます。

    マイグレーション済み環境で、両方のタイプのループに対して以前の動作を維持したい場合は、新しいカスタム・プロパティーを手動で追加する必要があります。管理コンソールで、**「サーバー」** → **「クラスター」** → **「WebSphere Application Server クラスター」** → **「*cluster_name*」**または**「サーバー」** → **「サーバー・タイプ」** → **「WebSphere Application Server」** → **「*server_name*」**をクリックした後、**「ビジネス・インテグレーション」**で**「Business Process Choreographer」**を展開し、**「Business Flow Manager」** → **「カスタム・プロパティー」**をクリックします。次に、

InlineHumanTasks.KeepOverMultipleWhileLoopIterations という名前のプロパティーを値 true で追加します。前の動作が必要なくなった場合は、このカスタム・プロパティーを削除する必要があります。

14. WebSphere Business Monitor を使用して Service Component Architecture (SCA) イベントをモニターする場合は、SCA イベントを使用可能にするカスタム・プロパティーを設定する必要があります。
    a. 管理コンソールで、「**サーバー**」 → 「**クラスター**」 → 「**WebSphere Application Server クラスター**」 → 「***cluster_name***」または「**サーバー**」 → 「**サーバー・タイプ**」 → 「**WebSphere Application Server**」 → 「***server_name***」をクリックした後、「**ビジネス・インテグレーション**」で「**Business Process Choreographer**」を展開し、「**Business Flow Manager**」 → 「**カスタム・プロパティー**」をクリックします。
    b. 「**新規**」をクリックして、新規カスタム・プロパティーを追加します。
    c. 名前 Compat.SCAMonitoringForBFMAPI および値 true を入力します。
    d. 変更を保管します。 この設定は、Business Process Choreographer が構成されているサーバーまたはクラスターを次に再始動したときにアクティブ化されます。

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

ビジネス・プロセス・ナビゲーションのパフォーマンスの向上

WebSphere が提供する Business Space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またはバージョン 6.2.0からバージョン 7.0 にマイグレーションした後には、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 または バージョン 6.2.0 からマイグレーションしており、Business Space が既に構成されている場合は、マイグレーション後に以下のステップを実行しなければBusiness Space を使用することはできません。

手順

1. Business Space バージョン 6.1.2 または バージョン 6.2.0 でカスタム・ウィジェットを使用していた場合は、これらのウィジェットを Business Space バージ

ョン 7.0 で作動可能にするために、いくつかの手動ステップを行う必要があります。詳しくは、「Business Space Development Guide」を参照してください。

2. リモート・エンドポイントで、Business Space ウィジェットが有効になっている場合、それらを手動でマイグレーションする必要があります。 これを行うには、以下の手順を使用します。
   a. 特定の BPM 製品ウィジェットのカタログ・レジストリー・ファイルを、ソースからターゲットのインストール済み環境にコピーします。
   b. 特定の BPM 製品ウィジェットのエンドポイント・ファイルを、ソースからターゲットにコピーします。
   c. ターゲット内の特定の BPM 製品ウィジェットのエンドポイント・ファイルを変更して、TNS URL を更新します。 手順については、クロスセル環境に対する Business Space ウィジェットの有効化を参照してください。
   d. Business Space 上の特定の BPM 製品ウィジェットに関するカタログ情報およびエンドポイント情報を、 `updateBusinessSpaceWidgets` コマンドを使用してターゲットのインストール済み環境に登録します。 詳しくは、『updateBusinessSpaceWidgets コマンド』を参照してください。

タスクの結果

Business Space バージョン 7.0 を使用できます。

注: Business Space バージョン 6.1.2 を使用していた場合は、Business Space バージョン 7.0 を使用する前に、必ずブラウザーのキャッシュを消去してください。これにより、不注意でBusiness Space 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。

ランタイム・マイグレーション・ツールのリファレンス

ランタイム・マイグレーション・ツールは、トポロジー構成、アプリケーション、およびデータベースを WebSphere Process Server バージョン 7.0 にマイグレーションするために使用します。

バージョン間マイグレーションを実行するために必要なランタイム・マイグレーション・ツールは、以下のカテゴリーに分類できます。

『BPM プロファイル・マイグレーション・ウィザード』

129 ページの『BPM プロファイル・コマンド行ツール』

130 ページの『BPM データベース・アップグレード・コマンド行ユーティリティー』

131 ページの『WebSphere Application Server コマンド行ユーティリティー』

BPM プロファイル・マイグレーション・ウィザード

BPM プロファイル・マイグレーション・ウィザードは、プロファイルをマイグレーションするプロセスの手引きを行うグラフィカル・ユーザー・インターフェース (GUI) です。このウィザードを起動するには、`BPMMigrate` コマンドを実行します。

`BPMMigrate` コマンドについて詳しくは、『BPMMigrate コマンド』のトピックを参照してください。

BPM プロファイル・マイグレーション・ウィザードの実行の詳細については、78 ページの『BPM プロファイル・マイグレーション・ウィザードを使用したプロファイルのマイグレーション』を参照してください。

BPM プロファイル・コマンド行ツール

BPMMigrate

BPMMigrate コマンドは、BPM プロファイルのマイグレーションをサポートする BPM プロファイル・マイグレーション・ウィザードを呼び出します。

BPMMigrate コマンドについて詳しくは、『BPMMigrate コマンド』のトピックを参照してください。

BPM プロファイル・マイグレーション・ウィザードの実行の詳細については、78 ページの『BPM プロファイル・マイグレーション・ウィザードを使用したプロファイルのマイグレーション』を参照してください。

BPMSnapshotSourceProfile

BPMSnapshotSourceProfile コマンドは、ソース・プロファイル内の構成ファイルを、プロファイル・マイグレーションのソースとして機能することになるスナップショット・ディレクトリーにコピーします。

BPMSnapshotSourceProfile コマンドについて詳しくは、BPMSnapshotSourceProfile コマンド行ユーティリティーのトピックを参照してください。

BPMCreateTargetProfile

BPMCreateTargetProfile コマンドは、BPMSnapshotSourceProfile コマンドを使用してバックアップされた基本構成情報の一部を使用して、ターゲット・マイグレーション・プロファイルを作成します。

BPMCreateTargetProfile コマンドについて詳しくは、『BPMCreateTargetProfile コマンド』のトピックを参照してください。

BPMMigrateProfile

BPMMigrateProfile コマンドは、ソース・プロファイルをスナップショット・ディレクトリーからターゲット・プロファイルにマイグレーションします。

BPMMigrateProfile コマンドについて詳しくは、『BPMMigrateProfile コマンド』のトピックを参照してください。

BPMMigrateCluster

BPMMigrateCluster コマンドは、クラスターを有効範囲とするアプリケーションと構成情報をマイグレーションします。

BPMMigrateCluster コマンドについて詳しくは、『BPMMigrateCluster コマンド』のトピックを参照してください。

BPMMigrationStatus

BPMMigrationStatus コマンドは、システム上で実行されたマイグレーションの状況を表示します。

BPMMigrationStatus コマンドについて詳しくは、『BPMMigrationStatus コマンド』のトピックを参照してください。

BPMCreateRemoteMigrationUtilities

BPMCreateRemoteMigrationUtilities コマンドは、すべてのコマンドとその前提条件が入っているアーカイブ・ファイルを作成します。それらのコマンドとその前提条件は、マイグレーション対象のソース・プロファイルが入っているシステム上で起動する必要があります。

BPMCreateRemoteMigrationUtilities コマンドについて詳しくは、『BPMCreateRemoteMigrationUtilities コマンド』のトピックを参照してください。

installBRManager

installBRManager コマンドは、ビジネス・ルール・マネージャーをマイグレーションします。

installBRManager コマンドについて詳しくは、『installBRManager コマンド』のトピックを参照してください。

BPM データベース・アップグレード・コマンド行ユーティリティー

migrateDB (Business Process Choreographer)

バージョン 6.1.x または 6.0.2.x からマイグレーションする場合は、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。新しいスキーマにより、ビジネス・プロセスおよびヒューマン・タスクの照会パフォーマンスが向上します。

migrateDB コマンドについて詳しくは、Business Process Choreographer データのマイグレーション・スクリプトのトピックを参照してください。

migrateSchema (Business Space)

migrateSchema コマンド行ユーティリティーは、Business Space データベース・スキーマをマイグレーションするために使用します。

migrateSchema コマンドについて詳しくは、Business Space データベース用の migrateSchema コマンド行ユーティリティーのトピックを参照してください。

updateBspaceData (Business Space)

migrateBSpaceData コマンド行ユーティリティーは、Business Space データをマイグレーションするために使用します。

migrateData コマンドについて詳しくは、migrateBSpaceData コマンド行ユーティリティーのトピックを参照してください。

upgradeSchema (共通データベース)

upgradeSchema コマンド行ユーティリティーは、共通データベース・スキーマをアップグレードするために使用します。

upgradeSchema コマンドについて詳しくは、共通データベース用の upgradeSchema コマンド行ユーティリティーのトピックを参照してください。

WebSphere Application Server コマンド行ユーティリティー

backupConfig

backupConfig コマンドは、ノードの構成をファイルにバックアップする単純なユーティリティーです。

backupConfig コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『backupConfig コマンド』のトピックを参照してください。

convertScriptCompatibility

convertScriptCompatibility コマンドは、WebSphere Application Server バージョン 5.1.x またはバージョン 6.0.x 管理スクリプトの後方互換性をサポートするモードから、完全なバージョン 7.0 構成モデルに構成を変換するために、管理者が使用します。

convertScriptCompatibility コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『convertScriptCompatibility コマンド』のトピックを参照してください。

migrationDisablementReversal

デプロイメント・セルまたは管理対象ノードをロールバックする必要がある場合は、wsadmin コマンドを使用して migrationDisablementReversal.jacl スクリプトを実行します。

migrationDisablementReversal.jacl スクリプトについて詳しくは、WebSphere Application Server インフォメーション・センターの『Network Deployment セルのロールバック』のトピックを参照してください。

restoreConfig

restoreConfig コマンドは、backupConfig コマンドを使用してノードの構成をバックアップした後に、その構成をリストアする場合に使用します。

restoreConfig コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『restoreConfig コマンド』のトピックを参照してください。

startManager

startManager コマンドは、スクリプト記述によってデプロイメント・マネージャーを操作するために使用します。

startManager コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『startManager コマンド』のトピックを参照してください。

startNode

startNode コマンドは、ノード・エージェント・プロセスの構成ファイルを読み取って、launch コマンドを構成します。

startNode コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『startNode コマンド』のトピックを参照してください。

startServer

startServer コマンドは、指定されたサーバー・プロセスの構成ファイルを読み取って、そのサーバー・プロセスを開始します。

startServer コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『startServer コマンド』のトピックを参照してください。

stopManager

stopManager コマンドは、Network Deployment マネージャー・プロセスの構成ファイルを読み取ります。

stopManager コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『stopManager コマンド』のトピックを参照してください。

stopNode

stopNode コマンドは、Network Deployment のノード・エージェント・プロセスの構成ファイルを読み取り、ノード・エージェントのシャットダウンを指示する Java Management Extensions (JMX) コマンドを送信します。

stopNode コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『stopNode コマンド』のトピックを参照してください。

stopServer

stopServer コマンドは、指定されたサーバー・プロセスの構成ファイルを読み取ります。このコマンドは、サーバーのシャットダウンを指示する Java management extensions (JMX) コマンドをサーバーに送信します。

stopServer コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『stopServer コマンド』のトピックを参照してください。

syncNode

syncNode コマンドは、ノードとそのノードが構成されているセルのデプロイメント・マネージャーとの間で、強制的に構成を同期させます。

ノード・エージェント・サーバーは、ノード構成とマスター・セル構成の同期を維持する構成同期サービスを実行します。ノード構成内の問題のためにノード・エージェントを実行できない場合は、再びノード構成をセル構成と強制的に同期させるために、ノード・エージェントが稼働していないときに、syncNode コマンドを使用して同期を実行できます。ノード・エージェントが稼働しており、syncNode コマンドを実行したい場合は、最初にノード・エージェントを停止する必要があります。

syncNode コマンドについて詳しくは、WebSphere Application Server インフォメーション・センターの『syncNode コマンド』のトピックを参照してください。

