



**お願い**

本書および本書で紹介する製品をご使用になる前に、335 ページの『特記事項』に記載されている情報をお読みください。

本書は、WebSphere Process Server for z/OS バージョン 6、リリース 0、モディフィケーション 2 (製品番号 5655-N53) および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

IBM 発行のマニュアルに関する情報のページ

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

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

(URL は、変更になる場合があります)

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

**原 典：** WebSphere Process Server for z/OS  
Version 6.0.2  
Administering WebSphere Process Server

**発 行：** 日本アイ・ビー・エム株式会社

**担 当：** ナショナル・ランゲージ・サポート

第1刷 2007.3

この文書では、平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、および平成角ゴシック体™W7を使用しています。この(書体\*)は、(財)日本規格協会と使用契約を締結し使用しているものです。フォントとして無断複製することは禁止されています。

注\* 平成明朝体™W3、平成明朝体™W7、平成明朝体™W9、平成角ゴシック体™W3、平成角ゴシック体™W5、平成角ゴシック体™W7

© Copyright International Business Machines Corporation 2006, 2007. All rights reserved.

© Copyright IBM Japan 2007

# 目次

<b>WebSphere Process Server の管理</b> . . . . .	<b>1</b>
WebSphere Process Server の管理の概要 . . . . .	1
WebSphere Process Server の管理コンソール . . . . .	4
WebSphere Process Server 管理環境ロードマップのセットアップ . . . . .	18
管理アーキテクチャーのセットアップ . . . . .	19
セル全体に対する設定の構成 . . . . .	20
WebSphere Process Server サーバー・ロードマップの管理 . . . . .	21
WebSphere Process Server リソース・ロードマップの構成 . . . . .	24
WebSphere Process Server 設定ロードマップの構成 . . . . .	25
イベント・サービスの管理 . . . . .	25
リレーシオンシップ・サービスの管理 . . . . .	27
クラスター環境の作成 . . . . .	30
Service Component Architecture アプリケーションをサポートするサーバーまたはクラスターの準備 . . . . .	34
アプリケーションおよびアプリケーション・サービスの管理 . . . . .	38
エンタープライズ・アプリケーションの管理 . . . . .	38
Service Component Architecture モジュールのデプロイ後の WebSphere MQ 宛先の変更 . . . . .	40
Application Scheduler . . . . .	43
WebSphere Process Server リソースの管理 . . . . .	49
アダプターの管理 . . . . .	49
バインディング . . . . .	88
SCA EIS バインディング . . . . .	89
JMS バインディング . . . . .	96
WebSphere MQ JMS バインディング . . . . .	113
データ・バインディング . . . . .	124
エクスポート・バインディングとイベント順序付け . . . . .	127
JMS ヘッダー . . . . .	130
JMS バインディングおよび MQ JMS バインディングでの障害の処理 . . . . .	131
JMS および MQ JMS バインディングのトラブルシューティング . . . . .	131
ビジネス・ステート・マシンのトラブルシューティングの概要 . . . . .	133
ビジネス・ルールおよびセレクターの概要 . . . . .	135
ターゲットの概要 . . . . .	174
拡張メッセージング・リソースの管理: 概要 . . . . .	179
Enterprise Service Bus の管理 . . . . .	187
バス環境の管理 . . . . .	189
サービス・アプリケーションの管理 . . . . .	242
リレーシオンシップの管理 . . . . .	290
リレーシオンシップ・タイプの表示 . . . . .	291
リレーシオンシップ詳細の表示 . . . . .	292
ロール詳細の表示 . . . . .	292
リレーシオンシップ・インスタンスの照会 . . . . .	293
リレーシオンシップ・インスタンスの結果の表示 . . . . .	294
リレーシオンシップ・インスタンス詳細の表示 . . . . .	295
リレーシオンシップ・インスタンスの詳細の編集 . . . . .	295
リレーシオンシップ・インスタンスの新規作成 . . . . .	296
リレーシオンシップ・インスタンスの削除 . . . . .	297
リレーシオンシップ・インスタンス・データのロールバック . . . . .	297
ロール・インスタンス詳細の表示 . . . . .	298
ロール・インスタンス・プロパティの編集 . . . . .	298
ロール・インスタンスの新規作成 . . . . .	299
ロール・インスタンスの削除 . . . . .	300

チュートリアル: Relationship Manager の管理 . . . . .	300
イベントの順次処理 . . . . .	302
例: イベント順序付け . . . . .	304
イベント順序付けを実装する場合の考慮事項 . . . . .	306
ロックの管理 . . . . .	310
イベント順序付けのトラブルシューティング . . . . .	312
WebSphere Process Server の失敗したイベントの管理 . . . . .	313
リカバリーのセキュリティー考慮事項 . . . . .	314
失敗したイベントの検索 . . . . .	315
失敗したイベントのデータの操作 . . . . .	322
失敗したイベントの再サブミット . . . . .	326
失敗したイベントに関連する Common Base Event の検索 . . . . .	327
失敗したイベントに関連するビジネス・プロセス・インスタンスの検索 . . . . .	327
失敗したイベントの削除 . . . . .	328
Failed Event Manager のトラブルシューティング . . . . .	328
WebSphere Process Server 管理のトラブルシューティング . . . . .	330
Failed Event Manager のトラブルシューティング . . . . .	330
Service Component Architecture と WebSphere MQ の通信のトラブルシューティング . . . . .	332
イベント順序付けのトラブルシューティング . . . . .	333
<b>特記事項 . . . . .</b>	<b>335</b>
プログラミング・インターフェース情報 . . . . .	337
商標 . . . . .	337

---

## WebSphere Process Server の管理

このセクションのトピックでは、WebSphere® Process Server のランタイム環境 (この環境にデプロイされるアプリケーションやリソースなども含む) を管理する方法について説明します。

WebSphere Process Server の資料 (PDF 形式)

WebSphere Process Server の管理について学ぶためのその他のリソースは、**WebSphere Business Process Management** モジュールの IBM Education Assistant Web サイトで利用できます。

ビジネス・プロセスおよびヒューマン・タスクの管理については、WebSphere Process Server for z/OS バージョン 6.0.2 インフォメーション・センターの以下のいずれかの場所を参照してください。

- 『WebSphere Process Server の管理』 > 『Business Process Choreographer の管理』
- 『WebSphere Process Server の管理』 > 『アプリケーションおよびアプリケーション・サービスの管理』 > 『ビジネス・プロセスおよびヒューマン・タスクの管理』
- *Business Process Choreographer* の PDF ファイル

Common Event Infrastructure (CEI) の管理については、WebSphere Process Server for z/OS バージョン 6.0.2 インフォメーション・センターの以下のいずれかの場所を参照してください。

- 『WebSphere Process Server の管理』 > 『Common Event Infrastructure の管理』
- *Common Event Infrastructure* の PDF ファイル

---

## WebSphere Process Server の管理の概要

WebSphere Process Server の管理には、アプリケーションおよびリソース自体の処理だけでなく、そのアプリケーションおよびリソースのデプロイ先である環境の準備、モニター、および変更も含まれます。

### 管理インターフェース

WebSphere Process Server では、ランタイム環境の管理用に、以下のような複数のインターフェースが提供されています。

#### 管理コンソール

管理コンソールは、ブラウザー・ベースのインターフェースで、これによりユーザーは、さまざまな種類のアプリケーション、サービス、およびリソースをモニター、更新、停止、および始動できます。また管理コンソールを使用して、リレーションシップを操作したり、失敗した WebSphere Process Server イベントを見つけて解決したりすることもできます。

管理コンソールでは、管理対象を広げて、WebSphere Application Server およびその他のユーザー定義製品を管理できるようになっています。

詳しくは、4 ページの『WebSphere Process Server の管理コンソール』を参照してください。

### ビジネス・ルール・マネージャー

ビジネス・ルール・マネージャーは、ビジネス・アナリストがビジネス・ルール値を表示および変更する際に使用する Web ベースのツールです。このツールは、WebSphere Process Server のオプションであり、このサーバーの初期インストール後にインストールできます。ビジネス・ルール・マネージャーを使用すると、リポジトリからビジネス・ルールのコピーを開いたり、ビジネス・ルールを表示して編集したり、ビジネス・ルールをリポジトリに公開したりすることができます。

### Business Process Choreographer Explorer

Business Process Choreographer Explorer は、スタンドアロン Web アプリケーションであり、ビジネス・プロセスおよびヒューマン・タスクを管理するための管理機能の基本セットを提供します。プロセス・テンプレート、プロセス・インスタンス、タスク・インスタンス、およびこれらの関連オブジェクトに関する情報を表示できます。これらのオブジェクトを処理することもできます。例えば、新規プロセス・インスタンスを開始したり、失敗したアクティビティを修復および再始動したり、作業項目を管理したり、完了したプロセス・インスタンスやタスク・インスタンスを削除したりすることができます。

### Business Process Choreographer Observer

Business Process Choreographer Observer は、ビジネス・プロセスおよびヒューマン・タスクの実行中に発生するイベントに関するレポートを作成する Web アプリケーションです。このレポートは、プロセスやアクティビティの有効性や信頼性を評価する目的で使用できます。

### スクリプト (wsadmin)

WebSphere 管理 (wsadmin) スクリプト・プログラムは、非グラフィカル・コマンド・インタープリター環境であり、スクリプト言語で管理オプションを実行したり、スクリプト言語プログラムの実行を依頼したりすることができます。これは、管理コンソールと同じタスクをサポートします。wsadmin は、実稼働環境および不在操作向けのツールです。

スクリプト・ツールに関して詳しくは、このインフォメーション・センターの「[参照](#)」>「[スクリプト・インターフェース \(Scripting interface\)](#)」を参照してください。

### コマンド行ツール

コマンド行ツールは、特定のタスクを実行するために、オペレーティング・システムのコマンド行プロンプトから実行する簡単なプログラムです。このツールを使用して、アプリケーション・サーバーの始動と停止、サーバー状況の確認、ノードの追加または除去、およびその他のタスクを実行できます。

WebSphere Process Server コマンド行ツールには、serviceDeploy コマンドが含まれており、このコマンドは、WebSphere Integration Developer 環境から

エクスポートされた .jar、.ear、.war および .rar ファイルを処理して、それらをプロダクション・サーバーにインストールするための準備をします。

コマンド行ツールに関して詳しくは、このインフォメーション・センターの「参照」>「API ドキュメンテーション」を参照してください。

## 管理プログラム

Java Management Extensions (JMX) 仕様に基づく一連の Java™ クラスおよびメソッドは、Service Component Architecture (SCA) およびビジネス・オブジェクトを管理するためのサポートを提供します。各プログラミング・インターフェースには、その目的の説明、インターフェースまたはクラスを使用する方法を示す例、および個々のメソッド説明への参照が含まれています。

プログラミング・インターフェースに関して詳しくは、このインフォメーション・センターの「参照」>「プログラミング・インターフェース」を参照してください。

## 構成情報

WebSphere Process Server 用のほとんどの構成データは、構成リポジトリ・ツリー (マスター・リポジトリ) のディレクトリに保持されている XML ファイルに格納されています。構成ファイルが配置されているディレクトリによって、その有効範囲 (データが適用される範囲) が決まります。

- 個々のサーバーのディレクトリ内のファイルは、そのサーバーにのみ適用されます。
- ノード・レベルのディレクトリ内のファイルは、そのノード上のすべてのサーバーに適用されます。
- セル・ディレクトリ内のファイルは、セル全体内のすべてのノード上のすべてのサーバーに適用されます。

表 1. WebSphere Process Server 構成ファイル

構成ファイル	説明
server-wbi.xml	プロセス・サーバーおよびそのコンポーネント (Adaptive Entity Service、拡張メッセージング・サービス、ビジネス・ルールとセレクター監査サービス、および WebSphere Business Integration Adapter Service 構成を含む) を識別します。
resources-wbi.xml	WebSphere Process Server 用のオペレーティング環境リソースを定義し、セル・スコープ、ノード・スコープ、およびサーバー・スコープに存在します。これには、拡張メッセージング・プロバイダーおよび WebSphere Business Integration Adapter が含まれています。
cell-wbi.xml	セルを識別します。このファイルは、リレーシオンシップ・サービス構成を格納するために使用され、セル・スコープにのみ存在します。

表 1. WebSphere Process Server 構成ファイル (続き)

構成ファイル	説明
server-bpc.xml	Business Process Choreographer コンテナおよびそのコンポーネント (Business Flow Manager、Human Task Manager、Staff Service、および Service Reference Service を含む) を識別します。
resources-bpc.xml	Staff Plugin Provider 用の構成情報を含む、Business Process Choreographer コンテナ用のオペレーティング環境リソースを定義します。このファイルは、セル・スコープ、ノード・スコープ、およびサーバー・スコープに存在します。
deployment-bpc.xml	ビジネス・プロセス・コンテナ用のアプリケーション・デプロイメント設定を構成します。
server-core.xml	Artifact Loader Service、Events Service、および Business Context Data Service を含む、コア WebSphere Process Server 構成用の構成情報を識別します。

WebSphere Process Server 構成ファイルは、管理コンソール、wsadmin、およびスクリプトを使用して編集できます。手動で編集する必要はありません。

サーバー構成ファイルの詳細については、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

## WebSphere Process Server の管理コンソール

管理コンソールは、WebSphere Process Server アプリケーション、サービス、およびその他リソースの管理に使用されるブラウザ・ベースのインターフェースです。管理コンソールは、セル・スコープ、ノード・スコープ、またはサーバー・スコープで管理するために使われます。管理コンソールは、スタンドアロン・プロセス・サーバーからも、ネットワーク環境内のセルですべてのサーバーを管理するデプロイメント・マネージャーからも使用できます。

**注:** WebSphere Process Server の管理コンソールは、さらに規模の大きい WebSphere Application Server 管理コンソールの一部です。そのため、多数の管理タスク (例えばセキュリティーの設定、ログの表示、およびアプリケーションのインストール) は、WebSphere Process Server および WebSphere Application Server の両方で同一の内容になっています。これらのタスクについては、WebSphere Application Server for z/OS インフォメーション・センターに記載されています。

### コンソールに関連付けられている WebSphere Process Server タスクについて

コンソールで実行される一般的な WebSphere Process Server タスクは、以下のとおりです。



- 管理アーキテクチャーおよび環境のセットアップ
- プロセス・サーバーおよびその設定の構成
- 新規アプリケーションのサーバーへのデプロイ
- 既存アプリケーションおよび構成の管理
- アプリケーション用リソース・プロバイダーの管理
- リレーションシップ、ビジネス・プロセス、タスク、アダプター、ビジネス・ルール、およびセレクターなどのサーバー・リソースの管理
- Business Process Choreographer の管理
- プロセス・サーバーで失敗したイベントの管理
- 製品セキュリティの構成
- トラブルシューティング用データの収集

## 管理コンソール・インターフェースについて

管理コンソールは、3 つの別個の領域に分かれています。

### タスクバー

タスクバーは、コンソールの最上位に配置されています。タスクバーは、コンソールからのログアウト、製品情報へのアクセス、およびサポートへのアクセスのオプションを提供します。

### ナビゲーション・ツリー

ナビゲーション・ツリーは、コンソールの左側にあります。ナビゲーション・ツリーは、サーバー、アプリケーション、およびその他リソースの作成および管理に使用するコンソール・ページへのリンクを提供します。

ナビゲーション・ツリーの項目の横にある正符号 (+) をクリックして展開するか、負符号 (-) をクリックしてその項目を縮小表示します。その項目自体をクリックして、展開状態および縮小状態を切り替えることもできます。

### ワークスペース

ワークスペースは、コンソールの右側に配置されています。ここには、サーバー、アプリケーション、およびその他のリソースの作成および管理に使用するページが表示されます。ナビゲーション・ツリーのリンクをクリックしたり、またはワークスペース・ページ自体の内部のリンクをクリックすることによって、これらのページにアクセスします。

ワークスペースに表示されるページのタイプについては、15 ページの『管理コンソール・ページ』を参照してください。

ワークスペースの一番右側は、ヘルプ・ポータルです。ヘルプ・ポータルでは、現在のページの各フィールドに関する概要情報が表示され、ヘルプ・ブラウザにある詳細な情報へのリンクも提供されています。

## 管理コンソールの WebSphere Process Server 固有領域の場所

WebSphere Process Server リソースは、管理コンソールのいくつかの領域にグループ化されています。以下のように、ナビゲーション・ツリーを使用して、これらのリソースを見つけます。

- 「**ガイド付きアクティビティ (Guided Activities)**」 > 「**Network Deployment 環境の構成**」 - Network Deployment 環境内のノード、データベース、クラスター、およびサーバーを構成および管理するためのガイド付きアクティビティを提供します。このオプションは、WebSphere Application Server Network Deployment をインストールしている場合にのみ使用可能です。
- 「**サーバー**」 > 「**アプリケーション・サーバー**」 > *server\_name* - ここから以下を利用できます。
  - ビジネス・プロセスおよびヒューマン・タスク用のコンテナ設定
  - Application Scheduler
  - ビジネス・ルール
  - イベント・サービス
  - 拡張メッセージング・サービス
  - サービス・コンポーネント・アーキテクチャーの構成
  - セレクター
  - スタッフ・サービス
  - WebSphere Business Integration Adapter サービス
  - Web サービス参照サービス
- 「**リソース**」 - 以下へのアクセスを提供します。
  - WebSphere Business Integration Adapter
  - Common Event Infrastructure プロバイダー
  - スタッフ・プラグイン・プロバイダー
  - 拡張メッセージング・プロバイダー
- 「**統合アプリケーション**」 - ここから以下を利用できます。
  - 失敗したイベント・マネージャー
  - リレーションシップ・マネージャー
  - Common Base Event ブラウザー

「ようこそ」ページでは、管理コンソール・ページを絞り込むために役立つタスク・フィルタリング・セレクターが提供されます。各フィルターにより、特定のタスクのセットに関連する管理コンソール機能のサブセット (アプリケーションの統合、Enterprise Service Bus の管理など) が提供されます。

## 管理コンソールからオンライン・ヘルプへのアクセス

管理コンソールのそれぞれのページおよびフィールドには、オンライン・ヘルプが提供されています。ヘルプは、以下のいずれかの方法で利用します。

- コンソール・タスクバーから「ヘルプ」をクリックすると、新しく Web ブラウザーが開いてオンライン・ヘルプが表示されます。

ヘルプ・ブラウザーから以下の操作を行うことができます。

- 「索引 (Index)」タブに表示するトピックをブラウズします。トピックのリンクをクリックすると、ブラウザーの右パネルにそのトピックが開きます。
- 「検索」タブに 1 つ以上のキー・ワードを指定して、トピックを検索します。一致したすべてのトピックが、ナビゲーション・ツリーに表示されます。トピックを表示するには、トピック・リンクをクリックします。
- フィールド上にカーソルを移動すると、そのフィールドの吹き出しヘルプが表示されます。
- フィールド上にカーソルを移動して、疑問符 (「?」) アイコンが表示されるのを待ちます。アイコンが表示されたら、フィールド名をクリックすると、ヘルプ・ポータル (ワークスペースの一番右側のパネル) にそのフィールドに関する簡潔なヘルプが表示されます。

そのフィールド、またはページ全体および関連タスクに関する拡張情報を表示するには、ヘルプ・ポータルの下部にある「このページの追加情報 (More information about this page)」リンクをクリックします。

#### 関連概念

『管理コンソール入門』

11 ページの『管理コンソール・タスク・フィルターの概要』  
タスク・フィルタリングを使用して不要なページを除去することにより、WebSphere 管理コンソールの表示をカスタマイズできます。

15 ページの『管理コンソール・ページ』

17 ページの『管理コンソールのボタン』

## 管理コンソール入門

管理コンソールを使用して WebSphere Process Server リソースの管理を開始するには、以下に示すタスクが役立ちます。

- **管理コンソールのサーバーを始動します。**

スタンドアロン・サーバー環境の場合、startServer コマンドを実行すると、指定されたアプリケーション・サーバーの構成ファイルが読み取られ、そのサーバーが始動されます。複数のサーバーのグループ (セル) をデプロイメント・マネージャーが管理している場合は、代わりに startManager コマンドを使用します。

startServer および startManager の使用については、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

- **管理コンソールを始動します。**

詳しくは、8 ページの『管理コンソールの始動と停止』を参照してください。

- **コンソールの設定を指定します。**

設定により、管理コンソールでのデータの表示方法と、ワークスペースの振る舞いを制御します。10 ページの『管理コンソール設定の変更』を参照してください。

- **コンソール・スコープを設定します。**

スコープは、リソースが管理コンソールに表示されるレベルを指定します。リソースは、セル、ノード、クラスター、サーバーいずれかのスコープでコンソール

ル・コレクション・テーブルに表示されます。スコープ設定の詳細については、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

- **情報を表示するためのフィルターを作成します。**

フィルターは、コレクション・ページの列にどのデータを表示するかを指定します。10 ページの『管理コンソール・フィルターの設定』を参照してください。

- **オプション: コンソール用のセッション・タイムアウトを設定します。**

デフォルトでは、非アクティブのまま 30 分経過すると、コンソール・セッションがタイムアウトになります。WebSphere Application Server for z/OS インフォメーション・センターに説明されているように、`deployment.xml` 構成ファイルを編集すれば、この値は変更できます。

- **自分の作業をマスター・リポジトリに保管します。**

変更内容をマスター・リポジトリに保管するまで、コンソールは、ローカル・ワークスペースを使用して変更内容を追跡します。変更内容を保管するには、「システム管理」>「変更内容をマスター・リポジトリに保管 (Save Changes to Master Repository)」をクリックして「保管」ページを表示してから、「保管」をクリックします。

#### 管理コンソールの始動と停止:

管理コンソールを利用するには、管理コンソールを始動してログインする必要があります。コンソールでの作業が終了したら、作業内容を保管してログアウトします。

コンソールを始動してログインするには、以下の手順を実行します。

1. 管理コンソールを始動します。
  - a. 管理コンソールへのアクセスに使用する Web ブラウザーで、Cookie を使用可能にします。
  - b. **オプション:** JavaScript を使用可能にします。管理コンソールのすべての機能が使用可能になるように、JavaScript™ を使用可能にすることをお勧めします。
  - c. Cookie を使用可能にした Web ブラウザーで、以下のように入力します。

`http://your_fully_qualified_server_name:portNumber/ibm/console`

ここで、`your_fully_qualified_server_name` には、管理サーバーが入っているマシンの完全修飾ホスト名を指定し、`portNumber` は、管理コンソールのポート番号です。管理コンソールがローカル・マシン上にある場合、セキュリティが有効になっていなければ、`your_fully_qualified_server_name` は `localhost` になります。

Windows® プラットフォームで `localhost` が認識されない場合は、実際のホスト名を使用してください。

セキュリティが有効になっていれば、要求は `https://your_fully_qualified_server_name:secure_portNumber/ibm/console` にリダイレクト

されます。ここで、*your\_fully\_qualified\_server\_name* は管理サーバーが入っているマシンの完全修飾ホスト名、*secure\_portNumber* は管理コンソールのセキュア・ポート番号です。

**注:** 非セキュア管理コンソールのデフォルト・ポート番号は、ポート 9060 です。セキュア管理コンソールのデフォルト・ポート番号は、9043 です。プロファイル作成中にデプロイされた新規の管理コンソールには、それぞれ新規の非セキュア・ポート番号が割り当てられます。プロファイル作成中にセキュリティーを有効にした場合には、新規のセキュア・ポート番号が割り当てられます。

- d. コンソール・アプリケーションを実行するサーバーの `System.Out.log` ファイルをチェックして、コンソール・アプリケーションが正常に開始したことを確認します。正常に開始した場合には、「CWWSR0221I: アプリケーションが開始されました: isclite (CWWSR0221I: Application started: isclite)」というメッセージが生成されます。

マシン上で既に実行中のアプリケーションとコンソール・ポートが競合するためにコンソールを始動できない場合は、以下のファイル内でポート番号を変更します。

- `install_root/profiles/profile_name/config/cells/cell_name/nodes/node_name/serverindex.xml`
- `install_root/profiles/profile_name/config/cells/cell_name/virtualhosts.xml`

プロファイル作成中に選択したポート (デフォルトでは 6090) をファイル内で検索し、そのポート番号をすべてコンソールのポートに変更します。または、競合するポートを使用している別のアプリケーションをシャットダウンした後で、管理コンソールを始動します。

管理コンソールがブラウザーにロードされ、ログイン・ページが表示されます。

## 2. コンソールにログインします。

- a. 「**ユーザー ID**」フィールドに、ユーザー名またはユーザー ID を入力します。

**注:** 既に使用中 (およびセッション中) の ID を入力すると、以下のいずれかを行うことを求めるプロンプトが出ます。

- 同じユーザー ID を持つもう一方のユーザーをログアウトすると、そのユーザーのセッション中に加えられた変更を回復できます。
- ログイン・ページに戻り、別のユーザー ID を入力する。

サーバー構成に加えられたすべての変更は、ユーザー ID に保管されます。セッションがタイムアウトになった場合も、サーバー構成はユーザー ID に保管されます。

- b. コンソールのセキュリティーが有効な場合は、「**パスワード**」フィールドにパスワードを入力する必要があります。
- c. 「**OK**」をクリックします。

ここで、管理コンソールに「ようこそ」ページが表示されます。

## 3. コンソールからログオフします。

- このセッション中の作業内容を保管するには、「システム管理」>「変更内容をマスター・リポジトリに保管 (Save Changes to Master Repository)」>「保管」をクリックしてから、「ログアウト」をクリックしてコンソールを終了します。
- 変更内容をリポジトリに保管しないでコンソールを終了するには、「ログアウト」をクリックします。

作業内容を保管する前にブラウザを閉じた場合は、同じユーザー ID を使用して次にログインしたときに、保管しなかった変更内容を回復できます。

#### 管理コンソール設定の変更:

管理コンソール設定によって、コレクション・ページ (データまたはリソースのコレクションをテーブルに表示するページ) 上のデータの表示をカスタマイズできます。設定はユーザー・レベルで指定され、通常、管理コンソールの領域ごとに別個に設定する必要があります。

コレクション・ページに対して以下の表示設定を指定できます。

- 「**最大行数**」 - コレクションの規模が大きい場合に、表示される行の最大数を指定します。行数が指定した最大値を超える場合、後続のページに表示されます。デフォルト値は 20 です。
- 「**フィルター基準の保存**」 - フィルター機能で入力された最後の検索基準が保存されるかどうかを指定します。保存が指定されている場合、コンソール・コレクション・ページでは、設定に従ってテーブルにデータを表示する際に最初に保存されているフィルター基準を使用します。詳しくは、『管理コンソール・フィルターの設定』を参照してください。
- 「**最大結果セットサイズ**」 - 検索の結果戻されるリソースの最大数を指定します。デフォルト値は 500 です。
- 「**最大列幅**」 - コレクション列で表示可能な文字の最大数を指定します。デフォルト値は 18 です。

以下のステップを実行して、コレクション・ページ用の表示設定を指定します。

1. 任意のコレクション・ページから「**設定**」をクリックします。

このページは設定フィールドを表示するために拡大されます。

2. 「**最大行数**」、「**フィルター基準の保存**」、「**最大結果セットサイズ**」、および「**最大列幅**」フィールドの値を必要に応じて変更します。
3. 「**適用**」をクリックします。

コレクション・テーブルが更新され、指定した値に従って表示されます。

ワークスペースの自動更新やデフォルトで使用するスコープなど、管理コンソールのグローバル設定も指定できます。管理コンソールで「設定」ページにアクセスするには、「システム管理」>「コンソール設定」>「設定」をクリックします。これらの設定の指定方法の資料については、WebSphere Application Server インフォメーション・センターを参照してください。

#### 管理コンソール・フィルターの設定:

管理コンソールのコレクション・ページの各テーブルには、WebSphere Process Server のデータまたはリソースのリストが表示されます。フィルターを使用すると、テーブルの特定の列に、どのリソースまたはデータを表示するかを正確に指定できます。1 つの列のみに対してフィルターを設定可能です。

1. テーブルの上部のボタンから「**ビューのフィルター (Filter the view)**」をクリックします。

「フィルター」ダイアログ・ボックスが、テーブルの一番上の行の上にかきまします。

2. 「**フィルター**」ドロップダウン・メニューを使用して、フィルターに含める列を選択します。
3. 「**検索語**」フィールドにフィルター基準を指定します。

この基準はストリングで、名前にこのストリングが含まれているテーブル項目のみが表示されます。ストリングでは、ワイルドカード文字としてパーセント記号 (%)、アスタリスク (\*)、または疑問符 (?) 記号を使用できます。例えば、「リソース・アダプター (Resource Adapter)」ページで、「名前」列のフィルター基準として \*JMS\* を入力すると、名前にストリング JMS が含まれるすべてのリソース・アダプターを検索できます。

検索を実行する正規表現エンジンが正しく検索基準との突き合わせを行えるように、ストリングの一部として現れる ( ) ^ \* % { } ¥ + & の各文字の前には円記号 (¥) を置きます。

例えば、プロバイダー名に (XA) が含まれているすべての Java DataBase (JDBC) プロバイダーを検索する場合は、次のようなストリングを「検索語」フィールドに指定します。

\*¥(XA¥)\*

4. 「**実行 (Go)**」をクリックします。

テーブルが更新され、選択された列でフィルター基準に適合する項目のみが表示されます。

## 管理コンソール・タスク・フィルターの概要

タスク・フィルタリングを使用して不要なページを除去することにより、WebSphere 管理コンソールの表示をカスタマイズできます。

タスク・フィルタリングによって、WebSphere 管理コンソール内のタスクをグループ化できます。タスク・フィルタリングには以下の 3 つのオプションがあります。

**すべて** すべての管理コンソール・ページを表示します。管理者は、このオプションを使用して、WebSphere Process Server および基盤となる WebSphere Application Server のすべての部分を管理します。

### アプリケーションの統合

以下のタスクに重点を置いたページを表示します。

- サービス統合バスの構成
- メディエーション・モジュール、ビジネス・モジュール、およびサービス・アプリケーションのデプロイメントをサポートするために使用されるメッセージ・エンジンとネットワーク・トポロジーの構成

- デプロイされたサービス・アプリケーション、ビジネス・モジュール、およびメディエーション・モジュールに必要なリソース (例えば、JMS 接続ファクトリーや Common Event Infrastructure プロファイル) の作成
- メディエーション・モジュールとサービス・アプリケーションの動作のモニター

管理者は、このオプションを使用して、ビジネス・モジュールとメディエーション・モジュールをデプロイおよび管理します。このオプションは、必要なバスとサーバーが存在していることを前提とします。

### サーバーおよびバス

以下のタスクに重点を置いたページを表示します。

- メディエーション・モジュール、ビジネス・モジュール、およびサービス・アプリケーションのデプロイメントをサポートするために必要とされるサービス統合バス、サーバー、サーバー・クラスター、メッセージング・エンジン、およびネットワーク・トポロジーの定義
- インフラストラクチャー・サービスを使用可能および使用不可に設定
- アプリケーション、ビジネス・モジュール、およびメディエーション・モジュールのインストール
- デプロイされたサービス・アプリケーションおよびメディエーション・モジュールに必要なリソース (例えば、JMS 接続ファクトリーや Common Event Infrastructure プロファイル) の作成
- バスおよびサーバー環境の動作のモニター

管理者は、このオプションを使用して、サービス・アプリケーションおよびメディエーション・モジュールのデプロイメントをサポートするために必要なバスおよびサーバー環境を管理します。管理には、ネットワークおよびバス・トポロジーの定義、適切なリソースの定義、ランタイム・システムのモニター、およびランタイム・エラーのトラブルシューティングが含まれます。このオプションを使用して、アプリケーションの技術環境を作成します。

最初に WebSphere 管理コンソールを開くと、デフォルトで「すべて」が選択されません。

### タスク・フィルターの使用:

タスク・フィルターを選択して、WebSphere 管理コンソールをカスタマイズします。

タスク・フィルターを選択して管理コンソール内のいくつかのページを除去することによって、WebSphere 管理コンソールをカスタマイズできます。例えば、特定のサーバー構成を表示する場合は、タスク・フィルターを使用して、そのサーバーに関連するページのみを表示できます。タスク・フィルターを適用するには、以下の手順を実行します。

1. ナビゲーション・ツリーで、「ようこそ」リンクをクリックします。
2. ページの下部近くにある「タスク・フィルタリング・セレクター」内で、使用可能なタスク・フィルターのリストからタスク・フィルターを選択します。
3. 「適用」をクリックします。



「適用」をクリックすると、そのタスク・フィルターは即時に有効になります。管理コンソールをシャットダウンしてから再始動する必要はありません。

### 関連概念

タスク・フィルターの概要

タスク・フィルタリングを使用して不要なページを除去することにより、WebSphere 管理コンソールの表示をカスタマイズできます。

## 管理コンソールのガイド付きアクティビティおよびウィザード

ガイド付きアクティビティは、特定の作業を支援する一連のステップです。作業の特定の部分の操作をガイドするために、ガイド付きアクティビティによってウィザードが開始される場合があります。

一部の機能 (Network Deployment など) をサポートするようクラスターまたはサーバーを構成する作業は複雑であるため、作業に精通していないと困難場合があります。IBM® 提供のガイド付きアクティビティは、作業について説明し、構成の手順をガイドするウィザードを開始します。これらの構成作業に精通するまでは、ガイド付きアクティビティを使用してください。

### ガイド付きアクティビティとは

ガイド付きアクティビティは、特定の目的を達成するために必要な論理的に関連のある一連の作業について説明します。ガイド付きアクティビティの各ステップは、構成の各部を実行するために必要な作業 (クラスターの構成など) に対応しています。構成をさらに簡素化するために、一部のステップには、そのステップで完了する必要がある一連の構成パネルの操作をガイドするウィザードが用意されています。図 1 に、ガイド付きアクティビティが表示された管理コンソールを示します。

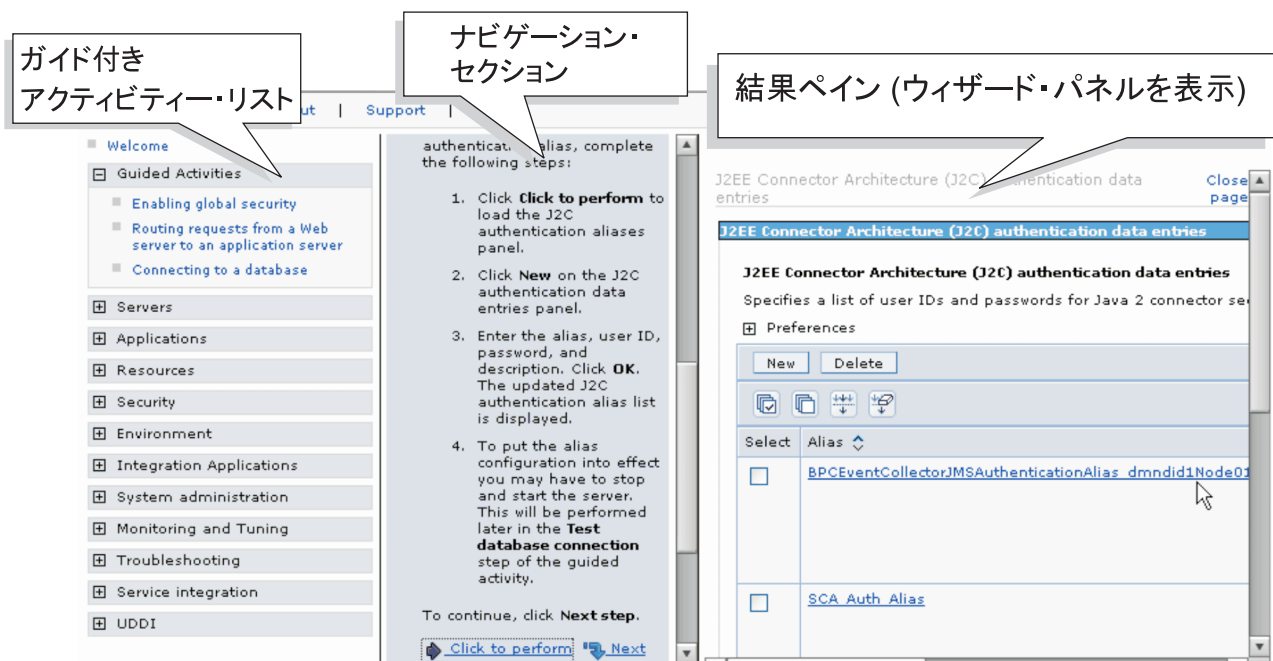


図 1. ガイド付きアクティビティ

## ウィザードとは

ウィザードは、構成の特定の目的を達成するために必要な一連の構成パネルの操作をガイドします。表示されるパネルの数は、作業の複雑さによって異なります。変更内容を保管すると、ガイド付きアクティビティーの次のステップに進むことができます。説明にある「**クリックしてステップを実行 (Click to perform step)**」を選択すると、ガイド付きアクティビティーによってウィザードが開始されます。

### ガイド付きアクティビティーとウィザードの使用:

ガイド付きアクティビティーとウィザードは、サーバーおよびクラスターの複雑な構成タスクの実行に役立ちます。これらのアクティビティーを使用すると、個別の構成タスク内のステップについての理解を深め、構成エラーを防止できます。タスクについて十分に理解するまではガイドを使用し、その後はガイドなしでタスクを実行できます。

タスクを実行するには、正しい権限を持つユーザー ID で管理コンソールにログインしている必要があります。

特定の構成タスクについて十分な知識がない場合またはそのタスクを実行する機会が少ない場合は、利用可能なガイド付きアクティビティーを使用すると役立つことがあります。これらのパネルは、実行しているタスクについての背景情報を表示するだけでなく、タスクの完了を支援するための段階的な指示も提供します。ガイド付きアクティビティーによって、説明されたタスクを実行するウィザードが起動されることもあります。

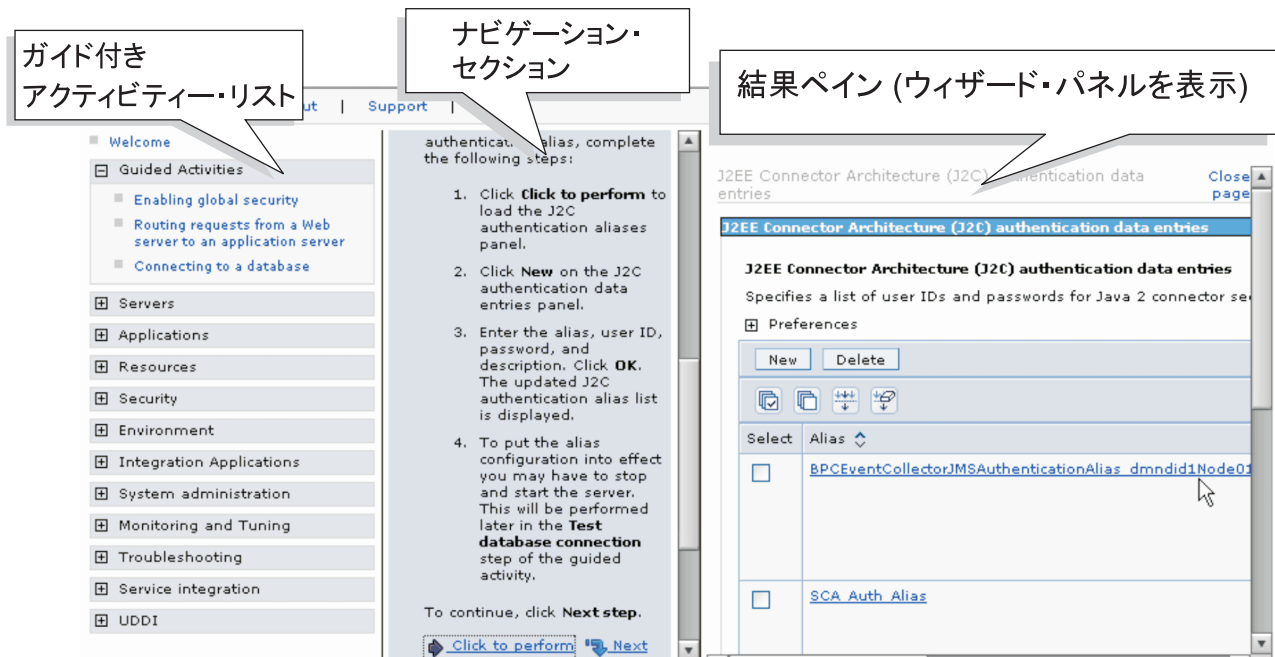


図2. ガイド付きアクティビティー

**ヒント:** ガイド付きアクティビティーを最初に使用するときには、ステップを実行する前にすべてのステップの説明を読んでください。アクティビティーの説明を通して読むと、実行する作業の概要および各ステップの間の相互作用

用を理解できます。最初に説明を読んでいるときには、「**クリックしてこのステップを実行 (Click to perform this step)**」を選択してウィザードを起動しないでください。

1. 管理コンソールの「**ガイド付きアクティビティ (Guided Activities)**」セクションを展開します。

「**ガイド付きアクティビティ (Guided Activities)**」の横にある正符号 (+) をクリックします。

2. 表示されるタスクのリストから、実行するタスクを選択します。

リスト内の項目をクリックします。コンソールに、2 つの追加セクション (ナビゲーション・セクションと結果ペイン) が表示されます。(『ガイド付きアクティビティ』を参照してください。) ナビゲーション・セクションにはタスクの概要が表示され、結果ペインにはタスクについての背景情報が表示されます。

3. ガイド付きアクティビティを開始するには、「**開始**」を選択します。

ナビゲーション・ペインに、タスクの次のステップが表示されます。

4. 説明を読み、必要な場合には説明されているアクションを実行します。「**クリックしてこのステップを実行 (Click to perform this step)**」が表示されている場合にはこのリンクを選択して、関連する構成ウィザードを起動します。

結果ペインに、ナビゲーション・ペインで説明されている構成タスクの最初のパネルが表示されます。

5. **オプション:** 構成パネルが使用しやすくなるように、ペインを調整します。
  - a. ペインを区切っている線をクリックします。
  - b. 線を左右にドラッグして、ペインの幅を調整します。
6. ウィザードから要求されるアクションを実行します。

ステップの実行を支援するウィザードを完了すると、ガイド付きアクティビティ内のステップに戻ります。

**ヒント:** 説明を読み直して、ステップの目標を達成できたかどうかを確認し、それらのアクションがアクティビティのフロー全体の中で占めている位置付けを再確認します。

7. タスクを完了するまで、ステップ 4 から 6 までを繰り返します。

## 管理コンソール・ページ

管理コンソール・ページは、コレクション・ページ、詳細ページ、およびウィザード・ページの 3 つのいずれかのフォーマットになっています。それぞれのタイプのページのレイアウトおよび振る舞いを理解すると、効果的に使用できるようになります。

## コレクション・ページ

コレクション・ページは既存の管理オブジェクト (例えば、リレーションシップ、失敗したイベント、またはリソース・アダプター) のコレクションを管理します。これには、次のエレメントが 1 つ以上含まれています。

## 有効範囲および設定

有効範囲および設定によって、どの管理オブジェクトを表に表示するか、およびそれらをどのように表示するかを決定できます。

## 既存オブジェクトの表

この表は、コレクション・ページで指定したタイプの、既存の管理オブジェクトを示しています。表の列は、これらのオブジェクトの主要な設定値をまとめたものです。オブジェクトがまだ存在していない場合は、表は空です。選択可能なボタンを使用して、新規オブジェクトを作成してください。

## アクションを実行するためのボタン

標準的なボタンは 17 ページの『管理コンソールのボタン』で説明されています。ほとんどの場合、コレクション・テーブルから 1 つ以上のオブジェクトを選択してから、ボタンをクリックする必要があります。アクションは、選択されたすべてのオブジェクトに適用されます。

## ソートのトグル・ボタン

表の各列見出しの後には、項目を昇順 (▲) または降順 (▼) にソートするためのアイコンがあります。デフォルトでは、オブジェクト名などの項目はアルファベット順に降順でソートされます。

## 詳細ページ

詳細ページは、オブジェクトに関する詳細を表示したり、特定のオブジェクト (アプリケーション・サーバーまたはリスナー・ポート拡張など) を構成するために使用されます。通常、これには次のエレメントが 1 つ以上含まれています。

### 構成タブの付いたページ

このタブ付きページは、管理オブジェクトの構成を変更するために使用されます。各構成ページには、オブジェクト固有の一連の一般プロパティが含まれています。構成している管理オブジェクトのタイプに応じて、ページに追加のプロパティを表示できます。

このタブ付きページへの変更を有効にするには、サーバーの再始動が必要になることがあります。

### ランタイム・タブの付いたページ

このタブ付きページには、管理オブジェクトの現在使用中の構成が表示されます。これは、読み取り専用であることがあります。一部の詳細ページにランタイム・タブがないことに注意してください。

このタブ付きページへの変更は、即時に有効になります。

### ローカル・トポロジー・タブの付いたページ

このタブ付きページには、管理オブジェクトの現在使用中のトポロジーが表示されます。異なるレベルのトポロジーを展開および縮小表示して、トポロジーを表示します。一部の詳細ページにローカル・トポロジー・タブがないことに注意してください。

## アクションを実行するためのボタン

特定のアクションを実行するボタンは、構成タブの付いたページとランタイム・タブの付いたページにのみ表示されます。標準的なボタンは 17 ページの『管理コンソールのボタン』で説明されています。

## ウィザード・ページ

ウィザード・ページは、いくつかのステップからなる構成プロセスの実行に役立ちます。ウィザードは、構成している特定のオブジェクトの特性に応じて、特定のステップを表示したり非表示にしたりすることができます。13 ページの『管理コンソールのガイド付きアクティビティおよびウィザード』を参照してください。

## 管理コンソールのボタン

管理コンソール・インターフェースには、現在表示されているページに応じて、多数のボタンが含まれています。このトピックでは、使用可能なコンソール・ボタンについて説明します。

以下のグラフィック・ボタンは、WebSphere Process Server リソースを表示するテーブルの上部に配置されています。

ボタン	結果のアクション
すべてにチェック・マークを付ける	テーブルにリストされているそれぞれのリソースを選択します (例えば、失敗したイベント、またはリレーションシップ・インスタンス)。これは、これらのリソースに対してアクションを実行するための準備です。
チェック・マークをすべて外す	すべての選択済みリソースをクリアします。これにより、リソースに対するアクションは実行されなくなります。
フィルター・ビューを表示する	フィルターを設定するためのダイアログ・ボックスが開きます。フィルターは、テーブルに表示するリソースのサブセットを指定するために使用します。10 ページの『管理コンソール・フィルターの設定』を参照してください。
フィルター・ビューを隠す	フィルターを設定するために使用するダイアログ・ボックスを非表示にします。
フィルター値をクリア	フィルターに対して行ったすべての変更をクリアし、最後に保管した値を復元します。

以下のボタンが、WebSphere Process Server 管理コンソール・ページの下部に表示されます。すべてのボタンがすべてのページに表示されるわけではありません。

**追加** 選択または入力した項目をリストに追加するか、または項目をリストに追加するためのダイアログ・ボックスを表示します。

**適用** ページを終了せずに、ページへの変更を保管します。

**戻る** シーケンスの直前のページまたは項目を表示します。管理コンソールでは、Web ブラウザーの「戻る」オプションおよび「進む」オプションの使用をサポートしていません。これらを使用すると、偶発的な問題が起こる可能性があります。代わりに、コンソールの「戻る」または「キャンセル」ボタンを使用します。

### キャンセル

現在のページまたはダイアログ・ボックスを終了します。保管されていないすべての変更は破棄されます。管理コンソールでは、Web ブラウザーの「戻る」オプションおよび「進む」オプションの使用をサポートしていません。これらを使用すると、偶発的な問題が起こる可能性があります。代わりに、コンソールの「戻る」または「キャンセル」ボタンを使用します。