ランタイム・マイグレーションのトラブルシューティング

WebSphere Process Server の旧バージョンからのマイグレーション時に問題が発生する場合は、このページでトラブルシューティングのヒントを参照してください。

以降のセクションでは、BPM ランタイム・バージョンのマイグレーションで発生する可能性のある特定のエラーおよび例外について説明し、これらの問題を理解して解決するために実行可能なステップを示します。

- 『アプリケーション・インストール・エラー』
- 134 ページの『アプリケーション・サーバー・エラー』
- 134 ページの『ビジネス・ルール・マネージャーが自動的にマイグレーションされない』
- 135 ページの『デプロイメント・マネージャーとの通信エラー』
- 135 ページの『ConnectorException』
- 135 ページの『例外: データベースの接続性、ロード、またはクラス欠落』
- 136 ページの『メモリー不足エラー』
- 136 ページの『プロファイル作成エラー』
- 137 ページの『プロファイル・マイグレーション・エラー』
- 138 ページの『サーブレット・エラー』
- 138 ページの『同期エラー』
- 139 ページの『WebSphere Process Server クライアント・マイグレーション』
- 139 ページの『WSDL 検証例外』

アプリケーション・インストール・エラー

マイグレーション・プロセスで、バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 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 7.0 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 6.2.0、6.1.2、6.1.0、または 6.0.2 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、BPMMigrateProfile 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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 7.0 構成に手動でインストールできます。

アプリケーション・サーバー・エラー

管理対象ノードを バージョン 7.0 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と、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。

ビジネス・ルール・マネージャーが自動的にマイグレーションされない

問題

バージョン 6.0.2 のビジネス・ルール・マネージャーが自動的にマイグレーションされない場合、ビジネス・ルール・マネージャーを起動すると、以下の例外が表示されます。

```
java.lang.ClassNotFoundException:
com.ibm.wbiserver.brules.BusinessRuleManager
```

これは、ビジネス・ルール・ランタイムがリファクタリングし、このクラス (com.ibm.wbiservers.brules.BusinessRuleManager)をバージョン 6.0.2 よりも後のリリースの新規パッケージに入れたために発生します。

説明

最後にマイグレーションしたノードが WebSphere Process Server プロファイルではない場合、ビジネス・ルール・リソースおよびビジネス・ルール・マネージャー・マイグレーション・スクリプトは、使用できません。そのため、ビジネス・ルール・マネージャーは、予期されるようにマイグレーション・プロセス中に自動的にマイグレーションされません。

解決策

システム全体のマイグレーション後に、WebSphere Process Server カスタム・ノードでビジネス・ルール・マネージャー・マイグレーション・スクリプトを実行します。詳しくは、installBRManager コマンド行ユーティリティーを参照してください。

デプロイメント・マネージャーとの通信エラー

場合によっては、マシン上のリソースが不十分なためにマイグレーション・プロセスが失敗する可能性があります。マイグレーションが失敗した場合は、ログ・ファイルを確認して、以下のメッセージがあるかどうかを調べてください。

```
"MIGR0494E: デプロイメント・マネージャーとの通信中に、予期しないエラーが発生しました。
マイグレーションは続行できません。エラーを解決してから WASPreUpgrade ツールを再実行して、
新しいバックアップ・ディレクトリーを作成してください。"
```

このメッセージがログ・ファイル内にある場合は、マシン上のディスク・スペース、メモリーおよび CPU の使用率を確認してください。可能であれば、マシン上の他のプロセスをいくつか停止して、マシン・リソースを解放してから、失敗したマイグレーション・コマンドを再実行してください。

ConnectorException

管理対象ノードをマイグレーションする際に、以下のような ConnectorException が表示された場合は、デプロイメント・マネージャーが動作していることを確認してからコマンドを再実行してください。

```
MIGR0380E: ポート 8879 上で SOAP のコネクター・タイプを使用したデプロイメント・マネージャー・ノード
qaxs06 での JMX 接続は、確立されませんでした。
(MIGR0380E: The JMX connection is not established with the deployment manager
node qaxs06, using connector type of SOAP on port 8879.) WASPostMigration プログラムを終了します。
(The WASPostMigration program is now closing.) ローカル Application Server 環境に変更は加えられません。
(No changes are made to the local Application Server environment.)
com.ibm.websphere.management.exception.ConnectorException:
ADMC0016E: システムはホスト qaxs06 にポート 8879 で接続するための SOAP
コネクターを作成できません。
(ADMC0016E: The system cannot create a SOAP connector to connect to host qaxs06 at port 8879.)
com.ibm.ws.migration.utility.UpgradeException:
com.ibm.websphere.management.exception.ConnectorException:
ADMC0016E: システムはホスト qaxs06 にポート 8879 で接続するための SOAP
コネクターを作成できません。
(ADMC0016E: The system cannot create a SOAP connector to connect to host qaxs06 at port 8879.)
```

例外: データベースの接続性、ロード、またはクラス欠落

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。

メモリー不足エラー

メモリー不足の問題により、BPMSnapshotSourceProfile または BPMMigrateProfile のいずれかのコマンド行ユーティリティーが失敗した場合には、ヒープ・サイズを、マイグレーションする環境のサイズとスコープ、およびマシンが許容するサイズを考慮した値に増やすことができます。

ヒープ・サイズを増やす方法については、「Handling certain Out of Memory conditions when migrating an earlier version of WebSphere Application Server to V6.0.2, V6.1, or 7.0」という技術情報の『Solution 4』で説明する手順を参照してください。

プロファイル作成エラー

構成をマイグレーションする場合、バージョン 7.0 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正しくない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・マイグレーション・エラー

マイグレーション・ウィザードを使用して、Solaris x64 プロセッサー・ベースのシステム上でプロファイルを WebSphere Process Server バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 から バージョン 7.0 にマイグレーションする場合は、BPMMigrateProfile 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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 7.0 のマイグレーション・ウィザードは、BPMMigrateProfile.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 6.2.0、6.1.2、6.1.0、または 6.0.2 ノードを停止すると、バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 用の JVM を 64 ビット・モードで実行しようとします。

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 6.2.0、6.1.2、6.1.0、または 6.0.2 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 BPMMigrateProfile.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. BPMMigrateProfile.sh または BPMMigrateProfile.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 7.0 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。
      - 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。
      - 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 7.0 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。

サーブレット・エラー

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、25 ページの『マイグレーションされるもの』トピックの『ビジネス・ルール・マネージャー』セクションを参照してください。

同期エラー

管理対象ノードの バージョン 7.0 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 7.0 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 7.0 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 のリポジトリーでバージョン 7.0 になっている。
- syncNode 操作を完了しなかったため、管理対象ノードがまだ完了していない。

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。

WebSphere Process Server クライアント・マイグレーション

ソース バージョン 6.2.0、6.1.2、6.1.0、または 6.0.2 からフル・サーバー WebSphere Process Server バージョン 7.0 インストール済み環境へ、WebSphere Process Server クライアント・プロファイルをマイグレーションするときに、ターゲット・プロファイル拡張が正しくありません。ターゲット・プロファイル上のアプリケーションが正しく機能していない可能性があります。問題を修正するには、manageprofiles コマンド行ユーティリティーを使用して、 INSTALL_ROOT/profileTemplates/SCA/*.sdo テンプレートの拡張を追加します。ここで「*」記号は、スタンドアロン・プロファイルの場合は「default」、統合プロファイルの場合は「managed」を表します。

WSDL 検証例外

BPMMigrateProfile コマンドが。以下の WSDL 検証例外を出して失敗した場合、それはインストールに失敗したアプリケーション内の WSDL ファイルに、オペレーション内で定義されていない input エレメント宣言が含まれていることを意味します。この問題を修正するには、input エレメント宣言を定義するか、WSDL ファイルから宣言を除去する必要があります。

WSDL 検証例外

```
java.io.IOException: javax.wsdl.WSDLException: WSDLException (at /wsdl:definitions
/wsdl:import/wsdl:definitions/wsdl:input): faultCode=INVALID_WSDL: Encountered
illegal extension element '{http://schemas.xmlsoap.org/wsdl/}input' in the context
of a 'javax.wsdl.Definition'. Extension elements must be in a namespace other than
WSDL's.
javax.wsdl.WSDLException: WSDLException (at /wsdl:definitions/wsdl:import/wsdl:
definitions/wsdl:input): faultCode=INVALID_WSDL: Encountered illegal extension
element '{http://schemas.xmlsoap.org/wsdl/}input' in the context of a 'javax.wsdl.
Definition'. Extension elements must be in a namespace other than WSDL's.
```

問題の修正方法

以下の手順を使用して、問題を修正します。

1. インストールに失敗したアプリケーションで WSDL ファイルを見つけます。検証で不合格となった WSDL ファイルには、オペレーション内で定義されていない input エレメント宣言が含まれています。

   不合格の WSDL ファイルのサンプル

   注: getLastSellPriceRequest の宣言が、wsdl:operation 宣言の下で定義されていません。

   ```
   wsdl:portType name="EnrollIntf"
   wsdl:operation name="Enrollment"
   wsdl:input message="tns:EnrollmentRequestMsg" name="EnrollmentRequest"/
   wsdl:output message="tns:EnrollmentResponseMsg" name="EnrollmentResponse"/
   /wsdl:operation
   /wsdl:portType

   wsdl:input name="getLastSellPriceRequest"
   wsdlsoap:header message="tns:EnrollmentRequest" part="soap_header" use="literal"/
   wsdlsoap:body parts="EnrollReq" use="literal"/
   /wsdl:input
   ```

2. input 宣言ファイルが必要であるかどうかに応じて、input 宣言に適切な変更を加えます。
   - input 宣言が必要な場合は、それを使用するオペレーションの下に宣言を移動します。
   - input 宣言が不要である場合は、WSDL ファイルからその宣言を除去します。
3. ソース環境でアプリケーションを更新します。
4. ソース環境でアプリケーションが機能することを確認します。
5. BPMSnapshotSourceProfile コマンドまたは BPM プロファイル・マイグレーション・ウィザードから始めて、マイグレーション・ステップを再度実行します。

使用すべきでないフィーチャー

このセクションでは、以下の製品で使用すべきでないフィーチャーの要約を示します。WebSphere Process Server バージョン 7.0、 バージョン 6.2.0、 バージョン 6.1.2、 バージョン 6.1.0、 バージョン 6.0.2、バージョン 6.0.1、およびバージョン 6.0。

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。

- 141 ページの『WebSphere Process Server バージョン 7.0 で使用すべきでないフィーチャー』
- 146 ページの『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』
- 150 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』
- 150 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』
- 155 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』

- 157 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』
- 157 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、非推奨が有効になった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 7.0 で使用すべきでないフィーチャー

バージョン間マイグレーション用コマンド行ユーティリティー

以下のバージョン間マイグレーション用のコマンド行ユーティリティーは、推奨されません。

これらの非推奨コマンド行ユーティリティーは、次の表に概要を示した新しいビジネス・プロセス管理コマンド行ユーティリティーによって置き換えられました。

表 2. 推奨されないバージョン間マイグレーション用のコマンド行ユーティリティー

推奨されないコマンド行ユーティリティー	代わりのコマンド行ユーティリティー
WBIPreUpgrade	*install_root*/bin/BPMSnapshotSourceProfile
WBIPostUpgrade	• *install_root*/bin/BPMCreateTargetProfile • *install_root*/bin/BPMMigrateProfile
WBIProfileUpdate.ant	*install_root*/bin/BPMMigrateCluster
install_root/bin/wbi_migration/wbi_migration	*install_root*/bin/BPMMigrate

Business Process Choreographer ウィジェット

これらのウィジェットは推奨されません。

- マイ・タスク
- 使用可能なタスク
- 作成したタスク
- タスクの作成

推奨されるマイグレーション・アクション

非推奨のウィジェットを使用してすべてのページについて以下のステップを実行した後、新しい「タスク・リスト」ウィジェットを使用します。

1. Widget Wiring Editor で非推奨ウィジェットとのワイヤリングがあれば、それを検査し、キャプチャーします。
2. そのウィジェットの固有の構成オプションを検査し、キャプチャーします。
3. そのウィジェットをページから削除します。