**クリア** 変更をクリアして、最後に保管された値を復元します。

**選択をクリア**

このタブ付きページの表で選択されているセルをすべてクリアします。

**閉じる** ダイアログを閉じます。

**削除** 選択したインスタンスを除去します。

**OK** 変更を保管して、ページを終了します。

**リセット**

タブまたはページ上の変更を消去し、最後に保管された値を復元します。

**保管** ローカル構成の変更をマスター構成に保管します。

管理コンソールで使用するボタン ( WebSphere Application Server と WebSphere Process Server の両方のリソースの管理用) の完全なリストについては、 WebSphere Application Server for z/OS の資料を参照してください。

---

## WebSphere Process Server 管理環境ロードマップのセットアップ

WebSphere Process Server 管理環境が正しくセットアップされるようにするには、特定のタスクを実行する必要があります。

WebSphere Process Server を正しくインストールした後に、管理環境およびサーバー環境をセットアップする必要があります。表 2 に、この環境のセットアップに関連するタスクをリストします。

表 2. WebSphere Process Server 管理環境のセットアップに関連するタスク

タスク	参照先
製品の構成	インストール後の製品の構成
ポートの構成	ポート競合の回避および での バージョンにおけるポート番号の設定 でのバージョンにおけるポート番号の設定ポート番号の設定
管理アーキテクチャーのセットアップ	管理アーキテクチャーのセットアップ
セル全体に対する設定を構成	セル全体に対する設定を構成
サーバー構成ファイルの変更	サーバー構成ファイルの使用
サーバーおよび Network Deployment 環境の管理	プロセス・サーバーの管理
WebSphere Process Server リソースの構成	WebSphere Process Server リソースの構成

表 2. WebSphere Process Server 管理環境のセットアップに関連するタスク (続き)

タスク	参照先
アプリケーションのデプロイおよび管理	アプリケーションのデプロイおよび管理およびモジュールの準備とインストールの概要
WebSphere Process Server サービスの管理	ビジネス・プロセスおよびヒューマン・タスクの管理、ビジネス・ルール、エンタープライズ・アプリケーションの管理、および Application Scheduler の構成

## 管理アーキテクチャーのセットアップ

ご使用の処理環境を構成するサーバーによってワークフローを制御する、サービス、デプロイメント・マネージャー、および環境のその他の部分をセットアップする方法。

このタスクでは以下の作業を完了していることが前提となります。

1. PDF 「WebSphere Process Server for z/OS Installation and Configuration」の『Loading the product code from the installation media onto z/OS』の説明に従って、テープから製品をロード済みである。
2. 「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『Running the installation script』の説明に従って、インストール・スクリプトを実行して製品定義を作成済みである。
3. 「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『Configuring the software』の説明に従って、ソフトウェアを構成済みである。
4. 「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『Avoiding port conflicts』および『Port number settings』の説明に従って、ポートを構成済みである。

始める前に、WebSphere Application Server for z/OS インフォメーション・センターの『管理アーキテクチャーのセットアップ』で説明されているタスクを確認します。

WebSphere Process Server のインストールおよびセットアップ後は、組み込まれたノードとそれらのノード上のリソースを、管理コンソールを使用してモニターし、制御する必要があります。

1. 管理サービス設定を使用して管理サービスを構成します。
2. データベース製品のインスタンスをインストールします。
3. データベース・インスタンス内に WebSphere Process Server データベースを作成します。
4. WebSphere Process Server for z/OS データベースを作成します。WebSphere Process Server for z/OS のインストールと構成のガイドの「ソフトウェアの構成」セクションにある「データベースおよびストレージ・グループの作成」を参照してください。

**DBUtility.sh** を使用してデータベースおよびデータベース表を作成できます。**DBUtility.sh** は、サポートされているデータベース・タイプの表を自動的にセ

ットアップします。DBUtility.sh は、拡張された WebSphere Application Server for z/OS<sup>®</sup> 構成 HFS 内にあります。例えば、/WebSphere/V6R0/AppServer/bin/DBUtility.sh です。

応答ファイルのパラメーターで、製品構成の一環としてデータベース表定義を作成するか、あるいは、後で DDL および SQL を手動で実行することにより定義を作成するかを指定します。DDL および SQL を実行するタイミングを決定する応答ファイルのプロパティの名前は **dbDelayConfig** です。このプロパティの設定方法の詳細については、WebSphere Process Server for z/OS インストールと構成のガイドの『ソフトウェアの構成』セクションにあるトピック『サンプル応答ファイル』を参照してください。

製品構成スクリプトを実行した後に、手動でデータベース表定義を作成する方法については、WebSphere Process Server for z/OS インストールと構成のガイドの『手動によるデータベースの構成』を参照してください。

5. セルを構成します。
6. デプロイメント・マネージャーを構成します。
7. 管理コンソールで、「環境」>「WebSphere 変数」> (ノードのスコープ) > *DatabaseName\_JDBC\_DRIVER\_PATH* と選択して、WebSphere Process Server データベースがある場所を定義します。

以下の変数のいずれかを使用して、データベースを定義します。

- DB2<sup>®</sup> の場合: DB2UNIVERSAL\_JDBC\_DRIVER\_PATH
- Cloudscape の場合: 分散環境ではサポートされていません。

詳しくは、『WebSphere 変数の構成』を参照してください。

8. ノードを管理します。
9. ノード・エージェントを管理します。
10. ノード・グループを管理します。
11. リモート・ファイル・サービスを構成します。

WebSphere Process Server 環境で作業できるようになります。

これで、アプリケーションをインストールできるようになりました (WebSphere Process Server の「モジュールの開発とデプロイ」の PDF の『モジュールの準備とインストールの概要』を参照してください)。

## セル全体に対する設定の構成

Web アプリケーション、Web コンテナ、およびアプリケーション・サーバー間の要求の処理を支援するために、仮想ホストおよび変数用のセル全体に対する設定を構成できます。

このタスクは、既に「管理アーキテクチャーのセットアップ」で説明する管理アーキテクチャーのセットアップを行っていることを前提としています。

Network Deployment 環境を設定する場合、またはクラスターにアプリケーションをインストールする計画の場合にも、セル全体に特定の設定を構成する必要があります。



- 仮想ホストの構成
- 変数の構成

WebSphere Process Server データベースを定義する変数をノードまたはセルのスコープで構成する必要があります。変数 `DatabaseName_JDBC_DRIVER_PATH` の場合、以下のいずれかを使用します。

- DB2 の場合: `DB2UNIVERSAL_JDBC_DRIVER_PATH`
- Cloudscape™ の場合: 分散環境ではサポートされていません。

## WebSphere Process Server サーバー・ロードマップの管理

WebSphere Process Server サーバー構成には、エンタープライズ・アプリケーションおよびそのコンポーネントを実行するサービスをサーバーがどのように提供するかを制御する設定があります。管理者は、既存アプリケーション・サーバー環境で、サーバーを作成および構成できます。

WebSphere Process Server 管理者は、1 つ以上のサーバーを構成し、以下のようなタスクを実行できます。

表 3. サーバー管理タスク

タスク	参照先
サーバーの作成	この PDF で「サーバーの作成」を参照してください。
アプリケーション・サーバーの管理	この PDF で「アプリケーション・サーバー・ロードマップの管理」を参照してください。
トランスポート・チェーンの構成	トランスポート・チェーンの構成
カスタム・サービスの開発	カスタム・サービスの開発
アプリケーション・サーバー用プロセスの定義	アプリケーション・サーバー・プロセスの定義
Java 仮想マシンの構成	JVM の構成

### サーバーの作成

WebSphere Process Server の場合、サーバーを作成するには、`wsadmin createApplicationServer` コマンドか、管理コンソールの「新規アプリケーション・サーバーの作成」ウィザードを使用します。

このタスクは、製品のインストールおよび構成が完了していることを前提にしています。

サーバーを作成して、WebSphere Process Server アプリケーションの処理環境を提供します。

方法を選択して、その方法のステップを実行します。

方法	実行するステップ
<p>「新規アプリケーション・サーバーの作成」ウィザード</p>	<ol style="list-style-type: none"> <li>1. 管理コンソールの「アプリケーション・サーバー」ページで、「新規」をクリックして「新規アプリケーション・サーバーの作成」ページを開きます。</li> <li>2. 以下のようにして、サーバーを定義します。 <ol style="list-style-type: none"> <li>a. サーバーのノードを選択します。</li> <li>b. サーバーの名前を入力します。この名前はノード内で固有でなければなりません。</li> <li>c. <b>defaultProcessServer</b> テンプレートを選択して、サーバーを作成します。既存アプリケーション・サーバーをテンプレートとして使用することもできます。新規プロセス・サーバーは、テンプレート・サーバーのすべてのプロパティを継承します。</li> </ol> </li> <li>3. 新規のサーバーが、HTTP トランスポートごとに固有のポートを持つようにするかどうかを選択します。 <p>デフォルトでは、このオプションは有効になっています。このオプションを選択すると、このサーバーと共に使用する予定の仮想ホストの別名リストを、上記の新しいポート値を含むように更新しなければならない可能性があります。(このリストには、ステップ 4 の <b>Business Process Choreographer</b> 構成に使用するホスト別名が含まれます。 <b>Business Process Choreographer</b> 自体はアプリケーションであるため、<code>host_alias</code> 管理が必要です。) このオプションを選択解除する場合は、デフォルトのポート値が同じワークステーション上の他のサーバーと競合しないようにしてください。</p> <p>注: 既存のアプリケーション・サーバーをモデルとして使用してサーバーを作成する場合、アプリケーションを既存のサーバーから新規のサーバーにマップする選択は行わないでください。デフォルトでは、このオプションは無効になっています。</p> </li> <li>4. <b>オプション:</b> サーバーでビジネス・プロセスまたはヒューマン・タスクを含むアプリケーションを実行する場合は、<b>Business Process Choreographer</b> を構成します。詳しくは、『<i>Configuring Business Process Choreographer</i>』を参照してください。</li> <li>5. 『<i>Business Process Choreographer の構成</i>』の説明に従って、<b>SchedulerCalendars</b> アプリケーションを 1 つ以上のサーバーにインストールします。</li> </ol>
<p><b>wsadmin</b> <b>createApplicationServer</b> コマンド</p>	<p>「AdminTask オブジェクトのコマンド」で <code>createApplicationServer</code> コマンド名を参照してください。</p>

## アプリケーション・サーバー・ロードマップの管理

管理コンソールの「アプリケーション・サーバー」パネル、または startServer および stopServer などのコマンド行ツールを使用して、サーバーを管理できます。管理コンソール・インターフェースについて説明します。

サーバーについての情報を表示するには、管理コンソールの「アプリケーション・サーバー」パネルを使用します。

注: セル内のノードの一部をアップグレードし、それ以外は旧リリース・レベルのままにしておくことができます。つまり、一定期間、現行リリースのサーバーと新リリースが稼働するサーバーを、同じセル内で管理できます。この混合環境では、旧リリース・レベルが稼働するサーバーで実行できることに制限があります。新リリース・レベルで稼働するサーバーで実行できることには、制限はありません。詳しくは、この PDF の『WebSphere Process Server サーバー・ロードマップの管理』セクションの『サーバーの作成』を参照してください。

表 4 に、アプリケーション・サーバーを管理するタスクを示します。

表 4. 管理コンソールで使用可能なサーバー管理タスク

タスク	実行方法
使用可能なサーバーの表示	コンソールのナビゲーション・ツリーで、「サーバー」>「アプリケーション・サーバー」をクリックします。  「アプリケーション・サーバー」ページには、サーバーと、そのサーバーを保持するセルおよびノードがリストされています。このページには、それぞれのサーバーの状況も表示されています。この状況は、サーバーが開始済みであるか、停止中であるか、または使用不可であることを示します。
特定のサーバーに関する情報の表示	「名前」の下の特定のサーバー名をクリックします。これにより、このサーバーの「アプリケーション・サーバー設定」ページにアクセスします。
プロセス・サーバーの作成	この PDF の『WebSphere Process Server サーバー・ロードマップの管理』セクションの『サーバーの作成』
サーバーの始動	アプリケーション・サーバーの始動
サーバー操作のモニター	ランタイム・コンポーネントによる問題の検出と処理
サーバーの停止	アプリケーション・サーバーの停止
サーバーの削除	<ol style="list-style-type: none"><li>1. コンソール・ナビゲーション・ツリーで「サーバー」&gt;「アプリケーション・サーバー」とクリックして、「アプリケーション・サーバー」ページにアクセスします。</li><li>2. 削除するには、サーバーの横のチェック・ボックスを選択します。</li><li>3. 「削除」をクリックします。</li><li>4. 削除を確認するために「OK」をクリックします。</li></ol>

## WebSphere Process Server リソース・ロードマップの構成

インストール後に、ご使用の WebSphere Process Server 環境の重要なリソースを構成する必要があります。例えば、Application Scheduler、ビジネス・プロセス、Common Event Infrastructure、ビジネス・ルール、セレクトター、およびリレーションシップなどです。

管理環境をセットアップするためのインストール後タスクで重要なステップとなるのは、重要なリソースの構成です。表 5 で、リソースを構成するための情報を見つけてみます。

表 5. リソース構成のステップ

リソース	構成方法
Application Scheduler	この PDF の『アプリケーションおよびアプリケーション・サービスの管理』セクションの『Application Scheduler の構成 (Configuring application scheduler)』を参照してください。
Business Process Choreographer	Business Process Choreographer の PDF の『Business Process Choreographer の構成 (Configuring Business Process Choreographer)』を参照してください。
Common Event Infrastructure	Common Event Infrastructure の PDF で「Configuring the Common Event Infrastructure」を参照してください。
ビジネス・ルール	この PDF の『ビジネス・ルールおよびセレクトターの概要』セクションの『ビジネス・ルール・マネージャー』を参照してください。
セレクトター・コンポーネント	この PDF の『ビジネス・ルールおよびセレクトターの概要』セクションの『セレクトター・コンポーネントの概要』を参照してください。
リレーションシップ・サービス	この PDF の『WebSphere Process Server 設定ロードマップの構成』セクションの『リレーションシップ・サービスの管理』を参照してください。

## サーバー構成ファイルの使用

必要に応じて、構成ファイルのデフォルト・ロケーションを変更できます。サーバー構成ファイルでは、使用可能なアプリケーション・サーバー、その構成、および内容が定義されています。

WebSphere Process Server のサーバー構成ファイルを処理するには、WebSphere Application Server for z/OS インフォメーション・センターの『サーバー構成ファイルの使用』トピックの手順に従ってください。

## アプリケーションのインストールと管理

製品のインストール後に、アプリケーションおよびその環境の管理に関する多数のインストール後タスクを実行する必要があります。

インストール後に、タスクのカスタマイズ、アプリケーションのデプロイ、サーバー環境の管理など、アプリケーションと環境の管理を行う必要があります。

- 詳しくは、WebSphere Application Server for z/OS バージョン 6.0.x インフォメーション・センターの『アプリケーションとその環境の管理』を参照してください。
- WebSphere Process Server 上へのアプリケーションのインストールについて詳しくは、『モジュールの準備とインストールの概要』を参照してください。
- アプリケーションの停止、開始、および変更については、WebSphere Application Server for z/OS バージョン 6.0.x インフォメーション・センターの『アプリケーションのデプロイおよび管理』を参照してください。

## アプリケーションおよびアプリケーション・サービスの管理

このセクションでは、管理インターフェースを使用して、ビジネス・プロセスとビジネス・タスク、ビジネス・ルール、スケジュールなど WebSphere Process Server のアプリケーションおよびアプリケーション・サービスを管理する方法を説明します。

### WebSphere Process Server 設定ロードマップの構成

ソフトウェアのインストール後に構成する WebSphere Process Server のサービスの一覧を示します。

WebSphere Process Server の正常なインストール時に、使用するさまざまなサービス用の設定を指定して構成を完了する必要があります。表 6 に、サービスおよび各サービスの設定の構成方法を説明したトピックをリストします。

表 6. 構成するサービス

サービス	参照先
Application Scheduler	『アプリケーションおよびアプリケーション・サービスの管理』セクションの『 <i>Application Scheduler</i> 』の『スタンドアロン・サーバー用 <i>Application Scheduler</i> の構成』を参照してください。
イベント・サービス	イベント・サービスの管理
拡張メッセージング・サービス	『拡張メッセージング・リソースの管理: 概要』セクションの『拡張メッセージング・サービスの使用可能化』を参照してください。
WebSphere Business Integration Adapter サービス	『アダプターの管理』セクションの『WebSphere アダプターの操作』を参照してください。
リレーションシップ・サービス	『リレーションシップ・サービスの管理』を参照してください。

## イベント・サービスの管理

「イベント・サービス」構成パネルには、イベント・インフラストラクチャーに渡される各イベントに WebSphere Process Server に関する情報が自動的に組み込まれるように設定するプロパティがリストされます。

管理コンソールを始動します。管理コンソールの始動またはアプリケーション・サーバーの表示方法に精通していない場合は、管理コンソールの関連トピックを参照してください。このタスクは管理コンソールで実行されます。

イベント・サービスがアクティブな場合、イベント・サービスは、イベント・インフラストラクチャーに渡されるイベントが自動的にサーバーの情報を組み込むようにします。「イベント・サービス」構成パネルを使用して、イベント・サービスの構成をオンまたはオフに設定します。

アプリケーション・サーバーのイベント・サービス開始プロパティを構成するには、以下のステップを実行します。

1. 構成するアプリケーション・サーバーを選択します。

ナビゲーション・パネルで、「サーバー」>「アプリケーション・サーバー」の順にクリックします。リストからアプリケーション・サーバーを選択します。

サーバー・プロパティが表示されます。

2. 「イベント・サービス」構成パネルを表示します。

ビジネス・インテグレーション・プロパティから「イベント・サービス」を選択します。

「イベント・サービス」構成パネルが表示されます。

3. 「始動時にサービスを使用可能にする (Enable service at startup)」フィールドを選択または選択解除します。

デフォルト値では、サービスは始動時に自動的に使用可能になります。このオプションが選択されていないときに、後続のアプリケーションでこのサービスの実行が必要となった場合、システム管理者がこのサービスを手動で開始する必要があります。あるいは、システム管理者がこのオプションを選択してサーバーを再始動することによって、このサービスを再開できます。

4. Java Naming and Directory Interface (JNDI) 名が正しいことを確認します。

「イベント・インフラストラクチャー・エミッター・ファクトリー JNDI 名」フィールドを検討します。代替プロファイルを作成していない場合、デフォルトの JNDI 名を受け入れることをお勧めします。

5. 必要に応じてカスタム・プロパティを選択します。

「カスタム・プロパティ」リンクをクリックして、リストからカスタム・プロパティを選択します。

6. 構成を保管します。

構成を保管し、「イベント・サービス構成」パネルに留まるには、「適用」をクリックします。構成を保管し、Business Integration パネルに戻るには、「OK」をクリックします。

7. 変更内容をサーバーに適用します。

アプリケーション・サーバーを停止して再始動し、変更内容を有効にします。

## リレーションシップ・サービスの管理

リレーションシップ・サービスは、システム内のリレーションシップおよびロールを保守します。これは、リレーションシップとロールの定義およびメタデータを管理します。また、リレーションシップの定義を指定して、定義から導き出されるインスタンスの操作を可能にします。

リレーションシップ・サービスにより、さまざまなオブジェクトの間のリレーションシップを取り込むことができます。リレーションシップの参加者は、果たすロールによって識別されます。例えば、Person オブジェクト「ジョー」は Car オブジェクト「スバル (ナンバー・プレート、XYZ 123)」との間に所有権リレーションシップを持つことができます。この例では、ジョーが「所有者」のロールでリレーションシップに参加し、自動車が「所有オブジェクト」のロールでリレーションシップに関与しています。

リレーションシップおよびロールは、WebSphere Integration Developer のリレーションシップ・エディター・ツールのグラフィカル・インターフェースで設計した定義で記述されます。リレーションシップ定義は、リレーションシップの概要を記述し、リレーションシップの各参加者が実行可能なロールを識別するテンプレートです。ロール定義により、参加者についての構造および制約要件を指定します。リレーションシップの定義は XML ファイルとして保管され、J2EE アプリケーションの一部として特定のサーバーにデプロイされます。

リレーションシップの作成、リレーションシップ・タイプの識別、およびリレーションシップ・エディターの使用に関する背景情報およびタスク情報については、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。

マップまたは WebSphere Process Server の他のコンポーネントの実行時にリレーションシップ・インスタンスが必要になると、リレーションシップのインスタンスは、シナリオに応じて作成または取得されます。リレーションシップおよびロール・インスタンス・データは、以下の 3 つの方法で操作できます。

- WebSphere Process Server コンポーネント Java スニペットによるリレーションシップ・サービス API の呼び出し
- WebSphere Process Server ビジネス・オブジェクト・マッピング・サービスでのリレーションシップ変換
- Relationship Manager ツールの使用

リレーションシップおよびロールのインスタンス・データは、リレーションシップ・サービスの構成時に指定したデフォルト・データ・ソースのデータベースに格納されるリレーションシップ・テーブルに保管されます。

リレーションシップ・サービスは、各サーバー上でセル・レベルで実行されます。「**Relationship Manager**」ホーム・ページの「**リレーションシップ・サービスについて (About)**」セクションに、リレーションシップ・サービスを実行中のセル内のサーバー数が表示されます。「**リレーションシップ**」セクションに、リレーションシップ・サービスを実行中の各サーバー名が表示されます。リレーションシップ・インスタンスで作業する前に、管理するリレーションシップおよびロールのインスタンスを持つサーバーを選択する必要があります。

Relationship Manager の使用について詳しくは、WebSphere Process Server インフォメーション・センターの Relationship Manager に関するトピックを参照してください。

以下のトピックでは、ご使用の WebSphere Process Server 環境のリレーションシップ・サービスのために実行する必要がある構成タスクについて説明します。

## リレーションシップ・サービスの構成

製品をインストールした後で、リレーションシップ・サービス用の構成プロパティを設定する必要があります。

**必要なセキュリティ・ロール:** このタスクを実行するには、コンフィギュレーターまたは管理者としてログインする必要があります。どの WebSphere セキュリティ・ロールでもこの構成を表示できます。

リレーションシップ・サービス用のデータ・ソース・プロパティおよびキャッシュ・サイズのヒント (リレーションシップ・インスタンスのカウント) プロパティを設定するには、このタスクを実行します。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「**統合アプリケーション**」>「**Relationship Manager**」をクリックします。
3. 「**リレーションシップ・サービス構成**」をクリックします。

タブ付きの構成ページが表示され、現在インストールされているリレーションシップ・サービスの名前とバージョン (読み取り専用) が示されます。

4. 「**キャッシュ・サイズのヒント (リレーションシップ・インスタンスのカウント)**」フィールドで、リレーションシップ・サービスがリレーションシップ照会用に確保する最大キャッシュを指定します。この設定により、照会結果セットのサイズが決定します。デフォルトでは、5000 個のリレーションシップ・インスタンスが一度に読み取られます。このフィールドによってサーバー・サイズ・メモリー使用量が制御されるため、管理者は、任意の照会が消費できるメモリー・リソースの量を一定の範囲内で制御できるようになります。
5. 「**データ・ソース**」フィールドで、セル・レベルで定義されたデータ・ソースの JNDI 名を入力して、リレーションシップ・サービスのデフォルトのデータ・ソースを指定します。これは、リレーションシップ・サービス用のテーブルが保管される場所です。デフォルトでは、各リレーションシップ関連スキーマはこのデータ・ソース内に作成されます。
6. 以下のオプションがあります。
  - 変更内容を保管し、前のページに戻る場合は、「**OK**」をクリックします。
  - 変更内容をクリアし、現在構成されている値または最後に保管された値を復元する場合は、「**リセット**」をクリックします。
  - そのページで保管されていない変更内容をすべて破棄し、前のページに戻る場合は、「**キャンセル**」をクリックします。



## リレーションシップ・サービスで管理されるリレーションシップの表示

このリレーションシップ・サービスで管理される既存のリレーションシップのリストを表示するには、このタスクを実行します。

**必要なセキュリティ・ロール:** どの WebSphere セキュリティ・ロールでもこの構成を表示できます。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」>「**Relationship Manager**」をクリックします。
3. 「リレーションシップ・サービス構成」>「リレーションシップ」をクリックします。

「リレーションシップ・コレクション」ページが表示されます。各行は、関連するリレーションシップのバージョンとデータ・ソースを示します。

**ヒント:** 一度に表示される行数をカスタマイズするには、「設定」をクリックします。「**最大行数**」フィールドの値を変更し、「**適用**」をクリックします。デフォルトは 25 です。このリレーションシップ・サービスによって管理されるリレーションシップの合計カウントがページの下部に表示されます。

リレーションシップの構成プロパティを表示するには、リレーションシップ・コレクション・テーブルのリレーションシップ名をクリックします。

## リレーションシップ・プロパティの表示

リレーションシップ・サービスがリレーションシップ・サービス・レベル (リレーションシップ・サービスに対して適用される時) と個々のリレーションシップ・レベル (個々のリレーションシップに適用される時) の両方のレベルで管理する構成プロパティを表示するには、このタスクを実行します。

**必要なセキュリティ・ロール:** どの WebSphere セキュリティ・ロールでもこの構成を表示できます。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」>「**Relationship Manager**」をクリックします。
3. 「リレーションシップ・サービス構成」>「リレーションシップ」をクリックします。
4. リレーションシップ・コレクション・テーブルで、プロパティを表示したいリレーションシップの名前をクリックします。

タブ付きの構成ページが表示され、そのリレーションシップの現在使用中の名前、バージョン、およびデータ・ソース (読み取り専用) が示されます。

**注:** バージョンは、マイグレーションの目的で使用されます。新しいシステム内で古いリレーションシップ・データを共存させる必要がある場合は、古いインフラストラクチャーのバージョンが古いバージョンに設定されます。それ以外の場合は、現行バージョンに設定されます。

5. 「リレーションシップ・コレクション」ページに戻るには、「戻る」をクリックします。

---

## クラスター環境の作成

クラスター環境を作成すると、WebSphere Process Server アプリケーションのフェイルオーバーとスケーリングの特性が強化されます。クラスター環境を確立するために、ユーザーおよびユーザーのチームは一定の手順を実行します。

クラスター環境を作成するには、以下のようないくつかの作業を事前に実行する必要があります。

- クラスタリングを正常にインプリメントするための十分なリソースがあることを確認します。
- クラスター環境にデプロイするサービス・アプリケーションを分析します。実行するオプションの手順の一部は、サービス・アプリケーションのニーズに応じて異なります。
- アプリケーション・ロジックがクラスター環境を許容することを確認します。例えば、以下について確認します。
  - 孤立した要求や順序通りでなく処理された要求に備える場合、アプリケーションが分割されたキューを許容すること。アプリケーション障害のために、これらのいずれかまたは両方の状況が発生することがあります。
  - システム全体で使われる値をローカルに保持していないこと。
- WebSphere Application Server for z/OS インフォメーション・センターのネットワーク・デプロイメントとクラスタリングの説明を確認しておきます。
- 手順を実行する前に、ここに記載する指示を確認しておきます。手順の全体的な概念を把握しておくこと、効率よく作業を進めることができます。
- WebSphere Application Server for z/OS がインストール済みであることを確認します。

WebSphere Application Server for z/OS インフォメーション・センターの製品および追加のソフトウェアのインストールを参照してください。

- WebSphere Process Server for z/OS クラスター環境をサポートするように、WebSphere Application Server for z/OS を構成します。

そのためには、以下のタスクを実行する必要があります。

- WebSphere Application Server for z/OS インフォメーション・センターで説明されているように Network Deployment セルを作成します。
- WebSphere Application Server for z/OS インフォメーション・センターで説明されているように、空の管理対象ノードを作成します。

**注:** デプロイメント・マネージャーには統合しないでください。すなわち、BBOWMNAN というジョブを、空の管理対象ノードを作成するプロセスの一部として実行しないでください。

- WebSphere Process Server for z/OS インストール・スクリプトを実行して、製品の定義を作成します。

この作業の説明については、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『*Running the installation script*』を参照してください。

- データベースおよびストレージ・グループを作成します。

WebSphere Process Server for z/OS には、データベースおよびストレージ・グループを作成するときに使用できるサンプル・ファイルが含まれています。

この作業の説明については、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『*Creating databases and storage groups*』を参照してください。

- デプロイメント・マネージャーの応答ファイルを編集します。

デプロイメント・マネージャーの応答ファイルには、使用環境に合わせて変更するデフォルトのプロパティ設定が含まれています。また、データベースおよびストレージ・グループの作成時に使用した値は、応答ファイル内の対応する値と一致する必要があります。詳しくは、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『*Working with response files*』を参照してください。

- デプロイメント・マネージャーの応答ファイルを使用して WebSphere Process Server for z/OS 構成スクリプトを実行します。

その結果、WebSphere Process Server for z/OS Network Deployment の構成データによってデフォルト・プロファイルが拡張されます。

また、WebSphere Process Server for z/OS スクリプトを実行すると、WebSphere Process Server for z/OS データベースの DDL のみが作成されます。WebSphere ESB 用のデータベースは作成されません。

この作業の説明については、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『*Creating a stand-alone configuration*』を参照してください。

- 管理対象ノードに対して WebSphere Process Server for z/OS インストール・スクリプトを実行します。

この作業の説明については、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『*Running the installation script*』を参照してください。

- 管理対象ノードの応答ファイルを使用して WebSphere Process Server for z/OS 構成スクリプトを実行します。

その結果、WebSphere Process Server for z/OS 管理対象ノードの構成データによってデフォルト・プロファイルが拡張されます。

この作業の説明については、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『*Creating an empty managed node configuration*』を参照してください。

- 管理対象ノードをデプロイメント・マネージャーに統合します。

BBOWMNAN というジョブを実行します。これは、空の管理対象ノードを最初に作成したときには実行しなかったジョブです。

この作業の説明については、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『*Federating an empty managed node to a deployment manager*』を参照してください。

単一サーバーで提供するより、さらに多くの容量と高度な可用性をアプリケーションが必要とする場合に、クラスター環境を作成します。クラスター環境には、この他に以下の利点があります。

#### ワークロード・バランシング

複数のサーバーでアプリケーション・イメージを実行することにより、クラスターが、クラスター内のサーバー間でアプリケーションのワークロードのバランスを取ります。

#### アプリケーションの処理能力

アプリケーションをサポートするクラスター・メンバーとして、サーバーのハードウェアを追加構成することによって、そのアプリケーションの処理能力を向上させることができます。

#### アプリケーションの可用性

1 つのサーバーで障害が起こった場合、アプリケーションがクラスター内の他のサーバー上で処理を続行するため、アプリケーション・ユーザーに影響を与えることなく、リカバリー作業を進めることができます。

#### 保守容易性

アプリケーションの処理を停止することなく、計画した保守を実行するためにサーバーを停止できます。

**柔軟性** 管理コンソールを使用して、必要に応じて容量を追加したり除去したりすることができます。

**注:** このページの末尾に、関連タスクのリストを記載しています。特定のステップに関連するタスクのタイトルを、そのステップの括弧付きのコメントとして示しています。そのタスクについて精通していない場合は、関連トピックで詳細情報を参照してください。

1. クラスター環境を設計します。
  - a. セルのトポロジーを設計します。セルに必要な物理的リソースと論理的リソースを決定します。
  - b. セルのさまざまなコンポーネント用にデータベースとスキーマのどちらを使用するかを決定します。
  - c. イベントをモニターする必要があるかどうか、およびモニター対象のイベントをどのサーバーでホストするかを決定します。
2. セルに必要なその他のデータベースを作成します。例えば、以下のようなデータベースを作成します。Business Process Execution Language と Common Event Infrastructure のデータベースおよびスキーマ
  - a. 環境変数を作成します。 以下を作成します。
    - DB2UNIVERSAL\_JDBC\_DRIVER\_PATH = /db2810/jcc/classes
    - DB2UNIVERSAL\_JDBC\_DRIVER\_NATIVEPATH = /db2810/jcc/lib
    - JDBC JCC プロパティ・ファイルを、DMGR/サーバーの付属カスタム・プロパティおよびサーバント・カスタム・プロパティに追加します。

「アプリケーション・サーバー」 > 「サーバー」 > 「プロセス定義」 > 「サーバント (Servant)」 > 「Java 仮想マシン (Java Virtual Machine)」 > 「カスタム・プロパティ」 > 「新規」

**db2.jcc.propertiesFile** (例えば /shared/db2810/jcc/properties/DB2JccConfiguration.properties)

**注:** このプロパティは、クラスター内で構成されたサーバーごとに必要になります。

3. **オプション:** イベントをモニターする必要がある場合は、モニター対象のイベントを処理するクラスターを作成します。
  - a. デフォルトの WebSphere Process Server テンプレートを使用して、クラスターとクラスター・メンバーを作成します。
  - b. ターゲットとして構成しているクラスターと、ステップ 2 (32 ページ) で作成した Common Event Infrastructure (CEI) データベースを使用して、CEI アプリケーションをインストールし、メッセージ駆動型 Bean (MDB) アプリケーションをインストールします。(Common Event Infrastructure の構成)
4. **オプション:** 管理アプリケーションにビジネス・ルールやセレクターが含まれていて、アプリケーションのデプロイ後に、アプリケーションでそのルールやセレクターに変更を加える必要がある場合は、管理アプリケーションを処理するクラスターを作成します。

**注:** ビジネス・ルールやセレクターの管理を、他の WebSphere Process Server コンポーネントを処理するクラスターにデプロイすることを決定した場合は、このステップをステップ 3 またはステップ 6 (34 ページ) と組み合わせて実行できます。

- a. デフォルトの WebSphere Process Server テンプレートを使用して、クラスターとクラスター・メンバーを作成します。
  - b. 「拡張構成」パネルを使用して、ビジネス・ルールのマネージャー・サーバーをデプロイします。
5. クラスター用のメッセージングを構成します。

**注:** 他の WebSphere Process Server コンポーネントを扱うクラスターによって管理メッセージングが処理されるように決定した場合は、このステップをステップ 3、ステップ 4、またはステップ 6 (34 ページ) と組み合わせて実行できます。

**重要:** このステップは、クラスターに最初のサービス・アプリケーションをインストールする前に完了する必要があります。

- a. デフォルトの WebSphere Process Server テンプレートを使用してクラスターを作成します。
- b. 「拡張構成」パネルの「デフォルト宛先ロケーション」を選択します。
- c. クラスターをイベント・バスのメンバーとして追加します。

ステップ 2 (32 ページ) で作成したデータ・ソースを使用します。ステップ 1b (32 ページ) で、1 つのデータベース内で複数のスキーマを使用することを決定した場合は、メッセージング・エンジン・データ・ソースの構成時に適切なスキーマを選択します。

- d. クラスタを Business Process Choreographer バスのバス・メンバーとして追加します。

ステップ 2 (32 ページ) で作成したデータ・ソースを使用します。ステップ 1b (32 ページ) で、1 つのデータベースで複数のスキーマを使用することを決定した場合は、メッセージング・エンジン・データ・ソースの構成時に適切なスキーマを選択します。

6. サービス・アプリケーション用のクラスタを作成します。
  - a. デフォルトの WebSphere Process Server テンプレートを使用して、クラスタとクラスタ・メンバーを作成します。
  - b. 「拡張構成」パネルを使用して、メッセージング・クラスタを選択します。

このステップをステップ 5 (33 ページ) と組み合わせて実行する場合は、そのステップに記載されたデフォルトのメッセージング・クラスタを選択してください。それ以外の場合は、適切なオプションを選択して、ステップ 5 (33 ページ) で作成したクラスタを選択します。

- c. **オプション:** ステップ 2 (32 ページ) で作成したデータベースまたはスキーマを使用して Business Process Choreographer のサポートを構成します。  
(Business Process Choreographer の構成)

アプリケーションに Business Process Execution Language (BPEL) またはビジネス・ステート・マシン・コンポーネントが含まれている場合は、このステップを実行してください。

- d. **オプション:** 「拡張構成」パネルを使用して、エミッター・プロファイルの Java Naming and Directory Interface (JNDI) 名を選択し、クラスタとステップ 2 (32 ページ) で作成したエミッター・ファクトリー・プロファイルを関連付けます。

サーバーおよびインストール済みのアプリケーションを始動させると、これらが正常に始動します。

これで、アプリケーションをクラスタにインストールできるようになりました。

## Service Component Architecture アプリケーションをサポートするサーバーまたはクラスタの準備

WebSphere Process Server Service Component Architecture (SCA) アプリケーションをサポートするよう、サーバーまたはクラスタを構成できます。サーバーおよびクラスタは、SCA アプリケーション、アプリケーション宛先、またはこれら両方をホストできます。

SCA アプリケーション (サービス・アプリケーションとも呼ばれる) は、SCA.APPLICATION.CellName.Bus、SCA.SYSTEM.CellName.Bus、または BPC.CellName.Bus のバスのうち 1 つ以上を使用する必要があるアプリケーションです。各アプリケーションは、一連の Java Message Service キュー (宛先と呼ばれる) を使用します。これらの宛先は構成済みのメッセージング・エンジンを必要とし、アプリケーションと同じサーバーまたはクラスタか、リモートのサーバーまたはクラスタ上でホストできます。

デフォルトでは、新規クラスターおよびサーバーは、SCA アプリケーションとその宛先をホストするように構成されていません。このサポートを有効にするには、管理コンソールの「Service Component Architecture」ページを使用します。SCA アプリケーションをサポートするようサーバーまたはクラスターを構成する方法には、3つの基本的なシナリオがあります。

- サーバーまたはクラスターが SCA アプリケーションをホストしますが、宛先はリモートのサーバーまたはクラスター上でホストされます。このシナリオでは、宛先をホストするために必要なメッセージング・エンジンを使って、リモート・サービス統合バス・メンバーを構成する必要があります。『宛先をホストしない Service Component Architecture アプリケーションのホスト』を参照してください。
- サーバーまたはクラスターが SCA アプリケーションおよびアプリケーション宛先をホストします。このシナリオでは、必要なメッセージング・エンジンをサーバーまたはクラスター上に構成します。これらのメッセージング・エンジンは、デプロイメント・ターゲットが異なるアプリケーションで必要とされる宛先もホストできます。36ページの『Service Component Architecture アプリケーションおよびアプリケーション宛先のホスト』を参照してください。
- サーバーまたはクラスターが、リモートのサーバーまたはクラスターにデプロイされたアプリケーションの宛先をホストします。サーバーまたはクラスターは SCA アプリケーションをホストしませんが、36ページの『Service Component Architecture アプリケーションおよびアプリケーション宛先のホスト』で説明されている方法でサービス・バス・メンバーを構成する必要があります。

## 宛先をホストしない Service Component Architecture アプリケーションのホスト

宛先がリモートのクラスター/サーバーに配置されている Service Component Architecture (SCA) アプリケーションのデプロイメント・ターゲットとしてサーバーまたはクラスターを構成できます。

宛先をホストするのに必要なメッセージ・エンジンで構成されているリモート・サービス統合バス・メンバーがあることを確認してください。バス・メンバーの構成方法については、36ページの『Service Component Architecture アプリケーションおよびアプリケーション宛先のホスト』を参照してください。

以下のタスクを実行するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

SCA アプリケーションをホストするようサーバーまたはクラスターを構成するには、以下の手順を実行します。

1. 管理コンソール内で、スコープに応じて以下のいずれかをクリックします。
  - サーバーを構成する場合: 「サーバー」 > 「アプリケーション・サーバー」 > `<server_name>` 「Service Component Architecture」
  - クラスターを構成する場合: 「サーバー」 > 「クラスター」 > `<cluster_name>` 「Service Component Architecture」
2. 「リモート宛先ロケーションの使用」をクリックします。
3. 関連付けられているドロップダウン・メニューを使用して、宛先をホストすることになるリモート・サーバーまたはクラスターの名前を指定します。

#### 4. 「OK」をクリックします。

以上で、サーバーまたはクラスターが SCA アプリケーションをホストできるようになりました。さらに、SchedulerCalendar アプリケーションがアプリケーションのホスティングをサポートできるようになりました。

- このクラスターまたはサーバー上で Business Process Execution Language (BPEL) によって定義されたビジネス・プロセスの使用を計画している場合は、ビジネス・プロセス・コンテナー・ウィザードを使用して必要な構成を実行することも必要です。
- ヒューマン・タスクが含まれるアプリケーションの使用を計画している場合、あるいはこのクラスターまたはサーバーで Business Process Choreographer Explorer の使用を計画している場合は、ヒューマン・タスク・コンテナー・ウィザードを使用して必要な構成を実行することも必要です。

## Service Component Architecture アプリケーションおよびアプリケーション宛先のホスト

Service Component Architecture (SCA) アプリケーションおよびアプリケーション宛先をホストするようにサーバーまたはクラスターを構成できます。これらの宛先を使用して、ローカルにインストールされた SCA アプリケーション、またはリモートのクラスターやサーバーにインストールされた SCA アプリケーションをサポートできます。

以下のタスクを実行するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

このタスクでは、システム・バスおよびアプリケーション・バスを構成する必要があります。一方または両方のバスに既存のデータ・ソースを使用するか、新規データ・ソースを構成します。データ・ソースについては、以下の考慮事項に注意してください。

- メッセージング・エンジンで DB2 データベースを使用している場合、システム・バスとアプリケーション・バスの両方のデータ・ソースに同一のデータベース・インスタンスを使用できますが、それぞれのバスに固有のスキーマ名を指定する必要があります。(スキーマにはデータ・ソースの表が含まれます。各メッセージング・エンジンはリソースを単一のスキーマに格納し、各データベース・スキーマは 1 つのメッセージング・エンジンのみによって使用されます。)
- その他のサポートされるデータベースを使用している場合は、システム・バスとアプリケーション・バスのデータ・ソースには異なるデータベース・インスタンスを使用する必要があります。

SCA アプリケーションおよび宛先をホストするようにサーバーまたはクラスターを構成するには、以下の手順を実行します。

1. 管理コンソール内で、スコープに応じて以下のいずれかをクリックします。
  - サーバーを構成する場合: 「サーバー」 > 「アプリケーション・サーバー」 > `server_name` > 「Service Component Architecture」
  - クラスターを構成する場合: 「サーバー」 > 「クラスター」 > `cluster_name` > 「Service Component Architecture」
2. 「宛先ロケーションの構成」をクリックします。



3. 「JDBC プロバイダー」ドロップダウン・メニューを使用して、JDBC プロバイダーに適切なテンプレートを指定します。選択した JDBC プロバイダーによって、システム・バスおよびアプリケーション・バスに使用できるデータベースのタイプが決まります。
4. 「ユーザー名」フィールドに、システム・バスおよびアプリケーション・バスのデータ・ソースへのアクセスに使用する ID を入力します。
5. ユーザー ID に関連付けられたパスワードを「パスワード」フィールドに入力します。
6. 以下のようにして、SCA システム・バスの構成を指定します。

使用するデータ・ソースのタイプ	実行するステップ
既存のデータ・ソース <b>注:</b> システム・バスとアプリケーション・バスには異なるデータ・ソースを使用する必要があります。	<ol style="list-style-type: none"> <li>1. 「既存のデータ・ソースの使用」をクリックします。</li> <li>2. ドロップダウン・メニューを使用して、使用するデータ・ソースを指定します。</li> </ol>
新規データ・ソース	<ol style="list-style-type: none"> <li>1. 使用しているデータベースがスキーマ名をサポートする場合は、「スキーマ名」フィールドを使用して、システム・バス・データ・ソース表を含むスキーマを指定します。</li> <li>2. 「データベース・プロパティ」フィールドに、SCA システム・バス・データ・ソースに必要なカスタム・プロパティを入力します。</li> </ol>

7. 以下のようにして、SCA アプリケーション・バスの構成を指定します。

使用するデータ・ソースのタイプ	実行するステップ
既存のデータ・ソース <b>注:</b> システム・バスとアプリケーション・バスには異なるデータ・ソースを使用する必要があります。	<ol style="list-style-type: none"> <li>1. 「既存のデータ・ソースの使用」をクリックします。</li> <li>2. ドロップダウン・メニューを使用して、使用するデータ・ソースを指定します。</li> </ol>
新規データ・ソース	<ol style="list-style-type: none"> <li>1. 使用しているデータベースがスキーマ名をサポートする場合は、「スキーマ名」フィールドを使用して、システム・バス・データ・ソース表を含むスキーマを指定します。</li> <li>2. 「データベース・プロパティ」フィールドに、SCA アプリケーション・バス・データ・ソースに必要なカスタム・プロパティを入力します。</li> </ol>

8. 「OK」をクリックします。

サーバーまたはクラスターは SCA アプリケーションおよびアプリケーション宛先をホストできます。さらに、SchedulerCalendar アプリケーションがアプリケーションのホスティングをサポートできるようになりました。

- このクラスターまたはサーバー上で Business Process Execution Language (BPEL) によって定義されたビジネス・プロセスの使用を計画している場合は、ビジネス・プロセス・コンテナー・ウィザードを使用して必要な構成を実行することも必要です。
- ヒューマン・タスクが含まれるアプリケーションの使用を計画している場合、あるいはこのクラスターまたはサーバーで Business Process Choreographer Explorer の使用を計画している場合は、ヒューマン・タスク・コンテナー・ウィザードを使用して必要な構成を実行することも必要です。

## アプリケーションおよびアプリケーション・サービスの管理

このセクションでは、管理インターフェースを使用して、ビジネス・プロセスとビジネス・タスク、ビジネス・ルール、スケジュールなど WebSphere Process Server のアプリケーションおよびアプリケーション・サービスを管理する方法を説明します。

### エンタープライズ・アプリケーションの管理

コンソールの「エンタープライズ・アプリケーション」ページ（「アプリケーション」>「エンタープライズ・アプリケーション」とクリックして表示）を使用して、サーバーにインストールされているエンタープライズ・アプリケーションの表示および管理を行います。

アプリケーションの構成で指定した値を表示するには、リストからアプリケーション名をクリックします。「アプリケーション詳細」ページが開き、アプリケーションの構成プロパティ、および該当する場合はローカル・トポロジーが表示されます。このページから既存値を変更して、アプリケーションを構成する追加コンソール・ページにリンクできます。

エンタープライズ・アプリケーションを管理するには、その名前の横にあるチェック・ボックスをクリックしてから、以下のいずれかのボタンを使用します。

表7. エンタープライズ・アプリケーションを管理するためのボタン

ボタン	結果のアクション
開始	<p>アプリケーションの実行を試みます。アプリケーションが正常に開始すると、アプリケーションの状態は以下のいずれかに変わります。</p> <ul style="list-style-type: none"> <li>• 開始済み - アプリケーションは、すべてのデプロイメント・ターゲットで開始しました。</li> <li>• 部分開始 - アプリケーションは、1 つ以上のデプロイメント・ターゲットでまだ開始中です。</li> </ul>
停止	<p>アプリケーションの処理の停止を試みます。アプリケーションが正常に停止すると、アプリケーションの状態は以下のいずれかに変わります。</p> <ul style="list-style-type: none"> <li>• 停止済み - アプリケーションは、すべてのデプロイメント・ターゲットで停止しました。</li> <li>• 部分停止 - アプリケーションは、1 つ以上のデプロイメント・ターゲットでまだ停止中です。</li> </ul>

表7. エンタープライズ・アプリケーションを管理するためのボタン (続き)