4. ページの同じ位置に「タスク・リスト」ウィジェットを追加します。
5. ウィジェットを非推奨ウィジェットの構成に一致するように構成します。フィルターまたはソートに使用したいプロパティーを可視として選択してあることを確認してください。
6. シナリオ・コンテキストを非推奨ウィジェットに一致するように構成します。
   - マイ・タスク」ウィジェットを置き換えるために、割り当てを受けたタスクを処理します。
   - 「使用可能なタスク」ウィジェットを置き換えるために、使用可能なタスクを評価します。
   - 「作成したタスク」ウィジェットを置き換えるために、開始されたタスク、サービス、およびプロセスの状況を検査します。
7. 前のワイヤリングに一致する明示的ワイヤーを追加します。
8. 明示的ワイヤーを追加してリストを更新し、ユーザー対話によって生じた「タスク情報」ウィジェット内のタスク状態を反映させます。具体的には、以前に非推奨ウィジェットの着信イベント用だったワイヤリングに一致する明示的ワイヤーを、「タスク・リスト」ウィジェットの com.ibm.widget.Refresh イベントに追加します。
   - `com.ibm.task.TaskCreated`
   - `com.ibm.task.TaskActivated`
   - `com.ibm.task.TaskClaimed`
   - `com.ibm.task.TaskReleased`
   - `com.ibm.task.TaskCompleted`
   - `com.ibm.task.TaskDelegated`
   - `com.ibm.task.TaskTerminated`
   - `com.ibm.task.TaskDeleted`
9. 「タスク情報」または「ヒューマン・ワークフロー・ダイアグラム」ウィジェット内でフォーカスがあるリスト内のタスクを強調表示させるために、明示的ワイヤーを追加します。
   - 「タスク情報」ウィジェットの `com.ibm.widget.TabChanged` イベントから、「タスク・リスト」ウィジェットの `com.ibm.widget.Highlight` イベントにワイヤーを追加します。
   - 「ヒューマン・ワークフロー・ダイアグラム」ウィジェットの `com.ibm.widget.FocusChanged` イベントから、「タスク・リスト」ウィジェットの `com.ibm.widget.Highlight` イベントにワイヤーを追加します。

「タスクの作成」ウィジェットを使用する代わりに、新しい「タスク定義リスト」ウィジェットを使用します。

1. そのウィジェットの固有の構成オプションを検査し、キャプチャーします。
2. ビジネス・カテゴリー・フィルターを構成してある場合は、対応するフィルターを使用して照会テーブルを定義し、デプロイします。
3. Widget Wiring Editor で、非推奨ウィジェットとの明示的なワイヤリングがあれば、それを検査し、キャプチャーします。
4. そのウィジェットをページから削除します。

5. ページの同じ位置に「タスク定義リスト」ウィジェットを追加します。
6. ウィジェットを非推奨ウィジェットの構成に一致するように構成します。
7. ビジネス・カテゴリー・フィルターを構成してある場合は、対応する照会テーブル用のタスク・リストを構成します。
8. シナリオ・コンテキストを構成して、タスク、サービス、およびプロセスを作成します。

インターフェース・マップ

インターフェース・マップ・コンポーネン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 Integration Developer 内の既存のインターフェース・マップ・モジュールを、メディエーション・フロー・コンポーネント内の機能を使用するようにマイグレーションすることができます。

サービス・データ・オブジェクト

以下のサービス・データ・オブジェクト・メソッドは推奨されません。

- `com.ibm.websphere.sca.sdo.DataFactory.create(Class interfaceClass);`

推奨されるマイグレーション・アクション

このメソッドが、ビジネス・オブジェクト・フレームワークのバージョン 7.0 を使用して呼び出された場合、「機能はサポートされません」という例外が発生します。ビジネス・オブジェクト・フレームワークのバージョン 6.2 を使用して呼び出した場合は、このメソッドは引き続き機能します。

Business Flow Manager

これらの EJB メソッドは推奨されていません。以下のリストに示す対応するメソッドを使用してください。

表 3. Business Flow Manager の非推奨メソッドと、マイグレーション用の関連メソッド

非推奨メソッド	推奨されるマイグレーション用のメソッド
`interface com.ibm.bpe.api. ExpirationBehavior`	`interface com.ibm.bpe.api. TimerBehavior`
`enum RESCHEDULE in com.ibm.bpe. api.ActivityInstanceActions`	`enum RESCHEDULE_TIMER in com. ibm.bpe.api.ActivityInstanceActions`
`enum RESCHEDULE in com.ibm.bpe. api.ActivityInstanceActionIndex`	`enum RESCHEDULE_TIMER in com. ibm.bpe.api.ActivityInstanceActionIndex`
`Enum REASON_POTENTIAL_SENDER in com.ibm.bpe.api.WorkItemData`	代わりはありません。このメソッドはまだ使用されておらず、今後も使用される予定はありません。

カスタム・プロパティー InlineHumanTasks.KeepOverMultipleWhileLoopIterations 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

CEI イベントまたは監査ログを使用して、同じ情報をキャプチャーしてください。

このカスタム・プロパティーは、前のバージョンとの互換性を維持する目的でバージョン 7.0 に導入されました。このプロパティーは、 Business Process Choreographer がループ内のインライン・ヒューマン・タスクを処理する方法に影響を与えます。バージョン 7.0 より前の動作は適切ではありませんが、この動作を必要とするユーザーもいる可能性があります。このプロパティーが設定されていない場合は、ループ内のインライン・ヒューマン・タスクを使用して履歴情報を取得することはできません。

HTTPdatabinding

推奨されない HTTPdatabinding メソッド、およびその推奨されるマイグレーション・メソッドを以下にリストします。

表 4. HTTPdatabinding の非推奨メソッドと、マイグレーションする関連メソッド

非推奨メソッド	推奨されるマイグレーション用のメソッド
HTTP SOAP メッセージ・データ・バインディング `com.ibm.websphere.http.data.bindings.HTTPStreamDataBinding SOAP`	`SOAPDataHandler`
HTTP XML メッセージ・データ `bindingcom.ibm.websphere.http.data.bindings.HTTPStreamDataBindingXML`	`UTF8XMLDataHandler`
HTTP サービス・ゲートウェイ・メッセージ・データ・バインディング `com.ibm.websphere.http.data.bindings.HTTPServiceGatewayDataBinding`	`NativeBodyDataHandler` と呼ばれる、Web サービス、HTTP、JMS、および WebSphere MQ のすべての着信メッセージを処理する単一のデータ・ハンドラーを使用できます。このハンドラーは、既存のプロトコル依存データ・バインディングと同じように機能します。

インストール

WebSphere Process Server のインストールに IBM Installation Manager が使用されるようになりました。IBM Installation Manager には、製品のインストール時にデプロイメント環境を作成するオプションはありません。

推奨されるマイグレーション・アクション

管理コンソールを使用すると、製品のインストール後にデプロイメント環境を構成することができます。

Oracle データベース・サポート

Oracle バージョン 9 は、バージョン 7.0 ではサポートされていません。

推奨されるマイグレーション・アクション

1. Oracle 9 を使用していて、データベースをまだ 10 または 11 にアップグレードしていない場合は、Oracle の資料の説明に従って、この時点でアップグレードを実行します。
2. ojdbc14.jar または ojdbc5.jar ドライバーを使用している場合、ORACLE_JDBC_DRIVER_PATH WebSphere 変数によって示されているディレクトリーに新規の ojdbc6.jar ドライバーをインストールする必要があります。

WebSphere Application Server にバンドルされた Data Direct ドライバー

WebSphere Application Server にバンドルされた組み込み Data Direct ドライバーは、WebSphere Process Server バージョン 7.0 ではサポートされません。組み込み Data Direct ドライバーのライセンスを購入するか、または MSSQL Server 用の Microsoft JDBC ドライバー (無料で入手可能) をダウンロードする必要があります。

Business Process Choreographer の管理スクリプト

以下の表に、非推奨の ProcessContainer MBean メソッドおよびその管理スクリプト・パラメーターと、推奨される代替メソッドおよびパラメーターを示します。

表 5. ProcessContainer MBean メソッド

非推奨メソッド	推奨されるマイグレーション用のメソッド
`ProcessContainer MBean メソッド deleteCompletedProcessInstances (String state, templateName, validFrom, completedBefore, startedBy)`	`ProcessContainer MBean メソッド deleteCompletedProcessInstances (String[] states, templateName, validFrom, completedAfter, completedBefore, startedBy)`

表 6. スクリプト・パラメーター

非推奨パラメーター	代替パラメーター
`deleteAuditLog.py` スクリプト・パラメーター **-time** および **processtime**。	**-timeUTC** および**-processtimeUTC** を使用してください。
`deleteCompletedProcess Instances.py` スクリプト・パラメーター **-validFrom** および **-completedBefore**	以下のパラメーターを使用してください。**-validFromUTC** および **-completedBeforeUTC**。
`deleteInvalidProcessTemplate.py` スクリプト・パラメーター **-validFrom**	**-validFromUTC** を使用してください。
`deleteInvalidTaskTemplate.py` スクリプト・パラメーター **-validFrom**	**-validFromUTC** を使用してください。
`observerDeleteProcessInstance Data.py` スクリプト・パラメーター **-validFrom**、**-deletedBefore**、および **-reachedBefore**	**-validFromUTC**、**-deletedBeforeUTC**、および **-reachedBeforeUTC**

Human Task Manager

以下の表に、ヒューマン・タスクに関する非推奨のメソッドと、モジュールのマイグレーション時に使用する代替メソッドを示します。

表 7. Human Task Manager の非推奨メソッドと、マイグレーションする関連メソッド

非推奨メソッド	推奨されるマイグレーション用のメソッド
HumanTaskManager.getAbsence()	HumanTaskManager.getUserSubstitution Detail()
HumanTaskManager.getAbsence (String userID)	HumanTaskManager.getUserSubstitution Detail(String userID)
HumanTaskManager.getSubstitutes()	HumanTaskManager.getUserSubstitution Detail()
HumanTaskManager.getSubstitutes (String userID)	HumanTaskManager.getUserSubstitution Detail(String userID)
HumanTaskManager.setAbsence (boolean absence)	Sequence: UserSubstitutionDetail retrievedDetail = HumanTaskManager.getUserSubstitution Detail();retrievedDetail.setStartDate(..);retrievedDetail.setEndDate(..); HumanTaskManager.setUserSubstitution Detail(retrievedDetail);
HumanTaskManager.setAbsence(String userID, boolean absence)	Sequence: UserSubstitutionDetail retrievedDetail = HumanTaskManager.getUserSubstitution Detail(userID);retrievedDetail.setStart Date(..);retrievedDetail.setEndDate(..); HumanTaskManager.setUserSubstitution Detail(userID, retrievedDetail);
HumanTaskManager.setSubstitutes (List substitutes)	Sequence: UserSubstitutionDetail retrievedDetail = HumanTaskManager.getUserSubstitution Detail();retrievedDetail.setSubstitutes (..);HumanTaskManager.setUserSubstitution Detail(retrievedDetail);
HumanTaskManager.setSubstitutes(String userID, List substitutes)	Sequence: UserSubstitutionDetail retrievedDetail = HumanTaskManager.getUserSubstitution Detail(userID);retrievedDetail.set Substitutes(..);HumanTaskManager.set UserSubstitutionDetail(userID, retrieved Detail);