ボタン	結果のアクション
インストール	ウィザードが開き、エンタープライズ・アプリケーションまたはモジュール (.jar, .war, または .ear ファイルなど) をサーバーにデプロイできるようにします。
アンインストール	WebSphere Application Server の構成リポジトリからアプリケーションを削除し、さらに、構成の保管後にアプリケーション・モジュールのインストール先のすべてのノードのファイル・システムから、そのアプリケーション・バイナリーを削除します。
更新	ウィザードが開き、サーバー上にデプロイされているアプリケーション・ファイルを更新できるようにします。アプリケーション全体を更新することも、単一モジュール、単一ファイル、あるいはアプリケーションの一部のみを更新することもできます。新規ファイルまたはモジュールの名前が、サーバー上に既に存在するファイルまたはモジュールの名前と同じである場合は、既存のファイルまたはモジュールが、その新規ファイルまたはモジュールに置き換えられます。同じ名前のファイルまたはモジュールがない場合は、デプロイされているアプリケーションに追加されます。
ファイルの除去	デプロイされたアプリケーションまたはモジュールから、ファイルを削除します。このボタンは、構成リポジトリ、およびファイルがインストールされているすべてのノードのファイル・システムから、ファイルを削除します。
エクスポート	「アプリケーション EAR ファイルのエクスポート」ページが開きます。このページでエンタープライズ・アプリケーションを EAR ファイルにエクスポートできます。デプロイされているアプリケーションをバックアップしたり、そのバインディング情報を保存したりする場合にも、「エクスポート」アクションを使用します。
DDL のエクスポート	「アプリケーション DDL ファイルのエクスポート」ページが開きます。このページでエンタープライズ・アプリケーションの EJB モジュール内の DDL ファイルをエクスポートできます。

アプリケーションの管理について詳しくは、WebSphere Application Server for z/OS の資料を参照してください。

### 不要な Service Component Architecture 宛先の除去

一般に、モジュールがアンインストールされると、サーバーはそのモジュールが使用しているすべての非アクティブな Service Component Architecture (SCA) 宛先を削除します。Java 仮想マシン (JVM) を始動し、システム変数 **SCA\_recycleDestinations** を true に設定して SCA 宛先の再使用を選択する場合は、SCA 宛先を手動で削除する必要があります。

このタスクは、管理コンソールを使用してあるいは startServer.bat または startServer.sh ファイルで、JVM カスタム変数 SCA\_recycleDestinations の値を true に設定していることを前提とします。

SCA 宛先のデフォルト処理を変更した場合は、SCA 宛先を含むモジュールをアンインストールした後で、すべての非アクティブな SCA 宛先を手動で除去する必要があります。

注:

#### アクティブな宛先

現在デプロイされているモジュールに属している宛先。

#### 非アクティブな宛先

現在デプロイされているモジュールに属していない宛先。

1. コマンド行から、`deleteSCADestinations.jacl` コマンドを入力します。特定のモジュールに関連付けられた宛先をその宛先がアクティブであっても、**-force** オプションを使用して削除します。
2. SCA 宛先を表示して、正しい宛先を削除したことを確認します。

宛先がサーバーから除去されます。

## Service Component Architecture モジュールのデプロイ後の WebSphere MQ 宛先の変更

ビジネス環境の発展に応じて、Service Component Architecture (SCA) モジュールがどの WebSphere MQ 宛先と通信するかを変更する必要があることがあります。このような変更を実行するには、このタスクを使用します。

変更する WebSphere MQ 宛先を、どの Service Component Architecture (SCA) モジュールが使用しているかを知っておく必要があります。デフォルトでは、これらの宛先は、SCA モジュールをサーバーまたはクラスターにインストールしたときに作成されます。宛先の形式は、以下のとおりです。

- 片方向のエクスポートの場合:
  - `modulename.exportname_MQEXPORT_CF`
  - `modulename.exportname_MQ_RECEIVE_D`
- これに加えて、両方向のエクスポートの場合:
  - `modulename.exportname_MQ_SEND_D`
- 片方向のインポートの場合:
  - `modulename.importname_MQIMPORT_CF`
  - `modulename.importname_MQ_SEND_D`
- これに加えて、両方向のインポートの場合:
  - `modulename.importname_MQ_RECEIVE_D`

このタスクは、管理コンソールを使用して構成を変更することを前提とします。

**制約事項:** 宛先を変更するときには、以下の事項に注意してください。

- ターゲット・クライアントを **MQ** に設定した状態でキュー宛先を構成する必要があります。
- TCP/IP クライアント接続を使用して、WebSphere MQ に接続する必要があります。詳しくは、「WebSphere MQ Intercommunication」を参照してください。

- チャネル圧縮は使用できません。
- 正しいデータ処理のために必要なチャネル出口を設定する必要があります。このためには、「WebSphere MQ メッセージング・プロバイダー (WebSphere MQ messaging provider)」 > 「WebSphere MQ 接続ファクトリー・オブジェクト (WebSphere MQ connection factory objects)」の「カスタム・プロパティ」にチャネル出口を以下のように設定します。
  - カスタム・プロパティ **SENDEXIT** が、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl` を持つように設定する必要があります。
  - カスタム・プロパティ **RECEXIT** が、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl` を持つように設定する必要があります。
  - オプションのカスタム・プロパティ **SENDEXITINIT** を、ユーザー指定のテキスト・ストリングに設定する必要があります。このプロパティを設定すると、トレースで出口を見分けやすくなります。
  - オプションのカスタム・プロパティ **RECEXITINIT** を、ユーザー指定のテキスト・ストリングに設定する必要があります。このプロパティを設定すると、トレースで出口を見分けやすくなります。
- 出口では **JMS MQ** バインディングは禁止されるため、WebSphere MQ バインディングと Java Message Service (JMS) MQ バインディングに同じ接続ファクトリーを使用することはできません。

SCA モジュールをデプロイした後で、新規のビジネス要件を満たすために、WebSphere MQ 宛先の変更が必要になることがあります。

1. 変更する宛先を使用しているすべてのアプリケーションを停止します。『エンタープライズ・アプリケーションの管理』で説明する手順を実行します。
2. 宛先が進行中のメッセージの処理を完了することを許可します。
3. WebSphere MQ 宛先を表示します。管理コンソール内でこのページにナビゲートするには、「リソース」 > 「JMS プロバイダー」 > 「WebSphere MQ」 > 「WebSphere MQ キュー宛先 (WebSphere MQ queue destinations)」を使用します。
4. 変更する宛先のスコープを選択します。

宛先を含む SCA モジュールが単一サーバー上にインストールされている場合は、「サーバー・スコープ (Server scope)」を選択します。

宛先を含む SCA モジュールがクラスター上にインストールされている場合は、「クラスター・スコープ (Cluster scope)」を選択します。

5. 変更する宛先をリストから選択します。
6. 次のページの各種フィールドを新規の値に変更します。
7. 変更する宛先ごとに、ステップ 5 と 6 を繰り返します。
8. 構成変更を保管します。

ステップ 1 (41 ページ) で停止したアプリケーションを再始動します。

## Service Component Architecture モジュールと WebSphere MQ

Service Component Architecture (SCA) モジュールは、他の SCA モジュールに対するのと同じように WebSphere MQ アプリケーションと通信できます。WebSphere MQ アプリケーションに要求を送信するモジュールは、そのアプリケーションに関連付けられた正しい応答および要求キューで構成されたインポートを使用します。同様に SCA モジュールは、適切なアプリケーション要求および応答キューで構成されたエクスポートを使用して、WebSphere MQ アプリケーションにサービスを提供できます。SCA モジュールと WebSphere MQ キューの間の接続は、WebSphere Integration Developer (WID) でモジュールをビルドするときに定義します。

WebSphere MQ キュー・マネージャーの観点からは、SCA モジュールは通常の MQ クライアントであるかのように見えます。SCA モジュールの側からは、WebSphere MQ キューは他のサービスと同じように見えます。SCA モジュールと WebSphere MQ キューとの間のシールドを更に強化するには、SCA モジュールと WebSphere MQ キューの間にメディエーション・モジュールを使用します。これにより、メディエーションが元の SCA 要求をターゲット・キューに適した正しい形式に変換し、応答が使用可能になったときにそれを処理できるようになります。

**制約事項:** インポートおよびエクスポート用に WebSphere MQ を構成する場合は、以下の点に注意してください。

- ターゲット・クライアントを **MQ** に設定した状態でキュー宛先を構成する必要があります。
- TCP/IP クライアント接続を使用して、WebSphere MQ に接続する必要があります。詳しくは、「WebSphere MQ Intercommunication」を参照してください。
- チャネル圧縮は使用できません。
- 正しいデータ処理のために必要なチャネル出口を設定する必要があります。このためには、「**WebSphere MQ メッセージング・プロバイダー (WebSphere MQ messaging provider)**」>「**WebSphere MQ 接続ファクトリー・オブジェクト (WebSphere MQ connection factory objects)**」の「**カスタム・プロパティ**」にチャネル出口を以下のように設定します。
  - カスタム・プロパティ **SENDEXIT** が、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl` を持つように設定する必要があります。
  - カスタム・プロパティ **RECEXIT** が、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl` を持つように設定する必要があります。
  - オプションのカスタム・プロパティ **SENDEXITINIT** を、ユーザー指定のテキスト・ストリングに設定する必要があります。このプロパティを設定すると、トレースで出口を見分けやすくなります。

- オプションのカスタム・プロパティ **RECEXITINIT** を、ユーザー指定のテキスト・ストリングに設定する必要があります。このプロパティを設定すると、トレースで出口を見分けやすくなります。
- 出口では **JMS MQ** バインディングは禁止されるため、**WebSphere MQ** バインディングと **Java Message Service (JMS) MQ** バインディングに同じ接続ファクトリーを使用することはできません。

#### 関連概念

88 ページの『バインディング』

バインディング情報は、サービスとアプリケーションとの接続方法および 対話方法を決定します。バインディングは、本質的にはインポートおよびエクスポートに 割り当てられたプロトコルおよびトランスポートです。

## Application Scheduler

Application Scheduler によって、管理者は、WebSphere Process Server にインストールされているアプリケーションの開始および停止をスケジュールに入れることができます。管理コンソールの「Application Scheduler」パネルを使用して、インストール済みアプリケーションのスケジューリングを制御します。管理コンソールの「Application Scheduler」パネルを使用して、これらのマイグレーション済みスケジューラー・エントリーを管理することもできます。

Network Deployment 環境では、Application Scheduler は、すべての管理対象サーバーおよび作成されたクラスター・メンバーに自動的にインストールされ、追加のアクションは不要です。新しい管理対象サーバーおよびクラスター・メンバーの作成手順については、WebSphere Application Server for z/OS インフォメーション・センターの『Network Deployment セルの計画』を参照してください。

スタンドアロン・サーバー環境では、Application Scheduler はオプションです。スタンドアロン・サーバーのプロファイルの作成時に、そのサーバーに Application Scheduler を構成およびインストールするチェック・ボックスを選択します。

### スタンドアロン・サーバー用 Application Scheduler の構成

Application Scheduler を使用するには、それがインストールされていることを確認する必要があります。Application Scheduler の構成については、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『Working with response files』を参照してください。

初めにプロファイルを拡張する必要があります。

これはオプション・コンポーネントであり、WebSphere InterChange Server スケジューラー・エントリーを WebSphere Process Server に移行するには、Application Scheduler を構成する必要があります。スタンドアロン・サーバー用にアプリケーション・サーバーをインストールするには、以下の手順に従います。

応答ファイル内の Application Scheduler の値を参照し、Application Scheduler を使用するために必要なプロパティ値を設定します。Application Scheduler に関するプロパティについては、「WebSphere Process Server for z/OS Installation and Configuration」の PDF の『Sample response files』を参照してください。

これでインストールは完了しました。

Application Scheduler が使用可能になります。

## Application Scheduler へのアクセス

Application Scheduler には、Application Scheduler Mbean インターフェースを使用してプログラマチックにアクセスするか、または管理コンソールの「Application Scheduler」パネルを使用してアクセスします。

Application Scheduler へのアクセスについては、以下を参照してください。

- 『Application Scheduler MBean インターフェースを使用した Application Scheduler へのアクセス』
- 45 ページの『管理コンソールを使用したスケジューラー・エントリーの表示』

## Application Scheduler MBean インターフェースを使用した Application Scheduler へのアクセス

Application Scheduler MBean を起動するには、コマンド行を使用してください。

Application Scheduler MBean を起動するには、以下を実行します。

1. クラス `com.ibm.wbiserver.migration.ics.Parameters` に、プロパティー `SOAP_HOSTNAME` および `SOAP_PORT` を設定します。このクラスは、`WAS_HOME\lib` ディレクトリーの `migration-wbi-ics.jar` ファイル内にあります。`SOAP_HOSTNAME` は、Application Scheduler が実行されているホストの名前です。`SOAP_PORT` は、Application Scheduler が実行されているポートです。

```
Parameters.instance.setProperty(Parameters.SOAP_HOSTNAME, "localhost");
Parameters.instance.setProperty(Parameters.SOAP_PORT, "8880");
```

**注:** セキュリティーがオンになっている場合、ロケーション

`WAS_HOME\profiles\profiles\properties\soap.client.props` にある SOAP プロパティー・ファイル内でユーザー ID およびパスワードを指定する必要があります。

このプロパティー・ファイル名を、以下に示す `Parameters` インスタンスに設定する必要があります。

```
Parameters.instance.setProperty(Parameters.SOAP_PROPERTIES,
"WAS_HOME\profiles\profiles\properties\soap.client.props");
```

2. `AppScheduler Mbean` への呼び出しを実装するクラス `com.ibm.wbiserver.migration.ics.utils.MBeanUtil` のインスタンスを作成します。

`MBeanUtil` のインスタンスを生成するには、`name`、`type`、`server name`、および `node name` に基づいて正しい `Mbean` を起動するコンストラクターに、この照会ストリングを渡す必要があります。

```
protected static final String WEBSHERE_MB_QUERY_CONSTANT = "WebSphere:*";
protected static final String NAME_QUERY_CONSTANT = ",name=";
protected static final String WBI_SCHED_MB_NAME = "WBISchedulerMB1";
protected static final String TYPE_QUERY_CONSTANT = ",type=";
protected static final String WBI_SCHED_MB_TYPE = "WBIScheduler";
protected static final String SERVER_QUERY_CONSTANT = ",process=";
protected static final String NODE_QUERY_CONSTANT = ",node=";

serverName = "server1";
```



```

nodeName = "myNode";

String queryString = new StringBuffer(WEBSphere_MB_QUERY_CONSTANT)
    .append(NAME_QUERY_CONSTANT).append(WBI_SCHED_MB_NAME).append(
    TYPE_QUERY_CONSTANT).append(WBI_SCHED_MB_TYPE).append(
    SERVER_QUERY_CONSTANT).append(serverName).append(
    NODE_QUERY_CONSTANT).append(nodeName).toString();

MBeanUtil mbs = new MBeanUtil(queryString.toString());

```

3. MBeanUtil インスタンスの invoke() メソッドを使用して Mbean メソッドを呼び出し、メソッドの名前を渡します。

以下に、Scheduler Mbean の createSchedulerEntry メソッドを起動する例を示します。最初のステップでは、SchedulerEntry を作成し、name、type、version、transition、entry status、recurrence type、recurrence week、recurrence period、initial date、repeat interval、および component id のような各種のパラメーターを設定します。

```

try
{
//First we set up the Schedule entry

ScheduleEntry entry1 = new ScheduleEntry();
entry1.setCName("BPEWebClient_localhost_server1");
entry1.setCType("Application");
entry1.setCVersion("ver1");
entry1.setCTransition("startApplication");
entry1.setSchedulerNumberOfRepeats(3); // Fire Three times
entry1.setScheduleEntryStatus(TaskStatus.SCHEDULED);
entry1.setRType(Recurrence.MINUTES);
entry1.setRWeekNumber(-1);
entry1.setRPeriod(2);
entry1.setInitialDate(new Date(System.currentTimeMillis()+SIXTY_SECOND_OFFSET));
entry1.setRepeatInterval(entry1.getInitialDate(), entry1.getRType(),
    entry1.getRWeekNumber(),
    entry1.getRPeriod());
entry1.setComponentID(entry1.getCName(), entry1.getCType(), entry1.getCVersion(),
    entry1.getCTransition());

```

その後、Mbean の createSchedulerEntry メソッドを起動します。ScheduleEntry クラスの名前と共に、スケジューラー・エントリー entry1 をパラメーターとして渡します。

そして、以下のように MBean の createScheduleEntry メソッドを起動します。

```

mbs.invoke(schedulerExtMBName, "createScheduleEntry", new Object[]{entry1},
    new String[]{"com.ibm.wbiserver.scheduler.common.ScheduleEntry"});

```

最後に、readAllScheduleEntries メソッドを呼び出して、追加したばかりのエントリーを含め、すべてのスケジュール・エントリーを読み込みます。

```

result = mbs.invoke("readAllScheduleEntries", null, null);
}
catch (MigrationException e)
{ e.printStackTrace();
}

```

## 管理コンソールを使用したスケジューラー・エントリーの表示

管理コンソールの「Application Scheduler」パネルを使用して、スケジューラー・イベントを作成、変更、または削除します。

このタスクを実行するには、ユーザーはサーバーの管理コンソールを使用している必要があります。

このパネルを表示し、既存のスケジューラー・イベントを表示するには、以下の手順に従います。

1. 「サーバー」>「アプリケーション・サーバー」>「サーバー名」を選択します。
2. 「Business Integration」から「Application Scheduler」を選択します。
3. 表示するエントリーの有効範囲 (セル、ノード、サーバー) を選択します。その有効範囲に対応する既存のスケジュール済みイベントがリストされます。

スケジューラー・イベントの編集、新規スケジューラー・イベントの作成、または既存のイベントの削除を実行できるようになりました。

### スケジュール済みイベントの作成

管理コンソールは、新規のスケジュール済みイベントを作成するためのパネルを提供します。

新規のスケジュール済みイベントを作成するには、サーバー用管理コンソールの「Application Scheduler」コレクション・パネルを使用する必要があります。詳しくは、45 ページの『管理コンソールを使用したスケジューラー・エントリーの表示』を参照してください。

特定のニーズに適合するようにイベントを作成しなければならないことがあります。新規のスケジュール済みイベントを作成するには、以下の手順に従います。

**注:** パネル上の「\*」が付いたフィールドは、必須フィールドです。

1. 「新規」をクリックします。「追加」パネルが開きます。
2. スケジュール済みイベントを構成します。
  - a. 「グループ・アプリケーション」を選択します。
  - b. 「状況」を選択します。
  - c. 「初期日付」を *month, dd, yyyy* という書式で入力します (*month* は省略形)。例えば、2005 年 4 月 15 日の場合、**Apr 15, 2005** と入力します。
  - d. 「初期時間」を、12 時間表記の *hh:mm* という書式で入力します。

**注:** **am** と **pm** のいずれであるか、および**時間帯**も示す必要があります。

**注:** このフィールドから移動した後、**次の起動時刻**が自動的に計算されます。

- e. 「アクション」を選択します。

**オプション: Recurrence** パラメーターに記入することもできます。

- 開始期間
- スケジュール・エントリーを指定した時刻に繰り返す必要があるかどうか
  - 1 分ごと、1 時間ごと、1 日ごと、1 カ月ごと、または 1 年ごとに 1 回以上
  - 1 カ月ごとまたは複数月ごとの特定の週 (第 1、第 2、第 3、第 4、または最後) の特定の曜日 (日曜から土曜)

- 1 カ月ごとまたは複数月ごとの最終日

3. 「適用」または「OK」をクリックして、イベントを設定します。

注: 別のイベントを作成するには、「リセット」をクリックしてパネルをクリアします。

Application Scheduler によって、新規のスケジュール済みイベントが作成され、「Application Scheduler」パネルに表示されます。

#### イベント状況およびアクションの説明:

各イベントには、状況とアクションが含まれていなければなりません。

#### 状況

状況フィールドは、モニター用にイベントの状況を示します。次の表にそれぞれの状況を示します。

状況	説明
スケジュール済み	タスクは事前に決定された日時および間隔で実行されます。2 回目以降の実行時刻は計算によって求められます。
中断	タスクは中断され、状況がスケジュール済みに変更されるまで実行されません。
完了	タスクは完了しています。
キャンセル済み	タスクがキャンセルされました。タスクは実行されず、再開できませんが、ページすることは可能です。
無効	通常、タスクの状況が無効になる理由は、タスクがページされているか、タスクを照会するために使用された情報が無効であるためです。
実行中	タスクは実行中です。 注: この状況はほとんど見られません。イベントが起動される非常に短い時間のみイベントをモニターするためです。

#### アクション

各イベントにはアクションが関連付けられていなければなりません。アクションは、イベントへの対処法を指示します。イベントに対して実行可能なアクションは以下の 2 つのみです。

- 「アプリケーションの始動」 - システムのデプロイメント・マネージャーの下にあるすべてのアプリケーションを始動します。
- 「アプリケーションの停止」 - システムのデプロイメント・マネージャーの下にあるすべてのアプリケーションを停止します。

## スケジュール済みイベントの変更

管理コンソールから、移行したスケジュール済みイベントまたは既存のスケジュール済みイベントを変更します。

スケジュール済みイベントを変更するには、サーバー用管理コンソールの「Application Scheduler」コレクション・パネルを使用する必要があります。詳しくは、45 ページの『管理コンソールを使用したスケジューラー・エントリーの表示』を参照してください。

ニーズに適合するようにイベントを変更しなければならないことがあります。イベントを変更するには、以下の手順に従います。

1. 変更する「イベント」をクリックします。「イベント」パネルが開きます。
2. 以下のフィールドのいずれかを変更します。

注: サーバー上のすべてのアプリケーションがリストされているため、既存のイベントの状況を変更する際には注意が必要です。サーバーで実行しているアプリケーションを停止する可能性があります。

- グループ・アプリケーション
- 状況
- 初期日付 (書式は *month, dd, yyyy*、*month* は省略形を使用)
- 初期時刻 (書式は 12 時間表記の *hh:mm* を使用)
- アクション

オプション: **Recurrence** パラメーターに記入することもできます。

3. 「適用」または「OK」をクリックして、イベントの変更を設定します。

注: スケジュール済みイベントを変更する場合、新規の「スケジュール・エントリー ID」がサーバーから割り当てられます。サーバーは、現在スケジュールされているイベントを削除して、新規 ID を持つ新規のイベントをスケジュールに入れます。

新規 ID を持つ変更済みイベントが、「Application Scheduler」コレクション・パネルに表示されます。

## スケジュール済みイベントの削除

Application Scheduler は、スケジュール済みイベントを削除するためのパネルを提供します。

スケジュール済みイベントを削除するには、サーバー用管理コンソールの「Application Scheduler」コレクション・パネルを使用する必要があります。詳しくは、45 ページの『管理コンソールを使用したスケジューラー・エントリーの表示』を参照してください。

イベントが使用されなくなった場合、コレクション・パネルのイベントのリストからそのイベントを削除できます。スケジュール済みイベントを削除するには、以下の手順に従います。

1. 「選択」列から削除するイベントを選択します。
2. 「削除」をクリックします。

3. プロンプトで「OK」をクリックします。

イベントは削除されます。

---

## WebSphere Process Server リソースの管理

管理インターフェースによって、セクター、ターゲット・コンポーネント、アダプター、拡張メッセージング・サービスなどの、WebSphere Process Server に関連するリソースを管理できます。

### アダプターの管理

WebSphere アダプター、バージョン 6.0、および WebSphere Business Integration Adapter (WebSphere Business Integration Framework、バージョン 2.6 を基にしたもの) は、アプリケーション・モジュール指向のエンタープライズ情報システム (EIS) 統合へのアプローチを提供します。

WebSphere Adapters (リソース・アダプター、JCA アダプター、J2C アダプターと呼ばれることもあります) は、J2EE Connector Architecture (JCA 1.5) に準拠しています。JCA は、EIS 接続に関する J2EE 標準です。EIS インポートおよび EIS エクスポートは、モジュール外部のサービスに関する一貫性のあるビューを、SCA アプリケーション・モジュールに提供します。これにより、コンポーネントは一貫性のある SCA プログラミング・モデルを使用して、さまざまな EIS システムと通信できます。

WebSphere アダプターは、インポートされた RAR ファイルから WebSphere Integration Developer でアセンブルされ、Enterprise Application Archive (EAR) ファイルとしてエクスポートされて、WebSphere Process Server にデプロイされます。

Service Component Architecture 内でのインポートおよびエクスポート機能は、サービス・モジュールが WebSphere Process Server 内で持つ外部インターフェースまたはアクセス・ポイントを定義します。インポートおよびエクスポートは、同じアプリケーション内の他のモジュールに対して、または EIS の他のアプリケーションに対して行うことができます。インポートはモジュール外部のサービスを確認し、そのサービスをモジュール内から呼び出し可能にします。エクスポートはモジュール内のコンポーネントがそのサービスを外部クライアントに提供できるようにします。モジュール・レベルのインポートまたはエクスポートは、モジュールが他のモジュールにアクセスすることを許可します。システム・レベルのインポートまたはエクスポートは、ご使用のアプリケーションが EIS 上のアプリケーションに対して、それらがあたかもローカル・コンポーネントであるかのようにアクセスすることを許可します。これにより、ご使用のアプリケーションが WebSphere アダプターおよび WebSphere Business Integration Adapter と連動することができます。

WebSphere Business Integration Adapter は、一連のソフトウェア、アプリケーション・プログラミング・インターフェース (API)、およびツールで構成され、これによって、アプリケーションは、統合ブローカー経由でビジネス・データを交換できます。それぞれのビジネス・アプリケーションがビジネス・インテグレーション・プロセスに関与するためには、そのアプリケーション固有のアダプターが必要です。現在の WebSphere Business Integration Adapter Framework および Development Kit System Manager ツールを使用して、アダプターをインストール、構成、およびテスト

トできます。WebSphere Integration Developer を使用して既存のビジネス・オブジェクトおよびコネクタ構成ファイルをインポートすることにより、成果物を生成し、WebSphere Process Server のためのソリューションをアSEMBルできます。WebSphere Business Integration Adapter 用の操作コマンドは、WebSphere Process Server 管理コンソールの一部です。これらのアダプターおよび WebSphere Process Server での作業について詳しくは、WebSphere Business Integration Adapter インフォメーション・センターを参照してください。

## WebSphere アダプターの操作

WebSphere Adapter は、管理コンソールを使用してインストールおよび管理できます。

このタスクを実行するには、管理コンソールへのアクセス権を持ち、コンソール内でサーバー設定を変更するためのセキュリティー権限を持っている必要があります。

WebSphere アダプターがアプリケーション内に組み込まれる場合も、組み込まれない場合も、WebSphere アダプターが存在すれば、このタスクを実行する必要があります。このアダプターの操作には、管理コンソールを使用します。

### 1. WebSphere アダプターをインストールします。

WebSphere アダプターのインストール・プロセスは、アダプターがアプリケーション内に組み込まれるかどうかによって異なります。

オプション	説明
スタンドアロン・アダプター	管理コンソールを使用して、アダプターをインストールします。 <b>注:</b> スタンドアロンの WebSphere アダプターは、WebSphere Process Server v6.0 ではサポートされません。
組み込みアダプター	このアダプターは、アプリケーションのインストールの一部としてインストールされます。

### 2. アダプターを管理します。

### 3. J2EE 接続ファクトリーを構成します。

## アダプターの概要

WebSphere Process Server で使用できるアダプターのタイプには、WebSphere Business Integration Adapter と WebSphere Adapter の 2 種類あります。これらのアダプターの機能について概説します。

アダプターが提供する接続機能により、統合インフラストラクチャーを強化する、データ、テクノロジー、およびプロトコルを利用できます。アダプターは、業界全体、および業界特定のパッケージ化されたアプリケーションからデータおよびトランザクション情報を抽出し、中央サーバーに接続します。

WebSphere Process Server では、次の 2 つのタイプのアダプターがサポートされています。

- **WebSphere Business Integration Adapter** は、一連のソフトウェア、アプリケーション・プログラム・インターフェース (API)、およびツールで構成され、これによって、アプリケーションは、統合ブローカー経由でビジネス・データを交換できます。
- **WebSphere アダプター** (Resource Adapter と呼ばれることもあります) は、優先テクノロジーです。これらのアダプターにより、WebSphere Process Server によってサポートされる J2EE コンポーネントと Enterprise Information System (EIS) 間の管理対象双方向接続が可能になります。

それぞれのビジネス・アプリケーションがビジネス・インテグレーション・プロセスに関与するためには、そのアプリケーション固有のアダプターが必要です。アダプターによって、Seibel または PeopleSoft などのさまざまな Enterprise Application から WebSphere Process Server への通信が可能になります。

WebSphere Business Integration Adapter によって、複数のアプリケーションの通信が可能になり、システムが完全に統合されます。Java 2 Enterprise Edition (J2EE) Connector Architecture (JCA) バージョン 1.5 に完全に準拠する WebSphere Adapter を使用するようお勧めします。WebSphere Business Integration Adapter は、JCA に準拠せず、アプリケーション・サーバーの外部で稼働します。

WebSphere Business Integration Adapters は、まだ WebSphere アダプターがユーザー・アプリケーションに存在しない場合、適切な選択です。WebSphere Business Integration Adapter ライセンスを購入している場合、そのライセンスを継続して使用できます。ただし、WebSphere アダプターは、一般にパッケージ化されているソフトウェアへの接続方法が更新されているため、WebSphere アダプターにマイグレーションすることが推奨されています。

アダプターを使用したアプリケーションの開発についての詳細は、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。

WebSphere Adapter のインストール、デプロイ、および構成についての詳細は、WebSphere Adapter インフォメーション・センターを参照してください。

WebSphere Process Server 用の WebSphere Adapters の開発について詳しくは、DeveloperWorks および「WebSphere Adapter Toolkit Guide」を参照してください。

### **WebSphere アダプターの概要:**

WebSphere Adapter は、WebSphere Process Server アプリケーションとエンタープライズ情報システム間の通信のための統合されたメカニズムを提供します。アプリケーションは、エンタープライズ情報システムに格納されたデータに対して機能します。

WebSphere Adapter は Java 2 Enterprise Edition (J2EE) Connector architecture (JCA) バージョン 1.5 を実装しています。これらのアダプターは、WebSphere アダプター、または Resource Adapter と呼ばれます。これらのアダプターは、Enterprise Information systems (EIS) と WebSphere Process Server によってサポートされる J2EE コンポーネント間の双方向接続を管理します。

JCA は、データ共有を容易にし、新規 J2EE アプリケーションをレガシーおよびその他 EIS に統合するように設計されています。JCA は、以下の機能を持つ WebSphere アダプターの開発方法を規定します。

- 任意の J2EE 準拠アプリケーション・サーバーにプラグインする。
- そのサーバー上で稼働するアプリケーションを EIS に接続する。
- J2EE アプリケーションと EIS 間でデータを交換できるようにする。

JCA 標準は、EIS とアプリケーション・サーバー内の J2EE コンポーネント間の対話を管理する一連の規約を定義することによって、これを実現します。JCA 標準に完全に準拠する WebSphere アダプターが、WebSphere Process Server で稼働するように開発されました。JCA への準拠には、以下のようないくつかの利点があります。

- JCA は、オープン・スタンダードです。
- JCA は、EIS 接続に関する J2EE 標準です。
- JCA は、管理フレームワークを提供します。

それぞれの WebSphere アダプターは、以下で構成されています。

- **ファウンデーション・クラス** このクラスは、WebSphere Process Server が、J2EE アプリケーションおよびすべての WebSphere アダプター間の相互作用を管理するために使用する、一連の汎用契約をインプリメントします。これらのサービス品質規約およびライフ・サイクル管理規約 (システム規約とも呼ばれる) は、サービス・プロバイダー・インターフェース (SPI) を定義します。例えば、システム規約は、セキュリティー信任状管理、接続プール、およびトランザクション管理パラメーターを指定します。
- **EIS サブクラス** これらの汎用 EIS 固有サブクラスは、Common Client Interface (CCI) 規約および EIS API 規約を定義します。例えば、活動化および接続仕様によって、WebSphere Process Server は、WebSphere Adapter の入力イベントおよび出力イベントを管理できます。
- **Enterprise Metadata Discovery** このユーティリティーは、EIS をイントロスペクトし、サービス・データ・オブジェクト (SDO) および、標準 Enterprise Application Archive (EAR) ファイルにコンパイルされているその他の成果物を生成します。

WebSphere アダプターの単純化された操作バージョンは、図 3 に図解されています。



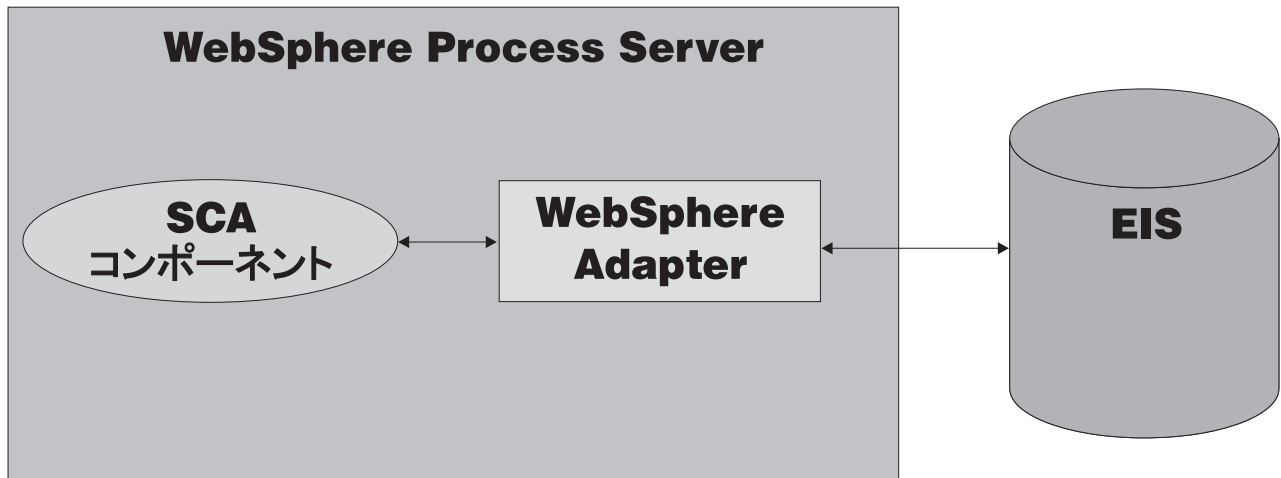


図 3. WebSphere アダプターの単純化した図解

**WebSphere アダプターおよび WebSphere Business Integration Adapter 間の差異:**

WebSphere Adapter および WebSphere Business Integration Adapter は、いずれもコンポーネントとエンタープライズ情報システム間の通信を仲介します。これら 2 種類のアダプターは、統合、JCA への準拠、データ・モデル、接続の管理などの点で異なります。

WebSphere アダプターおよび WebSphere Business Integration Adapter 間には、いくつかの差異があります。これらの違いは、アプリケーションの開発時に最も重要です。実行中のサーバーにアプリケーションをデプロイする場合、使用するアダプターの性質によって、実行する必要があるステップは異なってきます。

アダプターは、Enterprise Information System (EIS) と WebSphere アプリケーション間の通信メカニズムを提供します。アダプターの操作を説明するために、図 4 および図 5 では、2 つのアダプター・タイプについて WebSphere Process Server と EIS 間の通信について詳しく説明します。

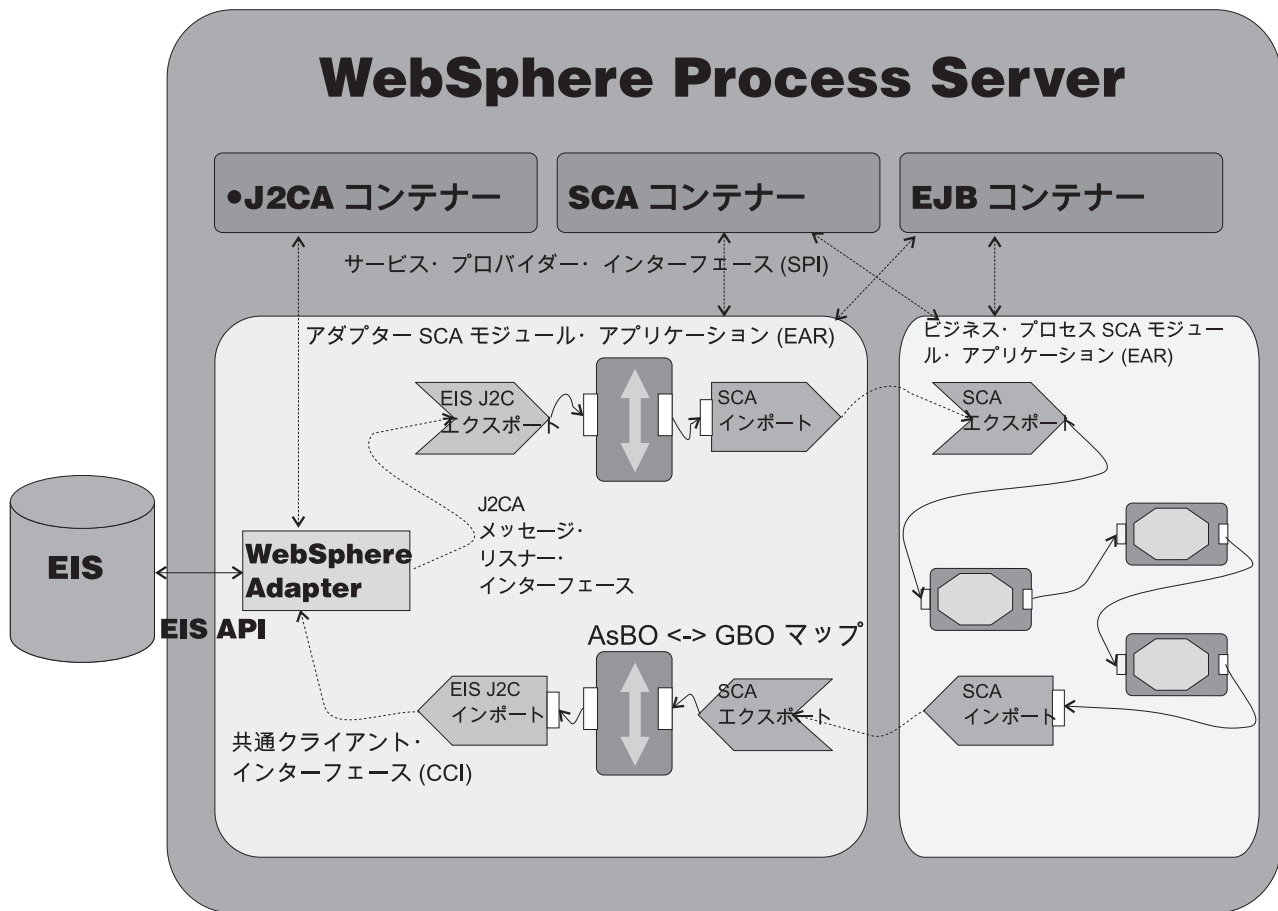


図 4. WebSphere アダプターの詳細な図式

図 4 は、WebSphere Process Server によってサポートされている J2EE コンポーネントと EIS 間の接続を管理する WebSphere アダプターを示しています。WebSphere アダプターは、WebSphere Process Server に常駐しています。

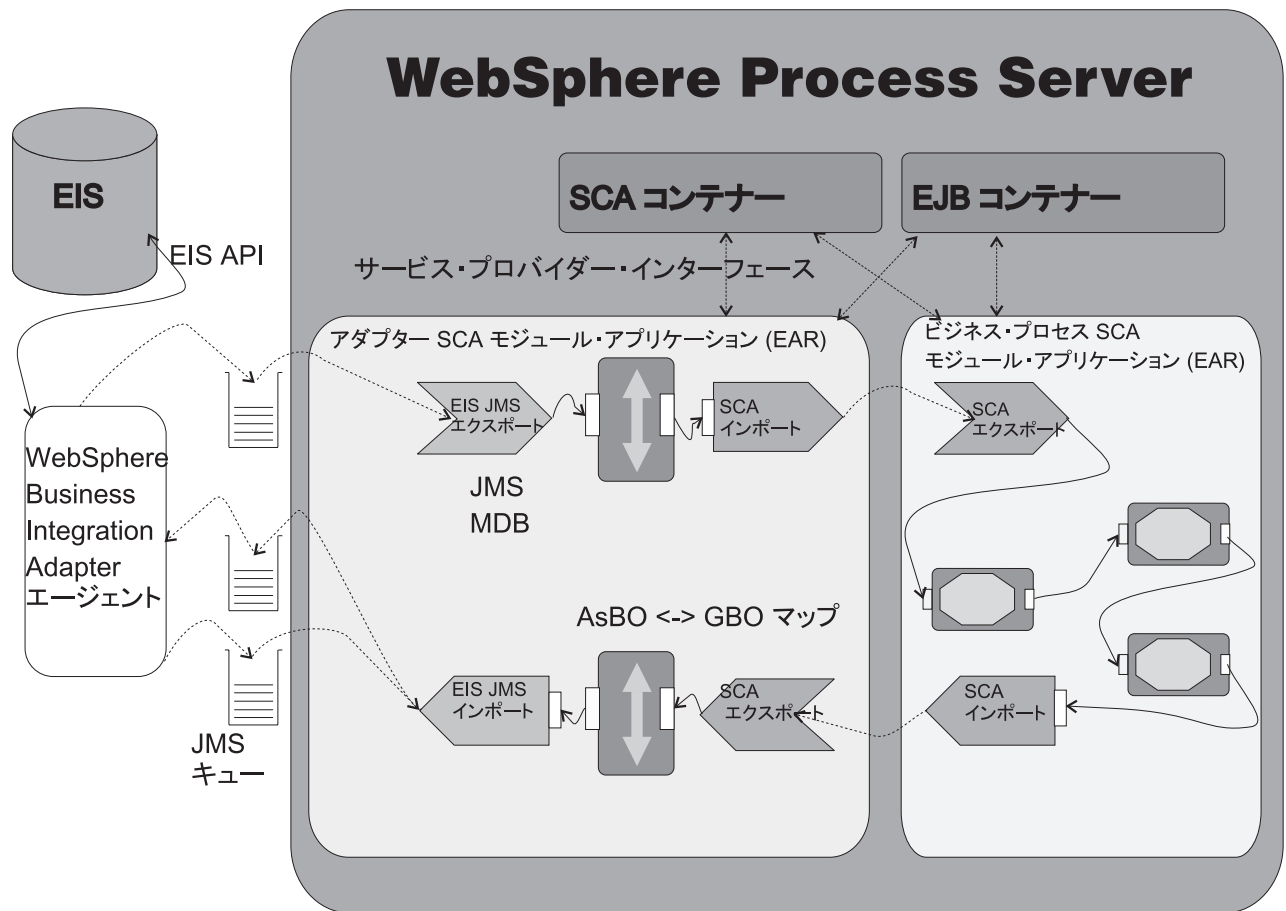


図 5. WebSphere Business Integration Adapter の詳細な図式

図 5 は、WebSphere Integration Broker と EIS 間の通信を仲介する WebSphere Business Integration Adapter を示しています。統合ブローカーは、Java Messaging Service (JMS) トランスポート・レイヤーを介して、WebSphere Business Integration Adapter と通信します。

2 つのアダプター・タイプ間の差異は、以下のとおりです。

表 8. WebSphere アダプターおよび WebSphere Business Integration Adapter 間の差異

フィーチャー	WebSphere Adapter	WebSphere Business Integration Adapter
JCA Compliance	JCA 完全準拠 (バージョン 1.5)	JCA 非準拠
Connectivity Manager	標準 JCA 規約を利用し、開始および停止などのライフ・サイクル・タスクを管理します。	WebSphere Adapter Framework を利用し、接続を管理します。
Event Notification	EventStore サブクラスを使用して、EIS からイベントを取り出します。	pollFor Events メソッドを使用してイベント通知を管理します。
Request Processing	クライアントは、いくつかの対話規約の 1 つを直接起動し、EIS のデータを照会または変更します。	組み込みサーバーおよび WebSphere Adapter Framework を利用して、処理要求を開始し、支援します。

表 8. WebSphere アダプターおよび WebSphere Business Integration Adapter 間の差異 (続き)

フィーチャー	WebSphere Adapter	WebSphere Business Integration Adapter
Data Models	Enterprise Metadata Discovery (EMD) ユーティリティを使用し、EIS を解析し、Service Data Object (SDO) およびその他有用な成果物を開発します。EMD は、WebSphere アダプターのインプリメンテーションの一部です。	別の Object Discovery Agent (ODA) を使用して、EIS をイントロスペクトし、ビジネス・オブジェクト定義スキーマを生成します。
Integration	WebSphere Process Server で実行します。	WebSphere Process Server の外側にあります。サーバーまたは統合ブローカーは、Java Message Service (JMS) トランSPORT・レイヤーを介してアダプターと通信します。

WebSphere Adapter は推奨ツールです。

#### WebSphere アダプターを使用する利点:

WebSphere Process Server では WebSphere Adapter を使用することをお勧めします。いくつかの重要な分野においては、WebSphere Business Integration Adapter を使用するより、WebSphere Adapter を使用する方が有利です。

WebSphere アダプターは、WebSphere Business Integration Adapter にいくつかの利点を提供します。以下に、これらの利点の要約を示します。

- 組み込み - WebSphere Adapters は、WebSphere Process Server に組み込まれ、WebSphere Business Integration Adapters は、アプリケーション・サーバーの外部になります。
- JCA 準拠 - WebSphere アダプターは、JCA バージョン 1.5 に完全に準拠しています。
- 要求処理 - WebSphere アダプターは、Enterprise Information Systems の要求を開始するために、WebSphere Adapter Framework も組み込みサーバーも利用しません。
- 接続 - WebSphere アダプターは、接続のために WebSphere Adapter Framework を利用していません。ただし、JCA 契約を使用してライフ・サイクル・タスクを管理します。
- データ・モデル - WebSphere アダプターは、エンタープライズ・サービス・ディスカバリー・ウィザードを使用して、EIS を解析し、Service Data Object (SDOs) を開発します。エンタープライズ・サービス・ディスカバリー・ウィザードは、WebSphere Adapter インプリメンテーションの一部です。WebSphere Business Integration Adapter は、分離したオブジェクトを使用します。
- イベント通知 - WebSphere アダプターは、EventStore のサブクラスを使用して、EIS からイベントを取得します。一方 WebSphere Business Integration Adapter は、pollFor Events メソッドを使用します。

使用可能な WebSphere アダプターは限定されていますが、可能な限りこれらのアダプターを使用することが推奨されています。

## WebSphere アダプターが組み込まれたアプリケーションのインストール

WebSphere Adapter が組み込まれたアプリケーションを作成した場合、アダプターはアプリケーションと共にデプロイされます。アダプターを別個にインストールする必要はありません。ここでは、アダプターが組み込まれたアプリケーションのインストール手順を説明します。

この作業は、アプリケーションに WebSphere アダプターが組み込まれて作成されている場合に限り実行できます。

1. アプリケーションとその中のリソース・アダプター・アーカイブ (RAR) モジュールをアセンブルします。『アプリケーションのアセンブル』を参照してください。
2. 新規アプリケーションのインストールのステップに従って、アプリケーションをインストールします。「モジュールをサーバーにマップ」ステップで、RAR ファイルごとにターゲット・サーバーまたはクラスターを指定します。RAR モジュール内で定義されているリソース・アダプターを使用するその他のモジュールはすべて、同じターゲットにマップしてください。また、このアプリケーションに対する要求のルーターとして機能するターゲットとして、Web サーバーを指定します。各 Web サーバーのプラグイン構成ファイル (plugin-cfg.xml) は、Web サーバーを経由して経路指定されるアプリケーションに基づいて生成されます。

**注:** サーバーに RAR ファイルをインストールする場合、WebSphere Process Server はコネクター・モジュールのマニフェスト (MANIFEST.MF) を検索します。最初に、RAR ファイルの connectorModule.jar ファイル内を検索し、connectorModule.jar ファイルからマニフェストをロードします。クラスパス・エントリーが connectorModule.jar ファイルのマニフェストにある場合、RAR はそのクラスパスを使用します。インストール済みコネクター・モジュールが必要なクラスおよびリソースを検出できるようにするには、コンソールを使用して、RAR のクラスパス設定を確認します。詳しくは、Resource Adapter の設定および WebSphere 関連のリソース・アダプターの設定を参照してください。

3. 変更を保管します。「終了」>「保管」をクリックします。
4. 新規にインストールされたアプリケーションの接続ファクトリーを作成します。
  - a. 管理コンソールを開きます。
  - b. 新規にインストールされたアプリケーションを選択します。「アプリケーション」>「エンタープライズ・アプリケーション」>「*application name*」をクリックします。
  - c. このページの「関連項目」セクションの「コネクター・モジュール」をクリックします。
  - d. RAR ファイルを選択します。「*filename.rar*」をクリックします。
  - e. このページの「追加プロパティ」セクションの「リソース・アダプター」をクリックします。
  - f. このページの「追加プロパティ」セクションの「J2C 接続ファクトリー」をクリックします。

- g. 「既存の接続ファクトリー」を更新する場合はそれをクリックし、新規に作成する場合は「新規」をクリックします。

**注:** WebSphere アダプターが EIS インポートまたは EIS エクスポートを使用して構成された場合、ConnectionFactory または ActivationSpec は存在し、更新することができます。

**Linux および UNIX:** ネイティブ・パス・エレメントを含むアダプターをインストールする場合、以下の点を考慮する必要があります。複数のネイティブ・パス・エレメントがあり、ネイティブ・ライブラリーの 1 つ (ネイティブ・ライブラリー A) が別のライブラリー (ネイティブ・ライブラリー B) に依存している場合、ネイティブ・ライブラリー B をシステム・ディレクトリーにコピーする必要があります。ほとんどの UNIX<sup>®</sup> システムには、ネイティブ・ライブラリーのロードを試行した場合に現行ディレクトリーが検索されないという制限があります。

接続ファクトリーを作成および保管した後で、必要に応じて、アプリケーションの各種モジュールで定義されているリソース参照を変更したり、接続ファクトリーの Java Naming and Directory Interface (JNDI) 名を指定したりできます。

**注:** 所定のネイティブ・ライブラリーは、Java 仮想マシン (JVM) のインスタンスごとに 1 度だけロードできます。アプリケーションごとに独自のクラス・ローダーがあるため、組み込み RAR ファイルを持つ 2 つのアプリケーションが両方とも同じネイティブ・ライブラリーを使用することはできません。2 番目のアプリケーションは、ライブラリーのロードを試行したときに例外を受け取ります。

アプリケーション・サーバーにデプロイされているアプリケーションが、ネイティブ・パス・エレメントを含む組み込み RAR ファイルを使用する場合は、未解決のトランザクションが残らないように、アプリケーション・サーバーを正常にシャットダウンする必要があります。正常にシャットダウンしなかった場合、アプリケーション・サーバーは、サーバーの再始動時にリカバリーを実行し、必要な RAR ファイルおよびネイティブ・ライブラリーをロードします。リカバリーが完了しても、アプリケーションに関する作業は行わないでください。サーバーをシャットダウンし、再始動します。この再始動時にはアプリケーション・サーバーによるリカバリーは実行されず、通常のアプリケーション処理を行うことができます。

### WebSphere Adapter:

WebSphere Adapter (JCA アダプターまたは J2C アダプター) は、エンタープライズ情報システム (EIS) に接続するために Java アプリケーションが使用するシステム・レベルのソフトウェア・ドライバです。WebSphere アダプターは、JCA 仕様のバージョン 1.5 に準拠しています。

WebSphere アダプターは、アプリケーション・サーバーにプラグインし、EIS、アプリケーション・サーバー、およびエンタープライズ・アプリケーションの間を接続します。

アプリケーション・サーバーのベンダーは、J2EE コネクター・アーキテクチャー (JCA) をサポートするように一度そのシステムを拡張すれば、その後は複数の EIS

へのシームレスな接続が保証されます。同様に、EIS のベンダーは、コネクタ・アーキテクチャをサポートする、アプリケーション・サーバーにプラグインする機能を備えた 1 つの標準 WebSphere アダプターを提供します。

WebSphere Process Server では、WebSphere リレーショナル・リソース・アダプター (RRA) のインプリメンテーションを提供します。この WebSphere アダプターは、JDBC 呼び出しによってデータベースに動的にアクセスするためのデータ・アクセスを提供します。接続管理は、JCA 接続管理アーキテクチャに基づいて行われます。これは、接続プール、トランザクション、およびセキュリティー・サポートを提供します。WebSphere Process Server バージョン 6.0 では、JCA バージョン 1.5 がサポートされています。

コンテナ管理パーシスタンス (CMP) Bean のデータ・アクセスは、WebSphere パーシスタンス・マネージャーによって間接的に管理されます。JCA 仕様により、パーシスタンス・マネージャーは、バックエンド・ストアについての特定の認識なしに WebSphere アダプターへのデータ・アクセスを代行することができます。リレーショナル・データベース・アクセスの場合、パーシスタンス・マネージャーは、リレーショナル・リソース・アダプターを使用して、データベースのデータにアクセスします。JDBC API 向けにサポートされるデータベース・プラットフォームに関しては、WebSphere Process Server の前提条件を記載した Web サイトを参照してください。

IBM は、WebSphere Process Server パッケージとは別に、各種のエンタープライズ・システム用のリソース・アダプターを用意しています。これには CICS<sup>®</sup>、Host On-Demand (HOD)、IMS<sup>™</sup>、および SAP R/3 などがあります。

WebSphere Process Server では、EIS インポートおよび EIS エクスポートは、WebSphere アダプターとのインターフェースで使用します。代わりに、WebSphere Adapter を使用するアプリケーションは、Rational<sup>®</sup> Application Developer などのツールを使用して EJB セッション Bean またはサービスを開発することによって作成できます。Session Bean は、javax.resource.cci インターフェースを使用して、WebSphere アダプターを介してエンタープライズ情報システムと通信します。

#### **WebSphere アダプターのデプロイメントの考慮事項:**

WebSphere Adapter のデプロイメントでは、スコープに関する特定のオプションが必要です。

管理コンソールを使用して、2 つの方法で WebSphere アダプターをデプロイできます。

- スタンドアロン - アダプターはノード・レベルでインストールされ、特定のアプリケーションとの関連付けはありません。

**注:** スタンドアロン WebSphere アダプターのデプロイメントは、WebSphere Process Server v6.0 ではサポートされていません。

- 組み込み - アダプターはアプリケーションの一部で、アプリケーションをデプロイすると、アダプターもデプロイされます。

組み込み WebSphere アダプター:

- RAR ファイルは、SCA モジュール (EIS インポートまたはエクスポートを含む) 内でアプリケーション・スコープを持つことができます。
- RAR ファイルは、非 SCA モジュール内でアプリケーション・スコープを持つことができます。EIS インポートおよびエクスポートを含むアプリケーション自体は、別個の SCA モジュールです。

スタンドアロン WebSphere アダプターは、インストールしないでください。

**注:** 管理コンソールでは、スタンドアロン WebSphere アダプターのインストールは除外されていませんが、このインストールは行わないでください。WebSphere アダプターは、アプリケーションに組み込む必要があります。

組み込み WebSphere アダプターのみ、WebSphere Process Server のデプロイメントに適しています。また、組み込み WebSphere アダプターのデプロイメントのみ、SCA モジュール内にアプリケーション・スコープされている RAR ファイルでサポートされています。非 SCA モジュールのデプロイメントはサポートされていません。

#### スタンドアロン WebSphere アダプターのインストール:

スタンドアロン WebSphere Adapter を使用する場合は、ここで説明する手順に従って、WebSphere Adapter をインストールする必要があります。組み込みアダプターを使用することもできます。組み込みアダプターは、関連アプリケーションをインストールすると自動的にインストールされます。

**注:** WebSphere アダプターは、アプリケーションに組み込む必要があります。スタンドアロン WebSphere アダプターは、WebSphere Process Server v6.0 ではサポートされていません。ここでの説明は単なる参考情報です。

アダプターをインストールする前に、データベースを構成する必要があります。

このタスクを実行するには、管理コンソールへのアクセス権と必要なセキュリティ・ロールを持っている必要があります。

1. 「RAR のインストール」ダイアログ・ウィンドウを開きます。管理コンソールで、以下を実行します。
  - a. 「リソース」を展開します。
  - b. 「リソース・アダプター」をクリックします。
  - c. このリソース・アダプターを定義する有効範囲を選択します。(この有効範囲は、ご使用の接続ファクトリーの有効範囲になります。) セル、ノード、クラスター、またはサーバーを選択できます。
  - d. 「**RAR のインストール**」をクリックします。

JCA コネクターのインストール、および JCA コネクター用の WebSphere アダプターの作成が可能なウィンドウが開きます。「新規」ボタンを使用することもできますが、「新規」ボタンで作成されるのは、新規リソース・アダプターのみです (JCA コネクターがシステムにインストール済みである必要があります)。

**注:** このダイアログを使用して RAR ファイルをインストールすると、「リソース・アダプター」ページで定義する有効範囲は、RAR ファイルのインストール・ロケーションでは無効になります。RAR ファイルは、ノード・レベ



ルでのみインストールできます。ファイルがインストールされるノードは、「RAR のインストール」ページの有効範囲により決定します。(新規リソース・アダプターの有効範囲は、「リソース・アダプター」ページで設定する有効範囲により決定します。新規リソース・アダプターは、サーバー、ノード、またはセルの各レベルでインストールできます。)

## 2. RAR ファイルをインストールします。

ダイアログから、以下の方法で WebSphere アダプターをインストールします。

- a. JCA コネクターのロケーションを参照します。RAR ファイルがローカル・ワークステーションに存在する場合は、「ローカル・パス (Local Path)」を選択し、ファイルを参照して指定します。RAR ファイルがネットワーク・サーバーに存在する場合は、「サーバー・パス (Server path)」を選択し、ファイルへの完全修飾パスを指定します。
- b. 「次へ」をクリックします。
- c. 「一般プロパティ」の下で、リソース・アダプター名およびその他の必要なすべてのプロパティを入力します。ネイティブ・パス・エレメントを含む J2C リソース・アダプターをインストールする場合、以下の点を考慮する必要があります。複数のネイティブ・パス・エレメントがあり、ネイティブ・ライブラリーの 1 つ (ネイティブ・ライブラリー A) が別のライブラリー (ネイティブ・ライブラリー B) に依存している場合、ネイティブ・ライブラリー B をシステム・ディレクトリーにコピーする必要があります。ほとんどの UNIX システムに制限があるため、ネイティブ・ライブラリーをロードしようとする際に現行ディレクトリーは調べません。
- d. 「OK」をクリックします。

### クラスター・メンバーとしての WebSphere アダプター・アプリケーション:

WebSphere Adapter モジュール・アプリケーションは、特定の条件でクラスターのメンバーとして複製できます。

WebSphere アダプター・モジュール・アプリケーションは、アダプターを経由した情報の流れに応じて、次の 3 つのタイプの内のいずれかになります。

- EIS エクスポートのみが組み込まれた WebSphere アダプター・アプリケーション - インバウンド・トラフィックのみ。
- EIS インポートのみが組み込まれた WebSphere アダプター・アプリケーション - アウトバウンド・トラフィックのみ。
- EIS インポートおよびエクスポートの両方が組み込まれた WebSphere アダプター・アプリケーション - 双方向トラフィック。

クラスターを使用することにより、Network Deployment 環境内のアプリケーションにスケラビリティと可用性が提供されます。

インバウンド・トラフィックまたは双方向トラフィックのいずれかを持つ WebSphere アダプター・モジュール・アプリケーションは、クラスターのメンバーとしてクローンすることができません。アウトバウンド・トラフィックのみを持つアプリケーションは、クラスターのメンバーとして複製できます。

インバウンドまたは双方向 WebSphere アダプター (すなわち、EIS エクスポートを含む) を持つアプリケーションには、HACMP™、Veritas、または Microsoft® Cluster Server などの、外部 Operating System High Availability (HA) 管理ソフトウェア・パッケージを使用することによって、Network Deployment 内で可用性を与えることもできます。

### クラスターのメンバーとしての WebSphere Business Integration Adapter アプリケーション:

WebSphere Business Integration Adapter モジュール・アプリケーションは、特定の条件でクラスターのメンバーとして複製できます。

WebSphere Business Integration Adapter モジュール・アプリケーションは、アダプターを経由した情報の流れに応じて、以下の 3 つのタイプの内のいずれかになります。

- EIS エクスポートのみが組み込まれた WebSphere Business Integration Adapter アプリケーション - インバウンド・トラフィックのみ。
- EIS インポートのみが組み込まれた WebSphere Business Integration Adapter アプリケーション - アウトバウンド・トラフィックのみ。
- EIS インポートおよびエクスポートの両方が組み込まれた WebSphere Business Integration Adapter アプリケーション - 双方向トラフィック。

クラスターを使用することにより、Network Deployment 環境内のアプリケーションにスケーラビリティと可用性が提供されます。

インバウンド・トラフィックまたは双方向トラフィックのいずれかを持つ WebSphere Business Integration Adapter モジュール・アプリケーションは、クラスターのメンバーとしてクローンすることができません。アウトバウンド・トラフィックのみを持つアプリケーションは、クラスターのメンバーとして複製できます。

インバウンドまたは双方向 WebSphere Business Integration Adapter (すなわち、EIS エクスポートを含む) を持つアプリケーションには、HACMP、Veritas、または Microsoft Cluster Server などの、外部 Operating System High Availability (HA) 管理ソフトウェア・パッケージを使用することによって、Network Deployment 内で可用性を与えることができます。

### 管理コンソールを使用した WebSphere アダプターの管理

WebSphere Process Server 管理コンソールを使用して、ご使用の WebSphere Adapter を管理および構成します。

このタスクを実行するには、管理コンソールで設定を変更できるセキュリティー権限を所有している必要があります。

WebSphere Adapter のインストールが完了した後で、管理コンソールを使用してアダプターを管理できます。

1. 管理するアダプター用のリソース・アダプター・コンソールを開きます。管理コンソールで、以下を実行します。
  - a. 「リソース」を展開します。
  - b. 「リソース・アダプター」をクリックします。

- c. 管理する WebSphere アダプターを選択します。
2. 「一般プロパティ」または「追加プロパティ」の見出しの下で必要に応じてプロパティを変更します。

変更が完了した後で、「適用」ボタンをクリックします。

マスター構成に保管されるまで、変更はローカル構成にのみ適用されます。

注: 変更を有効にするには、サーバーの再始動が必要になることがあります。

### WebSphere アダプター管理コンソールの設定:

ここでは、WebSphere Adapter の管理コンソールの設定と、それらのデフォルト値について説明します。

#### 目的

WebSphere Adapters 用の設定を編集するために、管理コンソールを使用します。

管理コンソールを表示するには、次のようにします。

- 「リソース」を展開します。
- 「リソース・アダプター」をクリックします。
- リストからリソース・アダプターを選択します。

各種のフィールドについて以下に説明します。

**有効範囲** このリソース定義が表示されるレベルを指定します。一般情報については、『関連参照』セクションの『管理コンソール有効範囲設定』を参照してください。「有効範囲」フィールドは、リソース・アダプターの特定の定義が配置されている場所を示す、読み取り専用ストリング・フィールドです。これは、リソース・アダプターがインストールされたとき (ノード・レベルでのみ可能)、または新規のリソース・アダプター定義が追加されたときに設定されます。

**名前** リソース・アダプター定義の名前を指定します。このプロパティは必須です。スペースを含まないストリングであり、リソース・アダプターにとって意味のあるテキスト ID となります。

プロパティ	値
データ型	ストリング

**説明** リソース・アダプターを説明するテキストを指定します。リソース・アダプターおよびその目的を説明するフリー・フォームのテキスト・ストリングです。

プロパティ	値
データ型	ストリング

**アーカイブ・パス (Archive path)** このリソース・アダプターのモジュールを含む RAR ファイルへのパスを指定します。このプロパティは必須です。

プロパティ	値
データ型	ストリング

クラスパス リソース・アダプター・クラスのロケーションを形成するパスまたは JAR ファイル名のリストを指定します。このリストには、リソース・アダプターが必要とする追加ライブラリーがすべて含まれています。リソース・アダプターのコード・ベースそのものは、自動的にクラスパスに追加されますが、RAR の外部で必要なものがあれば、ここで指定できます。

プロパティー	値
データ型	ストリング

ネイティブ・パス (Native path) リソース・アダプターのネイティブ・ライブラリーのロケーションを形成するパスのリストを指定します。リソース・アダプターのコード・ベースそのものは、自動的にクラスパスに追加されますが、RAR の外部で必要なものがあれば、ここで指定できます。

プロパティー	値
データ型	ストリング

スレッド・プール別名 (ThreadPool Alias) リソース・アダプターの作業マネージャーが使用するサーバー内で構成されるスレッド・プールの名前を指定します。サーバーにこの名前で構成されたスレッド・プールがない場合、デフォルトとして構成された Default という名前のスレッド・プール・インスタンスが使用されます。このプロパティーが必要となるは、このリソース・アダプターが作業マネージャーを使用する場合に限られます。

プロパティー	値
データ型	ストリング

## 接続ファクトリーの管理

接続ファクトリーは、アプリケーションとエンタープライズ情報システム (EIS) 間の通信で使用します。

アプリケーション・コンポーネントは、接続ファクトリーを使用して接続インスタンスにアクセスします。次いで、コンポーネントは、接続インスタンスを使用して基盤となる EIS に接続します。

接続の例には、データベース接続、Java Message Service 接続、および SAP R/3 接続などがあります。

### 管理コンソールでの J2EE コネクタ接続ファクトリーの構成:

接続ファクトリーは、アプリケーションとエンタープライズ情報システム (EIS) の間の通信を仲介するために使用されます。管理コンソールを使用して、接続ファクトリーを管理できます。

スタンドアロン・リソース・アダプターを使用している場合は、このタスクを実行する必要があります。組み込み WebSphere アダプターは少し異なる処理を必要とします。管理コンソールを編集するセキュリティ権限も必要です。

アプリケーションが非リレーショナル・データベースにアクセスする必要がある場合、JDBC プロバイダーおよびデータ・ソースではなく、リソース・アダプターおよび接続ファクトリーを構成する必要があります。

1. 管理するリソース・アダプターを選択します。管理コンソールの最上位から、以下のステップを実行します。
  - a. 「リソース」を展開します。
  - b. 「リソース・アダプター」を選択します。
  - c. 管理する WebSphere アダプターを選択します。
2. このアダプターの新規接続ファクトリーを作成します。

以下のように、新規接続ファクトリーを作成します。

- a. 「追加プロパティ」の下にある「JCA 接続ファクトリー (JCA Connection Factories)」を選択します。
- b. 「新規」をクリックします。
- c. 一般的なプロパティを指定します。
- d. 認証設定を指定します。
- e. コンポーネント管理の認証またはコンテナ管理の認証 (あるいはその両方) の別名を選択します。

別名が使用できない場合、または異なる別名を定義する場合は、以下を実行します。

- 1) 「適用」をクリックします。
  - 2) 「関連項目」の下にある「J2C 認証データ・エントリ」をクリックします。
  - 3) 「新規」をクリックします。
  - 4) 一般的なプロパティを指定します。
  - 5) 「OK」をクリックします。
  - f. 「OK」をクリックします。
3. 新規に作成した接続ファクトリーの接続プール・プロパティを変更し、接続プール・マネージャーの動作を最適化します。

以下のように、接続プール値を変更します。

- a. 新規接続ファクトリーを選択します。
- b. 「追加プロパティ」の見出しの下にある「接続プール・プロパティ」を選択します。
- c. プロパティ名をクリックして、必要な値をすべて変更します。
- d. 「OK」をクリックします。
- e. 「追加プロパティ」の下にある「カスタム・プロパティ」をクリックします。
- f. 任意のプロパティ名をクリックして、値を変更します。
- g. 「OK」をクリックします。

**関連タスク**

74 ページの『アプリケーション内の WebSphere アダプターの接続ファクトリーの構成』

WebSphere Adapter は、スタンドアロンにするか、またはアプリケーション内に組み込むことができます。接続ファクトリーを構成するプロセスは、アダプターがアプリケーション内にあるかどうかによって異なります。WebSphere アダプターがアプリケーションに組み込まれている場合、以下の説明に使用して接続ファクトリーを構成します。

#### 接続プール用の一般プロパティの設定:

管理コンソールを使用して、接続プールに一般設定を割り当てることができます。

接続プール用の一般設定を変更するには、管理コンソールで値を変更できるセキュリティ権限を所有している必要があります。

接続プールの一般プロパティを割り当てるためには、最初に WebSphere アダプター用の接続ファクトリーを作成する必要があります。

1. 管理コンソールの「接続プール・プロパティ」パネルを開きます。

管理コンソールの最上位から、以下を実行します。

- a. 「リソース」を展開します。
  - b. 「リソース・アダプター」をクリックします。
  - c. 管理する接続ファクトリーを持っている WebSphere Adapter をクリックします。
  - d. 「追加プロパティ」の見出しの下にある「J2C 接続ファクトリー」をクリックします。
  - e. 管理する接続ファクトリーを選択します。
  - f. 「追加プロパティ」の見出しの下にある「接続プール・プロパティ (Connection pool properties)」をクリックします。
2. タイムアウト、最大および最小接続数、ページ・ポリシーなどの一般接続プール・プロパティを、このパネルで変更できます。デフォルト値が提供されています。
  3. 変更が必要なプロパティを変更した後で、「OK」をクリックします。

#### 接続プール設定:

管理コンソールの「接続プール・プロパティ (Connection Pool Properties)」パネルで接続プールの各種のプロパティの値を変更できます。

#### 目的

「接続プール設定 (Connection Pool Settings)」パネルは接続プール用の一般的なプロパティ値を割り当てるために使用されます。タイムアウト、ページ・ポリシー、および接続限界などのプロパティを編集できます。

#### 「構成」タブ

#### 接続タイムアウト (Connection timeout)

接続要求がタイムアウトになり、`ConnectionWaitTimeoutException` がスローされるまでの時間間隔 (秒単位) を指定します。

この値は、通常、特定の接続プールの接続数が最大値に達したために、空きプール内に使用可能な接続がなく、また新規接続が作成できない時に、接続要求が待機する秒数を示します。例えば、「接続タイムアウト (Connection timeout)」が 300 に設定されている場合、最大数の接続がすべて使用中であれば、プール・マネージャーは物理接続が使用可能になるまで 300 秒間待機します。この時間内に物理接続が使用可能にならない場合は、プール・マネージャーが `ConnectionWaitTimeout` 例外を発生させます。`getConnection()` メソッドを再試行しても、通常は効果がありません。待機時間を長くする必要がある場合は、接続タイムアウトの設定値を大きくしてください。アプリケーションが `ConnectionWaitTimeout` 例外をキャッチした場合、管理者は予期されるアプリケーションの接続プール使用率を確認し、それに従って接続プールとデータベースを調整する必要があります。

「接続タイムアウト (Connection timeout)」が 0 に設定されている場合、プール・マネージャーは、接続が使用可能になるまで必要な限り待機します。アプリケーションがトランザクションを完了して接続をプールに戻したとき、または接続数が「最大接続数 (Maximum Connections)」の値を下回り、新規物理接続の作成が許可されたときに、接続が使用可能になります。

「最大接続数 (Maximum Connections)」が 0 に設定されていると、物理接続の数が無制限になるため、「接続タイムアウト (Connection timeout)」の値は無視されます。

プロパティ	値
データ型	整数
単位	秒
デフォルト	180
範囲	0 から最大の整数

### 最大接続数 (Maximum connections)

このプールで作成できる物理接続の最大数を指定します。

これらは、バックエンド・リソースへの物理接続です。この数に達すると、新規の物理接続は作成されず、要求元は、現在使用中の物理接続がプールに戻されるか、`ConnectionWaitTimeout` 例外が発行されるまで待機します。

例えば、「最大接続数 (Maximum connections)」の値が 5 に設定されているときに、5 つの物理接続が使用中の場合、プール・マネージャーは、「接続タイムアウト (Connection timeout)」に指定された時間、物理接続が空くまで待機します。

「最大接続数 (Maximum connections)」が 0 に設定されている場合は、接続プールを無限に大きくすることができます。このようにすると、「接続タイムアウト (Connection timeout)」の値が無視されるという副次作用もあります。

複数のスタンドアロン・アプリケーション・サーバーが同じデータ・ソースを使用している場合、アプリケーション・サーバーごとに 1 つのプールがあります。クローンを使用する場合、クローンごとに 1 つのデータ・プールが存在します。データ・プール数を知ることは、データベースの最大接続数を構成する場合に重要となります。

Tivoli® Performance Viewer を使用して、最適なプール内の接続数を見つけます。同時に待機している数が 0 より大きいにもかかわらず、CPU 負荷が 100% 近くではない場合は、接続プール・サイズを大きくすることを検討してください。通常のワークロードで使用パーセントの値が常に低い場合は、プール内の接続数を減らすことを検討してください。

プロパティ	値
データ型	整数
デフォルト	10
範囲	0 から最大の整数

### 最小接続数 (Minimum connections)

維持する物理接続の最小数を指定します。

接続プールのサイズが最小接続プール・サイズと同じかまたはそれより小さい場合、「未使用タイムアウト (Unused Timeout)」スレッドは物理接続を廃棄しません。しかし、最小接続プール・サイズを維持することを保証するために、プールが単独で接続を作成することはありません。また、「経過タイムアウト (Aged Timeout)」の値を設定すると、最小プール・サイズ設定にかかわらず、経過時間の有効期限が切れた接続は廃棄されます。

例えば、「最小接続数 (Minimum connections)」の値が 3 に設定されていて、1 つの物理接続が作成された場合、その接続が「未使用タイムアウト (Unused Timeout)」スレッドによって廃棄されることはありません。また、「最小接続数 (Minimum connections)」の設定値に達するように、スレッドが自動的に 2 つの追加の物理接続を作成することはありません。

プロパティ	値
データ型	整数
デフォルト	1
範囲	0 から最大の整数

### リープ時間 (Reap time)

プール維持スレッドが実行されてから次に実行されるまでの間隔 (秒) を指定します。

例えば、「リープ時間 (Reap time)」が 60 に設定されていると、プール維持スレッドは 60 秒ごとに実行されます。「リープ時間 (Reap time)」の間隔は、「未使用タイムアウト (Unused Timeout)」と「経過タイムアウト (Aged Timeout)」の設定値の精度に影響を与えます。間隔が短いほど精度は高まります。プール維持スレッドが使用可能である場合は、「リープ時間 (Reap time)」に、「未使用タイムアウト (Unused Timeout)」や「経過タイムアウト (Aged Timeout)」の値よりも小さい値を設定してください。プール維持スレッドが実行されると、接続数が「最小接続数 (Minimum Connections)」で指定された数になるまで、「未使用タイムアウト (Unused Timeout)」で指定された値よりも長時間使用されていない接続をすべて廃棄します。プール維持スレッドは、「経過タイムアウト (Aged Timeout)」で指定された値よりも長時間アクティブである接続も、すべて廃棄します。



「リープ時間 (Reap time)」の間隔は、パフォーマンスにも影響を与えます。間隔を短くすると、プール維持スレッドの実行回数が増え、パフォーマンスが低下することになります。

プール維持スレッドを使用不可にするには、「リープ時間 (Reap time)」を 0 に設定するか、または「未使用タイムアウト (Unused Timeout)」と「経過タイムアウト (Aged Timeout)」の両方を 0 に設定します。プール維持スレッドを使用不可にする方法として推奨されるのは、「リープ時間 (Reap time)」を 0 に設定する方法です。この場合「未使用タイムアウト (Unused Timeout)」と「経過タイムアウト (Aged Timeout)」は無視されます。ただし、「未使用タイムアウト (Unused Timeout)」と「経過タイムアウト (Aged Timeout)」が 0 に設定されている場合は、プール維持スレッドは実行されますが、タイムアウト値が非ゼロであるためにタイムアウトになった物理接続のみ廃棄されます。

プロパティ	値
データ型	整数
単位	秒
デフォルト	180
範囲	0 から最大の整数

#### 未使用タイムアウト (Unused Timeout)

未使用またはアイドル状態の接続が廃棄されるまでの時間 (秒) を指定します。

パフォーマンスを最適化するためには、「未使用タイムアウト (Unused Timeout)」に、「リープ時間 (Reap time)」より高い値を設定してください。未使用の物理接続が廃棄されるのは、現行の接続数が、「最小接続数 (Minimum connections)」の設定値を超える場合に限られます。例えば、未使用タイムアウト値が 120 に設定され、プール維持スレッドが使用可能 (「リープ時間 (Reap time)」が 0 でない) である場合、2 分間未使用の状態が続いた物理接続は廃棄されます。パフォーマンスと同様、このタイムアウトの精度も「リープ時間 (Reap time)」の値の影響を受けることに注意してください。詳しくは、『リープ時間』を参照してください。

プロパティ	値
データ型	整数
単位	秒
デフォルト	1800
範囲	0 から最大の整数

#### 経過タイムアウト (Aged timeout)

物理接続が廃棄されるまでの時間 (秒) を指定します。

「経過タイムアウト (Aged timeout)」を 0 に設定すると、アクティブな物理接続を無期限にプールに残しておくことができます。パフォーマンスを最適化するためには、「経過タイムアウト (Aged timeout)」に、「リープ時間 (Reap time)」より高い値を設定してください。例えば、「経過タイムアウト (Aged timeout)」の値が 1200 に設定され、「リープ時間 (Reap time)」の値が 0 でない場合は、1200 秒間 (20 分間) 存在し続けている物理接続はプールから廃棄されます。パフォーマンスと同様、このタイムアウトの精度も「リープ時間 (Reap time)」の値の影響を受けます。詳しくは、『リープ時間』を参照してください。

プロパティ	値
データ型	整数
単位	秒
デフォルト	0
範囲	0 から最大の整数

### ページ・ポリシー (Purge policy)

失効した接続または致命的接続エラーが検出されたときに、接続をページする方法を指定します。

有効な値は以下のとおりです。

- **EntirePool**: プール内の接続は、すべて失効したものとマークされます。使用されていない接続は、直ちに閉じられます。使用中の接続は閉じられ、次にその接続に操作が行われたときには、失効接続例外が発行されます。アプリケーションからの後続の `getConnection()` 要求が出されると、開かれているデータベースへの新規接続が作成されます。このページ・ポリシーを使用しているとき、プール内の一部の接続が、失効していない場合に不必要に閉じられる可能性があります。ただし、このようになることはまれです。ほとんどの場合は、ページ・ポリシーを **EntirePool** に設定することが最善の選択です。
- **FailingConnectionOnly**: 失効接続例外の発生原因である接続のみが閉じられます。この設定により、有効な接続が不必要に閉じられなくなる一方で、アプリケーション・パースペクティブからのリカバリーが複雑になります。現在障害が発生している接続のみが閉じられるので、アプリケーションからの次の `getConnection()` 要求により、プールからやはり失効した接続が戻され、結果的にさらに多くの失効接続例外が発生する可能性が極めて高くなります。接続の事前テスト機能では、無効なプール済み接続からのアプリケーションの分離を試みます。データベースなどのバックエンド・リソースが停止した場合、無効なプール済み接続が、空きプールに存在していることがあります。特に、ページ・ポリシーが **failingConnectionOnly** である場合にこのようになります。この場合には、障害のある接続は、そのプールから除去されず。障害の種類によっては、プール内の残りの接続が無効になっている場合があります。

プロパティ	値
データ型	ストリング
デフォルト	EntirePool

### 接続プール用の拡張プロパティの設定:

管理コンソールを使用して、接続プールに拡張設定を割り当てることができます。

接続プール用の拡張設定を変更するには、管理コンソールで値を変更できるセキュリティ権限を所有している必要があります。

接続プールの拡張プロパティを割り当てるためには、最初に、WebSphere アダプター用の接続ファクトリーを作成する必要があります。

1. 管理コンソールの「拡張接続プール・プロパティ (Advanced Connection Pool Properties)」パネルを開きます。

管理コンソールの最上位から、以下を実行します。

- a. 「リソース」を展開します。
  - b. 「リソース・アダプター」をクリックします。
  - c. 管理する接続ファクトリーを持っている WebSphere Adapter をクリックします。
  - d. 「追加プロパティ」の見出しの下にある「J2C 接続ファクトリー」をクリックします。
  - e. 管理する接続ファクトリーを選択します。
  - f. 「追加プロパティ」の見出しの下にある「接続プール・プロパティ (Connection pool properties)」をクリックします。
  - g. 「追加プロパティ」の見出しの下にある「拡張接続プール・プロパティ (Advanced Connection Pool Properties)」をクリックします。
2. 分割化、接続過負荷からの保護、応答していない接続の処理などの拡張接続プール設定を、このパネルで変更できます。デフォルト値が提供されていますが、パフォーマンスを改善できることがあるので、これらのプロパティを調整することをお勧めします。
  3. 変更が必要なプロパティをすべて変更した後で、「OK」をクリックします。

#### 接続プール拡張設定:

管理コンソールの「拡張接続プール設定 (Advanced Connection Pool Settings)」パネルで、接続プール用の拡張設定を変更できます。

#### 目的

「拡張接続プール設定 (Advanced Connection Pool Settings)」パネルは、接続プール用の値を割り当てるために使用されます。このパネルで、区分化、接続の過負荷の回避、および応答していない接続の処理などのプロパティを変更できます。

#### 「構成」タブ

##### 共用区画数 (Number of shared partitions)

各共用プールで作成される区画の数を指定します。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

##### 空きプール区画の数 (Number of free pool partitions)

空きプールのそれぞれに作成される区画の数を指定します。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

##### 空きプール分配テーブル・サイズ (Free pool distribution table size)

空きプール分配テーブル・サイズは、ハッシュ・テーブル内で Subject および CRI ハッシュ値を効果的に分配するために使用され、一致する空き接続の高速検索で発生する衝突を最小限にします。

信任状の異なる着信要求が多数ある場合、この値によってそのユーザーの接続用の空きプールの検索を分散できます。通常、リソースへのアクセスに多数の異なる信任状が使用されるインストールの場合は、値を大きくします。リソースに対するすべての着信要求に同一の信任状が適用される場合は、小さい値 (1) を使用します。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

### サージしきい値 (Surge Threshold)

作成された接続の数が、ここで指定された値に達すると、サージ保護がアクティブ化されます。

サージ保護は、同時に作成された接続の数が多すぎる場合に、データ・ソースの過負荷を防止するように設計されています。サージ保護は、サージしきい値およびサージ作成間隔という 2 つのプロパティで制御されます。

サージしきい値プロパティでは、サージ保護がアクティブ化されるまでに作成される接続の数を指定します。指定した接続数に達すると、サージ・モードに入ります。

サージ作成間隔プロパティでは、サージ・モードでの接続作成の間隔を秒数で指定します。

例えば、以下のように設定するものとします。

- maxConnections = 50
- surgeThreshold = 10
- surgeCreationInterval = 30 秒

接続プールが接続要求を 15 個受信すると、ほとんど同時に接続が 10 個作成されます。最初に 10 個の接続が作成されてから 30 秒が経過した後、11 番目の接続が作成されます。11 番目の接続が作成されてから 30 秒が経過した後、12 番目の接続が作成されます。新規の接続が必要でなくなるか、または maxConnections 値に達するまで、30 秒ごとに接続が作成され続けます。

サージしきい値が > -1、およびサージ作成間隔が > 0 である場合、サージ接続サポートが開始されます。サージしきい値プロパティのデフォルト値は -1 です。この値は、このプロパティをオフすることを意味します。

wsadmin の例

```
$AdminControl getAttribute $objectname surgeCreationInterval
$AdminControl setAttribute $objectname surgeCreationInterval 30
$AdminControl getAttribute $objectname surgeThreshold
$AdminControl setAttribute $objectname surgeThreshold 15
```

プロパティ	値
データ型	整数

プロパティ	値
デフォルト値	-1
範囲	-1 から最大の整数

### サージ作成間隔 (Surge creation interval)

サージ保護モードでの、接続作成の時間間隔を指定します。

サージしきい値プロパティで指定された数の接続が作成されると、新規の接続要求は、サージ作成間隔後に作成されるまで待機しなければなりません。このプロパティのデフォルト値は 20 です。すなわち、次の接続が作成されるまでに 20 秒以上経過していなければなりません。このプロパティの有効な値は、任意の正整数です。

プロパティ	値
データ型	整数
デフォルト値	20
範囲	0 から最大の整数

### 滞留タイマー時間 (Stuck timer time)

滞留接続は、接続プールに応答したり戻されたりすることのないアクティブな接続です。プールが滞留状態 (滞留しきい値に達した) になると、プールの滞留が解除されるまで、すべての新規接続要求に対してリソース例外が出されます。滞留タイマー時間プロパティは、タイマーの間隔です。これは、接続プールが滞留接続をチェックする間隔です。デフォルト値は 5 秒です。

**wsadmin** スクリプト・ツールを使用して滞留時間、滞留タイマー時間、または滞留しきい値プロパティの変更を試みて失敗した場合は、`IllegalState` 例外が発生します。この要求の間、プールに、アクティブな要求やアクティブな接続を置くことはできません。滞留した接続のサポートが開始されるには、3 つの滞留プロパティ値がすべて 0 より大きく、さらに最大接続数が 0 より大きくなければなりません。

また、滞留タイマー時間が設定されている場合は、滞留時間の値よりも小さくしなければなりません。滞留タイマー時間を、滞留時間の 4 分の 1 から 6 分の 1 にして、接続の滞留が宣言される前に、接続プールが滞留接続を 4 回から 6 回チェックするようにすることをお勧めします。これにより、誤って滞留であると判断する可能性が低くなります。

#### **wsadmin** の例

```
$AdminControl getAttribute $objectname stuckTime
$AdminControl setAttribute $objectname stuckTime 30
$AdminControl getAttribute $objectname stuckTimerTime
$AdminControl setAttribute $objectname stuckTimerTime 15
$AdminControl getAttribute $objectname stuckThreshold
$AdminControl setAttribute $objectname stuckThreshold 10
```

プロパティ	値
データ型	整数
デフォルト値	5
範囲	0 から最大の整数

### 滞留時間 (Stuck time)

滞留接続は、接続プールに応答したり戻されたりすることのないアクティブな接続です。プールが滞留状態 (滞留しきい値に達した) になると、プールの滞留が解除されるまで、すべての新規接続要求に対してリソース例外が出されます。滞留時間プロパティには間隔 (秒単位) を指定し、バックエンド・リソースへの単一のアクティブ接続の使用時間がこの値に達すると、その接続は滞留していると見なされます。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

#### 滞留しきい値 (Stuck Threshold)

滞留接続は、接続プールに応答したり戻されたりすることのないアクティブな接続です。プールが滞留状態 (滞留しきい値に達した) になると、プールの滞留が解除されるまで、すべての新規接続要求に対してリソース例外が出されます。アプリケーションは、この例外を明示的にキャッチして処理を続けることができます。滞留接続数がしきい値を超えると、プールは滞留接続の定期的なチェックを継続します。滞留接続数が滞留しきい値よりも小さくなると、プールは、定期チェック時にそのことを検出し、要求処理の再開が可能となります。滞留しきい値は、滞留していると見なされる接続がどれだけの数になれば、プールが滞留モードになるかを示すものです。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

#### アプリケーション内の WebSphere アダプターの接続ファクトリーの構成:

WebSphere Adapter は、スタンドアロンにするか、またはアプリケーション内に組み込むことができます。接続ファクトリーを構成するプロセスは、アダプターがアプリケーション内にあるかどうかによって異なります。WebSphere アダプターがアプリケーションに組み込まれている場合、以下の説明に使用して接続ファクトリーを構成します。

このタスクを実行するには、管理コンソールで設定を編集するために必要な権限を持つ必要があります。

1. アップロードおよびインストールする RAR ファイルを含む EAR ファイルを指定します。

管理コンソールの最上位から、以下の手順を実行します。

- a. 「アプリケーション」を展開します。
- b. 「新規アプリケーションのインストール」を選択します。
- c. ローカルまたはリモート・システム上の EAR ファイルを参照します。

リモートまたはローカル・ファイル・システムに関連するラジオ・ボタンを選択してから、EAR ファイルへのパスを指定します。

- d. 「次へ」をクリックします。

2. J2C 接続ファクトリーにマップすることを指定し、アプリケーションをインストールします。

J2C 接続ファクトリーに対するリソース参照マッピングを選択し、「次へ」をクリックします。

3. アプリケーションのインストール後に、新規にインストールされたアプリケーションに組み込まれている WebSphere アダプター用の接続ファクトリーを作成および構成します。

管理コンソールの最上位から、以下の手順を実行します。

- a. 「アプリケーション」を展開します。
- b. 「エンタープライズ・アプリケーション」を選択します。
- c. 新規にインストールされたアプリケーションを選択します。
- d. 「関連項目」の下にある「コネクタ・モジュール」をクリックします。
- e. RAR ファイルを選択します。
- f. 「追加プロパティ」の下にある「リソース・アダプター」を選択します。
- g. 「追加プロパティ」の下にある「J2C 接続ファクトリー」をクリックします。
- h. 「新規」をクリックします。
- i. すべての必要な一般プロパティを指定します。
- j. **オプション:** XA リカバリー用の認証別名を指定するか、コンポーネント管理の認証別名を使用します。このフィールドは、XA トランザクションをサポートするリソース用にのみ表示されます。
- k. **オプション:** コンポーネント管理の認証別名を選択します。

使用したい別名が使用できない場合、または異なる別名を定義する場合は、以下を実行します。

- 1) 「適用」をクリックします。
- 2) 「関連項目」の下にある「J2C 認証データ・エントリ」をクリックします。
- 3) 「新規」をクリックします。
- 4) 一般的なプロパティを指定します。
- 5) 「OK」をクリックして、「J2C 接続ファクトリー設定 (J2C Connection Factories Settings)」パネルに戻ります。

アプリケーションのいずれかのコンポーネントがこの接続ファクトリーから接続を取得するときに、空の引数を持つ `getConnection()` メソッドを使用し、アプリケーションまたは接続ファクトリーごとの認証をリソース参照で指定している場合は、別名を選択または定義します。

1. 新規接続ファクトリーの作成を完了します。「OK」をクリックします。
4. **オプション:** 新規接続ファクトリーの任意の接続プール・プロパティを変更します。

「J2C 接続ファクトリー・コレクション」パネルから、以下を実行します。

- a. 作成した接続ファクトリーを選択します。

- b. 「接続プール設定 (Connection Pool settings)」パネルを開きます。「追加プロパティ」の見出しの下にある「**接続プール・プロパティ (Connection pool properties)**」をクリックします。
  - c. プロパティ名をクリックして、任意のプロパティの値を変更します。
  - d. 接続プール設定へのこれらの変更を確認します。「**OK**」をクリックします。
5. **オプション:** 任意のカスタム・プロパティを変更します。

新規接続ファクトリーの「接続ファクトリー設定」パネルから、以下を実行します。

- a. 「追加プロパティ」の見出しの下にある「**カスタム・プロパティ**」をクリックします。
- b. 任意のプロパティ名をクリックして、値を変更します。

注: 「ユーザー名」および「パスワード」が存在する場合、これらは、前のステップで指定したコンポーネント管理の認証別名によってオーバーライドされます。

- c. 「**保管**」をクリックします。

#### 関連タスク

64 ページの『管理コンソールでの J2EE コネクター接続ファクトリーの構成』接続ファクトリーは、アプリケーションとエンタープライズ情報システム (EIS) の間の通信を仲介するために使用されます。管理コンソールを使用して、接続ファクトリーを管理できます。

### J2C 接続ファクトリー・コレクション:

「J2C 接続ファクトリー・コレクション (J2C connection factory collection)」パネルでは、選択可能な接続ファクトリーのリストが表示されます。

#### 目的

このパネルを使用して、接続ファクトリーを選択します。接続ファクトリーは、1 セットの接続構成値を表します。

エンタープライズ Bean などのアプリケーション・コンポーネントには、WebSphere Adapter ではなく、接続ファクトリーを参照するリソース参照記述子が含まれます。接続ファクトリーは、実際には構成プロパティ・リストのホルダーです。WebSphere アダプターのベンダーで定義されている任意の構成プロパティ・セットに加えて、接続ファクトリーに適用する複数の標準構成プロパティがあります。これらの標準プロパティは、アプリケーション・サーバー・ランタイムで Java 2 コネクターの接続プール・マネージャーによって使用され、ベンダーが提供する WebSphere Adapter コードでは認識されません。

「**接続ファクトリー・コレクション (Connection Factory collection)**」パネル

#### 名前

接続ファクトリー表示名のリストを指定します。



プロパティ データ型	値 文字列
---------------	----------

## JNDI 名

この接続ファクトリーの Java Naming and Directory Interface (JNDI) 名を指定します。

プロパティ データ型	値 文字列
---------------	----------

## 説明

この接続ファクトリーを説明するテキストを指定します。

プロパティ データ型	値 文字列
---------------	----------

## カテゴリ

この接続ファクトリーの分類またはグループ化に使用できる文字列を指定します。

プロパティ データ型	値 文字列
---------------	----------

## J2C 接続ファクトリーの設定:

管理コンソールの「J2C 接続ファクトリー設定 (J2C Connection Factory Settings)」パネルで接続ファクトリーの各種プロパティ用の設定を指定します。

## 目的

「J2C 接続ファクトリー設定 (J2C Connection Factory Settings)」パネルは、選択された接続ファクトリー用の一般的なプロパティ値を割り当てるために使用されます。タイムアウト、ページ・ポリシー、および接続限界などのプロパティを編集できます。

### 「構成」タブ

#### 接続ファクトリー・インターフェース (Connection Factory Interface)

リソース・アダプターがサポートする接続ファクトリー・インターフェースの完全修飾名を指定します。

これは必須プロパティです。新規のオブジェクトに対して、使用可能なクラスのリストがドロップダウン・リスト内のリソース・アダプターによって提供されます。接続ファクトリーが作成された後は、このフィールドは読み取り専用テキスト・フィールドです。

プロパティ データ型	値 ドロップダウン・リストまたはテキスト
---------------	-------------------------

#### XA リカバリーの認証別名 (Authentication Alias for XA recovery)

このオプション・フィールドを使用して、XA リカバリー処理時に使用される認証別名を指定します。

WebSphere アダプターが XA トランザクションをサポートしない場合、このフィールドは表示されません。デフォルト値には、アプリケーション認証用に選択された別名 (指定されている場合) が使用されます。

#### コンポーネント管理認証別名の使用 (Use Component-managed Authentication Alias)

このラジオ・ボタンを選択すると、コンポーネント管理認証に設定された別名を XA リカバリー時に使用することが指定されます。

プロパティ データ型	値 ラジオ・ボタン
---------------	--------------

#### 指定: (Specify:)

このラジオ・ボタンを選択することで、構成された別名のドロップダウン・リストから認証別名を選択できます。

プロパティ データ型	値 ラジオ・ボタン
---------------	--------------

#### 名前

接続ファクトリー表示名のリストを指定します。

これは必須プロパティです。

プロパティ データ型	値 ストリング
---------------	------------

#### JNDI 名

この接続ファクトリーの JNDI 名を指定します。

例えば、*eis/myECIConnection* といった名前を指定できます。

この値を設定した後、値を保管してサーバーを再始動します。dumpNameSpace ツールを実行すると、このストリングが表示されます。これは必須プロパティです。JNDI 名を指定しない場合は、「名前」フィールドを使用してデフォルトで入力されます。