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。

コピーと設定の両方を同時に行うと、コピーまたは設定で発生する可能性がある問題の一部がマスクされます。これは、コピーと設定を別々に実行するのと同様に簡単なので、回避策は単純です。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 6.2.0、6.1.2、6.1.0、または 6.0.2 で `manageprofiles` コマンド行ユーティリティーを使用していて、このコマンドを バージョン 7.0 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 8. 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	cdbSchemaName	DB2 z/OS
ceiDbSubSystemName	dbConnectionLocation	DB2 z/OS
ceiNativeJdbcClassPath	dbJDBCClasspath	DB2 iSeries® Native
ceiCollection	cdbSchemaName	DB2 iSeries Native
ceiToolboxJdbcClassPath	dbJDBCClasspath	DB2 iSeries Toolbox
ceiCollection	cdbSchemaName	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 コマンド行ユーティリティー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>
```

Java EE ロールの BPEAPIUser と TaskAPIUser のマップに使用する値「Everyone」

Java EE 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

Java EE 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 9. *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
com.ibm.wbiserver.manualrecovery.FailedEventWithParameters	com.ibm.wbiserver.manualrecovery.SCAEvent
com.ibm.wbiserver.manualrecovery.FailedEventManager	
• リスト getFailedEventsForDestination(String destModuleName, String destComponentName, String destMethodName, int pagesize) により FailedEventReadException を throw • リスト getFailedEventsForTimePeriod(Date begin, Date end, int pagesize) により FailedEventReadException を throw	リスト <FailedEvent> queryFailedEvents(QueryFilters queryFilters, int offset, int maxRows) により FailedEventReadException を throw
FailedEventWithParameters getFailedEventWithParameters (String msgId) により FailedEventDataException を throw	SCAEvent getEventDetailForSCA(FailedEvent failedEvent) により FailedEventDataException を throw
void discardFailedEvents(String[] msgIds) により DiscardFailedException を throw	void discardFailedEvents(List<FailedEvent> failedEvents) が DiscardFailedException により throw
void resubmitFailedEvents(String[] msgIds) により ResubmissionFailedException を throw	void resubmitFailedEvents(List failedEvents) により ResubmissionFailedException を throw
FailedEventManagerMBean.xml	
• getFailedEventsForDestination • getFailedEventsForTimePeriod	queryFailedEvents
getFailedEventWithParameters	getEventDetailForSCA
discardFailedEvents	次のパラメーターでの discardFailedEvents • name="failedEvents" • description="失敗したイベントのリスト" • type="java.util.List"
resubmitFailedEvents	resubmitFailedEvents • name="failedEvents" • description="失敗したイベントのリスト" • type="java.util.List"

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 エンティティー Bean を変換するか、または CMP エンティティー 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す。WebSphere Process Server は、このソフトウェアを使用せず、また依存関係もありませんが、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「cdbSchemaName」が追加されている) を使用してください。

- `public String observerForceRemoveInstanceData(String dataSourceName, String cdbSchemaName, String state, String templateName, String validFrom, String completedBefore )`
- `public String observerRemoveDeletedInstancesData(String dataSourceName, String cdbSchemaName, String completedBefore)`
- `public String observerRemoveInstanceDataOfTemplate(String dataSourceName, String cdbSchemaName, 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。

```
<sldap:attribute name="attribute name"
                 objectclass="LDAP object class"
                 usage="simple">
</sldap: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・アクティビティーを取得し、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、アクティビティー状態を確認します。

- 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) アクティビティー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す。

- 部門メンバーの E メール・アドレス
- グループ・メンバーの E メール・アドレス
- フィルターされたユーザーを除くグループ・メンバーの E メール・アドレス
- グループ検索の E メール・アドレス
- ロール・メンバーの E メール・アドレス
- ユーザーの E メール・アドレス
- ユーザー ID ごとのユーザーの E メール・アドレス

推奨されるマイグレーション・アクション

E メール・アドレスおよび優先言語は、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・コンテナーおよびヒューマン・タスク・コンテナーの構成)

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・コンテナーおよびヒューマン・タスク・コンテナーを構成することは、推奨されませ

ん。ビジネス・プロセス・コンテナーおよびヒューマン・タスク・コンテナーは、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・コンテナーおよびヒューマン・タスク・コンテナーの構成中に、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 iterfaceClass);
- 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」という例外が発生します。

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) は現在は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 レパ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、同様のレパ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ー・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)

    代わりのメソッド: createStoredQuery(String storedQueryName, String selectClause, String whereClause, String orderByClause, Integer threshold, TimeZone timezone)
  - getWorkListNames()

    代わりのメソッド: getStoredQueryNames()
  - deleteWorkList(String workListName)

    代わりのメソッド: deleteStoredQuery(String storedQueryName)
  - getWorkList(String workListName)

    代わりのメソッド: getStoredQuery(String storedQueryName)
  - executeWorkList(String workListName)

    代わりのメソッド: 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。

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()
- 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。

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()
- 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。

  getInputMessageTypeTypeSystemName()

  getOutputMessageTypeTypeSystemName()
- 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。

  getInputMessageTypeTypeSystemName()

推奨されるマイグレーション・アクション

代わりのメソッドがある場合は、そのメソッドを使用してください。

Human Task Manager の機能性の変更のため、WebSphere Process Server バージョン 6.0 の汎用プロセス API では、以下のメソッドは推奨されません。

- HumanTaskManager Bean では、以下のメソッドは使用すべきではありません。WebSphere Process Server バージョン 6.0 で使用する代わり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。

  getInputMessageTypeName()

  getOutputMessageTypeName()

推奨されるマイグレーション・アクション

代わりのメソッドがある場合は、そのメソッドを使用してください。

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 (アクティビティーおよび条件) 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ー内部のカスタム・プロパティーにアクセスする 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 2 章 マイグレーション: 継承製品

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、『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、 229 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、 229 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、WebSphere Process Server インフォメーション・センターの『マイグレーション: バージョン間』セクションで 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。

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

218 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

175 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、コンテナー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・プロパティーに名前を付けます。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 プロパティー名では、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ーを名前変更して、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・ユーティリティー、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 Java EE 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 コンテナー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 コンテナー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・ユーティリティー・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るためです。

可能な限り広い範囲で Activity Editor ツールを使用してください。これによって、ロジックはメタデータによって記述されるため、新規の成果物に変換しやすくなります。

開発が必要なあらゆる Java コード断片で、コードを可能な限り単純かつアトミックにすることを IBM ではお勧めします。Javaコードにおける高度化のレベルは、基本の評価、操作および計算を伴うスクリプト記述、データ・フォーマット付け、型変換などと同等である必要があります。より広範な、または高度なアプリケーショ

ン・ロジックが必要な場合は、WebSphere Application Server で稼働している EJB を使用してロジックをカプセル化すること、および Web サービス呼び出しを使用して、WebSphere InterChange Server 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 Java SE 環境で実行中であっても、Java EE 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、Java EE 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ようにしてください。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、コンテナー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 リポジトリーで、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: コネクター上の逆マップ

応答フローを持つアプリケーションのアウトバウンド・コネクターに逆マップが関連付けられているかどうかを判別します。関連付けられていない場合は、汎用 SMO が応答で返されます。

WebSphere InterChange Server または WebSphere Business Integration Server Express 用に作成した一部のアプリケーションには、アウトバウンド・コネクター・ポートの逆マップが組み込まれていない場合があります。応答によって返される結果の内容に関心がない場合に、このような状態が生じることがあります。ただし、WebSphere Process Server では、両方向呼び出しで、すべてのメディエーション・フロー・コンポーネントに有効な SMO が返されることが必要です。したがって、WebSphere Process Server は、逆マップを持たないメディエーション・フロー・コンポーネントに汎用 SMO を返します。

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 リポジトリー JAR ファイルが必要です。この JAR ファイルは、マイグレーション対象のアプリケーションに関して、必要なものを完備していなければなりません。つまり、JAR ファイル内のいずれかの成果物によって参照されるすべての成果物も、JAR ファイル内に含まれていることが必要です。

生成されるリポジトリー JAR ファイルが必要なものを完備するように、サーバー・リポジトリー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、リポジトリーが検証されます。リポジトリーが有効である場合、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。リポジトリー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。リポジトリー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、リポジトリー成果物をエクスポートし、リポジトリー 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ー定義ごとに 1 つずつ作成されます。ビジネス・オブジェクト、マップ、およびリレーションシップなど、その他の成果物では、生成された各 JAR ファイルに、入力 JAR ファイルから生成されたこれらすべての成果物のコピーが格納されます。マイグレーションされたコラボレーション・オブジェクトやコネクターがない場合は、共有されるすべての成果物から成るモジュールを格納した 1 つの JAR ファイルが作成されます。新しい

JAR ファイルが作成された後、**serviceDeploy** コマンドを使用して、WebSphere Process Server にデプロイできる EAR ファイルを生成します。

WebSphere InterChange Server 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 WBIScheduler 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ー定義ごとに、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ー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。

『セキュリティー』

177 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)』

177 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』

178 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』

179 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』

179 ページの『Access Enterprise JavaBean (EJB) サポート』

180 ページの『DynamicSend API 構成』

180 ページの『BaseCollaboration.dynamicSend メソッド呼び出しの使用可能化』

183 ページの『イベント順序付けマイグレーション』

183 ページの『失敗したイベント』

183 ページの『マップのマイグレーション』

183 ページの『コラボレーションのマイグレーション』

185 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』

185 ページの『WebSphere Process Server での logError API E メール通知の使用可能化』

186 ページの『WebSphere Process Server での非同期呼び出しの処理』

186 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』

187 ページの『WebSphere Process Server での相関値の処理』

187 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、189 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。共用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用ライブラリー・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 リポジトリー 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、Java EE 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、以前のバージョンとの互換性を保つために 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、 179 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 リポジトリー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 リポジトリー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 リポジトリーを、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ーを格納する 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。
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. 実装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_ExportName*Impl.java という名前を付けます。

   b. PreRoute_*ModuleName_ExportName*Impl.java を編集します。メソッド名を、locateService.TestBOInterfacePartner から locateService_*InterfaceName*Partner に変更します (*InterfaceName* はメソッドです)。TestBOInterfacePartner を *InterfaceName*Partner に変更します。

   c. PreRoute_*ModuleName_ExportName*Impl.java で「locateService_TestBOInterfacePartner」を検索し、その名前を locateService_*InterfaceName*Partner に変更します。

   d. *PreRoute_ModuleName_ExportName*Impl.java で「this.locateService_InterfaceNamePartner().invoke("Sync", tmpres)」を検索し、その名前を「this.locateService_InterfaceNamePartner.invoke ("Sync_ExportName", tmpres)」に変更して保存します。

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_ExportName*Impl 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、プロパティー・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、相関セットおよび非同期 in 呼び出しがある
- サポートされないもの:
  - マイグレーションでは、トリガー・ポートが 1 つ以上で、相関セットおよび非同期 in 呼び出しがある場合をサポートしません。この場合、結果として生成された成果物は、上述の最初の場合であるものとしてマイグレーションされます。さらに、欠落している SCA コンポーネントを手動で作成して、それらを適切に接続する必要があります。

SCA コンポーネント:

- エクスポート: コラボレーション・オブジェクトに関連付けられたコラボレーション・テンプレート内で定義されているトリガー・ポートのそれぞれに、エクスポートが作成されます。エクスポート名は *TriggeringPortName* です。
- BPEL へのエクスポート: データをエクスポートから BPEL ファイルへマップするインターフェース・マップが生成されます。インターフェース・マップ名は *Export_To_BPELname* です。トリガー・ポートがちょうど 1 つであり、コラボレーション・テンプレートに非同期 in 呼び出しがある場合、追加の SCA コンポーネントが作成されます。マイグレーションによって、インターフェース・マップが 1 つだけではなく 2 つ生成されます。1 つは同期呼び出し用、もう 1 つは非同期呼び出し用です。これら 2 つのインターフェース・マップのどちらに従うかを決定するには、Java コンポーネントを使用します。
- BPEL: すべてのトリガー・ポートで、エクスポートはインターフェース・マップに関連付けられ、そのインターフェース・マップは BPEL ファイルのインスタンスにマップされます。
- インポートする BPEL: トリガー・ポートでもそれ以外でも、すべてのポートに、BPEL ファイルをインポートへマップするインターフェース・マップがあります。インターフェース・マップ名は *BPEL_to_Port* です。

- インポート: 最後に、インポート・ファイルが作成されます。インポート名は *ConnectorName_BONameBG* です。

コラボレーション・テンプレートがどのように WebSphere Process Server BPEL ファイルにマイグレーションされるかについて詳しくは、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に、エラーのフラグが立てられます。**原因:** WebSphere Process Server の BPEL プロセスからパートナーを呼び出す場合は、呼び出しで使用されるすべてのオブジェクトを BPEL 変数として宣言し、使用されるオブジェクトのタイプを判別できるようにする必要があります。マイグレーション時には、コラボレーション・テンプレート内のポート宣言のみを調べて、宣言する必要がある BPEL 変数が判断されます。グローバル変数または ICS コラボレーション・テンプレート定義内の別の場所の断片で宣言される変数の場合は、マイグレーション・コードによってオブジェクト・タイプを確実に判別することはできないため、マイグレーションによって生成される BPEL ファイルでのオブジェクトに対しては BPEL 変数が宣言されません。**解決策:** マイグレーション後、呼び出し時に参照される変数を BPEL 変数として定義する必要があります。