プロパティ データ型 デフォルト	値 ストリング <i>eis/display name</i>
------------------------	---------------------------------------

#### 説明

この接続ファクトリーを説明するテキストを指定します。

プロパティ データ型	値 ストリング
---------------	------------

#### カテゴリ

この接続ファクトリーの分類またはグループ化に使用できるストリングを指定します。

プロパティ データ型	値 ストリング
---------------	------------

### コンポーネント管理の認証別名 (Component-managed Authentication Alias)

リソースに対するコンポーネント管理サインオンの認証データを指定します。

「セキュリティー (Security)」 > 「JAAS 構成 (JAAS Configuration)」 > 「J2C 認証データ (J2C Authentication Data)」 で定義された別名から選択します。

ピック・リストに表示されたことのない新規別名を定義するには、以下のようにします。

- 「適用」をクリックして「関連項目」を表示します。
- 「J2C 認証データ・エントリー (J2C Authentication Data Entries)」をクリックします。
- 別名を定義します。
- 「J2C 認証データ・エントリー (J2C Authentication Data Entries)」ページの最上部にある接続ファクトリー名をクリックして、接続ファクトリーのページに戻ります。
- 別名を選択します。

プロパティ データ型	値 ピック・リスト
---------------	--------------

### コンテナ管理認証別名 (Container-managed Authentication Alias)

リソースに対するコンテナ管理サインオンの認証データ (ユーザー ID およびパスワードに変換されるストリング・パス) を指定します。

**注:** res-auth=Container と設定されたコンポーネントでは、コンテナ管理認証別名は、デプロイメント時にリソース参照マッピングのログイン構成の指定によって置き換えられます。

「セキュリティー (Security)」 > 「JAAS 構成 (JAAS Configuration)」 > 「J2C 認証データ (J2C Authentication Data)」 で定義された別名から選択します。

まだリストに組み込まれていない新規別名を定義するには、以下のようにします。

- 「適用」をクリックして「関連項目」を表示します。
- 「J2C 認証データ・エントリー (J2C Authentication Data Entries)」をクリックします。
- 別名を定義します。
- 「J2C 認証データ・エントリー (J2C Authentication Data Entries)」ページの最上部にある接続ファクトリー名をクリックして、接続ファクトリーのページに戻ります。
- 別名を選択します。

プロパティ データ型	値 ピック・リスト
---------------	--------------

### 認証設定 (Authentication Preference)

この接続ファクトリーに定義される認証メカニズムを指定します。

**注:** 認証設定は、<res-auth> アプリケーション・コンポーネントのデプロイメント記述子設定と、デプロイメント時のリソース参照マッピングでのログイン構成の指定との組み合わせによって置き換えられます。

認証設定では、対応するリソース・アダプターに定義されている認証メカニズムのうちの、どの認証メカニズムをこの接続ファクトリーに適用するかを指定します。その値としては、リソース・アダプターの機能によって異なりますが、一般に KERBEROS、BASIC\_PASSWORD、および「なし」が使用されます。「なし」を選択した場合、アプリケーション・コンポーネントが、認証を管理することになります (<res-auth>Application</res-auth>)。この場合、ユーザー ID とパスワードは以下のうちのいずれかから取得されます。

- コンポーネント管理認証別名
- ユーザー名およびパスワードのカスタム・プロパティ
- getConnection メソッドで渡されるストリング

例えば、*ra.xml* 文書にあるリソース・アダプターに対して、以下の 2 つの認証メカニズム・エントリーが定義されているとします。

- <authentication-mechanism-type>BasicPassword</authentication-mechanism-type>
- <authentication-mechanism-type>Kerby5</authentication-mechanism-type>

認証設定では、コンテナ管理認証に使用されるメカニズムを指定します。この WebSphere アダプターがサポートしていないメカニズムが選択された場合は、サーバーの始動中に例外が発行されます。

プロパティ データ型 デフォルト値	値 ピック・リスト BASIC_PASSWORD
-------------------------	--------------------------------

### マッピング構成別名 (Mapping-Configuration Alias)

ユーザーが「セキュリティ (Security)」>「JAAS 構成 (JAAS Configuration)」>「アプリケーション・ログイン構成 (Application Logins Configuration)」リストから選択できるようにします。

**注:** res-auth=Container と設定されているコンポーネントでは、マッピング構成別名は、デプロイメント時のリソース参照マッピングでのログイン構成の指定によって置き換えられます。

DefaultPrincipalMapping JAAS 構成を使用すると、認証別名をユーザー ID およびパスワードにマップできます。他のマッピング構成を定義して使用することもできます。

プロパティ データ型	値 ピック・リスト
---------------	--------------

## J2C 接続ファクトリーの拡張設定:

管理コンソールの「J2C 接続ファクトリー拡張設定 (J2C Connection Factory Advanced Settings)」パネルで、J2C 接続ファクトリー用の拡張設定を変更できます。

### 目的

「J2C 接続ファクトリー拡張設定 (J2C Connection Factory Advanced Settings)」パネルは接続ファクトリーの拡張プロパティー用の値を割り当てるために使用されます。このパネルで、キャッシュされたハンドルおよび欠落トランザクションのロギングの管理に関するプロパティーを変更できます。

#### 「構成」タブ

##### キャッシュされたハンドルの管理 (Manage cached handles)

これにチェック・マークが付けられている場合、キャッシュされたハンドル (Bean の inst vars に保持されているハンドル) がコンテナによって追跡されません。

プロパティー	値
データ型	チェック・ボックス

##### トランザクション・コンテキストの欠落をログに記録 (Log missing transaction contexts)

これにチェック・マークが付けられている場合、コンテナは、接続の確立時にトランザクション・コンテキストの欠落をログに記録します。

プロパティー	値
データ型	チェック・ボックス

## WebSphere Business Integration Adapter の構成

WebSphere Business Integration Adapter の構成および使用の処理は 3 倍になります。

1. アプリケーション EAR ファイルをインストールします。

アプリケーション EAR ファイルをインストールすると、WebSphere Business Integration Adapter が動作するために必要なすべての成果物が作成されます。

2. WebSphere Business Integration Adapter の管理をセットアップします。

WebSphere Business Integration Adapter を管理するには、以下を実行する必要があります。

- a. 接続キュー・ファクトリーを作成します。
- b. WebSphere Business Integration Adapter リソースを作成します。
- c. WebSphere Business Integration Adapter サービスを使用可能にします。

**注:** WebSphere Business Integration Adapter サービスを使用可能または使用不可にする場合、変更を有効にするためにサーバーを再始動する必要があります。

### 3. WebSphere Business Integration Adapter を管理します。

管理コンソールを使用して、WebSphere Business Integration Adapter を管理します。

#### **WebSphere Business Integration Adapter が動作するために必要な成果物の作成:**

WebSphere Business Integration Adapter が動作するために必要な成果物を作成するには、アプリケーション EAR ファイルをインストールします。

WebSphere Process Server で WebSphere Business Integration Adapter を使用するために必要な成果物を作成するには、「モジュールの開発とデプロイ」の PDF の『実動サーバーへのモジュールのインストール』の指示に従う必要があります。

#### **WebSphere Business Integration Adapter の管理のセットアップ:**

WebSphere Business Integration Adapter を管理できるようにするには、いくつかの管理機能を実行する必要があります。必要な手順を以下に説明します。

アプリケーション EAR ファイルをインストールして、このタスクを実行する前に WebSphere Business Integration Adapter に必要な成果物を作成する必要があります。

WebSphere Business Integration Adapter を管理するには、まず以下の管理機能を実行する必要があります。

#### 1. キュー接続ファクトリーを作成します。

管理コンソールの最上位から、以下の手順を実行します。

- a. 「リソース」を展開します。
- b. 「JMS プロバイダー」を展開します。
- c. 「デフォルトのメッセージング」を選択します。
- d. 「JMS キュー接続ファクトリー」を選択します。

JMS 副見出しの下で、「JMS キュー接続ファクトリー」を選択します。

#### e. 新規 JMS キュー接続ファクトリーを作成します。

「新規」をクリックします。

#### f. 以下を例外として、すべてのデフォルト値を受け入れます。

- 名前: QueueCF
- JNDI 名: jms/QueueCF
- バス名: ご使用のバス名

#### g. 新規 JMS キュー接続ファクトリーの作成を完了します。

「OK」をクリックします。

「JMS キュー接続ファクトリー」パネルの上部にメッセージ・ウィンドウが表示されます。

#### h. ローカル構成レベルで行った変更をマスター構成に適用します。

メッセージ・ウィンドウで「保管」をクリックします。

2. WebSphere Business Integration Adapter リソースを作成します。

管理コンソールの最上位から、以下の手順を実行します。

- a. 「リソース」を展開します。
- b. 「WebSphere Business Integration Adapter」パネルを開きます。

「**WebSphere Business Integration Adapters**」を選択します。

- c. 新規 WebSphere Business Integration Adapter を作成します。

「**新規**」をクリックします。

- d. 以下を例外として、すべてのデフォルト値を受け入れます。

- 名前: EISConnector
- キュー接続ファクトリー JNDI 名: jms/QueueCF
- 管理入力キュー JNDI 名: *connectorName/AdminInQueue*
- 管理出力キュー JNDI 名: *connectorName/AdminOutQueue*

- e. WebSphere Business Integration Adapter の作成を完了します。

「**OK**」をクリックします。

WebSphere Business Integration Adapters パネルの上部にメッセージ・ウィンドウが表示されます。

- f. ローカル構成レベルで行った変更をマスター構成に適用します。

メッセージ・ウィンドウで「**保管**」をクリックします。

3. WebSphere Business Integration Adapter サービスを使用可能にします。

管理コンソールの最上位から、以下の手順を実行します。

- a. 「サーバー」を展開します。
- b. 「**アプリケーション・サーバー**」を選択します。
- c. サーバー・リストから、WebSphere Business Integration Adapter サービスを使用可能にするサーバーを選択します。

対象となるリソースをホストするサーバーの名前をクリックします。

- d. 「**WebSphere Business Integration Adapter サービス**」を選択します。

「**構成**」タブの副見出し「**ビジネス・インテグレーション**」の下で「**WebSphere Business Integration Adapter サービス**」を選択します。

- e. 「**始動時にサービスを使用可能にする (Enable Service at startup)**」チェック・ボックスが選択されていることを確認します。

- f. 「**OK**」をクリックします。

WebSphere Business Integration Adapters パネルの上部にメッセージ・ウィンドウが表示されます。

- g. WebSphere Business Integration Adapter サービスが使用可能な各サーバーで、ステップ 3c から 3f を繰り返します。
- h. ローカル構成レベルで行った変更をマスター構成に適用します。

メッセージ・ウィンドウで「保管」をクリックします。

**注:** WebSphere Business Integration Adapter サービスを使用可能または使用不可にする場合、変更を有効にするためにサーバーを再始動する必要があります。

### WebSphere Business Integration Adapter の管理:

WebSphere Business Integration Adapter が稼働している場合、管理コンソールの「WebSphere Business Integration Adapter リソースの管理」パネルを使用して WebSphere Business Integration Adapter を管理できます。

管理するためには WebSphere Business Integration Adapter が稼働している必要があります。

WebSphere Business Integration Adapter は管理コンソールから管理できます。

「WebSphere Business Integration Adapter リソースの管理」パネルを使用すると、1 つ以上のリソースを選択して、これらのリソースに対するさまざまな管理アクションを管理および実行することができます。

#### 1. 管理するリソース (1 つまたは複数) を選択します。

管理コンソールの最上位から、以下の手順を実行します。

- a. 「サーバー」を展開します。
- b. 「アプリケーション・サーバー」を選択します。
- c. サーバーのリストから、管理するリソースが存在するサーバーを選択します。

対象となるリソースをホストするサーバーの名前をクリックします。

#### d. 「WebSphere Business Integration Adapter サービス」を選択します。

「構成」タブの副見出し「ビジネス・インテグレーション」の下で「WebSphere Business Integration Adapter サービス」を選択します。

- e. 「WebSphere Business Integration Adapter リソースの管理」を選択します。
- f. リソースのリストから、管理するリソースを選択します。

管理するリソースに関連したチェック・ボックスを選択します。

#### 2. 選択したリソースを管理します。

選択したリソースに適用するコマンド・ボタンの 1 つをクリックします。

コマンド	説明
非アクティブ化	選択したリソースの状況をアクティブから休止または非アクティブに変更します。
アクティブ化	選択したリソースの状況を非アクティブからアクティブに変更します。
中断	選択したリソースの状況をアクティブから休止に変更します。



コマンド	説明
再開	選択したリソースの状況を休止からアクティブに変更します。
シャットダウン	選択したリソースの状況をアクティブから使用不可に変更します。

## サービス・コンポーネント定義および MQClientLink 構成ファイルの生成

WebSphere Business Integration Adapter を使用する前に、サービス・コンポーネント定義 (SCA 成果物) および MQClientLink 構成ファイル (.wbia ファイル) を生成する必要があります。これは WebSphere Integration Developer 環境の WebSphere Business Integration Adapter Artifact Importer で行います。

WebSphere Integration Developer は WebSphere Business Integration Adapter Artifact Importer を使用して WebSphere Business Integration Adapter Connector Configuration File および WebSphere Business Integration Adapter Business Objects ディレクトリを発見しインポートし、WebSphere Business Integration Adapter 用の指定した相互作用スタイルをサポートする希望のサービス・コンポーネント定義を生成します。

**注:** このタスクは WebSphere Integration Developer で実行されるため、ここでの説明は単なる参考情報です。詳しくは、WebSphere Integration Developer のインフォメーション・センターを参照してください。

1. 必要な構成ファイルおよびビジネス・オブジェクトを取得します。
2. WebSphere Business Integration Adapter Artifact Importer を使用して、必要なサービス・コンポーネント定義 (SCA 成果物) を生成します。

サービス・コンポーネント定義 (SCA 成果物) および WebSphere Process Server MQClientLink 構成ファイル (.wbia ファイル) が生成されます。

アプリケーションが WebSphere Process Server にデプロイされると、サービス・コンポーネント定義および MQClientLink 構成 (.wbia) ファイルは、デプロイメント・ツールによって自動的に処理されます。構成ファイルはデフォルト状態にしておくことをお勧めしますが、必要に応じてファイルを手動で編集できます。

## EIS アプリケーションのインストール

EIS アプリケーション・モジュールは、J2EE プラットフォームにデプロイできます。デプロイメントによって、EAR ファイルとしてパッケージされたアプリケーションがサーバーにデプロイされます。すべての J2EE 成果物およびリソースが作成され、アプリケーションが構成され、実行の準備が整います。

J2EE プラットフォームへのデプロイメントによって、以下の J2EE 成果物およびリソースが作成されます。

表9. バインディングから J2EE 成果物へのマッピング

SCA モジュール内のバインディング	生成された J2EE 成果物	作成された J2EE リソース
EIS インポート	モジュール・セッション EJB で生成されたリソース参照。	ConnectionFactory
EIS エクスポート	リソース・アダプターでサポートされるリスナー・インターフェースに応じて、生成またはデプロイされるメッセージ駆動型 Bean。	ActivationSpec
JMS インポート	ランタイムが提供するメッセージ駆動型 Bean (MDB) はデプロイされ、リソース参照はモジュール・セッション EJB 上で生成されます。MDB はインポートに受信宛先がある場合にのみ作成されることに注意してください。	<ul style="list-style-type: none"> <li>• ConnectionFactory</li> <li>• ActivationSpec</li> <li>• Destinations</li> </ul>
JMS エクスポート	ランタイムが提供するメッセージ駆動型 Bean はデプロイされ、リソース参照はモジュール・セッション EJB 上で生成されます。	<ul style="list-style-type: none"> <li>• ActivationSpec</li> <li>• ConnectionFactory</li> <li>• Destinations</li> </ul>

インポートまたはエクスポートが ConnectionFactory のようなリソースを定義する場合、リソース参照はモジュール・ステートレス・セッション EJB のデプロイメント記述子に生成されます。また、適切なバインディングは EJB バインディング・ファイルに生成されます。リソース参照がバインドされる名前は、ターゲット属性の値 (存在する場合) か、モジュール名およびインポート名に基づいてリソースに与えられるデフォルト JNDI 検索名のいずれかです。

デプロイメント時に、インプリメンテーションはモジュール・セッション Bean を配置し、それをリソースを検索するために使用します。

アプリケーションのサーバーへのデプロイメント時に、EIS インストール・タスクは、バインド先のエレメント・リソースが存在するかどうかを確認します。それが存在せずに、SCDL ファイルが少なくとも 1 つのプロパティを指定する場合、リソースは EIS インストール・タスクによって作成および構成されます。リソースが存在せずに、アクションが実行されない場合、アプリケーション実行の前にそのリソースが作成されることが想定されます。

JMS インポートが受信宛先を指定してデプロイされる場合、メッセージ駆動型 Bean (MDB) がデプロイされます。その MDB は、送信された要求への応答を listen します。MDB は、要求と共に送信された Destination に関連付けられています (この Destination を listen します)。この Destination は、JMS メッセージの JMSreplyTo ヘッダー・フィールドで指定されます。応答メッセージが到着したときに、MDB はその相関 ID を使用して、コールバック Destination に格納されているコールバック情報を検索し、コールバック・オブジェクトを呼び出します。

インストール・タスクでは、インポート・ファイルの情報から `ConnectionFactory` および 3 つの宛先を作成します。また、`ActivationSpec` を作成して、ランタイム MDB が受信 `Destination` 上の応答を `listen` できるようにします。`ActivationSpec` のプロパティは、`Destination/ConnectionFactory` プロパティから派生します。JMS プロバイダーが `SIBus` リソース・アダプターである場合、JMS `Destination` に対応する `SIBus Destination` が作成されます。

JMS エクスポートがデプロイされている場合、メッセージ駆動型 Bean (MDB) (JMS インポートでデプロイした MDB とは異なる MDB) がデプロイされます。その MDB は、受信 `Destination` 上で着信要求を `listen` し、SCA が処理する要求をディスパッチします。インストール・タスクは、JMS インポート用のセットに類似したリソースのセット (`ActivationSpec`、応答を送信するために使用される `ConnectionFactory`、および 2 つの `Destination`) を作成します。これらのリソースのすべてのプロパティが、エクスポート・ファイルで指定されます。JMS プロバイダーが `SIBus` リソース・アダプターである場合、JMS `Destination` に対応する `SIBus Destination` が作成されます。

## WebSphere アダプター・アプリケーションのリモート・デプロイメント

Network Deployment 環境では、WebSphere Adapter アプリケーションをリモート側のノードにインストールすると便利な場合があります。この場合のノードは、コンシュームする側のアプリケーション (BPEL プロセスを含むアプリケーションなど) が配置されているノードではなく、EIS に近いノード (EIS と同じマシン上など) にします。ここでは、このようなりモート・デプロイメントのベスト・プラクティスについて説明します。

このタスクを実行するには、アプリケーションに WebSphere アダプターを組み込んでいる必要があります。また、デプロイメント・マネージャーが作動している Network Deployment 環境で作業する必要があり、リモート・マシンにアプリケーションをインストールするための十分な特権も必要です。

BPEL プロセス・アプリケーションなど利用中のアプリケーションがある場合に、このタスクは適しています。このようなアプリケーションは、Network Deployment 環境で、WebSphere アダプター・アプリケーションを媒体として EIS と通信を行う必要があります。

1. セルの任意のノードに、コンシュームする側のアプリケーションをインストールします。
2. WebSphere アダプター・アプリケーションを、このアプリケーションが対話する EIS にごく接近したノードにリモート側でインストールします。

これで、利用中のプロセスは、WebSphere アダプター・アプリケーションを介して、EIS の情報にアクセスできるようになります。セル内のすべてのノードにアダプターをインストールする必要はありません。

WebSphere アダプター・アプリケーションを介して EIS と通信する必要がある BPEL 処理アプリケーションにとって、WebSphere アダプター・アプリケーションのリモート・デプロイメントは理想的です。WebSphere アダプター・アプリケーションは、通信する EIS の近くの便利な位置にあるセル内のノードに置くことができます。BPEL プロセス・アプリケーションと同じノードに配置する必要はありません。

ん。ベスト・プラクティスのガイドラインに従うと、WebSphere Process Server の Network Deployment 環境は、図 6 のようなアーキテクチャーになります。

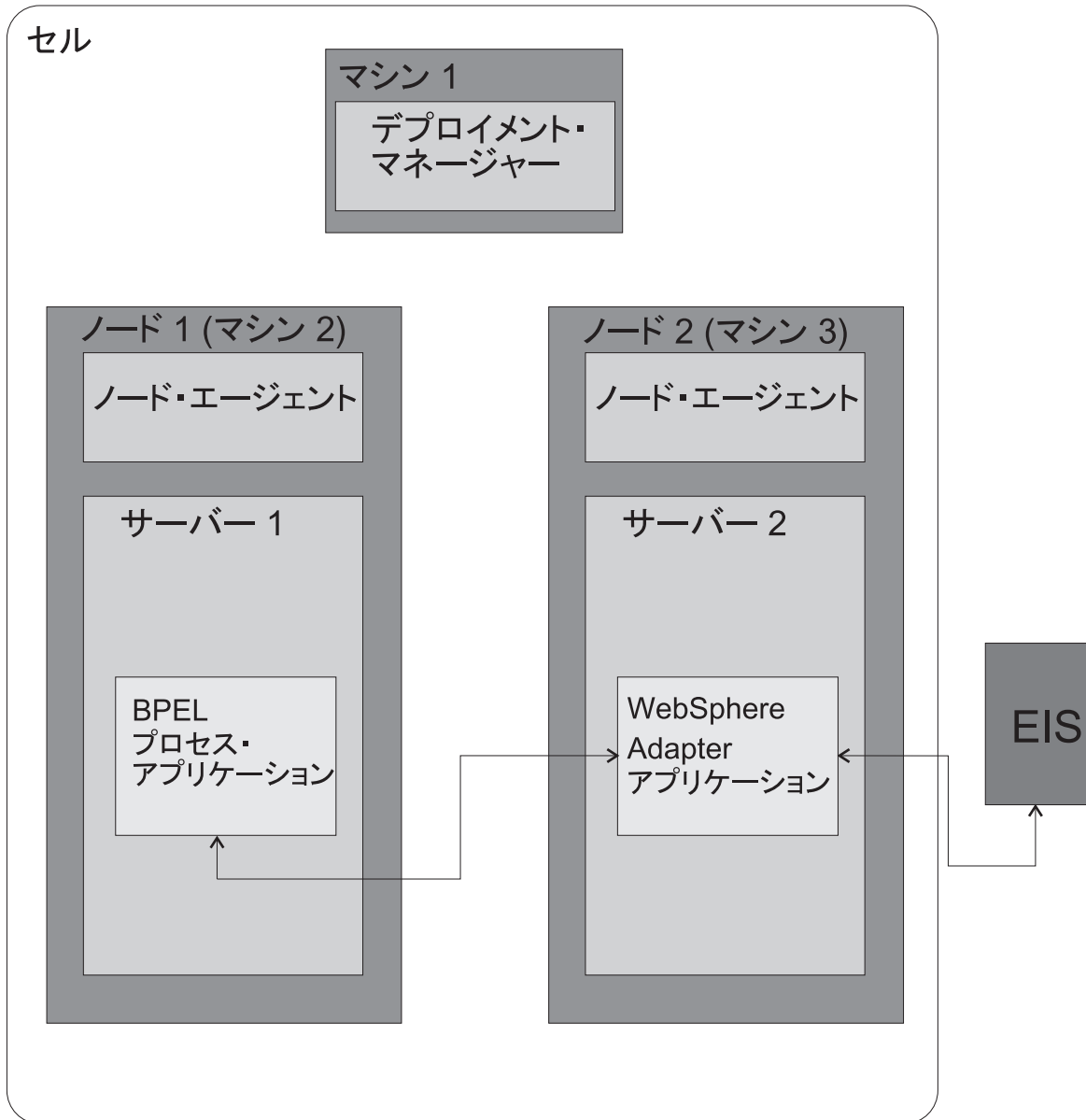


図 6. Network Deployment 環境に WebSphere アダプター・アプリケーションをリモート配置するセル・アーキテクチャー

## バインディング

バインディング情報は、サービスとアプリケーションとの接続方法および対話方法を決定します。バインディングは、本質的にはインポートおよびエクスポートに割り当てられたプロトコルおよびトランスポートです。

アプリケーションは、バインディング経由でサービスに接続します。

バインディングは、インポートおよびエクスポート・コンポーネントの重要な部分です。具体的には、バインディングは、インポートおよびエクスポート・コンポーネントが、そのインポートおよびエクスポート・コンポーネントを含むモジュールの外部にあるクライアントと対話する方法を決定します。バインディングは、特定のインターフェースに対するメッセージの形式およびプロトコルの詳細を指定します。

JMS バインディングは、信頼性が重要になる場合に使用します。信頼性が重要なビジネス・プロセスでは、多くの場合、メッセージング・キューを使用した非同期通信を使用します。メッセージ・コンシューマーおよび作成者が求める組み合わせ (JMS バインディングとインポートおよびエクスポートの各コンポーネントとの組み合わせ) があります。

IBM Web サービス・バインディングは、インポートおよびエクスポート・コンポーネントと共に使用することもできます。

インポート・コンポーネントのみの場合は、ステートレス・セッション・エンタープライズ Java Bean (EJB) バインディングを使用できます。

## EIS および JMS バインディング

EIS システムの場合の主要なバインディングは JMS バインディングと EIS バインディングです。

JMS バインディングはモジュール間で使用します。つまり、JMS バインディングを持つモジュールにエクスポート・コンポーネントを作成するか、JMS バインディングを持つモジュールにインポート・コンポーネントを作成します。EIS バインディングでこれを行うことはできません。JMS は非同期通信に使用するため、外部のメッセージング・ソースとモジュール内のコンポーネントの間のバインディングとして JMS を使用することもできます。

EIS バインディングは、EIS システムからアプリケーション、またはその逆で使用します。EIS システムからアプリケーションの場合は、EIS バインディングを持つエクスポート・コンポーネントを開発します。アプリケーションから EIS システムの場合は、EIS バインディングを持つインポート・コンポーネントを開発します。本来、EIS バインディングは、外部の EIS システムにアクセスしたり、WebSphere Process Server にデプロイされたアプリケーションに外部の EIS システムがアクセスしたりするために使用します。

### 関連概念

42 ページの『Service Component Architecture モジュールと WebSphere MQ』

## SCA EIS バインディング

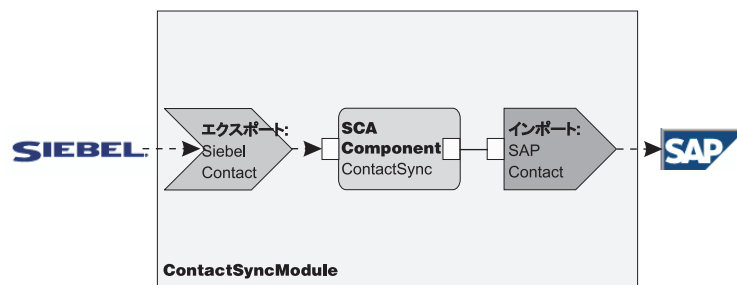
Service Component Architecture (SCA) エンタープライズ情報システム (EIS) のバインディングは、SCA コンポーネントと外部システム間の接続を提供します。この通信は、EIS エクスポートと EIS インポートによりメディアエーションされます。

SCA コンポーネントでは、外部 EIS との間でのデータの転送が必要になる場合があります。このような接続を必要とする SCA モジュールを作成する場合は、SCA コンポーネントに加えて、特定の外部 EIS との通信のための EIS インポートおよ

び EIS エクスポートを組み込む必要があります。EIS インポートおよびエクスポートを作成する方法についての詳細情報は、WebSphere Integration Developer インフォメーション・センターにあります。

## 例

ContactSyncModule という名前の標準的な SCA モジュールは、Siebel システムと SAP システム間の接続情報を同期化します。ContactSync という名前の SCA コンポーネントは、Siebel Contact という名前の EIS サービス・エクスポートを介して、Siebel への連絡に対する変更を listen します。ContactSync SCA コンポーネント自体は、SAP の接続情報を適宜更新するために、EIS サービス・インポートを介して SAP サービスを利用します。接続情報の保管用に使用されるデータ構造は、Siebel システムと SAP システムでは異なるため、ContactSync SCA がマッピングを行う必要があります。



### 関連概念

96 ページの『JMS バインディング』

Java Message Service (JMS) プロバイダーにより、Java Messaging Service に基づいたメッセージングが可能になります。JMS プロバイダーは、JMS 宛先への接続を作成する J2EE 接続ファクトリーを提供します。JMS バインディングは、EIS インポート およびエクスポート・バインディングを特殊化したものです。

## EIS バインディング: 全体像

データ・バインディングは、SCA アプリケーションに渡されるデータと EIS の間のマッピングを記述します。

クライアントは、SCA モジュールの適切なインポートまたはエクスポート・ファイルを使用して EIS インポートまたはエクスポートの機能を定義します。実行時には、インポートまたはエクスポート・ランタイム・コンポーネントを介して EIS システムとの対話を駆動するためにこのモデルが使用されます。

### JCA 対話仕様および接続仕様の動的プロパティ:

SCA EIS バインディングは、有効搭載量に付随する明確に定義された子データ・オブジェクトを使用することによって指定された InteractionSpec および ConnectionSpec に対する入力を受け入れることができます。これにより、InteractionSpec を介したリソース・アダプターとの動的要求応答対話と、ConnectionSpec を介したコンポーネント認証が可能になります。

javax.cci.InteractionSpec は、リソース・アダプターとの対話要求の処理方法に関する情報を保持します。また、要求後に対話を行う方法に関する情報も保持します。対話によるこれらの両方向通信は、会話とも呼ばれます。

SCA EIS バインディングは有効搭載量を必要とします。この有効搭載量は、「**properties**」という子データ・オブジェクトを格納するためのリソース・アダプターに対する引数になります。この **property** データ・オブジェクトは、名前と値のペアを含みます。この名前は、特定のフォーマットの対話仕様プロパティの名前になります。フォーマット設定の規則は以下のとおりです。

1. 名前はプレフィックス「IS」で開始し、その後にプロパティ名が続かなければなりません。例えば、InteractionId という Java Bean プロパティを持つ対話仕様は、プロパティ名を ISInteractionId と指定します。
2. 名前と値のペアは、対話仕様プロパティの名前と単純型の値を表します。

操作の入力が「Account」データ・オブジェクトであるインターフェースがあります。このインターフェースは、値が「xyz」である「workingSet」という動的 InteractionSpec プロパティを送受信するために、EIS インポート・バインディング・サービスを呼び出します。ここで、WebSphere Process Server のビジネス・グラフまたはビジネス・オブジェクトが、有効搭載量を持つプロトコル固有データを送信可能な下位の「properties」ビジネス・オブジェクトを含んでいることを利用できます。この **properties** ビジネス・オブジェクトは組み込みのものであり、ビジネス・オブジェクトを構成するときに XML スキーマで指定する必要はありません。単に作成するだけで使用できます。XML スキーマに基づいて独自のデータ型を定義している場合は、必要な名前と値のペアを含む「**properties**」エレメントを指定する必要があります。

```
BOFactory dataFactory = (BOFactory) ¥
serviceManager.locateService("com/ibm/websphere/bo/BOFactory");
//Wrapper for doc-lit wrapped stle interfaces,
//skip to payload for non doc-lit
DataObject docLitWrapper = dataFactory.createByElement /
("http://mytest/eis/Account", "AccountWrapper");
```

有効搭載量を作成します。

```
DataObject account = docLitWrapper.createDataObject(0);
DataObject accountInfo = account.createDataObject("AccountInfo");
//Perform your setting up of payload
```

```
//Construct properties data for dynamic interaction
```

```
DataObject properties = account.createDataObject("properties");
```

名前「workingSet」に対して予想される値「xyz」を設定します。

```
properties.setString("ISworkingSet", "xyz");
```

```
//Invoke the service with argument
```

```
Service accountImport = (Service) ¥
serviceManager.locateService("AccountOutbound");
DataObject result = accountImport.invoke("createAccount", docLitWrapper);
```

```
//Get returned property
```

```
DataObject retProperties = result.getDataObject("properties");  
String workingset = retProperties.getString("ISworkingSet");
```

**ConnectionSpec** プロパティは動的コンポーネント認証に使用できます。上記と同じ規則が適用されます。ただし、プロパティ名のプレフィックスは「CS」にする必要があります（「IS」ではありません）。**ConnectionSpec** プロパティは両方向ではありません。同じ「**properties**」データ・オブジェクトに IS プロパティと CS プロパティの両方を入れることができます。**ConnectionSpec** プロパティを使用するには、インポート・バインディングで指定する **resAuth** を「Application」とし、リソース・アダプターがコンポーネント許可をサポートする必要があります。詳しくは、「J2EE Connector Architecture Specification」の第 8 章を参照してください。

## EIS バインディングのサポート

EIS モジュールは、J2EE プラットフォームにデプロイできます。

EIS モジュール (EIS モジュール・パターンに従う SCA モジュール) は、J2EE プラットフォームにデプロイできます。

EIS モジュールを J2EE プラットフォームにデプロイすると、アプリケーションを実行する準備が整い、EAR ファイルとしてパッケージされ、サーバーにデプロイされます。すべての J2EE 成果物およびリソースが作成され、アプリケーションは構成され、実行の準備が整います。

## EIS インポート・バインディングの主な特徴

EIS インポートとは、Service Component Architecture (SCA) モジュール内のコンポーネントが、SCA モジュール外部で定義された EIS サービスを使用できるようにする SCA インポートです。EIS インポートは、SCA コンポーネントから外部 EIS へのデータ転送に使用されます。

### 目的

EIS インポートの役割は、SCA コンポーネントと外部 EIS システムの間のすき間を埋めることです。外部サービスを EIS エクスポートとして取り扱うことができます。この場合、外部サービスはそのデータを定期的な通知という形で送信します。

EIS インポートは、SCA コンポーネントに対して、モジュール外部のサービスの統一されたビューを提供します。これにより、コンポーネントは、SAP、Siebel、PeopleSoft などのさまざまな外部 EIS と、一貫性のある SCA モデルを使用して通信できます。

インポートのクライアント側には、EIS インポート・サービスにより公開される、1 つ以上のメソッド (それぞれがデータ・オブジェクトを引数および戻り値として取る) を使用するインターフェースがあります。実装側には、リソース・アダプターによって実装される共通クライアント・インターフェース (CCI) があります。

EIS インポートのランタイム実装は、クライアント側のインターフェースとこの CCI を接続します。インポートにより、インターフェース上のメソッドの呼び出しが CCI 上の呼び出しにマップされます。インターフェース・バインディング、それ



に含まれるメソッド・バインディングを使用する、そのメソッド・バインディングが次にデータ・バインディングを使用する、という 3 つのレベルのバインディングが作成されます。

インターフェース・バインディングは、インポートのインターフェースを、サービスを提供する EIS システムとの接続に関連付けます。これは、インターフェースによって表されるサービス・セットが EIS の特定のインスタンスによって提供され、接続によりこのインスタンスにアクセスできるという事実を示しています。バインディング・エレメントは、接続を作成するための十分な情報を指定したプロパティを持ちます (このプロパティは、`javax.resource.spi.ManagedConnectionFactory` インスタンスの一部です)。

メソッド・バインディングは、メソッドを EIS システムとの特定の対話に関連付けます。J2C では、この対話の特徴は、`javax.resource.cci.InteractionSpec` インターフェース実装のプロパティ・セットによって記述されます。メソッド・バインディングの対話エレメントには、これらのプロパティと共にクラスの名前が含まれています。これにより、対話を実行するための十分な情報を提供します。メソッド・バインディングでは、インターフェース・メソッドの引数と結果の EIS 表現とのマッピングを記述するデータ・バインディングを使用します。

EIS インポートのランタイム・シナリオを以下に示します。インポート・インターフェースのメソッドが、SCA プログラミング・モデルを使用して呼び出されます。EIS インポート・ハンドラーに到達する要求には、メソッドの名前とその引数が含まれています。最初にハンドラーはインターフェース・バインディングの実装を作成し、次にインポート・ファイルのデータを使用して `ConnectionFactory` を作成します。その後、この 2 つを関連付けます。つまり、ハンドラーはインターフェース・バインディングの `setConnectionFactory` を呼び出します。次のステップで、呼び出されたメソッドに対応するメソッド・バインディングの実装を作成します。

`javax.resource.cci.InteractionSpec` インスタンスが作成され、データが取り込まれます。その後データ・バインディングを使用して、メソッド引数をリソース・アダプターが認識する形式にバインドします。ここで、CCI インターフェースを使用して対話を実行されます。呼び出しが戻ったら、データ・バインディングにより、呼び出しの結果が作成され、結果が呼び出し元に戻されます。

アダプターの対話スタイルでは、EIS エクスポート・バインディングで、ターゲット・コンポーネントを非同期または同期式のどちらでも呼び出すことができます。デフォルトは非同期です。

### 関連概念

124 ページの『データ・バインディング』

データ・バインディングは、SCA アプリケーションに渡されるデータとエンタープライズ情報システム (EIS) の間のマッピングを記述します。つまり、EIS 固有の形式への SCA データ表現のバインディングを記述します。

### EIS インポート・バインディングの管理:

EIS インポート・バインディングは、SCA アプリケーションの一部として WebSphere Process Server にインストールされます。管理コンソールからインポートを管理します。

この作業を実行するには、マスター構成に対して変更を行う権限が必要です。

EIS インポート・モジュールを含んだインストール済みのアプリケーションが必要です。

1. 管理の対象とするインポートを表示します。「アプリケーション」を展開して「SCA モジュール」をクリックし、管理する SCA モジュールの名前をクリックします。「モジュール・コンポーネント」の見出しの下で「インポート」を展開して、管理するインポートを展開します。インターフェースまたはバインディングの管理を選択できます。
2. **オプション:** インポート・インターフェースの WSDL を表示します。「インターフェース」を展開して表示するインターフェースを選択します。インターフェースの WSDL が表示されます。WSDL は管理コンソールでは編集できませんが、テキスト・エディターを使用して変更できます。
3. **オプション:** インポート・バインディングを管理します。「バインディング」を展開して、管理するバインディングをクリックします。インポートされたサービスのポートまたは名前を変更できます。
4. 変更を有効にするために、変更内容を構成に保管します。

## EIS エクスポート・バインディングの主な特徴

EIS エクスポートは、Service Component Architecture (SCA) エクスポートです。このエクスポートは、SCA モジュール外部のクライアントからサービスを利用できるようにします。EIS エクスポートは、外部 EIS から SCA モジュールへのデータ転送に使用されます。EIS エクスポートは、EIS からの外部要求を listen するサブスクリプション・サービスと考えることができます。EIS エクスポートを利用する SCA コンポーネントは、EIS エクスポートをローカル・サービスと見なします。

### 目的

EIS エクスポートの役割は、SCA コンポーネントと外部 EIS の間のすき間を埋めることです。外部サービスを EIS エクスポートとして取り扱うことができます。この場合、外部サービスはそのデータを定期的な通知という形で送信します。

EIS エクスポートは、SCA コンポーネントに対して、モジュール外部のサービスの統一されたビューを提供します。これにより、コンポーネントは、SAP、Siebel、PeopleSoft などのさまざまな外部 EIS と、一貫性のある SCA モデルを使用して通信できます。

エクスポートは、EIS から要求を受け取るリスナー実装を特徴としています。リスナーは、JMS エクスポート用の `javax.jms.MessageListener` インターフェースまたはリソース・アダプター固有のリスナー・インターフェースのいずれかを実装します。また、エクスポートには、エクスポートを介して EIS に公開されるインターフェースを実装するコンポーネントも含まれます。

EIS エクスポートのランタイム実装は、インターフェースを実装するコンポーネントにリスナーを接続します。エクスポートにより、EIS 要求が、コンポーネント上の該当する操作の呼び出しにマップされます。リスナー・バインディング、リスナー・バインディングがそれに含まれるネイティブ・メソッド・バインディングを使用する、次にネイティブ・メソッド・バインディングがデータ・バインディングを使用する、という 3 つのレベルでバインディングが作成されます。

リスナー・バインディングは、要求を受け取るリスナーを、エクスポートを介して公開されるコンポーネントに関連付けます。エクスポート定義には、コンポーネントの名前が含まれています。これにより、ランタイムがコンポーネントを検索し、コンポーネントに要求を転送します。

ネイティブ・メソッド・バインディングは、ネイティブ・メソッドまたはリスナーが受け取ったイベント・タイプを、エクスポートを介して公開されたコンポーネントによって実装される操作に関連付けます。リスナーで呼び出されるメソッドとイベント・タイプの間には関係はなく、すべてのイベントはリスナーの 1 つ以上のメソッドを使用して受信されます。ネイティブ・メソッド・バインディングでは、エクスポートで定義された関数セレクターを使用して、インバウンド・データからネイティブ・メソッド名を取り出し、データ・バインディングを使用して、EIS のデータ形式をコンポーネントが認識する形式にバインドします。

EIS エクスポートのランタイム・シナリオを以下に示します。EIS 要求は、リスナー実装のメソッドの呼び出しを起動します。リスナーは、リスナーにすべての呼び出し引数 (例: JMS エクスポートに関する JMS メッセージ) を渡すエクスポート・ハンドラーを検索し、これを呼び出します。ハンドラーは、リスナー・バインディングの実装を作成します。ハンドラーは、次に関数セレクターをインスタンス化し、これをリスナー・バインディング上に設定します。次のステップで、エクスポート・ハンドラーは、ネイティブ・メソッド・バインディングを初期化して、リスナー・バインディングに追加します。ネイティブ・メソッド・バインディングごとに、データ・バインディングも初期化されます。最後に、ハンドラーはリスナー・バインディングを呼び出します。リスナー・バインディングは、エクスポートされたコンポーネントを見つけて、関数セレクターを使用しネイティブ・メソッド名を取得します。この名前を使用して、ネイティブ・メソッド・バインディングを探します。この後、ネイティブ・メソッド・バインディングによってターゲット・コンポーネントが呼び出されます。

#### **EIS エクスポート・バインディングの管理:**

EIS エクスポート・バインディングは、SCA アプリケーションの一部として WebSphere Process Server にインストールされます。管理コンソールからエクスポートを管理します。

この作業を実行するには、マスター構成に対して変更を行う権限が必要です。

EIS エクスポート・モジュールを含んだインストール済みのアプリケーションが必要です。

1. 管理の対象とするエクスポートを表示します。「アプリケーション」を展開して「SCA モジュール」をクリックし、管理する SCA モジュールの名前をクリックします。「モジュール・コンポーネント」の見出しの下で「エクスポート」を展開して、管理するエクスポートを展開します。インターフェースまたはバインディングの管理を選択できます。
2. **オプション:** エクスポート・インターフェースの WSDL を表示します。「インターフェース」を展開して表示するインターフェースを選択します。インターフェースの WSDL が表示されます。WSDL は管理コンソールでは編集できませんが、テキスト・エディターを使用して変更できます。

3. **オプション:** エクスポート・バインディングを管理します。「バインディング」を展開して、管理するバインディングをクリックします。エクスポートされたサービスのポートまたは名前を変更できます。
4. 変更を有効にするために、変更内容を構成に保管します。

## EIS バインディングを持つ外部クライアント

WebSphere Process Server は、EIS バインディングを使用して外部クライアントとの間でメッセージを送受信できます。

外部クライアント (Web ポータルや EIS など) は、WebSphere Process Server の SCA モジュールにメッセージを送信する必要があり、WebSphere Process Server 内からコンポーネントによって呼び出される必要があります。

クライアントは、動的起動インターフェース (DII) または Java インターフェースのいずれかを使用して他のサービスと同様に EIS インポートを呼び出します。

最初に、外部クライアントが ServiceManager のインスタンスを作成し、その参照名を使用して EIS インポートを検索します。検索の結果は Service インターフェースの実装です。次に、クライアントが入力引数 (総称 DataObject) を作成します。この引数は、データ・オブジェクト・スキーマを使用して動的に作成されます。このステップは、サービス・データ・オブジェクト (SDO) の DataFactory インターフェースの実装を使用して行われます。最後のステップで、外部クライアントが EIS を呼び出して必要な結果を取得します。

代わりに、クライアントは Java インターフェースを使用して EIS インポートを呼び出すこともできます。最初に、クライアントは ServiceManager のインスタンスを作成し、その参照名を使用して EIS インポートを検索します。検索の結果は、EIS インポートの Java インターフェースです。次に、クライアントは入力引数 (DataObject 型) を作成し、最後に EIS を呼び出して必要な結果を取得します。

EIS エクスポート・インターフェースは、外部の EIS サービスで使用可能なエクスポート済み SCA コンポーネントのインターフェースを定義します。このインターフェースは、SAP や PeopleSoft などの外部サービスが (EIS エクスポート・サービス・ランタイムの実装を介して) 呼び出すインターフェースと考えることができます。エクスポートは、EISExportBinding を使用してエクスポート済みサービスを外部の EIS サービスにバインドします。これにより、EIS サービス要求を listen するための SCA モジュールに含まれるサービスをサブスクライブできます。EIS エクスポート・バインディングは、リソース・アダプターによって (J2EE コネクター・アーキテクチャー・インターフェースを使用して) 認識されるインバウンド・イベントの定義と SCA 操作の呼び出しの間のマッピングを指定します。

EISExportBinding では、外部の EIS サービスが J2EE コネクター・アーキテクチャー 1.5 のインバウンド規約に基づいている必要があります。EISExportBinding では、DataBinding をバインディング・レベルまたはメソッド・レベルで指定する必要があります。

## JMS バインディング

Java Message Service (JMS) プロバイダーにより、Java Messaging Service に基づいたメッセージングが可能になります。JMS プロバイダーは、JMS 宛先への接続を

作成する J2EE 接続ファクトリーを提供します。JMS バインディングは、EIS インポートおよびエクスポート・バインディングを特殊化したものです。

JMS バインディングを介して提供されるサービスにより、サービス・コンポーネント・アーキテクチャー (SCA) モジュールは、外部システムに対する呼び出しを実行したりメッセージを受信したりできます。これらのシステムは、外部の JMS システムか WebSphere MQ ベースのシステムにすることができます。

JMS サービスは、使用可能な JMS JCA 1.5 ベースのリソース・アダプターを介して統合を提供します。SIB JMS リソース・アダプターは JCA 1.5 ベースのリソース・アダプターの一例であり、このアダプターの JMS 統合のための完全なサポートが提供されます。

代替の JMS プロバイダー JCA 1.5 ベースの JMS リソース・アダプターはサポートされません。

ユーザーは、WebSphere MQ 以外の外部システムと統合する場合に、JCA 1.5 リソース・アダプター用の JMS プロバイダーを使用します。SIB JMS リソース・アダプター外部サービスを使用すると、SCA コンポーネントと統合するためのメッセージを送受信できます。

#### 関連概念

89 ページの『SCA EIS バインディング』

Service Component Architecture (SCA) エンタープライズ情報システム (EIS) のバインディングは、SCA コンポーネントと外部システム間の接続を提供します。この通信は、EIS エクスポートと EIS インポートによりメディエーションされます。

### JMS バインディング: 全体像

JMS バインディングは、サービス・コンポーネント・アーキテクチャー (SCA) 環境と外部の JMS システムの間の接続を提供します。

#### JMS バインディング

JMS インポート・バインディングおよび JMS エクスポート・バインディングの主なコンポーネントは、以下のとおりです。

- リソース・アダプター - エンタープライズ情報システム (EIS) と J2EE コンポーネントの間の管理対象双方向接続を使用可能にします。
- 接続 - クライアントとプロバイダー・サービスの間の仮想接続をカプセル化します。
- 宛先 - 生成するメッセージの宛先またはコンシュームするメッセージの送信元を指定するためにクライアントが使用します。
- 認証データ - バインディングへのアクセスを保護するために使用します。