WebSphere Process Server での logError API E メール通知の使用可能化

問題: WebSphere Process Server へのマイグレーション後、WebSphere InterChange Server 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: 同じコネクターで 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ー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。同じコネクターでトリガー・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 コンポーネントの実装コードには、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 throw 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_root*AppScheduler/lib ディレクトリー、または WebSphere Process Server 6.0.2 の *install_root*/lib ディレクトリーから入手できます。その JAR ファイルを WebSphere Process Server 6.0.1.x の lib ディレクトリーにコピーして、既存の JAR ファイルと置き換えます。WebSphere Process Server は、lib ディレクトリー内のすべてのファイルを拡張子に関係なく JAR ファイルとして選択す

るので、既存の JAR ファイルの名前変更を行って lib ディレクトリーに残すことはしないでください。次に、サーバーまたはクラスターを再始動し、WebSphere Process Server が新しい JAR ファイルを選択するようにします。

WebSphere Process Server での相関値の処理

問題: WebSphere Process Server において、新規イベントが既存の相関値を使用しようとすると失敗します。そのような場合には、

```
CWWBE0074E: Correlation violation in activity 'null' for correlation set
'CorrelationSetA'java.sql.
SQLException: Could not insert new row - duplicate value in a UNIQUE INDEX column
```

というエラー・メッセージが表示されます。**原因:** WebSphere InterChange Server 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 リポジトリー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の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、リポジトリーを更新して再度マイグレーションするか、または出力された成果物を編集して、問題を取り除くことができます。

その他の検証エラーは、これらの成果物が何もないところから作成されるものとして解決する必要があります。一般的な成果物のエラーおよびその解決策の概要を説明するバリデーターの資料を参照してください。自動化マイグレーションでは、必然的に、ユーザーのプログラムの意図を完全に明らかにすることはできません。できるだけ実際に近い推測をすることのみ可能です。そのため、検証エラーがない場合でも、マイグレーション済み成果物が意図したとおりに実行されない可能性があります。すべての成果物を検討して、各成果物の意図した目的が、マイグレーション済みの内容によって達成されることを確認する必要があります。

関連概念

217 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。

218 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

173 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

164 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・セキュリティーの構成

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ー構成ステップを実行します。

始める前に

最初に、WebSphere Process Server のセキュリティーを構成する必要があります。『アプリケーションとその環境の保護』を参照してください。特に、『アダプターの保護』および『エンドツーエンド・セキュリティーの構築』に記載されたステップを必ず完了しておいてください。さらに、モジュールごとの EAR ファイルをインストールします。詳しくは、『セキュア・アプリケーションのデプロイ (インストール)』を参照してください。

このタスクについて

上記のタスクを実行すると、以下に示す構成ステップを実行できるようになります。

- メッセージ駆動型 Bean をアクティベーション・スペックにバインドする
- リソース参照をリソースにマップする
- セキュリティー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: セキュリティー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 プロパティー」の下で、「**Message Driven Bean リスナー・バインディング**」を選択します。
   d. インポートまたはエクスポート EJB (「_import」または「_export」で始まる EJB 名で示される) ごとに、「バインディング」列の下の「ActivationSpec 認証別名」フィールドで「**SCA_Auth_Alias**」を指定します。
   e. 「**OK**」を選択し、「**保管**」を選択します。
2. リソース参照をリソースにマップします。
   a. 管理コンソールで、「**アプリケーション**」>「**エンタープライズ・アプリケーション**」を選択します。
   b. 右側のパネルで、インストールしたばかりのアプリケーションの名前を選択します。 (名前の左側のチェック・ボックスではなく、名前を選択してください。)

c. 右側のパネルの「参照」の下で、「**リソース参照**」を選択します。

d. javax.jms.ConnectionFactory の下の「認証方式の指定:」フィールドで、「**デフォルト・メソッドの使用 (多対 1 のマッピング)**」ラジオ・ボタンを選択します。

e. 「**認証データ入力の選択 (Select authentication data entry)**」プルダウン・メニューで、「**SCA_Auth_Alias**」を選択します。

f. すべてのモジュールを選択するには、チェック・ボックスにチェック・マークを付けます。

g. 「**適用**」を選択してから、「**OK**」を選択し、次に「**保管**」を選択します。

3. セキュリティー役割をユーザー・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「詳細プロパティー」の下で、「**ユーザー/グループ・マッピングへのセキュリティー役割**」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「詳細プロパティー」の下で、「**ユーザー RunAs ロール**」を選択します。

d. ステップ 3 でマップしたロールの横にあるチェック・ボックスを選択します。

e. ステップ 3e で選択したユーザー名に対応するユーザー名とパスワードを、それぞれ「**ユーザー名**」フィールドと「**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

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・セキュリティーの構築

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、getByteArrayFromSDO()、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):BusinesObjectInterface
- createBusinessObject(String, Locale):BusinesObjectInterface
- createBusinessObject(String, String):BusinesObjectInterface
- createContainer(String):CxObjectContainerInterface
- generateMsg(int, int, int, int, int, Vector):String
- generateMsg(int, int, int, int, Vector):String
- getBlankValue():String
- getEncoding():String
- getIgnoreValue():String
- getLocale():String
- getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)
- getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)
- isBlankValue(Object):boolean
- isIgnoreValue(Object):boolean
- isTraceEnabled(int):boolean
- logMsg(String)
- logMsg(String, int)
- traceWrite(int, String)

JavaConnectorUtilDH
datahandler/
wbi/
ibm/
com/

- getSDOFromString(String inputString, String sdoName, String metaObjectName, String mimeType)
- getStringFromSDO(DataObject sdo, String metaObjectName, String mimeType)

BusObj
Collaboration/

- BusObj(DataObject)
- BusObj(String)
- BusObj(String, Locale)
- copy(BusObj)
- duplicate():BusObj
- equalKeys(BusObj):boolean
- equals(Object):boolean
- equalsShallow(BusObj):boolean
- exists(String):boolean
- get(int):Object
- get(String):Object
- getBoolean(String):boolean
- getBusObj(String):BusObj
- getBusObjArray(String):BusObjArray
- getCount(String):int
- getDouble(String):double
- getFloat(String):float
- getInt(String):int
- getKeys():String
- getLocale():java.util.Locale
- getLong(String):long
- getLongText(String):String
- getString(String):String
- getType():String
- getValues():String
- getVerb():String
- isBlank(String):boolean
- isKey(String):boolean
- isNull(String):boolean
- isRequired(String):boolean
- keysToString():String
- set(BusObj)
- set(int, Object)
- set(String, boolean)
- set(String, double)
- set(String, float)
- set(String, int)
- set(String, long)
- set(String, Object)
- set(String, String)

- setContent(BusObj)
- setDefaultAttrValues()
- setKeys(BusObj)
- setLocale(java.util.Locale)
- setVerb(String)
- setVerbWithCreate(String, String)
- setWithCreate(String, boolean)
- setWithCreate(String, BusObj)
- setWithCreate(String, BusObjArray)
- setWithCreate(String, double)
- setWithCreate(String, float)
- setWithCreate(String, int)
- setWithCreate(String, long):
- setWithCreate(String, Object)
- setWithCreate(String, String)
- toString():String
- validData(String, boolean):boolean
- validData(String, BusObj):boolean
- validData(String, BusObjArray):boolean
- validData(String, double):boolean
- validData(String, float):boolean
- validData(String, int):boolean
- validData(String, long):boolean
- validData(String, Object):boolean
- validData(String, String):boolean

BusObjArray
Collaboration/

- addElement(BusObj)
- duplicate():BusObjArray
- elementAt(int):BusObj
- equals(BusObjArray):boolean
- getElements():BusObj[]
- getLastIndex():int
- max(String):String
- maxBusObjArray(String):BusObjArray
- maxBusObjs(String):BusObj[]
- min(String):String
- minBusObjArray(String):BusObjArray
- minBusObjs(String):BusObj[]

- removeAllElements()
- removeElement(BusObj)
- removeElementAt(int)
- setElementAt(int, BusObj)
- size():int
- sum(String):double
- swap(int, int)
- toString():String

BaseDLM
DLM/

- BaseDLM(BaseMap)
- getDBConnection(String):CwDBConnection
- getDBConnection(String, boolean):CwDBConnection
- getName():String
- getRelConnection(String):DtpConnection
- implicitDBTransactionBracketing():boolean
- isTraceEnabled(int):boolean
- logError(int)
- logError(int, Object[])
- logError(int, String)
- logError(int, String, String)
- logError(int, String, String, String)
- logError(int, String, String, String, String)
- logError(int, String, String, String, String, String)
- logError(String)
- logInfo(int)
- logInfo(int, Object[])
- logInfo(int, String)
- logInfo(int, String, String)
- logInfo(int, String, String, String)
- logInfo(int, String, String, String, String)
- logInfo(int, String, String, String, String, String)
- logInfo(String)
- logWarning(int)
- logWarning(int, Object[])
- logWarning(int, String)
- logWarning(int, String, String)
- logWarning(int, String, String, String)
- logWarning(int, String, String, String, String)

- logWarning(int, String, String, String, String, String)
- logWarning(String)
- raiseException(RunTimeEntityException)
- raiseException(String, int)
- raiseException(String, int, Object[])
- raiseException(String, int, String)
- raiseException(String, int, String, String)
- raiseException(String, int, String, String, String)
- raiseException(String, int, String, String, String, String)
- raiseException(String, int, String, String, String, String, String)
- raiseException(String, String)
- releaseRelConnection(boolean)
- trace(int, int)
- trace(int, int, Object[])
- trace(int, int, String)
- trace(int, int, String, String)
- trace(int, int, String, String, String)
- trace(int, int, String, String, String, String)
- trace(int, int, String, String, String, String, String)
- trace(int, String)
- trace(String)

CwDBConnection
CwDBConnection/
CxCommon/

- beginTransaction()
- commit()
- executePreparedSQL(String)
- executePreparedSQL(String, Vector)
- executeSQL(String)
- executeSQL(String, Vector)
- executeStoredProcedure(String, Vector)
- getUpdateCount():int
- hasMoreRows():boolean
- inTransaction():boolean
- isActive():boolean
- nextRow():Vector
- release()
- rollback()

CwDBConstants
CwDBConnection/
CxCommon/

- PARAM_IN - 0
- PARAM_INOUT - 1
- PARAM_OUT - 2

CwDBStoredProcedureParam
CwDBConnection/
CxCommon/

- CwDBStoredProcedureParam(int, Array)
- CwDBStoredProcedureParam(int, BigDecimal)
- CwDBStoredProcedureParam(int, boolean)
- CwDBStoredProcedureParam(int, Boolean)
- CwDBStoredProcedureParam(int, byte[])
- CwDBStoredProcedureParam(int, double)
- CwDBStoredProcedureParam(int, Double)
- CwDBStoredProcedureParam(int, float)
- CwDBStoredProcedureParam(int, Float)
- CwDBStoredProcedureParam(int, int)
- CwDBStoredProcedureParam(int, Integer)
- CwDBStoredProcedureParam(int, java.sql.Blob)
- CwDBStoredProcedureParam(int, java.sql.Clob)
- CwDBStoredProcedureParam(int, java.sql.Date)
- CwDBStoredProcedureParam(int, java.sql.Struct)
- CwDBStoredProcedureParam(int, java.sql.Time)
- CwDBStoredProcedureParam(int, java.sql.Timestamp)
- CwDBStoredProcedureParam(int, Long)
- CwDBStoredProcedureParam(int, String)
- CwDBStoredProcedureParam(int, String, Object)
- getParamType():int getValue():Object

DataHandler (抽象クラス)
DataHandlers/
crossworlds/
com/

- createHandler(String, String, String):DataHandler
- getBO(InputStream, Object):BusinessObjectInterface
- getBO(Object, BusinessObjectInterface, Object)
- getBO(Object, Object):BusinessObjectInterface
- getBO(Reader, BusinessObjectInterface, Object) (抽象メソッド)
- getBO(Reader, Object):BusinessObjectInterface (抽象メソッド)

- getBO(String, Object):BusinessObjectInterface
- getBOName(InputStream):String
- getBOName(Reader):String
- getBOName(String):String
- getBooleanOption(String):boolean
- getEncoding():String
- getLocale():Locale
- getOption(String):String
- getStreamFromBO(BusinessObjectInterface, Object):InputStream (抽象メソッド)
- getStringFromBO(BusinessObjectInterface, Object):String (抽象メソッド)
- setConfigMOName(String)
- setEncoding(String)
- setLocale(Locale)
- setOption(String, String)
- traceWrite(String, int)

NameHandler (抽象クラス)
DataHandlers/
crossworlds/
com/

- getBOName(Reader, String):String) (抽象メソッド)

ConfigurationException (extends java.lang.Exception)
Exceptions/
DataHandlers/
crossworlds/
com/

MalformedDataException (extends java.lang.Exception)
Exceptions/
DataHandlers/
crossworlds/
com/

NotImplementedException (extends java.lang.Exception)
Exceptions/
DataHandlers/
crossworlds/
com/

BusinessObjectInterface
CxCommon/

- clone():Object
- dump():String
- getAppText():String

- getAttrCount():int
- getAttrDesc(int):CxObjectAttr
- getAttrDesc(String):CxObjectAttr
- getAttribute(String):Object
- getAttributeIndex(String):int
- getAttributeType(int):int
- getAttributeType(String):int
- getAttrName(int):String
- getAttrValue(int):Object
- getAttrValue(String):Object
- getBusinessObjectVersion():String
- getDefaultAttrValue(int):String
- getDefaultAttrValue(String):String
- getLocale():String
- getName():String
- getParentBusinessObject():BusinessObjectInterface
- getVerb():String
- getVerbAppText(String):String
- isBlank(int):boolean
- isBlank(String):boolean
- isIgnore(int):boolean
- isIgnore(String):boolean
- isVerbSupported(String):boolean
- makeNewAttrObject(int):Object
- makeNewAttrObject(String):Object
- setAttributeWithCreate(String, Object)
- setAttrValue(int, Object)
- setAttrValue(String, Object)
- setDefaultAttrValues()
- setLocale(Locale)
- setLocale(String)
- setVerb(String)

CxObjectAttr
CxCommon/

- BOOLEAN
- BOOLSTRING
- DATE
- DATESTRING
- DOUBLE

- DOUBSTRING
- FLOAT
- FLTSTRING
- INTEGER
- INTSTRING
- INVALID_TYPE_NUM
- INVALID_TYPE_STRING
- LONGTEXT
- LONGTEXTSTRING
- MULTIPLECARDSTRING
- OBJECT
- SINGLECARDSTRING
- STRING
- STRSTRING
- equals(Object):boolean
- getAppText():String
- getCardinality():String
- getDefault():String
- getMaxLength():int
- getName():String
- getRelationType():String
- getTypeName():String
- getTypeNum():String
- hasCardinality(String):boolean
- hasName(String):boolean
- hasType(String):boolean
- isForeignKeyAttr():boolean
- isKeyAttr():boolean
- isMultipleCard():boolean
- isObjectType():boolean
- isRequiredAttr():boolean
- isType(Object):boolean

CxObjectContainerInterface
CxCommon/

- getBusinessObject(int):BusinessObjectInterface
- getObjectCount():int
- insertBusinessObject(BusinessObjectInterface)
- removeAllObjects()
- removeBusinessObjectAt(int)

- setBusinessObject(int, BusinessObjectInterface)

DtpConnection
Dtp/
CxCommon/

- beginTran()
- commit()
- executeSQL(String)
- executeSQL(String, Vector)
- executeStoredProcedure(String, Vector)
- getUpdateCount():int
- hasMoreRows():boolean
- inTransaction():boolean
- isActive():boolean
- nextRow():Vector
- rollback()

DtpDataConversion
Dtp/
CxCommon/

- BOOL_TYPE - 4
- CANNOTCONVERT - 2
- DATE_TYPE - 5
- DOUBLE_TYPE - 3
- FLOAT_TYPE - 2
- INTEGER_TYPE - 0
- LONGTEXT_TYPE - 6
- OKTOCONVERT - 0
- POTENTIALDATALOSS - 1
- STRING_TYPE - 1
- UNKNOWN_TYPE - 999
- getType(double):int
- getType(float):int
- getType(int):int
- getType(Object):int
- isOKToConvert(int, int):int
- isOKToConvert(String, String):int
- toBoolean(boolean):Boolean
- toBoolean(Object):Boolean
- toDouble(double):Double
- toDouble(float):Double
- toDouble(int):Double

- toDouble(Object):Double
- toFloat(double):Float
- toFloat(float):Float
- toFloat(int):Float
- toFloat(Object):Float
- toInteger(double):Integer
- toInteger(float):Integer
- toInteger(int):Integer
- toInteger(Object):Integer
- toPrimitiveBoolean(Object):boolean
- toPrimitiveDouble(float):double
- toPrimitiveDouble(int):double
- toPrimitiveDouble(Object):double
- toPrimitiveFloat(double):float
- toPrimitiveFloat(int):float
- toPrimitiveFloat(Object):float
- toPrimitiveInt(double):int
- toPrimitiveInt(float):int
- toPrimitiveInt(Object):int
- toString(double):String
- toString(float):String
- toString(int):String
- toString(Object):String

DtpDate
Dtp/
CxCommon/

- DtpDate()
- DtpDate(long, boolean)
- DtpDate(String, String)
- DtpDate(String, String, String[], String[])
- addDays(int):DtpDate
- addMonths(int):DtpDate
- addWeekdays(int):DtpDate
- addYears(int):DtpDate
- after(DtpDate):boolean
- before(DtpDate):boolean
- calcDays(DtpDate):int
- calcWeekdays(DtpDate):int
- get12MonthNames():String[]
- get12ShortMonthNames():String[]

- get7DayNames():String[]
- getCWDate():String
- getDayOfMonth():String
- getDayOfWeek():String
- getHours():String
- getIntDay():int
- getIntDayOfWeek():int
- getIntHours():int
- getIntMilliSeconds():int
- getIntMinutes():int
- getIntMonth():int
- getIntSeconds():int
- getIntYear():int
- getMaxDate(BusObjArray, String, String):DtpDate
- getMaxDateBO(BusObj[], String, String):BusObj[]
- getMaxDateBO(BusObjArray, String, String):BusObj[]
- getMinDate(BusObjArray, String, String):DtpDate
- getMinDateBO(BusObj[], String, String):BusObj[]
- getMinDateBO(BusObjArray, String, String):BusObj[]
- getMinutes():String
- getMonth():String
- getMSSince1970():long
- getNumericMonth():String
- getSeconds():String
- getShortMonth():String
- getYear():String
- set12MonthNames(String[], boolean)
- set12MonthNamesToDefault()
- set12ShortMonthNames(String[])
- set12ShortMonthNamesToDefault()
- set7DayNames(String[])
- set7DayNamesToDefault()
- toString():String
- toString(String):String
- toString(String, boolean):String

DtpMapService
Dtp/
CxCommon/

- runMap(String, String, BusObj[], CxExecutionContext):BusObj[]

DtpSplitString
Dtp/
CxCommon/

- DtpSplitString(String, String)
- elementAt(int):String
- firstElement():String
- getElementCount():int
- getEnumeration():Enumeration
- lastElement():String
- nextElement():String
- prevElement():String
- reset()

DtpUtils
Dtp/
CxCommon/

- padLeft(String, char, int):String
- padRight(String, char, int):String
- stringReplace(String, String, String):String
- truncate(double):int
- truncate(double, int):double
- truncate(float):int
- truncate(float, int):double
- truncate(Object):int
- truncate(Object, int):double

BusObjInvalidVerbException (extends InterchangeExceptions)
Exceptions/
CxCommon/

- getFormattedMessage()

IdentityRelationship
relationship/
utilities/
crossworlds/
com/

- addMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- foreignKeyLookup(String, String, BusObj, String, BusObj, String, CxExecutionContext)
- foreignKeyXref(String, String, String, BusObj, String, BusObj, String, CxExecutionContext)

- maintainChildVerb(String, String, String, BusObj, String, BusObj, String, CxExecutionContext, boolean, boolean)
- maintainCompositeRelationship(String, String, BusObj, Object, CxExecutionContext)
- maintainSimpleIdentityRelationship(String, String, BusObj, BusObj, CxExecutionContext)
- updateMyChildren(String, String, BusObj, String, String, String, String, CxExecutionContext)

MapExeContext
Dtp/
CxCommon/

- ACCESS_REQUEST - "SUBSCRIPTION_DELIVERY"
- ACCESS_RESPONSE - "ACCESS_RETURN_REQUEST"
- EVENT_DELIVERY - "SUBSCRIPTION_DELIVERY"
- SERVICE_CALL_FAILURE - "CONSUME_FAILED"
- SERVICE_CALL_REQUEST - "CONSUME"
- SERVICE_CALL_RESPONSE - "DELIVERBUSOBJ"
- getConnName():String
- getGenericBO():BusObj
- getInitiator():String
- getLocale():java.util.Locale
- getOriginalRequestBO():BusObj
- setConnName(String)
- setInitiator(String)
- setLocale(java.util.Locale)

Participant
RelationshipServices/
Server/

- Participant(String, String, int, BusObj)
- Participant(String, String, int, String)
- Participant(String, String, int, long)
- Participant(String, String, int, int)
- Participant(String, String, int, double)
- Participant(String, String, int, float)
- Participant(String, String, int, boolean)
- Participant(String, String, BusObj)
- Participant(String, String, String)
- Participant(String, String, long)
- Participant(String, String, int)
- Participant(String, String, double)
- Participant(String, String, float)

- Participant(String, String, boolean)
- getBoolean():boolean
- getBusObj():BusObj
- getDouble():double
- getFloat():float
- getInstanceId():int
- getInt():int
- getLong():long
- getParticipantDefinition():String
- getRelationshipDefinition():String
- getString():String INVALID_INSTANCE_ID
- set(boolean)
- set(BusObj)
- set(double)
- set(float)
- set(int)
- set(long)
- set(String)
- setInstanceId(int)
- setParticipantDefinition(String)
- setRelationshipDefinition(String)
- setParticipantDefinition(String)
- setRelationshipDefinition(String)

Relationship
RelationshipServices/
Server/

- addMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- addParticipant(Participant):int
- addParticipant(String, String, boolean):int
- addParticipant(String, String, BusObj):int
- addParticipant(String, String, double):int
- addParticipant(String, String, float):int
- addParticipant(String, String, int):int
- addParticipant(String, String, int, boolean):int
- addParticipant(String, String, int, BusObj):int
- addParticipant(String, String, int, double):int
- addParticipant(String, String, int, float):int
- addParticipant(String, String, int, int):int
- addParticipant(String, String, int, long):int
- addParticipant(String, String, int, String):int

- addParticipant(String, String, long):int
- addParticipant(String, String, String):int
- create(Participant):int
- create(String, String, boolean):int
- create(String, String, BusObj):int
- create(String, String, double):int
- create(String, String, float):int
- create(String, String, int):int
- create(String, String, long):int
- create(String, String, String):int
- deactivateParticipant(Participant)
- deactivateParticipant(String, String, boolean)
- deactivateParticipant(String, String, BusObj)
- deactivateParticipant(String, String, double)
- deactivateParticipant(String, String, float)
- deactivateParticipant(String, String, int)
- deactivateParticipant(String, String, long)
- deactivateParticipant(String, String, String)
- deactivateParticipantByInstance(String, String, int)
- deactivateParticipantByInstance(String, String, int, boolean)
- deactivateParticipantByInstance(String, String, int, BusObj)
- deactivateParticipantByInstance(String, String, int, double)
- deactivateParticipantByInstance(String, String, int, float)
- deactivateParticipantByInstance(String, String, int, int)
- deactivateParticipantByInstance(String, String, int, long)
- deactivateParticipantByInstance(String, String, int, String)
- deleteMyChildren(String, String, BusObj, String, CxExecutionContext)
- deleteMyChildren(String, String, BusObj, String, Object, CxExecutionContext)
- deleteParticipant(Participant)
- deleteParticipant(String, String, boolean)
- deleteParticipant(String, String, BusObj)
- deleteParticipant(String, String, double)
- deleteParticipant(String, String, float)
- deleteParticipant(String, String, int)
- deleteParticipant(String, String, long)
- deleteParticipant(String, String, String)
- deleteParticipantByInstance(String, String, int)
- deleteParticipantByInstance(String, String, int, boolean)
- deleteParticipantByInstance(String, String, int, BusObj)
- deleteParticipantByInstance(String, String, int, double)