#### JMS インポート・バインディング

JMS インポート・バインディングは、サービス・コンポーネント・アーキテクチャー (SCA) アプリケーションから JMS プロバイダーへのアウトバウンド接続を提供します。

JMS インポートの一部である接続は接続ファクトリーから構成されます。接続ファクトリーとは、クライアントがプロバイダーへの接続を作成するために使用するオブジェクトです。接続ファクトリーは、管理者によって定義された一連の接続構成パラメーターをカプセル化します。各接続ファクトリーは、`ConnectionFactory`、`QueueConnectionFactory`、または `TopicConnectionFactory` インターフェースのインスタンスです。

JMS インポートの宛先エレメントは、宛先を開始する送信メッセージです。

サポートされる JMS インポート・バインディングは以下の 2 種類です。

- 片方向 - インポート・ファイルで `send` として指定された宛先にメッセージを送信します。JMS ヘッダーの `JMSReplyTo` フィールドには何も送信されません。
- 両方向 (要求応答) - インポート・ファイルで `send` として指定された宛先にメッセージを送信します。`receive` として指定された宛先には `JMSReplyTo` ヘッダー・プロパティーが送信されます。受信宛先で `listen` するためのメッセージ駆動型 Bean (MDB) をデプロイします。応答を受信すると、MDB はコールバック・オブジェクトを呼び出します。要求応答シナリオの場合、応答は送信メッセージの `messageID` に `correlationId` を設定し、デフォルト相関スキームを示す必要があります。

片方向と両方向のいずれを使用する場合も、動的および静的ヘッダー・プロパティーを指定できます。これらのプロパティーの一部は、SCA JMS ランタイムで特別な意味を持ちます。

## JMS エクスポート・バインディング

JMS エクスポート・バインディングは、JMS から SCA システムへのインバウンド接続を提供します。

JMS エクスポートの一部である接続はアクティベーション・スペックです。

JMS エクスポートには、送信、受信、およびコールバック宛先があります。受信宛先は、ターゲット・コンポーネントに対する着信メッセージを格納する宛先です。送信宛先は、着信メッセージが `JMSReplyTo` ヘッダー・プロパティーを使用してこの宛先を置き換えていない場合に応答を送信する宛先です。

エクスポート・バインディングで指定された `receive` 宛先に着信する要求を `listen` するため、MDB がデプロイされます。`send` フィールドで指定された宛先は、呼び出されたサービスが応答を返す場合にインバウンド要求に対する応答を送信するために使用されます。着信メッセージの `JMSReplyTo` フィールドで指定された宛先は、`send` で指定された宛先を上書きします。エクスポートの `callback` フィールドで指定された宛先は、コールバック情報の格納に使用されます。これは、JMS サービスのインターフェースで両方向 (要求応答) 操作を行うために必要です。

## JMS メッセージ配信と Failed Event Manager

JMS バインディングは、デフォルトで最大配信失敗回数を 5 として構成されます。JMS 宛先の基盤となる SIB 宛先のために、JMS バインディングの最大配信失敗回数の値は 2 以上でなければなりません。これは、インポートとエクスポートの両方の受信宛先で Failed Event Manager が失敗したメッセージを格納できるように

するために必要です。最大配信失敗回数の値を 2 未満に設定すると、失敗したターゲット・コンポーネントに対するメッセージが Failed Event Manager で使用できなくなります。ターゲット・コンポーネントでの失敗により、ランタイム例外が発生します。Failed Event Manager に到達しなかった失敗したメッセージは、基盤となる SIB 宛先によって指定される例外宛先に転送されます。

## JMS バインディングのサポート

Java Message Service (JMS) は、使用可能な JMS JCA 1.5 ベースのリソース・アダプターを介して統合を提供します。JMS 統合のための完全なサポートは、サービス統合バス (SIB) JMS リソース・アダプターに対して用意されています。

JCA 1.5 リソース・アダプター用の JMS プロバイダーは、WebSphere MQ 以外の外部システムと統合する場合に使用します。外部サービスは、SIB JMS リソース・アダプターを使用してサービス・コンポーネント・アーキテクチャー (SCA) コンポーネントと統合するためのメッセージを送受信できます。

他のプロバイダー固有の JCA 1.5 リソース・アダプターの使用はサポートされません。

JMS モジュールは J2SE 環境にデプロイできません。それらのモジュールは J2EE 環境にのみデプロイできます。

### 関連概念

115 ページの『MQ JMS バインディングのサポート』

WebSphere MQ JMS サービスは、WebSphere MQ ベースの JMS プロバイダーとの統合を提供します。

## JMS インポート・バインディングの主な特徴

JMS インポートには、いくつかの主な特徴があります。ヘッダーと作成される J2EE リソースが主な特徴です。

JMS インポート・サービスは、応答の処理に関して、ランタイム呼び出しの点で EIS インポートとは異なります。JMS の実装は非同期ですが、SCA プログラミング・モデルでは、JMS 実装が 3 つの呼び出し (片方向、両方向 (要求/応答とも呼ばれる))、およびコールバックをサポートすることが求められます。

JMS インポートがデプロイされる時、ランタイム環境によって提供されるメッセージ駆動型 Bean がデプロイされます。MDB は要求メッセージに対する応答を listen します。MDB は、要求と共に送信された宛先に関連付けられています (この宛先を listen します)。この宛先は、JMS メッセージの JMSReplyTo ヘッダー・フィールドで指定されます。

### 特殊ヘッダー

インポートにおいて、ターゲット・エクスポート・バインディングに対してメッセージの処理方法に関する情報を指示するために、特殊ヘッダー・プロパティーが使用されます。

### パラメーター

ネイティブ・メソッドから操作メソッドへのマッピングが、特殊ヘッダーによって提供されます。

#### **TargetFunctionName**

このヘッダーは、ネイティブ・メソッドから操作メソッドへのマッピングを行うために、エクスポート・バインディングで使用されます。String の JMS ヘッダー・プロパティ TargetFunctionName を設定すると、エクスポートの JMSFunctionSelectorImpl に対して、操作メソッドへのマッピングに使用するネイティブ・メソッドが指示されます。これを使用するためには、ネイティブ・メソッドを値と同様にメソッド・バインディング内で指定します。エクスポートで JMSFunctionSelectorImpl を使用させたい場合は、このヘッダーをインポート・バインディング定義内で JMS ヘッダー・プロパティとして設定する必要があります。

## **J2EE リソース**

JMS インポートを J2EE 環境にデプロイすると、いくつかの J2EE リソースが作成されます。

#### **パラメーター**

##### **ConnectionFactory**

クライアントが JMS プロバイダーとの接続を作成するために使用します。

##### **ActivationSpec**

要求に対する応答を受け取るために使用します。

##### **Destinations**

- 送信宛先 - 要求または出力メッセージが送信される宛先です。
- 受信宛先 - 応答または着信メッセージが格納される宛先です。
- コールバック宛先 - 関連情報に使用される SCA JMS システム宛先です。この宛先に対して、読み取りまたは書き込みを行わないでください。

インストール・タスクでは、インポート・ファイルの情報から ConnectionFactory および 3 つの宛先を作成します。さらに、ActivationSpec を作成して、ランタイム MDB が受信宛先上の応答を listen できるようにします。ActivationSpec のプロパティは、destination/ConnectionFactory プロパティから派生します。

#### **JMS インポート・バインディングの構成:**

JMS インポート・バインディングは、作成されるすべての必要な接続ファクトリーと共にインストールできます。あるいは、サーバー上で事前に構成された成果物セットを指すように設計できます。

通常、JMS インポート・バインディングは、WebSphere Integration Developer で作成されます。開発時に選択できる方法として、インストール時 (すなわちコンポーネントが WebSphere Process Server にインストールされる時) に、JMS インポートに必要な接続と宛先を作成できます。もう 1 つの方法として、JMS インポートで使用させる予定の WebSphere Process Server 上のリソースの JNDI 名を指定する方法があります。



このどちらの方法を選択するかによって、JMS インポート・バインディングの構成が異なります。

新しいメッセージ・プロバイダー・リソースが作成される場合（つまり、インストール時にサーバー上にリソースが作成される場合）は、リソースが存在するため、管理コンソールを使用してその位置を見つけ、管理できます。生成される成果物の JNDI 名を以下の表に示します。

表 10. JMS インポート・バインディングの WebSphere Process Server へのインストール時に作成されるリソースの名前および JNDI 名

リソース	モジュール名	インポート名	リソースのグローバル JNDI 名
接続ファクトリー	jms.module	my/import	jms.module/my/import_CF
アクティベーション・スベック	jms.module	my/import	jms.module/my/import_AS
宛先	jms.module	my/import	jms.module/my/import_SEND_D、 jms.module/my/import_RECEIVE_D、 jms.module/my/import_CALLBACK_D
SIB 宛先	jms.module	my/import	jms.module.my.import_SEND_D_SIB、 jms.module.my.import_RECEIVE_D_SIB、 jms.module.my.import_CALLBACK_D_SIB

注：リソースは、サーバー・スコープで作成されます。管理コンソールでのデフォルトのスコープはノードです。リソースを見つけ、管理するためには、スコープを変更する必要があります。

もう 1 つの方法を使用する場合、JMS インポートがサーバー上で使用するリソースを検出することになるため、このようなリソースをインストールしておくこと、およびインポート・ファイルにその JNDI 名が指定されていることが必要です。これにより、JMS インポートとリソースの間の関連付けが行われます。

### JMS インポート・バインディングの管理:

JMS インポート・バインディングを管理して、リソースを調整したり、リソースの特殊機能を設定したりすることができます。管理用タスクは、WebSphere 管理コンソールを使用して実行します。

管理コンソールでプロファイルに対して変更を行い、変更内容を保管する権限を持っている必要があります。

JMS インポートは、WebSphere Process Server プロファイル内の Service Component Architecture (SCA) アプリケーションの一部としてインストールする必要があります。

1. 管理コンソールのデフォルトのメッセージング・プロバイダー設定パネルを開きます。

「JMS プロバイダー」を展開して、「デフォルトのメッセージング」をクリックします（「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」）。

2. 有効範囲を「サーバー」に設定します。 JMS インポートで使用するリソースが、サーバー・レベルで作成されます。これらのリソースを表示し、管理するためには「有効範囲」を「サーバー」に設定します。

**注:** JMS インポート用のリソースを作成していて、そのリソースを異なる有効範囲レベルで作成している可能性がある場合は、有効範囲をそれに合わせて設定します。

3. **オプション:** JMS 接続ファクトリーを管理します (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS 接続ファクトリー」)。

接続ファクトリーのリストにある JMS 接続ファクトリーをクリックします。このパネルには、JMS 接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする JMS 接続ファクトリーをクリックするか、「新規」をクリックして新しい接続ファクトリーを作成します。

続いて表示されるパネルを使用して、デフォルトのメッセージング JMS プロバイダーで使用する、選択した JMS 接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS キューおよびトピックへの接続を作成する方法を制御します。

アプリケーションのリソース参照のために、このプロパティをバインディングに設定します。既存のアプリケーション用のバインディングを変更しない場合は、これらのプロパティを見つけることができる J2C パネルでこの接続ファクトリーを探します。

JMS 接続ファクトリーは、Point-to-Point メッセージングとパブリッシュ/サブスクライブ・メッセージングの両方において、対応する JMS 宛先の JMS プロバイダーとの接続を作成するために使用します。デフォルト・メッセージング・プロバイダーの JMS 接続ファクトリーを管理するには、接続ファクトリーの管理オブジェクトを使用します。

4. **オプション:** JMS キュー接続ファクトリーを管理します (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS キュー接続ファクトリー」)。

接続ファクトリーのリストにある JMS キュー接続ファクトリーをクリックします。このパネルには、JMS キュー接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。キュー接続ファクトリーを管理するには、そのキュー接続ファクトリーをクリックするか、「新規」をクリックして新しいキュー接続ファクトリーを作成します。

続いて表示されるパネルを使用して、デフォルトのメッセージング JMS プロバイダーで使用する、選択した JMS キュー接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS キューへの接続を作成する方法を制御します。

アプリケーションのリソース参照のために、このプロパティをバインディングに設定します。既存のアプリケーション用のバインディングを変更しない場合は、これらのプロパティを見つけることができる J2C パネルでこの接続ファクトリーを探します。

JMS キュー接続ファクトリーは、Point-to-Point メッセージングにおいて、JMS キューの関連する JMS プロバイダーへの接続を作成する場合に使用されます。デフォルト・メッセージング・プロバイダーの JMS キュー接続ファクトリーを管理するには、キュー接続ファクトリーの管理オブジェクトを使用します。

5. **オプション:** JMS トピック接続ファクトリーを管理します (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS トピック接続ファクトリー (JMS topic connection factory)」)。

接続ファクトリーのリストにある JMS トピック接続ファクトリーをクリックします。このパネルには、JMS トピック接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする JMS トピック接続ファクトリーをクリックするか、「新規」をクリックして新しい接続ファクトリーを作成します。

続いて表示されるパネルを使用して、デフォルトのメッセージング JMS プロバイダーで使用する、選択した JMS トピック接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS トピックへの接続を作成する方法を制御します。

アプリケーションのリソース参照のために、このプロパティをバインディングに設定します。既存のアプリケーション用のバインディングを変更しない場合は、これらのプロパティを見つけることができる J2C パネルでこの接続ファクトリーを探します。

JMS トピック接続ファクトリーは、パブリッシュ/サブスクライブ・メッセージングにおいて、JMS トピックの関連 JMS プロバイダーへの接続を作成する場合に使用されます。デフォルト・メッセージング・プロバイダーの JMS トピック接続ファクトリーを管理するには、トピック接続ファクトリーの管理オブジェクトを使用します。

6. **オプション:** JMS キューを管理します (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS キュー」)。

宛先のリストにある JMS キューをクリックします。このパネルには、JMS キューのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする JMS キューをクリックするか、「新規」をクリックして新しい JMS キューを作成します。

続いて表示されるパネルを使用して、JMS キューの設定を変更または入力します。

JMS キューは、Point-to-Point メッセージングの宛先として使用されます。デフォルト・メッセージング・プロバイダーの JMS キューを管理するには、JMS キュー宛先の管理オブジェクトを使用します。

7. **オプション:** JMS トピックを管理します (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS トピック」)。

宛先のリストにある JMS トピックをクリックします。このパネルには、JMS トピックのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする JMS トピックをクリックするか、「新規」をクリックして新しい JMS トピックを作成します。

続いて表示されるパネルを使用して、JMS トピックの設定を変更または入力します。

JMS トピックは、パブリッシュ/サブスクライブ・メッセージングの宛先として使用されます。デフォルト・メッセージング・プロバイダーの JMS トピックを管理するには、トピック宛先の管理オブジェクトを使用します。

8. **オプション: JMS アクティベーション・スペックを管理します** (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS アクティベーション・スペック (JMS Activation specification)」)。

アクティベーション・スペックのリストにある JMS アクティベーション・スペックをクリックします。パネルに JMS アクティベーション・スペックのリストが、その構成プロパティの要約と共に表示されます。

JMS アクティベーション・スペックは、1 つ以上のメッセージ駆動型 Bean (MDB) に関連付けられており、これらの Bean がメッセージを受信するのに必要な構成を提供します。JMS アクティベーション・スペックは、JMS エンドポイント (メッセージ駆動型 Bean アプリケーション) がデプロイされ、デフォルトのメッセージング・プロバイダーに対してアクティブ化されたときに構成情報を提供するために使用されます。

管理の対象とする JMS アクティベーション・スペックをクリックするか、「新規」をクリックして新しい JMS アクティベーション・スペックを作成します。

変更内容をマスター・プロファイルに保管し、必要に応じて WebSphere Process Server を再始動します。

## JMS エクスポート・バインディングの主な特徴

JMS エクスポート・バインディングには、いくつかの主な特徴があります。ヘッダーと作成される J2EE リソースが主な特徴です。

JMS エクスポート・バインディングが EIS エクスポート・バインディングと異なる点は、結果の戻りの取り扱いです。JMS エクスポートでは、着信メッセージに指定された JMSReplyTo 宛先に応答を明示的に送信します。この宛先が指定されていない場合は、送信宛先が使用されます。

JMS エクスポートがデプロイされる時、メッセージ駆動型 Bean (JMS インポートに使用されるものとは異なる MDB) がデプロイされます。この MDB は受信宛先で着信要求を listen し、次に、その要求を SCA ランタイムで処理するようにディスパッチします。

### 特殊ヘッダー

エクスポートにおいて、ターゲット・エクスポート・バインディングに対してメッセージの処理方法に関する情報を指示するために、特殊ヘッダー・プロパティが使用されます。

### パラメーター

特殊ヘッダーは、エクスポートにメッセージの処理方法を通知するために、エクスポート・バインディングに指定します。

### **TargetFunctionName**

このヘッダーは、ネイティブ・メソッドから操作メソッドへのマッピングを行うために、エクスポート・バインディングで使用されます。デフォルトの関数セレクター (JMSFunctionSelectorImpl) が、JMS メッセージからネイティブ・メソッド名を取り出す目的で使用します。

## **J2EE リソース**

JMS エクスポートを J2EE 環境にデプロイすると、いくつかの J2EE リソースが作成されます。

### **パラメーター**

#### **ConnectionFactory**

クライアントが JMS プロバイダーとの接続を作成するために使用します。

#### **ActivationSpec**

メッセージング・システムとの対話において MDB を表すメッセージ・エンドポイントを構成します。

### **宛先**

- 受信宛先 - 着信メッセージまたは要求メッセージが格納される宛先です。
- 送信宛先 - 着信メッセージの JMSReplyTo ヘッダーで置き換えられない場合に、応答メッセージの送信先となる宛先です。
- コールバック宛先 - 関連情報の保管に使用される SCA JMS システム宛先です。この宛先に対して、読み取りまたは書き込みを行わないでください。

インストール・タスクにより、JMS インポート用のリソース・セットと同様のリソース・セット、すなわち ActivationSpec、応答を送信するために使用される ConnectionFactory、および送信、受信、コールバックの 3 つの宛先が作成されます。これらのリソースのすべてのプロパティが、エクスポート・ファイルで指定されます。

### **JMS エクスポート・バインディングの構成:**

JMS エクスポート・バインディングは、作成されるすべての必要な接続ファクトリーと共にインストールできます。あるいは、サーバー上で事前に構成された成果物セットを指すように設計できます。

通常、JMS エクスポート・バインディングは、WebSphere Integration Developer で作成されます。開発時に選択できる方法として、インストール時 (すなわちコンポーネントが WebSphere Process Server にインストールされる時) に、JMS エクスポートに必要な接続と宛先を作成できます。もう 1 つの方法として、JMS エクスポートで使用させる予定の WebSphere Process Server 上のリソースの JNDI 名を指定する方法があります。

このどちらの方法を選択するかによって、JMS エクスポート・バインディングの構成が異なります。

新しいメッセージ・プロバイダー・リソースが作成される場合 (つまり、インストール時にサーバー上にリソースが作成される場合) は、リソースが存在するため、管理コンソールを使用してその位置を見つけ、管理できます。生成される成果物の

JNDI 名を以下の表に示します。

表 11. JMS エクスポート・バインディングの WebSphere Process Server へのインストール時に作成されるリソースの名前および JNDI 名

リソース	モジュール名	エクスポート名	リソースのグローバル JNDI 名
接続ファクトリー	jms.module	my/export	jms.module/my/export_CF
アクティベーション・パック	jms.module	my/export	jms.module/my/export_AS
宛先	jms.module	my/export	jms.module/my/export_SEND_D、 jms.module/my/export_RECEIVE_D、 jms.module/my/export_CALLBACK_D
SIB 宛先	jms.module	my/export	jms.module.my.export_SEND_D_SIB、 jms.module.my.export_RECEIVE_D_SIB、 jms.module.my.export_CALLBACK_D_SIB

注: リソースは、サーバー・スコープで作成されます。管理コンソールでのデフォルトのスコープはノードです。リソースを見つけ、管理するためには、スコープを変更する必要があります。

もう 1 つの方法を使用する場合、JMS エクスポートがサーバー上で使用するリソースを検出することになるため、このようなリソースをインストールしておくこと、およびエクスポート・ファイルにその JNDI 名が指定されていることが必要です。これにより、JMS エクスポートとリソースの間の関連付けが行われます。

#### 関連タスク

110 ページの『外部クライアントの使用』

外部クライアント (すなわち、WebSphere Process Server の外側) が、WebSphere Process Server 内にインストールされたアプリケーションと対話する必要がある場合があります。

129 ページの『JMS エクスポート・バインディングでのイベントの順序付けの有効化』

JMS エクスポート・バインディングは、イベント順序付けをサポートしています。イベントを受信順に処理するためには、バインディングのプロパティを構成する必要があります。

#### JMS エクスポート・バインディングの管理:

JMS エクスポート・バインディングを管理して、リソースを調整したり、リソースの特殊機能を設定したりすることができます。管理用タスクは、WebSphere 管理コンソールを使用して実行します。

管理コンソールでプロファイルに対して変更を行い、変更内容を保管する権限を持っている必要があります。

JMS エクスポートは、WebSphere Process Server プロファイル内の Service Component Architecture (SCA) アプリケーションの一部としてインストールする必要があります。

1. 管理コンソールのデフォルトのメッセージング・プロバイダー設定パネルを開きます。

「JMS プロバイダー」を展開して、「デフォルトのメッセージング」をクリックします（「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」）。

2. 有効範囲を「サーバー」に設定します。JMS エクスポートで使用するリソースが、サーバー・レベルで作成されます。これらのリソースを表示し、管理するためには「有効範囲」を「サーバー」に設定します。

**注:** JMS エクスポート用のリソースを作成していて、そのリソースを異なる有効範囲レベルで作成している可能性がある場合は、有効範囲をそれに合わせて設定します。

3. **オプション:** JMS 接続ファクトリーを管理します（「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS 接続ファクトリー」）。

接続ファクトリーのリストにある JMS 接続ファクトリーをクリックします。このパネルには、JMS 接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする JMS 接続ファクトリーをクリックするか、「新規」をクリックして新しい接続ファクトリーを作成します。

続いて表示されるパネルを使用して、デフォルトのメッセージング JMS プロバイダーで使用する、選択した JMS 接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS キューおよびトピックへの接続を作成する方法を制御します。

アプリケーションのリソース参照のために、このプロパティをバインディングに設定します。既存のアプリケーション用のバインディングを変更しない場合は、これらのプロパティを見つけることができる J2C パネルでこの接続ファクトリーを探します。

JMS 接続ファクトリーは、Point-to-Point メッセージングとパブリッシュ/サブスクライブ・メッセージングの両方において、対応する JMS 宛先の JMS プロバイダーとの接続を作成するために使用します。デフォルト・メッセージング・プロバイダーの JMS 接続ファクトリーを管理するには、接続ファクトリーの管理オブジェクトを使用します。

4. **オプション:** JMS キュー接続ファクトリーを管理します（「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS キュー接続ファクトリー」）。

接続ファクトリーのリストにある JMS キュー接続ファクトリーをクリックします。このパネルには、JMS キュー接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。キュー接続ファクトリーを管理するには、そのキュー接続ファクトリーをクリックするか、「新規」をクリックして新しいキュー接続ファクトリーを作成します。

続いて表示されるパネルを使用して、デフォルトのメッセージング JMS プロバイダーで使用する、選択した JMS キュー接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS キューへの接続を作成する方法を制御します。

アプリケーションのリソース参照のために、このプロパティをバインディングに設定します。既存のアプリケーション用のバインディングを変更しない場合は、これらのプロパティを見つけることができる J2C パネルでこの接続ファクトリーを探します。

JMS キュー接続ファクトリーは、Point-to-Point メッセージングにおいて、JMS キューの関連する JMS プロバイダーへの接続を作成する場合に使用されます。デフォルト・メッセージング・プロバイダーの JMS キュー接続ファクトリーを管理するには、キュー接続ファクトリーの管理オブジェクトを使用します。

5. **オプション: JMS トピック接続ファクトリーを管理します** (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS トピック接続ファクトリー (JMS topic connection factory)」)。

接続ファクトリーのリストにある JMS トピック接続ファクトリーをクリックします。このパネルには、JMS トピック接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする JMS トピック接続ファクトリーをクリックするか、「新規」をクリックして新しい接続ファクトリーを作成します。

続いて表示されるパネルを使用して、デフォルトのメッセージング JMS プロバイダーで使用する、選択した JMS トピック接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS トピックへの接続を作成する方法を制御します。

アプリケーションのリソース参照のために、このプロパティをバインディングに設定します。既存のアプリケーション用のバインディングを変更しない場合は、これらのプロパティを見つけることができる J2C パネルでこの接続ファクトリーを探します。

JMS トピック接続ファクトリーは、パブリッシュ/サブスクライブ・メッセージングにおいて、JMS トピックの関連 JMS プロバイダーへの接続を作成する場合に使用されます。デフォルト・メッセージング・プロバイダーの JMS トピック接続ファクトリーを管理するには、トピック接続ファクトリーの管理オブジェクトを使用します。

6. **オプション: JMS キューを管理します** (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS キュー」)。

宛先のリストにある JMS キューをクリックします。このパネルには、JMS キューのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする JMS キューをクリックするか、「新規」をクリックして新しい JMS キューを作成します。

続いて表示されるパネルを使用して、JMS キューの設定を変更または入力します。



JMS キューは、Point-to-Point メッセージングの宛先として使用されます。デフォルト・メッセージング・プロバイダーの JMS キューを管理するには、JMS キュー宛先の管理オブジェクトを使用します。

7. **オプション: JMS トピックを管理します** (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS トピック」)。

宛先のリストにある JMS トピックをクリックします。このパネルには、JMS トピックのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする JMS トピックをクリックするか、「新規」をクリックして新しい JMS トピックを作成します。

続いて表示されるパネルを使用して、JMS トピックの設定を変更または入力します。

JMS トピックは、パブリッシュ/サブスクライブ・メッセージングの宛先として使用されます。デフォルト・メッセージング・プロバイダーの JMS トピックを管理するには、トピック宛先の管理オブジェクトを使用します。

8. **オプション: JMS アクティベーション・スペックを管理します** (「リソース」 > 「JMS プロバイダー」 > 「デフォルトのメッセージング」 > 「JMS アクティベーション・スペック (JMS Activation specification)」)。

アクティベーション・スペックのリストにある JMS アクティベーション・スペックをクリックします。パネルに JMS アクティベーション・スペックのリストが、その構成プロパティの要約と共に表示されます。

JMS アクティベーション・スペックは、1 つ以上のメッセージ駆動型 Bean (MDB) に関連付けられており、これらの Bean がメッセージを受信するのに必要な構成を提供します。JMS アクティベーション・スペックは、JMS エンドポイント (メッセージ駆動型 Bean アプリケーション) がデプロイされ、デフォルトのメッセージング・プロバイダーに対してアクティブ化されたときに構成情報を提供するために使用されます。

管理の対象とする JMS アクティベーション・スペックをクリックするか、「新規」をクリックして新しい JMS アクティベーション・スペックを作成します。

変更内容をマスター・プロファイルに保管し、必要に応じて WebSphere Process Server を再始動します。

## 外部クライアント

WebSphere Process Server は、JMS バインディングを使用して外部クライアントとの間でメッセージを送受信できます。

外部クライアント (Web ポータルや EIS など) は、WebSphere Process Server の SCA モジュールにメッセージを送信する必要があり、WebSphere Process Server 内からコンポーネントによって呼び出される必要があります。

JMS エクスポート・コンポーネントは、エクスポート・バインディングで指定された receive 宛先に着信する要求を listen するためのメッセージ駆動型 Bean (MDB) をデプロイします。send フィールドで指定された宛先は、呼び出されたサービスが

応答を返す場合にインバウンド要求に対する応答を送信するために使用されます。したがって、外部クライアントは、エクスポート・バインディングを介してアプリケーションを呼び出すことができます。

JMS インポートは外部クライアントにバインドし、外部クライアントにメッセージを配信できます。このメッセージは、外部クライアントからの応答を要求してもしなくても構いません。

#### 外部クライアントの使用:

外部クライアント (すなわち、WebSphere Process Server の外側) が、WebSphere Process Server 内にインストールされたアプリケーションと対話する必要がある場合があります。

外部クライアントの処理を説明するために、外部クライアントが WebSphere Process Server 上の一般アプリケーションとの対話を求める、非常に単純なシナリオを考えてみます。以下の図は、標準的な単純なシナリオを表しています。

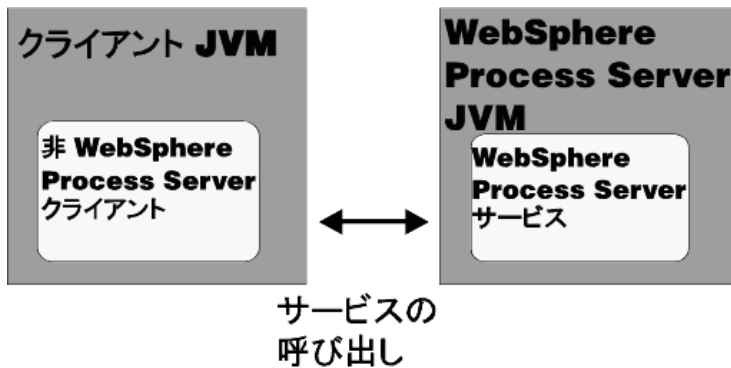


図7. 単純なユース・ケースのシナリオ。外部クライアントが WebSphere Process Server アプリケーションと対話する。

サービスは、JMS バインディングを使用してエクスポートされ、これによって外部クライアントがサービスを使用できるようになります。

外部クライアントが WebSphere Process Server とは別個の Java 仮想マシン (JVM) 内にある場合、JMS エクスポートとの接続および対話を作成するためにいくつかのステップを実行する必要があります。クライアントは、正しい値が指定された InitialContext を取得して、JNDI を使用してリソースを検索します。次にクライアントは、JMS 1.1 仕様のクライアントを使用して宛先にアクセスし、宛先の送信メッセージおよび受信メッセージにアクセスします。ランタイムにより自動作成されるリソースのデフォルトの JNDI 名は、テーブル xyz にリストされます。事前に作成済みのリソースがある場合は、その JNDI 名を使用します。

1. メッセージを送信するための JMS の宛先と接続ファクトリーを構成する必要があります。
2. SIB リソース・アダプター用の正しい JNDI コンテキストとポート、およびメッセージング・ブートストラッピング・ポートが必要です。

WebSphere Process Server はいくつかのデフォルト・ポートを使用しますが、それより多くのサーバーがそのマシンにインストールされている場合は、他のサーバー・インスタンスとの競合を避けるために、インストール時に代替りのポートが作成されます。管理コンソールを使用して、WebSphere Process Server が使用するポートを調べることができます。「サーバー」 > 「アプリケーション・サーバー」 > サーバーの名前 > 「構成」と進んで、「通信 (Communication)」の下で「ポート」をクリックします。「ポート」または「詳細」をクリックして、使用中のポートを編集できます。

3. クライアントは、正しい値が指定された初期コンテキストを取得し、JNDI を使用してリソースを検索します。
4. クライアントは、JMS 1.1 仕様を使用して宛先にアクセスし、宛先で送信メッセージおよび受信メッセージにアクセスします。

ランタイムにより自動作成されるリソースのデフォルトの JNDI 名については、関連トピックで説明します。

事前に作成済みのリソースがある場合は、作成したリソースの JNDI 名を使用します。

#### 例: 注文処理アプリケーションの要求/応答操作

要求/応答操作のクライアント・ソース・コードを以下に示します。

```
ConnectionFactory conFact;
Connection conn;
Session sess;
Destination requestQueue, responseQueue;
MessageProducer requestProducer;
MessageConsumer replyConsumer;
String jmsRequestQName = "jms/ExportIn";
String jmsReplyQName = "jms/ExportOut";
String jmsFactoryName = "jms/MyCF";
// JNDI 初期コンテキストを取得するために使用する初期コンテキスト・ファクトリー
String jndiFactoryName = "com.ibm.websphere.naming.WsnInitialContextFactory";
// ブートストラッピング用の URL。WPS ホスト・マシンおよび
// ブートストラッピング・ポートの IP アドレスを含みます。デフォルトは 2809 で、
// これを host:port の値に変更する必要があります。
String jmsContextURL = "iiop://9.26.237.246:2809/";
TextMessage jmsRequestMessage, jmsReplyMessage;
// これは直列化されたビジネス・オブジェクト「order」です。
// このサンプルに対する直列化を必要以上に複雑にしないため、
// 既に String として直列化させています。
String message = " " String message = "<order:Order xsi:type=%>order:Order%>"
xmlns:xsi=%>http://www.w3.org/2001/XMLSchema-instance%>"
xmlns:order=%>http://OrderProcessing%>"
+ "<orderId>123</orderId>"
+ "<customerId>456</customerId>"
+ "<orderDate>2002-01-01Z</orderDate>"
+ "<orderStatus>Reject</orderStatus>"
+ "<totalPrice>1000.0</totalPrice>"
+ "</order:Order>";

System.out.println("JMS Factory name:" + jmsFactoryName);
System.out.println("Queue Factory name:" + jmsRequestQName);
System.out.println("Initial Context Provider URL:" + jmsContextURL);
System.out.println();

try {
    Hashtable jndiHash = new Hashtable();
    jndiHash.put(javax.naming.Context.INITIAL_CONTEXT_FACTORY,
```

```

jndiFactoryName);
jndiHash.put(javax.naming.Context.PROVIDER_URL, jmsContextURL);
javax.naming.Context ctx = new InitialContext(jndiHash);
System.out.println("Got Initial Context from provider: " +
jmsContextURL);

conFact = (ConnectionFactory) ctx.lookup(jmsFactoryName);
if (conFact != null) {
    System.out.println("Got Connection Factory: " + jmsFactoryName);
    System.out.println(conFact.toString());
}

conn = conFact.createConnection();
conn.start();
System.out.println("Created Queue Connection...");
System.out.println(conn.toString());

sess = conn.createSession(true, Session.AUTO_ACKNOWLEDGE);
System.out.println("created Session...");
System.out.println(sess.toString());
System.out.println(conn.toString());

requestQueue = (Queue) ctx.lookup(jmsRequestQName);
if (requestQueue != null) {
System.out.println("Found JMS request Destination " + jmsRequestQName);
    System.out.println(requestQueue.toString());
}
responseQueue = (Queue) ctx.lookup(jmsReplyQName);
if (responseQueue != null) {
    System.out.println("Found JMS reply-to Queue " + jmsReplyQName);
}

requestProducer = sess.createProducer(requestQueue);
System.out.println("Created Sender...");

jmsRequestMessage = sess.createTextMessage();
System.out.println("Creating Message...");
jmsRequestMessage.setText(message);
jmsRequestMessage.setJMSReplyTo(responseQueue);
jmsRequestMessage.setStringProperty("TargetFunctionName",
"submitOrderRequestResponse");

requestProducer.send(jmsRequestMessage);
sess.commit();

System.out.println("MESSAGE SENT!");
System.out.println();
System.out.println("JMS Correlation ID: ");
String corrID = jmsRequestMessage.getJMSCorrelationID();
if (corrID != null) {
    System.out.println(jmsRequestMessage.getJMSCorrelationID());
}
else {
    System.out.println("NULL");
}
System.out.println();
System.out.println("JMS Message: ");
System.out.println(jmsRequestMessage.toString());

System.out.println();
System.out.println("WAITING FOR THE RESPONSE...");
System.out.println();

responseQueue = (Queue) ctx.lookup(jmsReplyQName);
if (responseQueue != null) {
    System.out.println("Found JMS response Destination " +
jmsReplyQName);
}

```

```

    System.out.println(responseQueue.toString());
}
else {
    System.out.println("JMS response Destination not found: " +
jmsReplyQName);
}

replyConsumer = sess.createConsumer(responseQueue);
jmsReplyMessage= (TextMessage)replyConsumer.receive(10000);
if (jmsReplyMessage != null) {
    System.out.println("MESSAGE RECEIVED!!");
    System.out.println();
    System.out.println("JMS Message: ");
    System.out.println(jmsRequestMessage.toString());
}
else {
    System.out.println("DIDN'T FIND MESSAGE IN THE REMOTE QUEUE: " +
jmsReplyQName);
}
sess.commit();
requestProducer.close();
replyConsumer.close();
sess.close();
conn.close();

} catch (Exception j) {
    System.out.println("There was an exception, (for whole stack trace look
in SystemErr.log): " + j.getMessage());
    j.printStackTrace();
}
}

```

### 関連概念

105 ページの『JMS エクスポート・バインディングの構成』  
**JMS** エクスポート・バインディングは、作成されるすべての必要な接続ファク  
トリーと共にインストールできます。あるいは、サーバー上で事前に構成された  
成果物セットを指すように設計できます。

## WebSphere MQ JMS バインディング

WebSphere MQ JMS サービスは、WebSphere MQ ベースの JMS プロバイダーとの  
統合を提供します。

WebSphere MQ JMS プロバイダー・サービスは、WebSphere Process Server 環境か  
ら WebSphere MQ ベースのシステムと直接統合する場合に使用します。これによ  
り、サービス統合バスの MQ リンクまたはクライアント・リンク機能を使用する必  
要がなくなります。

WebSphere MQ からサービスをインポートするときには、データの送信先の宛先  
と、応答を受信できる宛先を使用します。データと JMS メッセージの間の相互変  
換は、JMS データ・バインディング・エッジ・コンポーネントを介して行います。

WebSphere MQ にサービスをエクスポートするときには、要求を受信でき、応答を  
送信できる宛先を使用します。データと JMS メッセージの間の相互変換は、JMS  
データ・バインディングを介して行います。関数セレクターは、呼び出すターゲッ  
ト・コンポーネントに対する操作へのマッピングを提供する役割を果たします。

## WebSphere MQ JMS バインディング: 全体像

WebSphere MQ JMS サービスは、WebSphere MQ ベースの JMS プロバイダーとの統合を提供します。

### WebSphere MQ 管理用タスク

WebSphere MQ 管理者は WebSphere MQ に基盤となるキュー・マネージャーおよびキューを作成する必要があります。これは、WebSphere MQ JMS バインディングが含まれるアプリケーションを実行する前に、これらのバインディングによって使用されます。

### WebSphere MQ JMS インポート・バインディング

WebSphere MQ JMS インポート・バインディングは、サービス・コンポーネント・アーキテクチャー (SCA) アプリケーションから WebSphere MQ ベースの JMS プロバイダーへのアウトバウンド接続を提供します。サポートされるバージョンの WebSphere MQ を使用していることを確認してください。詳細なハードウェアおよびソフトウェア要件については、IBM サポート・ページを参照してください。

以下の 2 種類の WebSphere MQ JMS インポート・バインディングがサポートされます。

- 片方向 - インポート・ファイルで *send* として指定された宛先にメッセージを送信します。JMS ヘッダーの *replyTo* フィールドには何も送信しません。
- 両方向 (要求応答) - インポート・ファイルで *send* として指定された宛先にメッセージを送信します。*receive* として指定された宛先は JMSReplyTo ヘッダー・フィールドで送信します。*receive* 宛先で *listen* するためのメッセージ駆動型 Bean (MDB) をデプロイします。応答を受信すると、MDB がコールバック・オブジェクトを呼び出します。要求応答シナリオの場合、応答は送信メッセージの *JMSMessageID* に *JMSCorrelationId* を設定し、デフォルト相関スキームを示す必要があります。

片方向と両方向のいずれを使用する場合も、動的および静的ヘッダー・プロパティを指定できます。これらのプロパティの一部は、SCA JMS ランタイムで特別な意味を持ちます。

### WebSphere MQ JMS エクスポート・バインディング

WebSphere MQ JMS エクスポート・バインディングは、WebSphere MQ ベースの JMS プロバイダーから SCA システムへのインバウンド接続を提供します。

エクスポート・バインディングで指定された *receive* 宛先に着信する要求を *listen* するため、MDB がデプロイされます。*send* フィールドで指定された宛先は、呼び出されたサービスが応答を返す場合にインバウンド要求に対する応答を送信するために使用されます。応答メッセージの JMSReplyTo フィールドで指定された宛先は、*send* フィールドで指定された宛先を上書きします。

### WebSphere MQ JMS メッセージ配信と Failed Event Manager

WebSphere MQ JMS バインディングは、デフォルトで最大再試行回数の値を 1 として構成されます。着信メッセージに使用する基盤となるリスナー・ポートのため

に、WebSphere MQ JMS バインディングの最大再試行回数の値は 1 以上でなければなりません。これは、インポートとエクスポートの両方の受信宛先で Failed Event Manager が失敗したメッセージを格納できるようにするために必要です。最大再試行回数の値を 1 未満に設定すると、失敗したターゲット・コンポーネントに対するメッセージが Failed Event Manager で使用できなくなります。ターゲット・コンポーネントでの失敗により、ランタイム例外が発生します。

## MQ JMS バインディングのサポート

WebSphere MQ JMS サービスは、WebSphere MQ ベースの JMS プロバイダーとの統合を提供します。

WebSphere MQ JMS プロバイダー・サービスは、J2EE プログラミング環境内にある WebSphere MQ ベースのシステムと直接統合する場合に使用します。これにより、サービス統合バスの MQ リンクまたはクライアント・リンク機能が不要になります。

WebSphere MQ からサービスをインポートするときには、データの送信先の宛先と、応答を受信できる宛先を使用します。データと JMS メッセージの間の相互変換は、JMS データ・バインディング・エッジ・コンポーネントを介して行います。

WebSphere MQ にサービスをエクスポートするときには、要求を受信でき、応答を送信できる宛先を使用します。データと JMS メッセージの間の相互変換は、JMS データ・バインディングを介して行います。関数セクターは、呼び出すターゲット・コンポーネントに対する操作へのマッピングを提供する役割を果たします。

WebSphere MQ キュー・マネージャーを構成するのは WebSphere MQ システム管理者の役割です。ただし、WebSphere MQ の MQ\_INSTALL\_ROOT 環境変数を、WebSphere Process Server によってサポートされる WebSphere MQ バージョンに変更して、サーバーを再始動する必要があります。これにより、サポートされるバージョンの WebSphere MQ のライブラリーが使用されることが保証されます。詳細なハードウェアおよびソフトウェア要件については、IBM サポート・ページを参照してください。

### 関連概念

99 ページの『JMS バインディングのサポート』

Java Message Service (JMS) は、使用可能な JMS JCA 1.5 ベースのリソース・アダプターを介して統合を提供します。JMS 統合のための完全なサポートは、サービス統合バス (SIB) JMS リソース・アダプターに対して用意されています。

## MQ JMS インポート・バインディングの主な特徴

MQ JMS インポートには、いくつかの主な特徴があります。ヘッダー、J2EE 成果物、および作成される J2EE リソースが主な特徴です。

### ヘッダー

JMS メッセージ・ヘッダーには、クライアントとプロバイダーの両方がメッセージの識別と発送に使用する値を指定する定義済みフィールドがいくつかあります。

### パラメーター

### **JMSCorrelationID**

関連メッセージへのリンクです。通常このフィールドは、応答の対象となるメッセージのメッセージ ID のストリングに設定されます。

### **TargetFunctionName**

このヘッダーは、ネイティブ・メソッドから操作メソッドへのマッピングを行うために、エクスポート・バインディングで使用されます。String の JMS ヘッダー・プロパティ TargetFunctionName を設定すると、エクスポートの JMSFunctionSelectorImpl に対して、操作メソッドへのマッピングに使用するネイティブ・メソッドが指示されます。これを使用するためには、ネイティブ・メソッドを値と同様にメソッド・バインディング内で指定します。エクスポートで JMSFunctionSelectorImpl を使用させたい場合は、このヘッダーをインポート・バインディング定義内で JMS ヘッダー・プロパティとして設定する必要があります。

## **関連スキーム**

WebSphere MQ JMS インポート・バインディングは、要求メッセージと応答メッセージを関連させる方法を決定するためのさまざまな関連スキームを提供します。

### **パラメーター**

以下の応答メッセージ関連スキームがあります。

#### **RequestMsgIDToCorrelID**

応答サービスは、要求の JMSMessageID を応答メッセージの JMSCorrelationID フィールドにコピーすることになります。これはデフォルト設定です。

#### **RequestCorrelIDToCorrelID**

応答サービスは、要求の JMSCorrelationID を応答メッセージの JMSCorrelationID フィールドにコピーすることになります。

## **J2EE リソース**

MQ JMS インポートを J2EE 環境にデプロイすると、いくつかの J2EE リソースが作成されます。

### **パラメーター**

#### **MQ 接続ファクトリー**

クライアントが MQ JMS プロバイダーとの接続を作成するために使用します。

#### **応答接続ファクトリー**

送信宛先が受信宛先とは異なるキュー・マネージャー上にある場合に、SCA MQ JMS ランタイムが使用します。

#### **リスナー・ポート**

接続ファクトリー、つまり宛先とメッセージ駆動型 Bean の間の関連を指定します。これにより、ポートに関連付けられているデプロイ済みのメッセージ駆動型 Bean が、宛先からメッセージを取得できます。

### **宛先**

- 送信宛先 - 要求または出力メッセージが送信される宛先です。
- 受信宛先 - 応答または着信メッセージが格納される宛先です。



## MQ JMS インポート・バインディングの構成:

MQ JMS インポート・バインディングは、作成されるすべての必要な接続ファクトリーと共にインストールできます。あるいは、サーバー上で事前に構成された成果物セットを指すように設計できます。

通常、MQ JMS インポート・バインディングは、WebSphere Integration Developer で作成されます。開発時に選択できる方法として、インストール時 (すなわちコンポーネントが WebSphere Process Server にインストールされる時) に、MQ JMS インポートに必要な接続と宛先を作成できます。もう 1 つの方法として、MQ JMS インポートで使用させる予定の WebSphere Process Server 上のリソースの JNDI 名を指定する方法があります。

このどちらの方法を選択するかによって、MQ JMS インポート・バインディングの構成が異なります。

新しいメッセージ・プロバイダー・リソースが作成される場合 (つまり、インストール時にサーバー上にリソースが作成される場合) は、リソースが存在するため、管理コンソールを使用してその位置を見つけ、管理できます。生成される成果物の JNDI 名を以下の表に示します。

表 12. MQ JMS インポート・バインディングの WebSphere Process Server へのインストール時に作成されるリソースの名前および JNDI 名

リソース	モジュール名	インポート名	リソースのグローバル JNDI 名
アウトバウンド接続ファクトリー	mqjms.module	my/import	mqjms.module/my/import_MQ_CF
応答リスナー・ポート	mqjms.module	my/import	mqjms.module.my.import_RESP_LP (注: これは JNDI ではなく、単なる名前です)
応答接続ファクトリー	mqjms.module	my/import	mqjms.module/my/import_RESP_CF
送信	mqjms.module	my/import	mqjms.module/my/import_MQ_SEND_D
受信	mqjms.module	my/import	mqjms.module/my/export_MQ_RECEIVE_D
SIB コールバック宛先	mqjms.module	my/import	mqjms.module/my/import_MQ_CALLBACK_D
SIB コールバック接続ファクトリー	すべてのモジュール	my/import	SCA.MQJMS/Callback_CF

**注:** リソースは、サーバー・スコープで作成されます。管理コンソールでのデフォルトのスコープはセルです。リソースを見つけ、管理するためには、スコープを変更する必要があります。

**注:** SIB コールバック宛先および SIB コールバック接続ファクトリーは、SIB JMS リソースです。表の中のその他の項目は、MQ JMS リソースです。2 つのタイプのリソースが管理されます。

もう 1 つの方法を使用する場合、MQ JMS インポートがサーバー上で使用するリソースを検出することになるため、このようなリソースをインストールしておくこ

と、およびインポート・ファイルにその JNDI 名が指定されていることが必要です。これにより、MQ JMS インポートとリソースの間の関連付けが行われます。

### MQ JMS インポート・バインディングの管理:

MQ JMS インポートを管理して、リソースを調整したり、リソースの特殊機能を設定したりすることができます。管理用タスクは、WebSphere 管理コンソールを使用して実行します。

管理コンソールでプロファイルに対して変更を行い、変更内容を保管する権限を持っている必要があります。

キューおよびキュー・マネージャーは自動的に生成されないため、WebSphere MQ 管理者が WebSphere MQ 内で作成する必要があります。

MQ JMS インポートは、WebSphere Process Server プロファイル内の Service Component Architecture (SCA) アプリケーションの一部としてインストールする必要があります。

1. 管理コンソールのデフォルトのメッセージング・プロバイダー設定パネルを開きます。

「JMS プロバイダー」を展開して、「WebSphere MQ」をクリックします。

2. **オプション:** WebSphere MQ 接続ファクトリーを管理します。

追加プロパティのリストにある WebSphere MQ 接続ファクトリーをクリックします。このパネルには、WebSphere MQ 接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする WebSphere MQ 接続ファクトリーをクリックするか、「新規」をクリックして新しい接続ファクトリーを作成します。

続いて表示されるパネルを使用して、JMS プロバイダーとしての WebSphere MQ で使用する、選択した JMS 接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS キューおよびトピックへの接続を作成する方法を制御します。

アプリケーションのリソース参照のために、このプロパティをバインディングに設定します。既存のアプリケーション用のバインディングを変更しない場合は、これらのプロパティを見つけることができる J2C パネルでこの接続ファクトリーを探します。

WebSphere MQ メッセージング・プロバイダーで構成された統合 JMS 接続ファクトリーでは、Point-to-Point とパブリッシュ/サブスクライブの両方のメッセージングが可能です。

3. **オプション:** WebSphere MQ キュー接続ファクトリーを管理します。

追加プロパティのリストにある WebSphere MQ キュー接続ファクトリーをクリックします。このパネルには、WebSphere MQ キュー接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする WebSphere MQ キュー接続ファクトリーをクリックするか、「新規」をクリックして新しいキュー接続ファクトリーを作成します。

続いて表示されるパネルを使用して、WebSphere MQ の JMS プロバイダーで使用する、選択したキュー接続ファクトリーの構成を表示または変更します。この構成プロパティは、関連した JMS キューへの接続を作成する方法を制御します。

WebSphere MQ キュー接続ファクトリーは、WebSphere MQ が Point-to-Point メッセージング用に提供するキューへの JMS 接続を作成するために使用されません。WebSphere MQ の JMS プロバイダー用のキュー接続ファクトリーを管理するには、WebSphere MQ キュー接続ファクトリーの管理オブジェクトを使用します。

#### 4. オプション: WebSphere MQ キュー宛先を管理します。

追加プロパティのリストにある WebSphere MQ キューの宛先をクリックします。このパネルには、WebSphere MQ キュー宛先のリストが、その構成プロパティの要約と共に表示されます。管理の対象とするキューの宛先をクリックするか、「新規」をクリックして新しい WebSphere MQ キューの宛先を作成します。

続いて表示されるパネルを使用して、WebSphere MQ をメッセージング・プロバイダーとする Point-to-Point メッセージングのための、選択した JMS キュー宛先の構成プロパティを表示または変更します。

WebSphere MQ のキュー宛先は、JMS キューのプロパティを構成するために使用されます。キューへの接続は、メッセージング・プロバイダーとしての WebSphere MQ 用の関連する JMS キュー接続ファクトリーによって作成されません。

#### 5. オプション: WebSphere MQ トピック接続ファクトリーを管理します。

追加プロパティのリストにある WebSphere MQ トピック接続ファクトリーをクリックします。このパネルには、WebSphere MQ トピック接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする WebSphere MQ トピック接続ファクトリーをクリックするか、「新規」をクリックして新しい接続ファクトリーを作成します。

続いて表示されるパネルを使用して、JMS プロバイダーとしての WebSphere MQ で使用する、選択したトピック接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS トピック宛先への接続を作成する方法を制御します。

WebSphere MQ JMS プロバイダー用の JMS トピック接続ファクトリーを管理するには、WebSphere MQ トピック接続ファクトリーの管理オブジェクトを使用します。

#### 6. オプション: WebSphere MQ トピック宛先を管理します。

追加プロパティのリストにある WebSphere MQ トピックの宛先をクリックします。このパネルには、WebSphere MQ トピック接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とするトピックの宛先をクリックするか、「新規」をクリックして新しいトピックの宛先を作成します。

続いて表示されるパネルを使用して、WebSphere MQ をメッセージング・プロバイダーとするパブリッシュ/サブスクライブ・メッセージングのための、選択した JMS トピック宛先の構成プロパティを表示または変更します。

WebSphere MQ のトピック宛先は、メッセージング・プロバイダーとしての WebSphere MQ 用の JMS トピックのプロパティを構成するために使用されます。トピックへの接続は、関連付けられたトピック接続ファクトリーによって作成されます。

7. 変更内容をマスター・プロファイルに保管し、必要に応じて WebSphere Process Server を再始動します。

## MQ JMS エクスポート・バインディングの主な特徴

MQ JMS エクスポートには、いくつかの主な特徴があります。ヘッダー、J2EE 成果物、および作成される J2EE リソースが主な特徴です。

### 関連スキーム

WebSphere MQ JMS エクスポート・バインディングは、要求メッセージと応答メッセージを関連させる方法を決定するためのさまざまな関連スキームを提供します。

#### パラメーター

以下の応答メッセージ関連スキームがあります。

##### RequestMsgIDToCorrelID

WebSphere MQ JMS サービスは、要求の JMSMessageID を、送信する応答メッセージの JMSCorrelationID フィールドにコピーします。これはデフォルト設定です。

##### RequestCorrelIDToCorrelID

WebSphere MQ JMS サービスは、要求の JMSCorrelationID を応答メッセージの JMSCorrelationID フィールドにコピーします。

### J2EE リソース

MQ JMS エクスポートを J2EE 環境にデプロイすると、いくつかの J2EE リソースが作成されます。

#### パラメーター

##### MQ 接続ファクトリー

クライアントが MQ JMS プロバイダーとの接続を作成するために使用します。

##### 応答接続ファクトリー

送信宛先が受信宛先とは異なるキュー・マネージャー上にある場合に、SCA MQ JMS ランタイムが使用します。

##### リスナー・ポート

接続ファクトリー、つまり宛先とメッセージ駆動型 Bean の間の関連を指定します。これにより、ポートに関連付けられているデプロイ済みのメッセージ駆動型 Bean が、宛先からメッセージを取得できます。

##### 宛先

- 受信宛先 - 着信メッセージまたは要求メッセージが格納される宛先です。

- 送信宛先 - 着信メッセージの JMSReplyTo ヘッダーで置き換えられない場合に、応答メッセージの送信先となる宛先です。

## パラメーター

### MQ JMS エクスポート・バインディングの構成:

MQ JMS エクスポート・バインディングは、作成されるすべての必要な接続ファクトリーと共にインストールできます。あるいは、サーバー上で事前に構成された成果物セットを指すように設計できます。

通常、MQ JMS エクスポート・バインディングは、WebSphere Integration Developer で作成されます。開発時に選択できる方法として、インストール時 (すなわちコンポーネントが WebSphere Process Server にインストールされる時) に、MQ JMS エクスポートに必要な接続と宛先を作成できます。もう 1 つの方法として、MQ JMS エクスポートで使用させる予定の WebSphere Process Server 上のリソースの JNDI 名を指定する方法があります。

このどちらの方法を選択するかによって、MQ JMS エクスポート・バインディングの構成が異なります。

新しいメッセージ・プロバイダー・リソースが作成される場合 (つまり、インストール時にサーバー上にリソースが作成される場合) は、リソースが存在するため、管理コンソールを使用してその位置を見つけ、管理できます。生成される成果物の JNDI 名を以下の表に示します。

表 13. MQ JMS エクスポート・バインディングの WebSphere Process Server へのインストール時に作成されるリソースの名前および JNDI 名

リソース	モジュール名	エクスポート名	リソースのグローバル JNDI 名
リスナー・ポート	mjms.module	my/export	mjms.module.my.export_LP (注: これは JNDI ではなく、単なる名前です)
インバウンド接続ファクトリー	mjms.module	my/export	mjms.module/my/export_LIS_CF
応答接続ファクトリー	mjms.module	my/export	mjms.module/my/export_RESP_CF
受信	mjms.module	my/export	mjms.module/my/export_MQ_RECEIVE_D
送信	mjms.module	my/export	mjms.module/my/export_MQ_SEND_D
SIB コールバック宛先	mjms.module	my/export	mjms.module/my/export_MQ_CALLBACK_D
SIB コールバック接続ファクトリー	すべてのモジュール	my/export	SCA.MQJMS/Callback_CF

注: リソースは、サーバー・スコープで作成されます。管理コンソールでのデフォルトのスコープはセルです。リソースを見つけ、管理するためには、スコープを変更する必要があります。

注: SIB コールバック宛先および SIB コールバック接続ファクトリーは、SIB JMS リソースです。表の中のその他の項目は、MQ JMS リソースです。2 つのタイプのリソースが管理されます。

もう 1 つの方法を使用する場合、MQ JMS エクスポートがサーバー上で使用するリソースを検出することになるため、このようなリソースをインストールしておくこと、およびインポート・ファイルにその JNDI 名が指定されていることが必要です。これにより、MQ JMS エクスポートとリソースの間の関連付けが行われます。

### MQ JMS エクスポート・バインディングの管理:

MQ JMS エクスポートを管理して、リソースを調整したり、リソースの特殊機能を設定したりすることができます。管理用タスクは、WebSphere 管理コンソールを使用して実行します。

管理コンソールでプロファイルに対して変更を行い、変更内容を保管する権限を持っている必要があります。

MQ JMS エクスポートは、WebSphere Process Server プロファイル内の Service Component Architecture (SCA) アプリケーションの一部としてインストールされます。

1. 管理コンソールのデフォルトのメッセージング・プロバイダー設定パネルを開きます。

「JMS プロバイダー」を展開して、「WebSphere MQ」をクリックします。

2. オプション: WebSphere MQ 接続ファクトリーを管理します。

追加プロパティのリストにある WebSphere MQ 接続ファクトリーをクリックします。このパネルには、WebSphere MQ 接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする WebSphere MQ 接続ファクトリーをクリックするか、「新規」をクリックして新しい接続ファクトリーを作成します。

続いて表示されるパネルを使用して、JMS プロバイダーとしての WebSphere MQ で使用する、選択した JMS 接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS キューおよびトピックへの接続を作成する方法を制御します。

アプリケーションのリソース参照のために、このプロパティをバインディングに設定します。既存のアプリケーション用のバインディングを変更しない場合は、これらのプロパティを見つけることができる J2C パネルでこの接続ファクトリーを探します。

WebSphere MQ メッセージング・プロバイダーで構成された統合 JMS 接続ファクトリーでは、Point-to-Point とパブリッシュ/サブスクライブの両方のメッセージングが可能です。

3. オプション: WebSphere MQ キュー接続ファクトリーを管理します。

追加プロパティのリストにある WebSphere MQ キュー接続ファクトリーをクリックします。このパネルには、WebSphere MQ キュー接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする

CWebSphere MQ キュー接続ファクトリーをクリックするか、「新規」をクリックして新しいキュー接続ファクトリーを作成します。

続いて表示されるパネルを使用して、WebSphere MQ の JMS プロバイダーで使用する、選択したキュー接続ファクトリーの構成を表示または変更します。この構成プロパティは、関連した JMS キューへの接続を作成する方法を制御します。

WebSphere MQ キュー接続ファクトリーは、WebSphere MQ が Point-to-Point メッセージング用に提供するキューへの JMS 接続を作成するために使用されます。WebSphere MQ の JMS プロバイダー用のキュー接続ファクトリーを管理するには、WebSphere MQ キュー接続ファクトリーの管理オブジェクトを使用します。

#### 4. オプション: WebSphere MQ キュー宛先を管理します。

追加プロパティのリストにある WebSphere MQ キューの宛先をクリックします。このパネルには、WebSphere MQ キュー宛先のリストが、その構成プロパティの要約と共に表示されます。管理の対象とするキューの宛先をクリックするか、「新規」をクリックして新しい WebSphere MQ キューの宛先を作成します。

続いて表示されるパネルを使用して、WebSphere MQ をメッセージング・プロバイダーとする Point-to-Point メッセージングのための、選択した JMS キュー宛先の構成プロパティを表示または変更します。

WebSphere MQ のキュー宛先は、JMS キューのプロパティを構成するために使用されます。キューへの接続は、メッセージング・プロバイダーとしての WebSphere MQ 用の関連する JMS キュー接続ファクトリーによって作成されます。

#### 5. オプション: WebSphere MQ トピック接続ファクトリーを管理します。

追加プロパティのリストにある WebSphere MQ トピック接続ファクトリーをクリックします。このパネルには、WebSphere MQ トピック接続ファクトリーのリストが、その構成プロパティの要約と共に表示されます。管理の対象とする WebSphere MQ トピック接続ファクトリーをクリックするか、「新規」をクリックして新しい接続ファクトリーを作成します。

続いて表示されるパネルを使用して、JMS プロバイダーとしての WebSphere MQ で使用する、選択したトピック接続ファクトリーの構成プロパティを表示または変更します。この構成プロパティは、関連した JMS トピック宛先への接続を作成する方法を制御します。

WebSphere MQ JMS プロバイダー用の JMS トピック接続ファクトリーを管理するには、WebSphere MQ トピック接続ファクトリーの管理オブジェクトを使用します。

#### 6. オプション: WebSphere MQ トピック宛先を管理します。

追加プロパティのリストにある WebSphere MQ トピックの宛先をクリックします。このパネルには、WebSphere MQ トピック接続ファクトリーのリスト

が、その構成プロパティの要約と共に表示されます。管理の対象とするトピックの宛先をクリックするか、「新規」をクリックして新しいトピックの宛先を作成します。

続いて表示されるパネルを使用して、WebSphere MQ をメッセージング・プロバイダーとするパブリッシュ/サブスクライブ・メッセージングのための、選択した JMS トピック宛先の構成プロパティを表示または変更します。

WebSphere MQ のトピック宛先は、メッセージング・プロバイダーとしての WebSphere MQ 用の JMS トピックのプロパティを構成するために使用されます。トピックへの接続は、関連付けられたトピック接続ファクトリーによって作成されます。

変更内容をマスター・プロファイルに保管し、必要に応じて WebSphere Process Server を再始動します。

## 外部クライアント

WebSphere Process Server は、WebSphere MQ JMS バインディングを使用して外部クライアントとの間でメッセージを送受信できます。

外部クライアント (Web ポータルや EIS など) は、WebSphere Process Server の SCA モジュールにメッセージを送信する必要があり、WebSphere Process Server 内からコンポーネントによって呼び出される必要があります。

WebSphere MQ JMS エクスポート・コンポーネントは、エクスポート・バインディングで指定された receive 宛先に着信する要求を listen するためのメッセージ駆動型 Bean (MDB) をデプロイします。send フィールドで指定された宛先は、呼び出されたサービスが応答を返す場合にインバウンド要求に対する応答を送信するために使用されます。したがって、外部クライアントは、エクスポート・バインディングを介してアプリケーションを呼び出すことができます。

WebSphere MQ JMS インポートは外部クライアントにバインドし、外部クライアントにメッセージを配信できます。このメッセージは、外部クライアントからの応答を要求してもしなくても構いません。

WebSphere MQ を使用して外部クライアントと対話する方法については、WebSphere MQ インフォメーション・センターを参照してください。

## データ・バインディング

データ・バインディングは、SCA アプリケーションに渡されるデータとエンタープライズ情報システム (EIS) の間のマッピングを記述します。つまり、EIS 固有の形式への SCA データ表現のバインディングを記述します。

### 関連資料

92 ページの『EIS インポート・バインディングの主な特徴』

EIS インポートとは、Service Component Architecture (SCA) モジュール内のコンポーネントが、SCA モジュール外部で定義された EIS サービスを使用できるようにする SCA インポートです。EIS インポートは、SCA コンポーネントから外部 EIS へのデータ転送に使用されます。



## EIS サービスのデータ・バインディング

データ・バインディングを使用すると、EIS インポートまたは EIS エクスポートの実装で引数データを J2C または JMS 実装が必要とする固有の形式に変換できます。

データ・バインディング実装クラスは、インポートまたはエクスポート・ファイルで定義できます。

DataBinding は、ルート・データ・バインディング・インターフェースです。これにより、DataObject の設定が可能です。これは、J2C と JMS の両方に適用できます。

### JCA 1.5 レコード・ベースのデータ・バインディング:

データ・バインディング実装クラスは、インポートまたはエクスポート・ファイルで定義できます。recordObased データ・バインディングを使用する場合は、固有のインターフェースを実装する必要があります。

RecordDataBinding インターフェースは基本インターフェースを拡張して javax.resource.cci.Record を追加します。データ・バインディングを J2EE コネクター・アーキテクチャー・ベースのサービスと共に使用する場合は、このインターフェースを実装する必要があります。これは、レコードとして入力および出力でリソース・アダプター対話の execute メソッドに渡すことができます。

リソース・アダプターが対話の execute メソッドの入力専用バージョンをサポートする場合は、このインターフェースでは不十分です。この制限は、入力専用の実行で結果が cci Record として返されるためです。このインスタンスでは、レコードをデータ・バインディングで設定した後に変換後のデータ・オブジェクトを取得できるようになります。リソース・アダプターが execute メソッドの入力専用バージョンをサポートする場合には、データ・バインディング・プロバイダーが RecordHolderDataBinding インターフェースを実装する必要があります。

インバウンド J2C 通信の場合は、RecordHolderDataBinding インターフェースも実装する必要があります。リスナー・インターフェースはリソース・アダプターによって呼び出され、実装側の MDB はネイティブ・データとして渡されます。このデータは、ビジネス・オブジェクトから取得できるように、データ・バインディングで設定する必要があります。リスナーの引数と戻りの型が javax.resource.cci.Record である場合は、RecordHolderDataBinding で十分です。リスナーの引数または戻りの型が cci Record 以外である場合は、ユーティリティー・インターフェース InboundNativeDataRecord が提供されます。

InboundNativeDataRecord インターフェースでは、リスナーに対する任意の数および型の引数をオブジェクト配列として設定できます。実装 (cci Record) はその後で RecordHolderDataBinding インターフェースで設定できるため、すべてのリスナー引数が渡され、データ・バインディングは引数を取得してビジネス・オブジェクトを作成できます。戻り時には、データ・バインディングの実装が対称的な方法で戻り値を設定します。次に、ランタイムが get メソッドを使用してオブジェクト配列を返します。このとき、指標 0 の引数は、呼び出し側 (リソース・アダプター) に返される戻り値です。

## JMS および MQ JMS サービスのデータ・バインディング

JMS および WebSphere MQ JMS サービスに定義されるデータ・バインディングは、JMSDataBinding または JMSObjectBinding インターフェースを実装する必要があります。

JMS サービスの JMSDataBinding インターフェースにより、DataObject またはビジネス・オブジェクトと JMS メッセージの間でデータを変換できます。

### 直列化:

JMS および MQ JMS サービスとエクスポートの実装は、JMS または MQ JMS 形式との間での DataObject の直列化を提供するデータ・バインディングのデフォルト実装を提供できます。

JMS サービスとエクスポートの実装は、以下の JMS データ・バインディングに対するデフォルト実装を提供できます。

- JMSDataBindingImplJava - JMSObjectMessage をサポートし、JMSMessage のオブジェクト・フィールドとの間で DataObject を直列化します。メッセージのオブジェクト内容は、Serializable インターフェースを実装する必要があります。JMSDataBindingImplJava は、java.lang.Object をメッセージに直列化できるようにする JMSObjectBinding も実装します。
- JMSDataBindingImplXML - JMSTextMessage をサポートします。このバインディングは、DataObject を XML 形式に直列化して JMSMessage のテキスト・フィールドに設定します。直列化の実装は、SCA 実装 (クラス com.ibm.ws.sca.data.impl.DataBindingImplXML) で公開されるネイティブの SDO オブジェクト直列化を使用しています。

XML スキーマにグローバル・エレメントなしで定義されたタイプがある場合、JMS バインディング (JMSDataBindingImplXML および JMSDataBindingImplJava) はこのタイプをエレメントに解決できません。

### スキーマ:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">
<!-- global element required but missing -->

<complexType name="Quote">
<sequence>
<element name="symbol" type="string"></element>
<element name="price" type="float"></element>
</sequence>
</complexType>

</schema>
```

以下のような例外を受信した場合、

```
com.ibm.websphere.sca.ServiceRuntimeException:
caused by: java.lang.IllegalArgumentException:
{Quote}Quote is not corresponding to a global element.
```

または

```
[8/25/06 10:20:40:938 PDT] 00000054 FFDC Z
com.ibm.ws.sca.databinding.impl.DataBindingImplXML
com.ibm.ws.sca.databinding.impl.DataBindingImplXML#002 Exception:
org.eclipse.emf.ecore.xmi.FeatureNotFoundException:
Feature 'Quote' not found. (sca:/dataObject.xml, 2, 126)
```

これは、グローバル・エレメントを以下のように定義する必要があることを示している場合があります。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">

<element name="Quote" type="tns:Quote"></element> <!-- global element required -->

<complexType name="Quote">
<sequence>
<element name="symbol" type="string"></element>
<element name="price" type="float"></element>
</sequence>
</complexType>

</schema>
```

### ランタイム環境でのデータ・バインディングの使用:

アプリケーションをアセンブルすると、データ・バインディング・ジェネレーターによって、すべてのタイプのすべての操作を対象とする `DataBinding` の実装が生成され、インポート・ファイルまたはエクスポート・ファイルの `inDataBindingType` 属性または `outDataBindingType` 属性にそれぞれの名前が保管されます。

データ・バインディングを使用すると、EIS インポートまたは EIS エクスポートの実装で引数データを J2C または JMS 実装が必要とする固有の形式に変換できます。データ・バインディング・クラスは、以下に示すさまざまなインターフェースを実装します。

`DataBinding` は、ルート・データ・バインディング・インターフェースです。これにより、`DataObject` の設定が可能です。これは、J2C と JMS の両方に適用できます。

ユーザーは、事前に構成された JMS リソースを使用して、ユーザー固有のバインディングを構成できます。接続ファクトリーと宛先を事前に定義してから、それらを指し示すだけです。

## エクスポート・バインディングとイベント順序付け

エクスポートがメッセージを受信順に処理および配信するように規定できます。イベント順序付けを可能にするには、エクスポート・バインディングを適切に構成する必要があります。

イベント順序付けは、SCA コンポーネントに対する着信イベントを順序付ける機能を提供します。EIS および JMS エクスポートは、ターゲット SCA コンポーネントに対するエントリー・ポイントです。イベント順序付けを使用可能にするには、エクスポートがメッセージの受信順序と同じ順序でメッセージを処理および配信する必要があります。

イベント順序付けを使用可能にするために必要な固有の設定は、使用するエクスポート・バインディングのタイプによって異なります。

EIS および JMS エクスポートの他に、ネイティブ WebSphere MQ エクスポート・バインディングもイベント順序付けをサポートします。イベント順序付け用のネイティブ WebSphere MQ エクスポート・バインディングの構成は WebSphere Integration Developer で行います。

#### 関連概念

307 ページの『エクスポート・バインディングでのイベント順序付けの使用』イベント順序付けは、EIS エクスポート・バインディング、JMS エクスポート・バインディング、ネイティブ MQ、および WebSphere MQ JMS エクスポート・バインディングでサポートされます。エクスポートに正しい順序でメッセージを処理および配信させるには、エクスポート・バインディングを適切に構成する必要があります。

#### 関連情報



WebSphere Integration Developer

## EIS エクスポート・バインディングでのイベントの順序付けの有効化

EIS エクスポート・バインディングでは、イベントの順序付けがサポートされます。バインディングのアクティベーション・スペックを適切に構成する必要があります。

WebSphere Adapter は、アクティベーション・スペック・プロパティを指定することにより、イベントの順序付けを可能にするメカニズムを提供します。詳しくは、資料を参照してください。JCA 1.5 リソース・アダプターでは、イベントの順序付けを可能にするアダプターの構成方法の詳細を、特定のプロバイダーの資料で調べる必要があります。

WebSphere Adapter のアクティベーション・スペック・プロパティの構成方法については、WebSphere Adapter インフォメーション・センターのアクティベーション・スペックについてのトピックを参照してください。

クラスター環境の構成方法、および Network Deployment 環境内の特定のリソース・アダプターで、イベントの順序付けがサポートされるかどうかについては、WebSphere Adapter インフォメーション・センターを参照してください。

#### 関連概念



WebSphere Adapter

307 ページの『エクスポート・バインディングでのイベント順序付けの使用』イベント順序付けは、EIS エクスポート・バインディング、JMS エクスポート・バインディング、ネイティブ MQ、および WebSphere MQ JMS エクスポート・バインディングでサポートされます。エクスポートに正しい順序でメッセージを処理および配信させるには、エクスポート・バインディングを適切に構成する必要があります。

## JMS エクスポート・バインディングでのイベントの順序付けの有効化

JMS エクスポート・バインディングは、イベント順序付けをサポートしています。イベントを受信順に処理するためには、バインディングのプロパティを構成する必要があります。

JMS エクスポート・バインディングでのイベントの順序付けは、クラスター環境ではサポートされません。

JMS エクスポート・バインディングでのイベントの順序付けを可能にするには、着信メッセージを処理する並行メッセージ駆動型 Bean (MDB) の数を制限する必要があります。

SIB JMS Resource Adapter を使用する場合は、ActivationSpec の maxConcurrency カスタム・プロパティの値を 1 に設定します。

1. 管理コンソールのナビゲーション・ペインで、「リソース」を展開して、「リソース・アダプター (Resource Adapters)」をクリックします。
2. コンテンツ・ペインで、「SIB JMS Resource Adapter」をクリックします。
3. 「追加プロパティ」の下で「J2C アクティベーション・スペック (J2C activation specification)」をクリックしてから、「エクスポート」アクティベーション・スペックを選択します。アクティベーション・スペックの名前は、エクスポート・バインディングのインストール時に定義されます。命名規則については、関連トピックを参照してください。
4. 「追加プロパティ」の下で、「J2C アクティベーション・スペック・カスタム・プロパティ (J2C activation specification custom properties)」をクリックします。
5. maxConcurrency を 1 に設定します。値フィールドをクリックして、値 1 を入力した後、「OK」をクリックします。
6. 変更をマスター構成に保管します。

### 関連概念

105 ページの『JMS エクスポート・バインディングの構成』

JMS エクスポート・バインディングは、作成されるすべての必要な接続ファクトリーと共にインストールできます。あるいは、サーバー上で事前に構成された成果物セットを指すように設計できます。

307 ページの『エクスポート・バインディングでのイベント順序付けの使用』  
イベント順序付けは、EIS エクスポート・バインディング、JMS エクスポート・バインディング、ネイティブ MQ、および WebSphere MQ JMS エクスポート・バインディングでサポートされます。エクスポートに正しい順序でメッセージを処理および配信させるには、エクスポート・バインディングを適切に構成する必要があります。

## WebSphere MQ JMS エクスポート・バインディングでのイベントの順序付けの有効化

WebSphere MQ JMS エクスポート・バインディングは、イベント順序付けをサポートしています。イベントを受信順に処理するためには、バインディングのプロパティを構成する必要があります。

WebSphere MQ JMS エクスポート・バインディングでのイベントの順序付けは、クラスター環境ではサポートされません。

WebSphere MQ JMS エクスポートのイベント順序付けを有効にするには、メッセージ駆動型 Bean にメッセージを配信する並行リスナー・スレッドの数を制限する必要があります。

1. WebSphere MQ JMS エクスポート・バインディングの `maxSessions` プロパティを 1 に設定します。
2. 変更をマスター構成に保管します。

#### 関連概念

307 ページの『エクスポート・バインディングでのイベント順序付けの使用』イベント順序付けは、EIS エクスポート・バインディング、JMS エクスポート・バインディング、ネイティブ MQ、および WebSphere MQ JMS エクスポート・バインディングでサポートされます。エクスポートに正しい順序でメッセージを処理および配信させるには、エクスポート・バインディングを適切に構成する必要があります。

## JMS ヘッダー

JMS ヘッダーは、JMS メッセージ・プロパティのすべてのプロパティを含むサービス・データ・オブジェクト (SDO) です。これらのプロパティはインバウンド・メッセージからのものか、アウトバウンド・メッセージに適用されるプロパティにすることができます。

JMS ヘッダーは、システム・プログラミング・インターフェースを使用して提供されるため、サービス・コンポーネント・アーキテクチャー・メッセージにアクセスすることにより JMS ヘッダーを設定または取得します。JMS ヘッダーは、クライアント・プログラムからは設定または取得できません。ただし、ESB メディエーションを使用して JMS ヘッダー値を取得したり、発信要求で JMS ヘッダー値を設定できます。

JMS ヘッダーおよび JMS メッセージ・プロパティが使用されるのは、基本サービス・コンポーネント・アーキテクチャー SCDL バインディング・スイッチがオンになっている場合のみです。スイッチがオンになっているとき、コンテキスト情報が伝搬されます。デフォルトでは、このスイッチはオンになっています。コンテキスト情報の伝搬を回避するには、値を `false` に変更します。

コンテキスト伝搬を使用可能にすると、ヘッダー情報をメッセージまたはターゲット・コンポーネントに流すことができます。コンテキスト伝搬をオン/オフにするには、インポートおよびエクスポート・バインディングの `contextPropagationEnabled` 属性に `true` または `false` を指定します。以下に例を挙げます。

```
<esbBinding xsi:type="eis:JMSImportBinding" contextProgagationEnabled="true">
```

デフォルトは `true` です。

## JMS バインディングおよび MQ JMS バインディングでの障害の処理

JMS バインディングおよび MQ JMS バインディングで、障害が発生することがあります。このような障害を処理するメカニズムが用意されています。

JMSDataBinding および JMSObjectBinding に渡されることが想定されている引数は、インターフェース操作と入力、出力、および障害のタイプによって異なります。

障害では、メソッド・バインディングで指定された `outDataBindingType` が使用されます。指定されていない場合は、すべての直列化および非直列化に対してバインディング・レベル `dataBindingType` が使用されます。

障害タイプが単純である場合は、障害メッセージを表す `JMS DataBinding` にストリングが設定されます。さらに、`IsBusinessException` が `true` に設定されます。

障害タイプがデータ・オブジェクトである場合は、障害メッセージを表す `JMS DataBinding` にデータ・オブジェクトが設定されます。このシナリオでは、`JMSDataBinding` を使用する必要があります。

障害を含んだメッセージは、JMS データ・バインディングによって処理されます。ブール値のヘッダー・プロパティ `IsBusinessException` は、データ・バインディングによってインターセプトされます。値が `true` の場合、データ・バインディングにより、有効搭載量に障害データが含まれていることがランタイムに通知されます。

カスタム・データ・バインディングを使用している場合は、障害を正しく処理するために以下のステップを実行する必要があります。デフォルトの実装では、障害がユーザーの介入なしに処理されます。

1. JMS エクスポートでは、`JMSDataBinding` インターフェースの `setBusinessException(boolean isBusinessException)` メソッドを使用して、データ・バインディングに指定されたデータ・オブジェクトまたはオブジェクトが障害オブジェクトであり、このバインディングによって作成されたメッセージを適宜構成する必要があることを示します。データ・バインディングが、`isBusinessException` を適切に指定しなければなりません。
2. JMS インポートでは、`JMSDataBinding` インターフェースの `isBusinessException()` メソッドを使用して、メッセージに障害が含まれているかどうかを示します。

データ・バインディングでは、有効搭載量に定義された障害を示すヘッダー・プロパティの値が取得されます。ランタイムにより `JMS` メッセージがデータ・バインディングに渡された後、データ・バインディングの `isBusinessException()` が呼び出されます。戻り値が `false` の場合はメッセージが正常に処理され、それ以外の場合は、呼び出し元に `ServiceBusinessException` が戻されます。バインディングによって生成されたデータ・オブジェクトやオブジェクトは `ServiceBusinessException` に設定され、これが呼び出し元に戻されます。

## JMS および MQ JMS バインディングのトラブルシューティング

JMS バインディングまたは MQ JMS バインディングで発生した問題を診断し、修正できます。

JMS と MQ JMS のインポートおよびエクスポート実装は、さまざまなエラー状態に応じて、以下の 2 種類の例外のうちのいずれかを戻すことがあります。

- `SCABusinessException` - サービス・ビジネス・インターフェース (WSDL ポート・タイプまたは Java インターフェース) で指定された例外が発生した場合に、この例外が戻されます。
- `SCARuntimeException` - その他のすべてのケースで生成されます。ほとんどの場合、cause 例外には、元の例外が含まれます。J2C の場合は `ResourceException` であり、JMS の場合は `JMSEException` になります。

XML スキーマにグローバル・エレメントなしで定義されたタイプがある場合、JMS バインディング (`JMSDataBindingImplXML` および `JMSDataBindingImplJava`) はこのタイプをエレメントに解決できません。

スキーマ:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">
  <!-- global element required but missing -->

  <complexType name="Quote">
    <sequence>
      <element name="symbol" type="string"/>
      <element name="price" type="float"/>
    </sequence>
  </complexType>
</schema>
```

以下のような例外を受信した場合、

```
com.ibm.websphere.sca.ServiceRuntimeException:
caused by: java.lang.IllegalArgumentException:
{Quote}Quote is not corresponding to a global element.
```

または

```
[8/25/06 10:20:40:938 PDT] 00000054 FFDC Z
com.ibm.ws.sca.databinding.impl.DataBindingImplXML
com.ibm.ws.sca.databinding.impl.DataBindingImplXML#002 Exception:
```

```
org.eclipse.emf.ecore.xmi.FeatureNotFoundException:
Feature 'Quote' not found. (sca:/dataObject.xml, 2, 126)
```

これは、グローバル・エレメントを以下のように定義する必要があることを示している場合があります。

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.ibm.com" xmlns:tns="http://www.ibm.com">

  <element name="Quote" type="tns:Quote"/> <!-- global element required -->

  <complexType name="Quote">
    <sequence>
      <element name="symbol" type="string"/>
      <element name="price" type="float"/>
    </sequence>
  </complexType>

</schema>
```



## Failed Event Manager に表示されない JMS ベースの SCA メッセージ

JMS の対話の失敗によって SCA メッセージが発生した場合は、Failed Event Manager でこのメッセージを見つけることになります。Failed Event Manager にこのようなメッセージが表示されない場合は、JMS 宛先の基盤となる SIB 宛先の最大配信失敗回数の値が 1 よりも大きいことを確認してください。この値を 2 以上に設定すると、JMS バインディングに対する SCA 呼び出しの間の Failed Event Manager との対話が可能になります。

## Failed Event Manager に表示されない WebSphere MQ JMS ベースの SCA メッセージ

WebSphere MQ JMS の対話の失敗によって SCA メッセージが発生した場合は、Failed Event Manager でこのメッセージを見つけることになります。Failed Event Manager にこのようなメッセージが表示されない場合は、基盤となるリスナー・ポートの最大再試行回数プロパティの値が 1 以上であることを確認してください。この値を 1 以上に設定すると、MQ JMS バインディングのための SCA 呼び出しの間の Failed Event Manager との対話が可能になります。

## ビジネス・ステート・マシンのトラブルシューティングの概要

ビジネス・ステート・マシンのインスタンスのデバッグおよび管理を行うために、**相関セット**の値および表示状態の変数を表示できます。

ビジネス・ステート・マシンは、イベント・ドリブン・ビジネス・プロセスを表すために使用されます。ビジネス・ステート・マシン内には多くのインスタンスがあります。以下を使用すると、ビジネス・ステート・マシンのインスタンスの管理およびデバッグが可能です。

- 相関セットのプロパティ
- 表示状態

### 相関セットのプロパティ

ビジネス・ステート・マシンのインスタンスを相互に区別するために、**相関セット**を使用してステート・マシンのインスタンスを一意的に識別します。例えば、顧客 ID と状態を相関セットのプロパティにすることができます。特定のインスタンスを管理する場合は、相関セット・プロパティの値が必要です。相関セット・プロパティは WebSphere Integration Developer で定義され、Business Process Choreographer Explorer に表示されます。

WebSphere Integration Developer で定義できる相関セットは 1 つのみです。複数の相関セットは使用できません。

### 表示状態

表示状態変数は、特定のビジネス・ステート・マシンのインスタンスの現行の状態を示します。最後にコミットされた状態を調べると、ビジネス・ステート・マシンのデバッグや管理に役立ちます。表示状態は WebSphere Integration Developer で定義され、Business Process Choreographer Explorer に表示されます。

表示状態変数は、必ずしもビジネス・ステート・マシンのインスタンスの最新の状態を示すとは限りません。インスタンスが次々とイベントを処理している場合は、表示状態変数のメモリー内コピーが最後にコミットされた値と異なる場合があります。Business Process Choreographer Explorer に表示されるのは、最後にディスクに書き込まれた表示状態の値です。ビジネス・ステート・マシンのインスタンスがイベントを処理している場合には、トランザクションが完了するまで、変数のメモリー内の値はディスクに書き込まれません。

## ビジネス・ステート・マシンのインスタンスの検索

特定のビジネス・ステート・マシンのインスタンスを検索して管理するための、相関セット・プロパティを表示します。

WebSphere Integration Developer 内で相関セットを定義し、モジュールを保管します。モジュールをサーバーにデプロイします。

相関セット・プロパティの値によって、1 つのビジネス・ステート・マシンのインスタンスがライフ・サイクルにわたって他のインスタンスから区別されます。特定のビジネス・ステート・マシンのインスタンスを終了させる必要がある場合、相関セット・プロパティの値によって正しいインスタンスを識別できます。Business Process Choreographer Explorer を使用して、相関セット・プロパティを表示するには、以下の手順を実行します。

**制約事項:** 1 つのビジネス・ステート・マシンに対して定義できる相関セットは 1 つのみです。複数の相関セットは使用できません。

1. 「プロセス・テンプレート」の下で、使用しているビジネス・ステート・マシンを表すプロセス・テンプレートを選択します。
2. 「プロセス・テンプレート名」の下で、プロセス・テンプレートを選択し、「インスタンス」をクリックして、システム内で現在もアクティブな、存在するすべてのインスタンスを表示します。
3. インスタンスごとに、そのインスタンスをクリックしてから「照会プロパティ」タブをクリックして、「プロパティ名」の下に相関セット・プロパティを表示します。

管理用タスクを実行します。

## 表示状態の表示

ビジネス・ステート・マシンのインスタンスを管理したり、デバッグしたりする場合に、表示状態を表示します。

WebSphere Integration Developer で表示状態変数を初期化して、モジュールを保管します。モジュールをサーバーにデプロイします。

表示状態変数を使用すると、アクティブなビジネス・ステート・マシンのインスタンスの現在の状態を表示できます。例えば、あるビジネス・ステート・マシンのインスタンスが予想通りに応答しない場合は、アクティブなビジネス・ステート・マシンのインスタンスを表示して、現在の状態を判別し、問題をデバッグできます。そのアクティブなビジネス・ステート・マシンのインスタンスの相関セット・プロ

パティyの値が必要です。アクティブなビジネス・ステート・マシンのインスタンスの現在の状態を表示するには、Business Process Choreographer Explorer で次の手順を実行します。

1. 「プロセス・テンプレート」の下で、使用しているビジネス・ステート・マシンを表すプロセス・テンプレートを選択します。
2. 「プロセス・テンプレート名」の下で、プロセス・テンプレートを選択し、「インスタンス」をクリックして、システム内で現在もアクティブな、存在するすべてのインスタンスを表示します。
3. インスタンスごとに、そのインスタンスをクリックしてから「照会プロパティ」タブをクリックして、「プロパティ名」の下に相関セット・プロパティと表示状態を表示します。

管理用タスクを実行します。

## ビジネス・ルールおよびセレクトyの概要

ビジネス・ルールおよびセレクトyは、基準に基づいて処理の結果を変更できるため、ビジネス・プロセスに柔軟性が提供されます。ビジネス・ルールおよびセレクトyのコンポーネントを含むアプリケーションをインストールする前に、ビジネス・ルール動的リポジトリをインストールする必要があります。

ビジネス・ルールまたはセレクトyを含むモジュールをインストールしたり、またはサーバーのビジネス・ルールおよびセレクトyを変更するたびに、システム・ログ、またはビジネス・ルールおよびセレクトy監査ロギングの構成時に指定した別のログに、更新内容が記録されます。

### ビジネス・ルールとセレクトyの監査ログの構成

ビジネス・ルールとセレクトyの新規作成、変更、および削除を追跡するログの値として、デフォルト値とは異なる値を使用するように、サーバーを構成できます。構成を変更すると、サーバー上のリソースを効率的に使用できます。

このタスクを実行するには、管理コンソールを使用している必要があります。

サーバーを実稼働環境で一定期間実行した後で、ビジネス・ルールとセレクトyの監査ログのためにサーバーが使用しているデフォルト値を調整する必要があると判断することがあります。

1. 「ビジネス・ルールとセレクトy監査」ページに移動します。このページにナビゲートするには、「サーバー」>「アプリケーション・サーバー」>*serverName*を選択します。
2. 「ビジネス・インテグレーション」-「追加プロパティ」の下で、「ビジネス・ルールとセレクトy監査」を選択します。

3. 必要な変更のタイプに応じて、以下のいずれかを実行します。

変更のタイプ	アクション
即時	<ol style="list-style-type: none"> <li>「ランタイム」タブを選択します。</li> <li>必要な変更をページに入力します。</li> <li><b>オプション:</b> 永続的に変更するには、「構成へのランタイム変更も保管」を選択して、変更内容をリポジトリにコピーします。</li> <li>変更を適用してから前の画面に戻るには、「OK」をクリックします。変更の適用後もこの画面を表示したままにするには、「適用」をクリックします。</li> </ol>
遅延	<ol style="list-style-type: none"> <li>「構成」タブを選択します。</li> <li>必要な変更をページに入力します。</li> <li>変更を適用してから前の画面に戻るには、「OK」をクリックします。変更の適用後もこの画面を表示したままにするには、「適用」をクリックします。</li> <li>変更内容を有効にするには、サーバーを再始動します。</li> </ol>

指定した属性が監査ログに適用されます。

### コマンドを使用したビジネス・ルールとセレクターの監査の構成

サーバーの実行中に特性を変更する必要がある場合には、コマンドを使用して、ビジネス・ルールとセレクターの監査を構成します。

これらのコマンドは、サーバーのコマンド行環境から実行する必要があります。

ビジネス・ルールとセレクターを監査するサーバーの数を変更する必要があるにもかかわらず、関連するサーバーを再始動できない場合があります。コマンド行を使用して、サーバーの構成をバッチ・モードで自動化できます。以下の手順は、コマンドを使用して 1 つのサーバーを構成する方法を示しています。

**重要:** これらの設定は、サーバーを再始動した場合には保管されません。これらのコマンドを入力した後で構成を保管するには、管理コンソールを使用する必要があります。(「サーバー」>「アプリケーション・サーバー」>*servername*>「ビジネス・インテグレーション」>「ビジネス・ルール」>「ビジネス・ルールとセレクター監査」>「ランタイム」または「サーバー」>「アプリケーション・サーバー」>*servername*>「ビジネス・インテグレーション」>「セレクター」>「ビジネス・ルールとセレクター監査」>「ランタイム」)

**注:** 以下の手順では、サーバー *server1* を構成します。ステップ 2 (137 ページ) 以降のすべてのステップは、jacl スクリプトに組み込んで実行できます。

1. 管理環境に入ります。

```
wsadmin
```

2. 監査ロギングを構成するか、既存の構成を変更するかを決定します。

タスク	コマンド
監査ロギングの構成	set mbean [\$AdminControl queryNames *:*,name=CustomizationAuditMBean,process=server1]
監査ロギング構成の変更	set auditconfig [\$AdminConfig list AuditLog]

3. 該当するコマンドを入力します。

### 監査ロギングを構成または変更するコマンド

**重要:** 既存の構成を変更するコマンドを入力する場合は、変更内容を保管する必要があります。また、サーバーを再始動しないと、変更内容は有効になりません。

以下は、入力可能なコマンドです。

#### **\$AdminControl invoke \$mbean getSeparateAuditLogEnabled**

別個の監査ログへのロギングを実行するかどうかを判別するために使用します。

#### **\$AdminControl invoke \$mbean setSystemOutAuditLogEnabled {boolean}**

SystemOut.log ファイルへのロギングを使用可能または使用不可にするために使用します。Boolean には、true または false を指定できます。

#### **\$AdminControl invoke \$mbean getSeparateAuditLogFileName**

別個の監査ログのファイル名を判別するために使用します。

#### **\$AdminControl invoke \$mbean setSeparateAuditLogFileName {filename}**

新規のログ・ファイルの名前 (MyAudit.log など) を設定するために使用します。

#### **\$AdminControl invoke \$mbean getSeparateAuditLogFileRolloverSize**

監査ログのサイズを判別するために使用します。

#### **\$AdminControl invoke \$mbean setSeparateAuditLogFileRolloverSize integer**

システムが監査ログをヒストリー・ファイルへロール・オーバーする条件となる、監査ログのサイズを設定するために使用します。サイズはメガバイト単位です。

#### **\$AdminControl invoke \$mbean getSeparateAuditLogFileMaxNumberOfBackupFiles**

監査ログ・ヒストリー・ファイルの数を判別するために使用します。

#### **\$AdminControl invoke setSeparateAuditLogFileMaxNumberOfBackupFiles integer**

監査ログ・ヒストリー・ファイルの数を設定するために使用します。

#### **\$AdminControl invoke \$mbean setSeparateAuditLogEnabled {boolean}**

別個のログ・ファイルへのロギングを開始または停止するために使用します。Boolean には、true または false を指定できます。

#### **\$AdminConfig showall \$auditconfig**

現在の監査ログ構成を表示するために使用します。

#### **\$AdminConfig modify \$auditconfig {{separateAuditLogEnabled true}}**

別個の監査ログへのロギングを使用可能にするために使用します。

**\$AdminConfig modify \$auditconfig {{systemOutAuditLogEnabled false}}**  
system.Out ファイルへの監査を使用不可にするために使用します。

**\$AdminConfig modify \$auditconfig {{customAuditLog {{maxNumberOfBackupFiles 7} {rolloverSize 7}}}}**  
監査ログ・ヒストリー・ファイルの数と監査ログ・ファイルのサイズを変更するために使用します。

**\$AdminConfig modify \$auditconfig {{customAuditLog {{fileName MyAudit.log}}}}**  
監査ログ・ファイルの名前を変更するために使用します。

**\$AdminConfig save**  
構成を保管するために使用します。

これらの変更内容を保管する場合は、管理コンソールに移動するか (「サーバー」 > 「アプリケーション・サーバー」 > *servername* > 「ビジネス・インテグレーション」 > 「ビジネス・ルール」 > 「ビジネス・ルールとセクター監査」 > 「ランタイム」 または 「サーバー」 > 「アプリケーション・サーバー」 > *servername* > 「ビジネス・インテグレーション」 > 「セクター」 > 「ビジネス・ルールとセクター監査」 > 「ランタイム」)、または \$AdminConfig save と入力します。