- deleteParticipantByInstance(String, String, int, float)
- deleteParticipantByInstance(String, String, int, int)
- deleteParticipantByInstance(String, String, int, long)
- deleteParticipantByInstance(String, String, int, String)
- getNewID(String):int
- maintainCompositeRelationship(String, String, BusObj, Object, CxExecutionContext)
- maintainSimpleIdentityRelationship(String, String, BusObj, BusObj, CxExecutionContext)
- retrieveInstances(String, boolean):int[]
- retrieveInstances(String, BusObj):int[]
- retrieveInstances(String, double):int[]
- retrieveInstances(String, float):int[]
- retrieveInstances(String, int):int[]
- retrieveInstances(String, long):int[]
- retrieveInstances(String, String):int[]
- retrieveInstances(String, String, boolean):int[]
- retrieveInstances(String, String, BusObj):int[]
- retrieveInstances(String, String, double):int[]
- retrieveInstances(String, String, float):int[]
- retrieveInstances(String, String, int):int[]
- retrieveInstances(String, String, long):int[]
- retrieveInstances(String, String, String):int[]
- retrieveInstances(String, String[], boolean):int[]
- retrieveInstances(String, String[], BusObj):int[]
- retrieveInstances(String, String[], double):int[]
- retrieveInstances(String, String[], float):int[]
- retrieveInstances(String, String[], int):int[]
- retrieveInstances(String, String[], long):int[]
- retrieveInstances(String, String[], String):int[]
- retrieveParticipants(String):Participant[]
- retrieveParticipants(String, String):Participant[]
- retrieveParticipants(String, String[]):Participant[]
- retrieveParticipants(String, int):Participant[]
- retrieveParticipants(String, String, int):Participant[]
- retrieveParticipants(String, String[], int):Participant[]
- updateMyChildren(String, String, BusObj, String, String, String, String, CxExecutionContext)
- updateParticipant(String, String, BusObj)
- updateParticipantByInstance(Participant)
- updateParticipantByInstance(String, String, int)

- updateParticipantByInstance(String, String, int, BusObj)

UserStoredProcedureParam
Dtp/
CxCommon/

- UserStoredProcedureParam(int, String, Object, String, String)
- getParamDataTypeJavaObj():String
- getParamDataTypeJDBC():int
- getParamIndex():int
- getParamIOType():String
- getParamName():String
- getParamValue():Object
- setParamDataTypeJavaObj(String)
- setParamDataTypeJDBC(int)
- setParamIndex(int)
- setParamIOType(String)
- setParamName(String)
- setParamValue(Object)
- PARAM_TYPE_IN - "IN"
- PARAM_TYPE_OUT - "OUT"
- PARAM_TYPE_INOUT - "INOUT"
- DATA_TYPE_STRING - "String"
- DATA_TYPE_INTEGER - "Integer"
- DATA_TYPE_DOUBLE - "Double"
- DATA_TYPE_FLOAT - "Float"
- DATA_TYPE_BOOLEAN - "Boolean"
- DATA_TYPE_TIME - "java.sql.Time"
- DATA_TYPE_DATE - "java.sql.Date"
- DATA_TYPE_TIMESTAMP - "java.sql.Timestamp"
- DATA_TYPE_BIG_DECIMAL - "java.math.BigDecimal"
- DATA_TYPE_LONG_INTEGER - "Long"
- DATA_TYPE_BINARY - "byte[]"
- DATA_TYPE_CLOB - "Clob"
- DATA_TYPE_BLOB - "Blob"
- DATA_TYPE_ARRAY - "Array"
- DATA_TYPE_STRUCT - "Struct"
- DATA_TYPE_REF - "Ref"

BaseCollaboration
Collaboration/

- BaseCollaboration(com.ibm.bpe.api.ProcessInstanceData)

- AnyException - "AnyException"
- AppBusObjDoesNotExist - "BusObjDoesNotExist"
- AppLogOnFailure - "AppLogOnFailure"
- AppMultipleHits - "AppMultipleHits"
- AppRequestNotYetSent - "AppRequestNotYetSent"
- AppRetrieveByContentFailed - "AppRetrieveByContent"
- AppTimeOut - "AppTimeOut"
- AppUnknown - "AppUnknown"
- AttributeException - "AttributeException"
- existsConfigProperty(String):boolean
- getConfigProperty(String):String
- getConfigPropertyArray(String):String[]
- getCurrentLoopIndex():int
- getDBConnection(String):CwDBConnection
- getDBConnection(String, boolean):CwDBConnection getLocale():java.util.Locale
- getMessage(int):String
- getMessage(int, Object[]):String
- getName():String
- implicitDBTransactionBracketing():boolean
- isCallerInRole(String):boolean
- isTraceEnabled(int):boolean
- JavaException - "JavaException"
- logError(int)
- logError(int, Object[])
- logError(int, String)
- logError(int, String, String)
- logError(int, String, String, String)
- logError(int, String, String, String, String)
- logError(int, String, String, String, String, String)
- logError(String)
- logInfo(int)
- logInfo(int, Object[])
- logInfo(int, String)
- logInfo(int, String, String)
- logInfo(int, String, String, String)
- logInfo(int, String, String, String, String)
- logInfo(int, String, String, String, String, String)
- logInfo(String)
- logWarning(int)
- logWarning(int, Object[])

- logWarning(int, String)
- logWarning(int, String, String)
- logWarning(int, String, String, String)
- logWarning(int, String, String, String, String)
- logWarning(int, String, String, String, String, String)
- logWarning(String)
- not(boolean):boolean ObjectException - "ObjectException"
- OperationException - "OperationException"
- raiseException(CollaborationException)
- raiseException(String, int)
- raiseException(String, int, Object[])
- raiseException(String, int, String)
- raiseException(String, int, String, String)
- raiseException(String, int, String, String, String)
- raiseException(String, int, String, String, String, String)
- raiseException(String, int, String, String, String, String, String)
- raiseException(String, String)
- ServiceCallException - "ConsumerException"
- ServiceCallTransportException - "ServiceCallTransportException"
- SystemException - "SystemException"
- trace(int, int)
- trace(int, int, Object[])
- trace(int, int, String)
- trace(int, int, String, String)
- trace(int, int, String, String, String)
- trace(int, int, String, String, String, String)
- trace(int, int, String, String, String, String, String)
- trace(int, String)
- trace(String)
- TransactionException - "TransactionException"

CxExecutionContext
CxCommon/

- CxExecutionContext()
- getContext(String):Object
- MAPCONTEXT - "MAPCONTEXT"
- setContext(String, Object)

CollaborationException
Collaboration/

- getMessage():String

- getMsgNumber():int
- getSubType():String
- getText():String
- getType():String
- toString():String

Filter
crossworlds/
com/

- Filter(BaseCollaboration)
- filterExcludes(String, String):boolean
- filterIncludes(String, String):boolean
- recurseFilter(BusObj, String, boolean, String, String):boolean
- recursePreReqs(String, Vector):int

Globals
crossworlds/
com/

- Globals(BaseCollaboration)
- callMap(String, BusObj):BusObj

SmartCollabService
crossworlds/
com/

- SmartCollabService()
- SmartCollabService(BaseCollaboration)
- doAgg(BusObj, String, String, String):BusObj
- doMergeHash(Vector, String, String):Vector
- doRecursiveAgg(BusObj, String, String, String):BusObj
- doRecursiveSplit(BusObj, String):Vector
- doRecursiveSplit(BusObj, String, boolean):Vector
- getKeyValues(BusObj, String):String
- merge(Vector, String):BusObj
- merge(Vector, String, BusObj):BusObj
- split(BusObj, String):Vector

StateManagement
crossworlds/
com/

- StateManagement()
- beginTransaction()
- commit()
- deleteBO(String, String, String)

- deleteState(String, String, String, int)
- persistBO(String, String, String, String, BusObj)
- recoverBO(String, String, String):BusObj
- releaseDBConnection()
- resetData()
- retrieveState(String, String, String, int):int
- saveState(String, String, String, String, int, int, double)
- setDBConnection(CwDBConnection)
- updateBO(String, String, String, String, BusObj)
- updateState(String, String, String, String, int, int)

EventKeyAttrDef
EventManagement/
CxCommon/

- EventKeyAttrDef()
- EventKeyAttrDef(String, String)
- public String keyName
- public String keyValue

EventQueryDef
EventManagement/
CxCommon/

- EventQueryDef()
- EventQueryDef(String, String, String, String, int)
- public String nameConnector
- public String nameCollaboration
- public String nameBusObj
- public String verb
- public int ownerType

FailedEventInfo
EventManagement/
CxCommon/

- FailedEventInfo()
- FailedEventInfo(String x6, int, EventKeyAttrDef[], int, int, String, String, int)
- public String nameOwner
- public String nameConnector
- public String nameBusObj
- public String nameVerb
- public String strTime
- public String strMessage
- public int wipIndex

- public EventKeyAttrDef[] strbusObjKeys
- public int nKeys
- public int eventStatus
- public String expirationTime
- public String scenarioName
- public int scenarioState

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。

ビジネス・プロセスでの補正処理 (Compensation handling in business processes)
補正処理は、プロセス・モデルで補正が定義された実行中のプロセス・インスタンスにおける、障害処理の手段の 1 つです。補正は、障害が発生した時点までにコミットされた操作の影響をリバースし、整合した状態に戻します。

関連資料

175 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。

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

217 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

175 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。

164 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「一般プロパティー」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。

WebSphere Process Server バージョン 7.0 での継承 API の新規動作

WebSphere Process Server のバージョン 7.0 では、継承 API が、以前は BusinessObjectInterface インターフェースによって保管されていた属性の状態およびデータを WebSphere Process Server サービス・データ・オブジェクトを使用して保管します。そのため、BusinessObjectInterface および CxObjectContainerInterface インターフェースの一部のメソッド呼び出しの動作が変更されています。

WebSphere Process Server バージョン 7.0 での継承 API (HAPI) の主要な変更は、WebSphere InterChange Server BusinessObjectInterface インターフェースが HAPI のルート・ストレージ・オブジェクトではなくなったことです。代わりに WebSphere Process Server サービス・データ・オブジェクト (SDO) を使用して、属性の状態およびデータを保管するようになっています。

Java 等価演算子と緩く型付けされた属性の原則を使用する場合は、以下のセクションで説明するように、BusinessObjectInterface インターフェースと CxObjectContainerInterface インターフェースでのメソッド呼び出しの動作が異なります。

- 『設定操作に続いて取得操作を実行する場合の Java 等価演算子の使用』
- 223 ページの『BusinessObjectInterface オブジェクトを複数のターゲット属性に設定する場合の Java 等価演算子の使用』
- 225 ページの『BusinessObjectInterface オブジェクトを CxObjectContainerInterface インターフェースに設定して取得する場合の Java 等価演算子の使用』
- 226 ページの『BusObj クラスの validData メソッドでの緩く型付けされた属性データ型の使用』

設定操作に続いて取得操作を実行する場合の Java 等価演算子の使用

1 つのターゲット属性に対して BusinessObjectInterface オブジェクトの設定操作、続いて取得操作を実行すると、異なる BusinessObjectInterface オブジェクトが返されます。以下の表に、以前の動作と現在の動作についての説明、および設定操作に続いて取得操作を実行するときに、以前使用していた Java 等価演算子を何に変更するべきかを説明する例を記載します。

表 10. 動作の変更: 設定操作および取得操作での Java 等価演算子の使用

動作タイプ	説明
WebSphere Process Server バージョン 7.0 より前の動作	取得される BusinessObjectInterface コンテナーは、設定されたコンテナーと同じで、Java 等価演算子「==」を使用して、2 つが同じであるかどうかを判断することができました。 例: `boolean b = (JavaObjectA == JavaObjectB)`
WebSphere Process Server バージョン 7.0 以降の動作	元の BusinessObjectInterface コンテナーは破棄され、取得操作を実行して BusinessObjectInterface オブジェクトを取得すると、新規コンテナーが作成されます。返されるコンテナーは同じオブジェクトではありませんが、このコンテナーがラップするルート・オブジェクトは、同じオブジェクトです。isEquivalent という新規メソッドが BusinessObjectInterface クラスに追加されました (BusinessObjectInterface.isEquivalent(BOI))。 2 つの BusinessObjectInterface オブジェクトが同じであるかどうかを判断するには、isEquivalent メソッドを使用して比較を行います。

表 10. 動作の変更: 設定操作および取得操作での Java 等価演算子の使用 (続き)

動作タイプ	説明

新規動作の例	以下の例で、isEquivalent の使用方法を示します。 BusinessObjectInterface オブジェクトの型が MasterBusinessObject で、属性 Attr_Nine が設定されているとします。これは、HelloWorld 型の BusinessObjectInterface オブジェクトです。
	`BusinessObjectInterface mboBOI, hw1BOI, hw2BOI;`
	`hw1BOI.setAttrValue("Message", "hw1BOI_message");`
	`hw1BOI.setVerb("Create");`
	`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`
	`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`
	
	以前は以下を使用していました。
	
	`boolean result = (hw1BOI == hw2BOI); assertTrue(result);`
	
	代わりに以下を使用します。
	
	`boolean result = hw1BOI.isEquivalent(hw2BOI); assertTrue(result);`

BusinessObjectInterface オブジェクトを複数のターゲット属性に設定する場合の Java 等価演算子の使用

BusinessObjectInterface オブジェクトを複数のターゲット属性に設定すると、複製されたオブジェクトが設定されます。これは、BusObjArray クラスの要素と、複数のターゲット属性の両方に適用されます。以下の表に、以前の動作と現在の動作についての説明、および BusinessObjectInterface オブジェクトを複数のターゲット属性に設定するときに、以前使用していた Java 等価演算子を何に変更するべきかを説明する例を記載します。

表 11. 動作の変更: 複数のターゲット属性での Java 等価演算子の使用

動作タイプ	説明
WebSphere Process Server バージョン 7.0 より前の動作	BusinessObjectInterface オブジェクトを複数のロケーションに設定し、そのすべてのロケーションに元の BusinessObjectInterface オブジェクトへの参照を含めることができました。ある BusinessObjectInterface オブジェクトで属性を変更すると、その変更は、そのオブジェクトの他のすべての参照にも反映されました。

表 11. 動作の変更: 複数のターゲット属性での Java 等価演算子の使用 (続き)

動作タイプ	説明
WebSphere Process Server バージョン 7.0 以降の動作	サービス・データ・オブジェクト (SDO) の規則により、同じ SDO を複数のターゲット・プロパティーに設定することはできません。SDO を複数のターゲット・プロパティーに設定すると、SDO が 1 つの属性から次の属性へと移り、前の属性ロケーションには「NULL」の値が残ることになります。BusinessObjectInterface オブジェクトを 2 番目、3 番目のロケーションへと設定しているときには、「NULL」の値を残す代わりに、オブジェクトが複数のロケーションに複製されるようになっています。 例えば、型が MasterBusinessObject で、HelloWorld 型の Attr_Nine 属性と Attr_Eleven 属性を設定した BusinessObjectInterface オブジェクトがあるとします。同じ HelloWorld オブジェクトを両方の属性に設定すると、Attr_Nine は元のオブジェクトに割り当てられ、Attr_Eleven にはクローンが割り当てられます。このクローンは、オブジェクトが複製された時点でのオブジェクトのスナップショットです。 2 つの BusinessObjectInterface オブジェクトが同じであるかを判断するには、Java 等価演算子を使用しないでください。代わりに、isEquivalent メソッドを指標して比較を行います。

新規動作の例	以下の例に、isEquivalent およびクローンの使用方法を示します。型が MasterBusinessObject で、HelloWorld 型の Attr_Nine 属性と Attr_Eleven 属性を設定した BusinessObjectInterface オブジェクトがあるとします。
	
	`BusinessObjectInterface mboBOI;`
	`BusinessObjectInterface hw1BOI, hw2BOI, hw3BOI;`
	`hw1BOI.setAttrValue("Message", "hw1BOI_message");`
	`hw1BOI.setVerb("Create");`
	`mboBOI.setAttrValue("Attr_Nine", hw1BOI);`
	`mboBOI.setAttrValue("Attr_Eleven", hw1BOI);`
	`hw2BOI = mboBOI.getAttrValue("Attr_Nine");`
	`hw3BOI = mboBOI.getAttrValue("Attr_ Eleven ");`
	
	以前は以下を使用していました。
	
	`boolean result = hw2BOI == hw3BOI); assertTrue(result);`
	
	代わりに isEquivalent を使用します。
	
	`boolean result = hw2BOI.isEquivalent(hw3BOI); assertTrue(result);`
	
	複製されたオブジェクトは参照を共用しません。そのため、元の BusinessObjectInterface オブジェクトに対する変更は、複製された BusinessObjectInterface オブジェクトには反映されません。
	
	`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`
	`boolean result = hw1BOI.isEquivalent(hw2BOI);`
	`assertTrue(result);`
	`boolean result = hw1BOI.isEquivalent(hw3BOI);`
	`assertFalse(result);`
	`boolean result = hw2BOI.isEquivalent(hw3BOI);`
	`assertFalse(result);`

BusinessObjectInterface オブジェクトを CxObjectContainerInterface インターフェースに設定して取得する場合の Java 等価演算子の使用

以下の表に、以前の動作と現在の動作についての説明、および BusinessObjectInterface オブジェクトを CxObjectContainerInterface インターフェースに設定して取得するときに、以前使用していた Java 等価演算子を何に変更するべきかを説明する例を記載します。

表 12. 動作の変更: CxObjectContainerInterface インターフェースでの Java 等価演算子の使用

動作タイプ	説明
WebSphere Process Server バージョン 7.0 より前の動作	BusinessObjectInterface オブジェクトを設定して CxObjectContainerInterface インターフェースから取得するときには、Java 等価演算子「==」を使用できました。これは、取得される BusinessObjectInterface コンテナーは、設定された BusinessObjectInterface コンテナーと同じであるためです。
WebSphere Process Server バージョン 7.0 以降の動作	BusinessObjectInterface.isEquivalent(BOI) メソッドを使用する必要があります。

表 12. 動作の変更: CxObjectContainerInterface インターフェースでの Java 等価演算子の使用 (続き)

動作タイプ	説明

新規動作の例	以下の JUnit テスト・コードで、以前の動作と新規動作を説明します。
	`CxObjectContainerInterface testCxObjectContainerInt;`
	`BusinessObjectInterface mB01, mB02, mB03;`
	`testCxObjectContainerInt.insertBusinessObject(mB01);`
	`testCxObjectContainerInt.setBusinessObject(1, mB01);`
	`BusinessObjectInterface mB02 = testCxObjectContainerInt.`
	`getBusinessObject(0));`
	`BusinessObjectInterface mB03 = testCxObjectContainerInt.`
	`getBusinessObject(1));`
	`assertTrue(mB01 == mB02);`
	`assertTrue(mB01 == mB03);`
	`assertTrue(mB02 == mB03);`
	この Java 等価演算子は機能しなくなりました。CxObjectContainerInterface.getBusinessObject(int index) から返される BusinessObjectInterface オブジェクトは、CxObjectContainerInterface に設定された Java オブジェクトとは同じではないためです。
	以下のコードでは、等価演算子を BusinessObjectInterface.isEquivalent(BOI) メソッドに置き換えています。
	`boolean result1 = mB01.isEquivalent(mB02)`
	`assertTrue(result1);`
	`boolean result2 = mB01.isEquivalent(mB03)`
	`assertFalse(result2);`
	`boolean result3 = mB02.isEquivalent(mB03)`
	`assertFalse(result3);`
	複製されたオブジェクトは参照を共用しません。そのため、元の BusinessObjectInterface オブジェクトに対する変更は、複製された BusinessObjectInterface オブジェクトには反映されません。
	`hw1BOI.setAttrValue("Message", "hw1BOI_message changed");`
	`boolean result = mB01.isEquivalent(mB02;`
	`assertTrue(result);`
	`boolean result = mB01.isEquivalent(mB02;`
	`assertFalse(result);`
	`boolean result = mB02.isEquivalent(hw3BOI);`
	`assertFalse(result);`

BusObj クラスの validData メソッドでの緩く型付けされた属性データ型の使用

以下の表に、以前の動作と現在の動作についての説明、および BusObj クラスの validData メソッドに使用する場合、以前使用していた WebSphere InterChange Server または WebSphere Business Integration Server Express の緩く型付けされた属性データ型を何に変更するべきかを説明する例を記載します。

表 13. 動作の変更: BusObj クラスの validData メソッドでの緩く型付けされた属性データ型の使用

動作タイプ	説明
WebSphere Process Server バージョン 7.0 より前の動作	BusObj クラスの validData メソッドでは、属性データ型は WebSphere InterChange Server または WebSphere Business Integration Server Express に緩く型付けされていました。そのため、一致しないデータと型の組み合わせを使用することができました。例えば、ビジネス・オブジェクトの属性がブール型であるが、ストリング・パラメーターを持つ設定メソッドを使用している場合、ブール型である属性に、「ブール値ではない」ストリングを設定することが可能でした。getString メソッドを使用する限り、「ブール値ではない」ストリングを取得することができました。
WebSphere Process Server バージョン 7.0 以降の動作	現在、属性データ型は厳格に型付けされるようになっています。以前有効だったデータ型が、現在有効でない場合には、メッセージ番号 1802 の CollaborationException 例外が throw されます。WebSphere Process Server は厳格に型付けされているため、ストリング値をブール型の属性に組み込むことはできません。 Java 変換を使用してストリングを *true* および *false* のブール値にしたとしても、「ブール値ではない」元の値を返す手段がありません。指定可能な戻り値は、*true* または *false* のみです。 したがって、属性は double と float または int と long に厳格に型付けされるようになっています。これらの型は、Java が自動キャストを行う場合には、相互に置き換えて使用できます。ただし、あらゆる型のキャストの場合と同じく、フィールドが短い型に変換されるときには、精度がある程度失われる可能性があります。型が、設定されている属性には有効でないが、WebSphere InterChange Server または WebSphere Business Integration Server Express では有効だった場合には、メッセージ番号 1802 の CollaborationException 例外が throw されます。これは新規のメッセージ番号です。このメッセージの定義は、InterchangeSystem.txt メッセージ・ファイルにあります。

表 13. 動作の変更: BusObj クラスの validData メソッドでの緩く型付けされた属性データ型の使用 (続き)

動作タイプ	説明

新規動作の例	型が、設定されている属性には有効でないが、WebSphere InterChange Server では有効だった場合には、メッセージ番号 1802 の CollaborationException が throw されます。これは新規のメッセージ番号です。このメッセージの定義は、InterchangeSystem.txt メッセージ・ファイルにあります。
	`try`
	`{`
	`BusObj mBO = new BusObj("MasterBusinessObject");`
	`mBO.set("Attr_Two", "xxx");`
	`fail("Expected CollaborationException not thrown");`
	`}`
	`catch (CollaborationException e)`
	`{`
	`int a = e.getMsgNumber();`
	`String b = e.getSubType();`
	`String c = e.getMessage();`
	`String d = e.toString();`
	`assertEquals("exception_msgNumber", 1802, a);`
	`assertEquals("exception_type", "AttributeException", b);`
	`assertEquals("exception_message",`
	`"Error 1802 The attribute ¥"Attr_Two¥" in SDO MasterBusinessObject is of type boolean and is not allowed to be set with a value ¥"xxx¥" of type String. Error1802", c);`
	
	`assertEquals("exception_toString", "AttributeException: Error 1802 The attribute ¥"Attr_Two¥" in SDO MasterBusinessObject is of type boolean and is not allowed to be set with a value ¥"xxx¥" of type String. Error1802", d);}`

Microflow が補正されない

Microflow がサービスを呼び出したときにプロセスが失敗しましたが、元に戻すサービスが呼び出されません。

解決方法

Microflow の補正を起動するには、さまざまな条件を満たす必要があります。次の点を確認します。

1. Business Process Choreographer Explorer にログオンし、「**失敗した補正**」をクリックして、補正サービスが失敗しており、修復する必要があるかどうかを確認します。
2. Microflow の補正は、Microflow のトランザクションがロールバックした場合にのみ起動されます。この場合に該当するかどうか確認してください。
3. Microflow の compensationSphere 属性を「必須」に設定する必要があります。
4. 補正サービスが実行されるのは、対応する転送サービスが Microflow のトランザクションにかかわっていない場合のみです。転送サービスがナビゲーション・トランザクションにかかわっていないことを確認してください。例えば、プロセス・コンポーネントの参照時には、Service Component Architecture (SCA) の修飾子 suspendTransaction を True に設定します。

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

Printed in Japan

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