## スタンドアロン・サーバー用ビジネス・ルール動的リポジトリのインストール

インストールおよび構成スクリプトの実行時に、動的リポジトリがスタンドアロン・サーバーにインストールされます。

ビジネス・ルールをサポートしないデータベースを選択すると、システムはデフォルトのデータベースである Cloudscape を使用します。

DB2 for z/OS バージョン 7 では、基本キーのサイズが 255 バイトに制限されます。ビジネス・ルールやセクターの成果物用の動的成果物リポジトリでは、ターゲットのネーム・スペース、名前、タイプのいずれかを使用して基本キーを生成します。DB2 for z/OS バージョン 7 を使用するようにシステムを構成した場合は、名前を以下のように制限する必要があります。

- ターゲットのネーム・スペース = 170 バイト
- 名前の最大長 = 75 バイト
- タイプの最大長 = 10 バイト (この値はランタイム・システムにより設定されます)

注: DB2 for z/OS バージョン 8 では、この制限は適用されません。

## Network Deployment 用ビジネス・ルール動的リポジトリのインストール

ビジネス・ルールが含まれているアプリケーションをインストールするには、事前にビジネス・ルール用の動的リポジトリをインストールしておく必要があります。

動的リポジトリでは、中央構成がサポートされています。この構成では、同一のリポジトリを使用するようにすべてのサーバーを構成できるので、アプリケーションはすべて同じデータを使用できます。これは、ビジネス・ルール・マネージャ

ー・ツールを使用してこのデータを動的に更新するユーザーにとって重要です。中央リポジトリにより、以下を実行できるようになります。

- 一度だけ行われる変更
- すべてのサーバー・インストールで有効になる変更

DB2 for z/OS バージョン 7 では、基本キーのサイズが 256 バイトに制限されます。ビジネス・ルールやセクターの成果物用の動的成果物リポジトリでは、ターゲットのネーム・スペース、名前、タイプのいずれかを使用して基本キーを生成します。DB2 for z/OS バージョン 7 を使用するようにシステムを構成した場合は、名前を以下のように制限する必要があります。

- ターゲットのネーム・スペース = 171 バイト
- 名前の最大長 = 75 バイト
- タイプの最大長 = 10 バイト (この値はランタイム・システムにより設定されます)

注: DB2 for z/OS バージョン 8 では、この制限は適用されません。

### **ビジネス・ルールおよびセクターを含むモジュールの考慮事項**

このトピックでは、ビジネス・ルールおよびセクターを含むモジュールをインストールまたは削除する場合に考慮すべき情報について説明します。

ビジネス・ルールおよびセクターによって、モジュールの柔軟性が増します。柔軟性が増したことにより、モジュールをインストールまたは削除する方法に影響が及びます。サーバーが、中央リポジトリにビジネス・ルールおよびセクターを保管しているためです。

### **ビジネス・ルールまたはセクターの変更に関する考慮事項**

ビジネス・ルールおよびセクターは、影響があるモジュールを再構成および再インストールすることなく、実稼働環境で変更できます。これらの変更は、直接リポジトリに対して行われ、ビジネス・ルールまたはセクターを含むファイルにはコピーされません。ビジネス・ルールまたはセクターを変更後、そのビジネス・ルールまたはセクターをエクスポートし、開発環境にインポートしてください。ビジネス・ルールおよびセクターのエクスポートおよびインポートに習熟していない場合は、これらのタスクについて説明しているトピックを参照してください。

### **ビジネス・ルールまたはセクターを含むモジュールの置換に関する考慮事項**

ビジネス・ルールまたはセクターを含むモジュールを置換する場合、サーバーはリポジトリ内のビジネス・ルールおよびセクターのコピーを上書きします。動的に行ったすべての変更は、モジュールを置換するときに失われます。変更を失わないようにするには、モジュールが使用するビジネス・ルールおよびセクターをエクスポートして、開発環境に再度インポートし、実動システム上のモジュールを置換する前に、そのモジュールを再ビルドしてください。

あるモジュールによって実装されているビジネス・ルールまたはセクターを変更した場合、サーバー内で稼働中のその他のモジュールには、そのビジネス・ルールまたはセクターの現在のコピーが必要です。この場合、更新したモジュールをサ

サーバーにインストールする時に、このモジュールが他のモジュールに影響を与えないよう、別のリポジトリを構成する必要があります。『環境の構成』トピックでは、データベースの構成について説明します。

## ビジネス・ルールまたはセクターを含むモジュールの削除に関する考慮事項

ビジネス・ルールまたはセクターを含むモジュールをサーバーから削除しても、サーバーはビジネス・ルールおよびセクターをリポジトリから削除しません。サーバーは、他のアプリケーションまたはモジュールがこれらのビジネス・ルールを必要としているかどうかを判別できないため、これらの成果物を保持します。

ビジネス・ルールまたはセクターを必要とするものがないと判断した場合は、リポジトリから除去してください。『リポジトリからのビジネス・ルールおよびセクター・データの除去』では、不要なビジネス・ルールまたはセクターの消去方法について説明します。

## データベース構成に関する考慮事項

ビジネス・ルールやセクターの成果物用の動的成果物リポジトリでは、ターゲットのネーム・スペース、名前、タイプのいずれかを使用して基本キーを生成します。DB2 for z/OS バージョン 7 では、基本キーのサイズが 255 バイトに制限されます。

DB2 for z/OS バージョン 7 を使用するようにシステムを構成した場合は、名前を以下のように制限する必要があります。

- ターゲットのネーム・スペース = 170 バイト
- 名前の最大長 = 75 バイト
- タイプの最大長 = 10 バイト

注: DB2 for z/OS バージョン 8 では、この制限は適用されません。

### リポジトリからのビジネス・ルールおよびセクターのデータの除去:

ビジネス・ルールやセクターを使用するアプリケーションをアンインストールする場合、サーバーはリポジトリからその成果物を除去しません。このタスクによって、リポジトリからビジネス・ルールおよびセクターの不要な成果物を除去します。

除去しようとしているビジネス・ルールまたはセクターを使用するアプリケーションのすべてのコピーを、すべてのサーバーから確実にアンインストールしてください。

ビジネス・ルールまたはセクターの成果物を含むアプリケーションをインストールする場合、サーバーはこれらの成果物をデータベース・テーブルに格納するため、アプリケーションを変更しなくてもその成果物を動的に更新できます。また、これによって、他のサーバーはその成果物を共有できます。アプリケーションをアンインストールする場合、サーバーはデータベース・テーブルからその成果物を自動的に除去しません。アプリケーションがまだ別のサーバーにインストールされていて、実行されている可能性があるためです。その場合、データベースから成果物



を削除してしまうと、アプリケーションの別の実行中のコピーがビジネス・ルールまたはセレクターの使用を試行したときにそれらのアプリケーションは失敗します。

未使用の成果物をデータベースから削除するには、その成果物を使用するすべてのアプリケーションをアンインストールした後で、手動で削除する必要があります。リポジトリのデータベース・プラットフォームで提供されるツールを使用して成果物を除去します。

1. データベースのロケーションを探します。

データベースのロケーションは、データベース・プラットフォームにより異なります。

#### データベース・プラットフォーム

##### Cloudscape

##### その他のデータベース

#### ロケーション

WASHOME¥cloudscape¥databases¥RepositoryDB  
サーバーのインストールおよび構成時に構成されたロケーションにより異なります。例えば、サーバーを自動的に構成し、デフォルト・データベース名を選択した場合、データベースの名前は WPSDB となります。

2. 行を削除する以下のデータベース・テーブルのロケーションを探します。

#### BYTESTORE

ビジネス・ルールおよびセレクターの成果物を含むメイン・テーブル

#### BYTESTOREOVERFLOW

メイン・テーブル用のオーバーフロー・テーブル

#### APPTIMESTAMP

ビジネス・ルールおよびセレクターの成果物を含むインストール済みアプリケーション

3. アプリケーション用の成果物を削除します。

ご使用のデータベース・プラットフォーム用のツールを使用して、所定のアプリケーション用のすべてのビジネス・ルールおよびセレクターの成果物を削除するには、以下のステップを実行します。

- a. BYTESTORE テーブルで、**APPNAME** 列の値がアプリケーション名と同じ行をすべて検索します。
- b. 検索されたすべての行の基本キー列の値を記録します。BYTESTORE テーブルの基本キー列は、**ARTIFACTTNS**、**ARTIFACTNAME**、および **ARTIFACTTYPE** です。
- c. BYTESTORE テーブルから、ステップ 3a で検索された行を削除します。
- d. ステップ 3b で記録した基本キー値の各セットに対して、対応する列に同じ値を持つ BYTESTOREOVERFLOW テーブルの行を探します。

注: 基本キー値の各セットに対して、BYTESTOREOVERFLOW テーブルに対応する行がない場合や、1 つ以上対応する場合があります。

- e. BYTESTOREOVERFLOW テーブルから、ステップ 3d で検索された行を削除します。

- f. **APPNAME** 列がアプリケーションの名前と同じである **APPTIMESTAMP** テーブルの行を削除します。

データベース・テーブルからビジネス・ルールおよびセクターの不要な成果物を除去しました。

## ビジネス・ルールの概要

ビジネス・ルールを使用して、ビジネス・プラクティスの振る舞いを制御します。(ビジネス・ルールの構築、デプロイについての詳細は、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。)

## ビジネス・ルールとは

ビジネス・プラクティスに構造を与えたり、ビジネス・プラクティスの振る舞いを制御したりするものはすべてビジネス・ルールです。ルールは、ビジネス・ポリシーの適用、組織内での共通のガイドラインの確立、ビジネス環境でのアクセスの制御を実行できます。

## ビジネス・ルールを使用する場合

ビジネス・ルールは、頻繁に変わるビジネス・プラクティス (ビジネスに伴って当然生じる場合もあれば、監督官庁などの外部から要求される場合もある) に対応するために使用します。ビジネス・ルールの標準的な使用法は次のとおりです。

- 現在の金利を確認する
- 製品の割引を計算する
- 消費税を計算する
- 高齢者や得意先など特別なグループを判別する

## ビジネス・ルールの使用法

ビジネス・ルールの作成および変更は、Eclipse ベースの WebSphere Integration Developer ツールを使用して行います。ビジネス・ルール値の管理および変更は、Web ベースのビジネス・ルール・マネージャー・ツールを使用して行います。

### ビジネス・ルール・コンポーネントの表示:

ビジネス・ルール・グループを管理する際には、まず、ビジネス・ルール・コンポーネントを表示します。その表示画面から、ビジネス・ルール・グループの一部またはすべてをエクスポートまたはインポートしたり、ビジネス・ルール・グループを構成するテーブルを表示したりすることができます。

このタスクを実行するには、ユーザーは WebSphere Process Server の管理コンソールを使用していなければなりません。

このタスクを実行して、サーバー上に存在するビジネス・ルール・グループを判別します。

1. 管理コンソールで「**サーバー**」>「**アプリケーション・サーバー**」を選択します。
2. **サーバー名** をクリックして、ビジネス・ルールを表示するサーバー・リストからサーバーを選択します。

3. 「Business Integration」の下の「ビジネス・ルール」をクリックします。

コンソールに、各グループの説明を使用して定義されたすべてのビジネス・ルール・コンポーネントのリストが表示されます。

#### 管理コンソールを使用したビジネス・ルールのエクスポート:

ビジネス・ルール・コンポーネントをエクスポートすることで作成されたファイルを、開発環境にインポートします。これにより、開発の成果物を実動システムの成果物と同期させることができます。

このタスクを開始するには、『ビジネス・ルール・コンポーネントの表示』に説明されているように、ビジネス・ルール・コンポーネントが既に表示されている必要があります。

ビジネス・ルール・テーブルを変更したら、ビジネス・ルール・コンポーネントをエクスポートして、開発環境と実稼働環境を同期させます。このタスクは、ビジネス・ルール・コンポーネント表示画面から開始します。

**ヒント:** コマンド行を使用してビジネス・ルールをエクスポートすることもできます。『`exportBusinessRuleArtifacts.jacl` コマンド』を参照してください。

1. エクスポートするビジネス・ルール・グループを選択します。

ビジネス・ルール・グループの横のチェック・ボックスをクリックしてから、「エクスポート」をクリックします。ブラウザには、選択したビジネス・ルール・グループへの HTML リンクのリストが表示されます。(これは「エクスポートするビジネス・ルール」パネルです。) 各ビジネス・ルール・グループには、ファイル拡張子 `.zip` が付加されています。

2. ファイルをダウンロードします。

各ファイル名をクリックすると、プロンプトが出てファイルを保管するように要求されます。プロンプトが出たら、「OK」をクリックして、ファイル・システムにファイルを保管します。

**注:** ダウンロードするときファイル名を変更することもできます。

3. ビジネス・ルール表示パネルに戻ります。

「戻る」をクリックして、ビジネス・ルール・グループのリストに戻ります。

指定した場所にファイルが保管されます。その後、そのファイルをテスト・システムにコピーできます。

このファイルは WebSphere Integration Developer 環境にインポートする必要があります。WebSphere Integration Developer について詳しくは、インフォメーション・センターを参照してください。

#### 管理コンソールを使用したビジネス・ルールのインポート:

ビジネス・ルールをインポートすると、アプリケーションを再インストールしなくても、インストール済みのビジネス・ルールを更新できます。

管理コンソールで作業する必要があるため、エクスポート機能によって作成された圧縮ファイルのロケーションを知っている必要があります。

インストール済みのアプリケーションが使用しているビジネス・ルールを変更した後で、それらの変更内容を別のクラスターまたはサーバーにも適用したい場合に、ビジネス・ルールをインポートします。ご使用の開発環境と実稼働環境内の変更内容を同期させる場合にも、この機能を使用できます。

**ヒント:** ビジネス・ルールは、コマンド行を使用してインポートすることもできます。『importBusinessRuleArtifacts.jacl コマンド』を参照してください。

1. ビジネス・ルールのインポート先のサーバー上のビジネス・ルールを表示します。
2. 「インポート」をクリックします。
3. 「ビジネス・ルールのインポートの準備」パネル上でファイルへのパスを指定します。

ビジネス・ルールを表示して、変更されたルールを確認します。

## ビジネス・ルール・マネージャー

ビジネス・ルール・マネージャーは、ビジネス・アナリストがビジネス・ルール値を表示および変更する際に使用する Web ベースのツールです。このツールは WebSphere Process Server のオプションであり、サーバーの初期インストール後にインストールします。

ビジネス・ルールは、if/then ルール・セットおよびデシジョン・テーブルを使用して操作を実装することにより、WebSphere Integration Developer で設計および開発します。ルール・セットおよびデシジョン・テーブルはテンプレートに設定します。テンプレートは、変更可能なビジネス・ルールの側面と変更の程度を制御します。それらによって、デシジョン・テーブルの if/then ルール、条件ケース、およびアクションの構造を定義します。

テンプレートは、ビジネス・ルール・マネージャーでのビジネス・ルール・ランタイム・オーサリングのメカニズムを提供します。テンプレートを使用すると、ビジネス・ルールの値を変更したり、ルール・セット内に新しいルールを作成したり、デシジョン・テーブル内に新しい条件またはアクションを作成したり、実行時にビジネス・ルール定義に対する変更を公開したりできます。

ビジネス・ルールはビジネス・ルール・グループとして編成されます。これらはルール・ブックを構成します。ビジネス・ルール・グループは、ルールに対するインターフェースとして、ルールを呼び出すために使用されます。ルール・セットおよびデシジョン・テーブルが直接呼び出されることはありません。

ビジネス・ルールの構築、デプロイについての詳細は、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。

### ビジネス・ルール・マネージャーが機能する仕組み:

ビジネス・ルール・マネージャーは、ビジネス・アナリストがランタイム・ルール・オーサリングに使用する主要な WebSphere Process Server ツールです。

ビジネス・ルール・マネージャーを使用して、以下のタスクを実行します。

- リポジトリからビジネス・ルールのコピーを開く
- ビジネス・ルールを参照および編集する
- リポジトリへビジネス・ルールを公開する

図 8 は、ビジネス・ルール・マネージャーがルールを呼び出し、公開する仕組みを示しています。

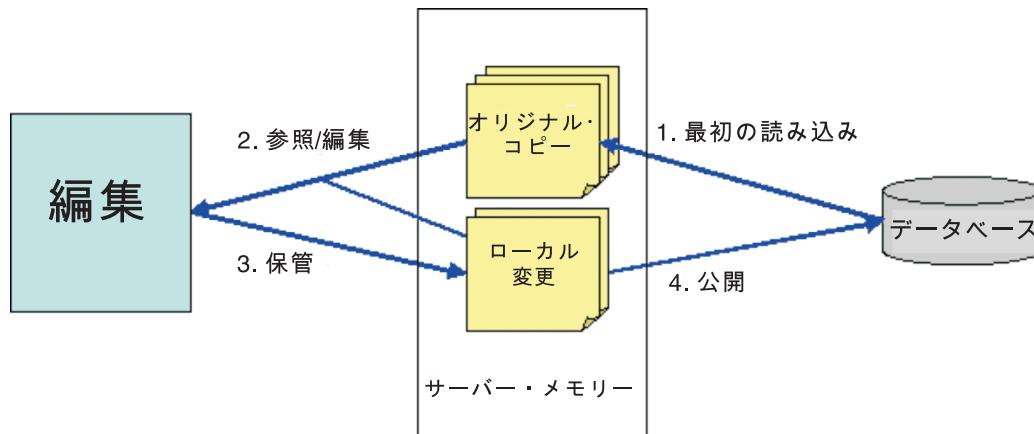


図 8. ビジネス・ルール・マネージャーのイベント・シーケンス

ビジネス・ルール・マネージャーにログオンした後に、ビジネス・ルールを変更すると、以下のイベントが発生します。

1. ビジネス・ルールを選択すると、ビジネス・ルール・マネージャーは、データベースからルールにアクセスし、それをオリジナルのコピーとしてサーバー・メモリーに保管します。
2. これで、ルールを参照または編集できるようになります。
3. ビジネス・ルールをコピーとしてローカル・サーバー・メモリーに保管します。
4. ローカル・コピーを再びデータベースに公開します。あるいは、サーバーに対する更新を実行せずに、変更を取り消すことも可能です。

#### ビジネス・ルール・マネージャーのインストール:

ビジネス・ルール・マネージャーは、WebSphere Process Server 用のエンタープライズ・アプリケーションとしてインストールされるため、ランタイム中にビジネス・ルールを管理できます。インストールには管理コンソールを使用できます。または、オプションとして、Windows、Linux、iSeries、または z/OS 用の .jac1 スクリプトを使用できます。

**必要なセキュリティー・ロール:** ビジネス・ルール・マネージャーをインストールするには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールからビジネス・ルール・マネージャーをインストールするには、「ビジネス・ルール・マネージャーの構成」ページの「ビジネス・ルール・マネージャーのインストール」チェック・ボックスを選択します。

- a. 管理コンソール内から、「サーバー」を展開し、「アプリケーション・サーバー」をクリックします。
  - b. サーバーの名前をクリックします。
  - c. 「構成」ページの「ビジネス・インテグレーション」で、「ビジネス・ルール」を展開し、「ビジネス・ルール・マネージャーの構成」をクリックします。
  - d. 「一般プロパティ」から、「ビジネス・ルール・マネージャーのインストール」チェック・ボックスを選択します。
  - e. 「コンテキスト・ルート」フィールドで、デフォルト・コンテキスト・ルートの `br/webclient` を受け入れるか、またはビジネス・ルール・マネージャー URL のカスタム・コンテキスト・ルートを入力します。
  - f. 「OK」をクリックします。
2. オプション: コマンド行を使用してビジネス・ルール・マネージャーを手動でインストールする場合 (管理コンソールを使用しない場合) は、最初に WebSphere Process Server が始動していることを確認する必要があります。 `WPS_HOME/bin` ディレクトリーに移動し、ご使用のオペレーティング・システムのシェルまたはコマンド・プロンプトを開き、ご使用のオペレーティング・システムに固有のインストール・コマンドを実行します。
- Windows で実行するコマンド: `wsadmin.bat -f installBRManager.jacl [-s servername -n nodename | -cl clustername] -c cellname -r rootname -a applicationname。`
  - Linux で実行するコマンド: `./wsadmin.sh -f ./installBRManager.jacl [-s servername -n nodename | -cl clustername] -c cellname -r rootname -a applicationname。`
  - iSeries で実行するコマンド: `wsadmin -f ./installBRManager.jacl [-s servername -n nodename | -cl clustername] -c cellname -r rootname -a applicationname。`
  - z/OS で実行するコマンド: `wsadmin.sh -f ./installBRManager.jacl [-s servername -n nodename | -cl clustername] -c cellname -r rootname -a applicationname。`

各部の意味は、次のとおりです。

*servername*

アプリケーション・サーバーの名前。

Network Deployment 構成内でクラスターが指定されていない場合には、引数のペア「-s servername」が必要です。

*nodename*

インストール・ノードの名前。

Network Deployment 構成内でクラスターが指定されていない場合には、引数のペア「-n nodename」が必要です。

*clustername*

アプリケーションをインストールするクラスターの名前。

Network Deployment 構成内でサーバー名とノード名が指定されていない場合には、引数のペア「-cl clustername」が必要です。

**注:** ノードとサーバーを指定するか、またはクラスターを指定する必要があります。両方は指定しないでください。

*cellname*

インストール・セルの名前。

引数のペア「-c *cellname*」はオプションです。

*rootname*

アプリケーション・ルート・ディレクトリの名前。

引数のペア「-r *rootname*」はオプションです。欠落している場合、*rootname* のデフォルト値は「br/webclient」です。

*applicationname*

アプリケーションの名前。

引数のペア「-a *applicationname*」はオプションです。欠落している場合、*applicationname* のデフォルト値は「BusinessRulesManager」です。

**注:** WebSphere Process Server が単一サーバー環境で構成されている場合、これらの引数のペアのすべてはオプションです。WebSphere Process Server が Network Deployment 環境用に構成されている場合、以下の引数ペアのいずれかが必要です。

- -s *servername* と -n *nodename*
- または -cl *clustername*

他の引数ペアはオプションです。

**注:** 引数が欠落している場合のデフォルト値は、以下のとおりです。

- 「server1」 = *servername*
- 「br/webclient」 = *rootname*
- 「BusinessRulesManager」 = *applicationname*

**ビジネス・ルール・マネージャーのインストールの考慮事項:**

ビジネス・ルール・マネージャーを分散環境で使用する場合、セル、ノード、クラスターの概念、および実行時のパフォーマンスを最適にするビジネス・ルール・マネージャーのセットアップ方法を理解する必要があります。

アプリケーション・サーバーは、セル、ノード、およびサーバーの概念に基づいて編成されます。スタンドアロン・サーバー構成では、1つのセルには1つのノードが含まれ、各ノードには1つのサーバーが含まれます。システム管理アプリケーションとユーザー・アプリケーションはすべて同じサーバーで稼働します。スタンドアロン・サーバー構成では、ビジネス・ルール・マネージャーを同じアプリケーション・サーバーにインストールできます。また、ビジネス・ルール・マネージャーにはデフォルトの URL からアクセスできます。

分散サーバー構成では、1つのセルに複数のノードが含まれ、各ノードに複数のアプリケーション・サーバーが含まれるように構成できます。各セルは、単一の管理ドメインを構成します。この構成では、ドメイン全体で中央管理、ワークロード管理、およびフェイルオーバー構成を使用できます。

分散サーバー構成でパフォーマンスを最適にするには、ビジネス・ルール・マネージャーを管理デプロイメント・ターゲット（ビジネス管理サービスが中央にホストされているセル内のアプリケーション・サーバー）にインストールします。通常、このサーバーは Common Event Infrastructure サービスをホストするサーバーです。

セル内では、ビジネス・ルール・データベースが同じセル内のすべてのサーバーで使用され、共有されます。ビジネス・ルール・データベースは、中央で管理されるビジネス・ルール情報用のストレージです。ビジネス・ルール・データベースにアクセスすると、ビジネス・アプリケーションがどの場所にインストールされているかにかかわらず、すべての動的ビジネス・ルール成果物の定義にアクセスできます。

実行時にセル内のすべてのビジネス・ルールが使用できるこの中央ストレージがあることによって、セル内のどのアプリケーション・サーバーにもビジネス・ルール・マネージャーをデプロイできます。また、ビジネス・ルール・マネージャーは、セル内のすべてのビジネス・ルールの一貫性のあるビューを提供します。ただし、高可用性を考慮すると、ビジネス・ルール・マネージャーを管理デプロイメント・ターゲット（ビジネス管理サービスが中央でホストされているセル内の専用アプリケーション・サーバー）にデプロイすることを、システム管理者にお勧めします。管理デプロイメント・ターゲット・サーバーは、Common Event Infrastructure サービスやその他のビジネス管理アプリケーションがインストールされているのと同じサーバーです。この構成では、高可用性が必要な場合、管理デプロイメント・ターゲット・サーバーをクラスター化して、アプリケーション・ユーザーに拡張が容易なソリューションを提供できます。

#### **ビジネス・ルール・マネージャーのサーバー・セキュリティの構成:**

サーバーでセキュリティを使用する場合は、ビジネス・ルール・マネージャーを使用しているサーバーを構成する必要があります。サーバーでセキュリティを使用可能に設定しない場合は、追加の構成なしでビジネス・ルール・マネージャーを使用できます。

1. ユーザー ID にセキュリティを設定します。

ユーザー ID の作成時には、グローバル・セキュリティを設定する必要があります。ユーザー ID を作成して、それらを BusinessRuleUser または business analyst のロールにマップします。各 ID にロールを割り当てる必要があります。

2. Cookie を使用してセッションを追跡するように、セッション・トラッキング・メカニズムを設定します。
3. 最低でも、適切なセッション・タイムアウト値を必ず設定します。
4. **オプション:** サーバーにセキュリティを設定します。

異なるロールまたはユーザー ID がある場合、サーバーの構成時にグローバル・セキュリティを設定する必要があります。

#### **ビジネス・ルール・マネージャーを正常に動作させるための Web ブラウザーの構成:**



サーバーは、ビジネス・ルール・マネージャーのインストール中に自動的にクライアントを構成しますが、ビジネス・ルール・マネージャーが正しく動作するように、Web ブラウザーが正しく構成されていることを確認する必要があります。

1. Web ブラウザーでスクリプトが使用可能であることを確認します。

ビジネス・ルール・マネージャーが動作するには、スクリプトが機能する必要があります。

2. Cookie が使用可能であることを確認します。

ビジネス・ルール・マネージャーを使用する場合、必要に応じてセッションの追跡に Cookie が使用されます。このため、セッションを追跡できるように、ブラウザで Cookie を使用可能にします。Cookie を使用可能にする場合、必ずシステム管理者に連絡してください。

### ビジネス・ルール・マネージャーへのアクセス:

ビジネス・ルール・マネージャーには、Web ブラウザーを使用してアクセスします。

サーバーとクライアントが両方とも正しく構成されていることを確認してください。

ビジネス・ルール・マネージャーにアクセスできるデフォルトの URL は、以下のとおりです。URL は環境によって異なる場合があります。

- <https://hostname:9443/br/webclient> (セキュリティーが使用可能である場合)
- <http://hostname:9080/br/webclient> (セキュリティーが使用不可である場合)

ここで、hostname はホストの名前です。

**注:** サーバーのグローバル・セキュリティーが使用可能な場合に限り、「ログイン」ページが開きます。グローバル・セキュリティーが使用不可の場合に、ビジネス・ルール・マネージャーにアクセスすると、「ルール・ブック」ページが開きます。

グローバル・セキュリティーが使用可能な場合、ログインするには、以下のステップを実行します。

1. 「ログイン」ページで、「**ユーザー ID**」を入力します。
2. 「**パスワード**」を入力します。
3. 「**ログイン**」をクリックします。

ビジネス・ルール・マネージャーの初期ページが開き、ナビゲーション・ペインに既存のルール・ブックがリスト表示されます。

ビジネス・ルール操作をブラウズおよび編集し、ビジネス・ルールをテンプレート化できます。

「**ルール・ブック**」ページおよび「**ビジネス・ルール・マネージャー**」ページのレイアウト:

ビジネス・ルール・マネージャーが開くと、「ルール・ブック」ページが表示されます。ここでは、ビジネス・ルール・グループと定義済みの操作が表示されます。

「ルール・ブック」ページはナビゲーションの第 1 レベルです。このページのレイアウトには、他のビジネス・ルール・マネージャー・ページにも共通する多くのエレメントが含まれます。

## ツールバー

ツールバーの内容は以下のとおりです。

### ようこそ

ユーザー名が表示されます。

### ユーザー識別

「ウェルカム・ユーザー名 (Welcome User Name)」で始まる現在のユーザー名を示します。

### ログアウト

グローバル・セキュリティーが使用可能である場合に、「ログイン」ページを開きます。

**重要:** 公開せずにログアウトしようとする、ポップアップ・ダイアログに確認メッセージが表示されます。

**ヘルプ** WebSphere Process Server インフォメーション・センターのビジネス・ルールに関するトピックにアクセスできます。

## ナビゲーション・ペイン

ナビゲーション・ペインは、ページの左側にあるペインです。「公開および復帰」ページおよび使用可能なルール・ブックへのアクセスを提供します。このナビゲーション・ツリーを使用して、必要なルール・レベルにドリルダウンできます。

**注:** ナビゲーション・ペインは、編集モードのページには表示されません。

### 公開および復帰

「公開および復帰」ページを開きます。このページでは、データベースにルール変更を公開したり、データベースにあった元のコピーにルールを復帰したりすることができます。

### ルール・ブック

「ルール・ブック」ページを開きます。このページはブラウザのトップレベルです。ルール・ブックはナビゲーション・ツリーにリストされます。ルール・ブック名の横にある正符号 (+) または負符号 (-) をクリックするとルール・ブックが展開または縮小され、すべてのルールが表示されます。左のペインのナビゲーション・ツリーでルール・ブックを選択すると、右のペインにすべての子ルール・ページ (ビジネス・ルール操作) と関連するすべてのルール・セットやデシジョン・テーブルが表示されます。これらをクリックすると、対応する編集用ページが開きます。

## コンテンツ領域

ページの右側ペインはコンテンツ領域であり、メインの表示および編集領域です。ここには「タイトル」セクション、「一般情報」セクション、およびページ固有のセクションがあります。

**注:** コンテンツ領域に表示される情報は、「ルール・ブック」ページ、「ルール・セット」ページ、「デシジョン・テーブル」ページのいずれを表示しているかによって異なります。

### 「タイトル」セクション

「タイトル」セクションには以下の情報があります。

#### パス情報

ページに対するパス (ルール・ブックおよび「ルール」ページの名前など) を以下の形式で表示します。

BusinessRuleGroup01 > Table1\_operation1

**例:** CalculateDiscountBRG > CalculateDiscount

#### ルール・タイトル

ビジネス・ルールのリソース名およびタイプを以下の形式で表示します。

Ruleset112 - Ruleset

**例:** calculateDiscount-Rule Page、CalculateDiscountRS - Ruleset

#### 機能ボタン

ページに表示される機能ボタンは、各ページの機能によって異なります。ページで使用可能な機能ボタンを以下の表にリストします。

表 14. ボタン・リスト

ボタン名	機能
戻る	前のページに戻ります。
編集	「ルール」ページ、デシジョン・テーブル、またはルール・セットの編集を使用可能にします。
保管	変更内容をローカル・コピーに保管して、前のページに戻ります。稼働中のサーバーの状態は変更されていないことに注意してください。サーバーの状態を変更する方法については、「公開」を参照してください。
キャンセル	リソースへの変更を破棄して前のページに戻ります。
コピー	新規デシジョン・テーブルまたはルール・セットを作成するため、デシジョン・テーブルまたはルール・セットをコピーします。新規ルールを作成するには、既存ルールをコピーしてから、値を変更する必要があります。
公開	サーバーに「ルール」ページを公開します。

表 14. ボタン・リスト (続き)

ボタン名	機能
復帰	ローカル環境に保管されたルールに対する変更をすべて取り消し、データベースに存在する元のコピーにルールを戻します。公開後にルールを復帰することはできません。

### メッセージ・フィールド

ルールに対してとられたアクションの状況またはエラーが発生したことを表示します。状況メッセージの例を以下に示します。

"calculateDiscount" が一時的に保管されました。

「公開および復帰」ページから変更を公開できます。

### 「一般情報」セクション

この「一般情報」セクションには以下の情報があります。

注: 「ルール・ブック」ページには、このセクションはありません。

#### 最後に公開

「ルール」ページ、ルール・セット、またはデシジョン・テーブルの最終公開日を表示します。

**状況** 「ルール」ページ、ルール・セット、またはデシジョン・テーブルが編集モードにあるか公開されたかを表示します。

**説明** 「ルール」ページ、ルール・セット、またはデシジョン・テーブルの要旨を表示します。

**制約事項:** ビジネス・ルール・マネージャーでビジネス・ルール・グループ・コンポーネントおよびビジネス・ルールの説明フィールドを編集するときには、CDATA タグを使用しないでください。使用すると、ビジネス・ルール・グループおよびビジネス・ルールが編集不能になってしまいます。CDATA タグが存在する場合は、XML エディターでビジネス・ルール・グループまたはビジネス・ルールを開き、説明フィールドから CDATA タグを手動で除去します。

### ページ固有情報のセクション

ページ固有情報セクションの内容は、「ルール・ブック」ページ、「ルール」ページ、「ルール・セット」ページ、「デシジョン・テーブル」ページのいずれを表示しているかによって異なります。これらの各ページに関する具体的な説明については、個々のトピックを参照してください。

「ルール・ブック」ページの場合、このセクションには以下の情報が表示されません。

#### ビジネス・ルール・リソース (Business Rule Resources)

「ルール」ページ、ルール・セット、およびデシジョン・テーブルの名前をリストします。

**説明** 「ルール」ページ、ルール・セット、およびデシジョン・テーブルの要旨または名前を表示します。

#### **アクション**

対応するビジネス・ルールのリストについて選択可能なアクションを表示します。最初は空になっていますが、グループを展開すると、それぞれのルールの横に「編集」ボタンが表示されます。

#### **「公開および復帰」ページ:**

「公開および復帰」ページは、ローカル環境に保管されているルールに対する変更内容をデータベースに公開するためのものです。また、ローカル環境に保管する前にデータベースにあった元のコピーにルールを戻すためのものです。

コンテンツ領域のページ固有情報セクションには、以下のエレメントがあります。

#### **「変更されたビジネス・ルール・リソース」セクション**

公開または復帰が可能なビジネス・ルール、ルール・セット、およびデシジョン・テーブルのリストを表示します。

#### **ビジネス・ルール・リソース (Business Rule Resources)**

「ルール」ページ、ルール・セット、およびデシジョン・テーブルの名前をリストします。公開の準備ができているリソースの横にはチェック・ボックスが表示されます。

**状況** リソースが元の状態であるかローカル環境で変更されたかを示します。

**説明** リソースの要旨を提供します。

#### **アクション**

復帰可能なリソースを示します。リソースのルールの「アクション」列に「復帰」ボタンがあります。

#### **「ルール・ブック」ページ:**

「ルール・ブック」ページには、WebSphere Process Server システム内のすべてのビジネス・ルール・リソースがリストされます。

「ルール・ブック」ページから、ビジネス・ルール・リソースを参照できます。

コンテンツ領域のページ固有情報セクションには、以下のエレメントがあります。

#### **「ルール・ブック」セクション**

ビジネス・ルール・グループに関連する「ルール」ページ、ルール・セット、およびデシジョン・テーブルのリストを表示します。

#### **ビジネス・ルール・リソース (Business Rule Resources)**

「ルール」ページ、ルール・セット、およびデシジョン・テーブルの名前をリストします。

**説明** 「ルール」ページ、ルール・セット、およびデシジョン・テーブルの要旨または名前を表示します。

## アクション

対応するビジネス・ルール・リストについて選択可能なアクションを表示します。最初は空になっていますが、グループを展開すると、それぞれのルールの横に「編集」ボタンが表示されます。

### 「ルール」ページ:

「ルール」ページには、ルール・ロジック選択レコードのビジネス・ルールの値を変更するためのインターフェースがあります。情報は表形式で表示されます。

「ルール」ページからは、ビジネス・ルールの有効な日付の表示、変更、追加、分割、または削除などの操作を実行できます。また、既存のビジネス・ルールをコピーすることにより、新しいビジネス・ルールを作成することもできます。

コンテンツ領域のページ固有情報セクションには、以下のエレメントがあります。

### 「ルール・ロジック選択レコード」セクション

ルールの構築ブロックである有効なビジネス・ルール・リストを提供します。ここでは、ビジネス・ルール・レコードの処理 (選択レコードの追加やソートなど) が可能です。

#### 開始日/時刻

特定の日付または「開始日無し」のいずれかを選択できます。

注: 「開始日無し」は、ターゲット・ルール・ロジックが終了日より前のすべての日に対して有効であることを意味します。

#### 終了日/時刻

特定の日付または「終了日無し (no end date)」のいずれかを選択できます。

注: 「終了日無し (no end date)」は、ルール・ロジックが開始日およびそれ以降のすべての日に対して有効であることを意味します。

#### 有効なルール・ロジック

対応する時間フレーム内で有効なルール・セットまたはデシジョン・テーブルを指定します。

#### アクション

ビジネス・ルール・レコードを分割および削除するためのオプションを提供します。

#### デフォルト・ルール・ロジック

他のルール・ロジックを適用できない場合のデフォルトのルール・ロジックを提供します。これは、他の選択レコードのいずれとも日付が一致しない場合に選択されます。

注: ビジネス・ルール・マネージャーのルール・ロジック選択日時の値は、現地時間 (Web ブラウザーを実行しているクライアントの時間帯を使用します)、または協定世界時 (UTC) のいずれかで指定できます。

## 使用可能なルール・ロジック

特定のビジネス・ルールとアクションに適用可能なルール・セットまたはデシジョン・テーブルのリストを、ビジネス・ルールに関連する説明と共に表示します。ここでは、ルールを容易に編集またはコピーできます。

### 「ルール・セット」 ページ:

「ルール・セット」 ページには、ビジネス・ルールのルール「インスタンス」とそれらの実行順序、およびそのルール・セットに関連付けられているテンプレートがリストされます。

「ルール・セット」 ページからは、テンプレートを使用して既存のルール・インスタンスを表示または編集したり、選択したテンプレートから新規ルール・インスタンスを作成したり、ルールの実行順序を指定したり、ルールまたはルール・セットの名前を変更したり、ルール・セットの説明を表示または編集したり、ルール・セットを作業用コピーとして保管したり、ルールを削除したりすることができます。

コンテンツ領域のページ固有情報セクションには、以下のエレメントがあります。

### 「ルール」 セクション

以下の情報と共に関連するルールのリストを表示します。

**名前** ルールの名前を提供します。

**ルール** ルールを定義する変数、制約、範囲、および列挙をリストします。

### アクション

ルールの再配列および削除が可能です。

### 「テンプレート」 セクション

既存のテンプレートを使用して新規ルールを容易に作成できます。ルールの以下の情報を指定するための以下のフィールドがあります。

### テンプレート名

既存のテンプレートの名前を提供します。

**名前** ルールの名前を入力するためテキスト域を提供します。

**ルール** ルールを定義する変数、制約、範囲、および列挙を指定するためテキスト域を提供します。

### アクション

ルールをテンプレートに追加できます。

### 「デシジョン・テーブル」 ページ:

「デシジョン・テーブル」 ページには、条件ケースとアクション、その方向 (行と列)、およびデシジョン・テーブルに関連付けられたテンプレートがあります。「デシジョン・テーブル」 ページは「ルール」 ページから開きます。

「デシジョン・テーブル」 ページでは、テンプレートを使用して既存の条件またはアクションの表示または編集、そのデシジョン・テーブルに定義されたテンプレートを使用した新規の条件の追加、条件の削除、条件の順序の変更、方向の変更、関

連付けられているテンプレートを使用した初期化アクション・ルールの変更、説明の表示または編集、デシジョン・テーブルの保管 (作業用コピーとして) が可能です。

コンテンツ領域のページ固有情報セクションには、以下のエレメントがあります。

### 「デシジョン・テーブル」セクション

テーブル内で、条件ケース (行および列の見出しで表します) とアクション (条件ケースの交点として表します) を提供します。「方向」アイコンを使用して、条件行の方向を水平方向から垂直方向、またはその逆に切り替えることができます。

#### 初期設定のルール

このデシジョン・テーブルの初期設定のルールを示します。初期設定のルールは、デシジョン・テーブル・ロジックが発行される前に直接呼び出されます。初期設定のルールを使用して、デシジョン・テーブルで使用される変数やアクションを初期化できます。ビジネス・ルール定義が WebSphere Integration Developer 内で初期化アクションを指定して設計されている場合にのみ、初期設定のルールが表示されます。

#### その他の場合

このデシジョン・テーブルの「その他の場合」条件を示します。「その他の場合」条件は、デシジョン・テーブル内のほかの条件が適用されない場合にデフォルトで入力される特別な条件です。「その他の場合」条件は、デシジョン・テーブル定義が WebSphere Integration Developer 内で「その他の場合」条件を指定して設計されている場合にのみ表示されます。ビジネス・ルール・マネージャーで、デシジョン・テーブルの「その他の場合」条件の列を動的に追加または除去することはできません。

### 「テンプレート」セクション

既存のテンプレートを使用して新規ルールを容易に作成できます。

#### ビジネス・ルール・レコードの操作:

ビジネス・ルール・レコードは、ルールの有効な日付やルールに関連付けられた if/then ルール・セットまたはデシジョン・テーブルなど、ルールに関する情報を識別します。

ビジネス・ルール・マネージャーを使用して、ビジネス・ルール・レコードの作成、変更、または削除を行います。

#### ビジネス・ルール・レコードの作成:

既存のレコードからビジネス・ルール・レコードを作成します。

ルールを作成するには、「編集」モードにする必要があります。

新規ビジネス・ルール・レコードを作成するには、以下のステップを実行します。

1. 「ルール」ページで、「選択レコードの追加」をクリックします。



「開始日/時刻」および「終了日/時刻」フィールドが「1月1日」に設定された新規ビジネス・ルール・レコードが、リストの下部に追加されます。また、日付/時刻フィールド値が無効であることを示すメッセージが「メッセージ」フィールドに表示されます。

2. 「開始日/時刻」フィールドを設定します。
  - a. ドロップダウン・リストから月を選択します。
  - b. ドロップダウン・リストから日を選択します。
  - c. 年を入力します。
  - d. 時刻を入力します (24 時間形式)。
3. 「終了日/時刻」フィールドを設定します。
  - a. ドロップダウン・リストから月を選択します。
  - b. ドロップダウン・リストから日を選択します。
  - c. 年を入力します。
  - d. 時刻を入力します (24 時間形式)。

**制約事項:** 任意の一時点で 1 つのルール・ロジックのみが有効です。複数のルールの日付/時刻の範囲を重複させることはできません。

**注:** 日付/時刻の範囲に間隔が空いても構いません。デフォルトのビジネス・ルールが指定されていた場合は、この間隔にそのルールが使用されます。ベスト・プラクティスとして、常にデフォルトのルールを指定してください。

4. ドロップダウン・リストから「**有効なルール・ロジック**」を選択します。
5. 「**保管**」をクリックします。

レコードが一時的に保管されたことや、変更内容を「公開および復帰」ページから公開できることを示すメッセージがメッセージ・フィールドに表示されます。

#### 関連タスク

166 ページの『ビジネス・ルール・レコードの削除』

既存のビジネス・ルール選択レコードを、ルール選択ロジック・テーブルから削除できます。選択レコードを削除した場合、それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

#### ビジネス・ルール・レコードの変更:

既存のビジネス・ルール選択レコードの日付および時刻の値を変更できます。

ルールを変更するには、「**編集**」モードにする必要があります。

ビジネス・ルールを変更するには、以下のステップを実行します。

1. 「ルール」ページで、選択レコードの「**開始日/時刻**」を編集します。
  - a. ドロップダウン・リストから月を選択します。
  - b. ドロップダウン・リストから日を選択します。
  - c. 年を入力します。

- d. 時刻を入力します (24 時間形式)。
2. 選択レコードの「終了日/時刻」を編集します。
- a. ドロップダウン・リストから月を選択します。
  - b. ドロップダウン・リストから日を選択します。
  - c. 年を入力します。
  - d. 時刻を入力します (24 時間形式)。

**制約事項:** 任意の一時点で 1 つのルール・ロジックのみが有効です。複数のルールの日付/時刻の範囲を重複させることはできません。

**注:** 日付/時刻の範囲に間隔が空いても構いません。デフォルトのビジネス・ルールが指定されていた場合は、この間隔にそのルールが使用されます。ベスト・プラクティスとして、常にデフォルトのルールを指定してください。

3. 「保管」をクリックします。

**注:** 「日付/時刻」フィールドが無効な場合、そのフィールドは赤色に変わり、さらに日付/時刻フィールド値が無効なことを示すメッセージがメッセージ・フィールドに表示されます。

レコードがローカル側で保管されます。レコードをデータベースに公開する準備ができました。詳しくは、167 ページの『ビジネス・ルールの公開と復帰』を参照してください。

ビジネス・ルールの日付の設定について詳しくは、159 ページの『ビジネス・ルールの日付の分割』を参照してください。

#### 関連タスク

166 ページの『ビジネス・ルール・レコードの削除』

既存のビジネス・ルール選択レコードを、ルール選択ロジック・テーブルから削除できます。選択レコードを削除した場合、それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

#### 日付/時刻の選択:

ビジネス・ルールを選択するには、日付/時間を指定します。

日付は、ビジネス・ルール・グループの演算パラメーターの一部として定義されるか、実行時に派生されます。日付は、常に UTC で表現され、特定の時点を示します。ある一時点で有効にできるビジネス・ルール・インプリメンテーションは、操作ごとに 1 つだけです。ある一時点に、他の有効なビジネス・ルールがない場合は、デフォルトのビジネス・ルールが使用されます。

ビジネス・ルール・グループは、以下の日付/時刻オプションをサポートします。これらのオプションにアクセスするには、「開始日/時刻」および「終了日/時刻」フィールドのアイコンをクリックします。

#### 「日付/時刻の指定」

日付を手動で指定します。

### 「連続」

終了日を選択レコードよりも後の一番早い開始日に設定する自動日付計算を使用します。連続的な日付選択は、「終了日/時刻」フィールドでのみ可能です。

注: 連続選択は、2 つのビジネス・ルール選択レコードの日付範囲が連続している場合に使用します。連続属性は、最初の選択レコードの終了日に設定されます。この属性が設定されている場合は、2 番目の選択レコードの開始日は、最初の選択レコードの終了日に設定されます。そのため、両方の日付を指定する必要はありません。

### 「開始日なし」または「終了日なし」

開始または終了 (選択内容に応じて決まります) の境界を設定しません。

**制約事項:** ビジネス・ルール・グループは有効な日付のみをサポートします。他の種類の選択を行う必要がある場合は、セレクター・コンポーネントを使用してください。

### ビジネス・ルールの日付の分割:

ビジネス・ルールの日付を分割すると、ビジネス・ルールを別の目的のために簡単に変更できます。

ルールを変更するには、「編集」モードにする必要があります。

ビジネス・ルール・レコードを分割するには、以下のステップを実行します。

1. レコードの横にある「分割」をクリックします。

開始日が 1 月 1 日で、各フィールドが赤色の新規レコードが作成されます。また、日付/時刻フィールド値が無効であることを示すメッセージが「メッセージ」フィールドに表示されます。

2. 新規レコードの開始日/時刻を選択します。

元のレコードの終了日/時刻が「継続」から新規レコードの開始日/時刻に変更され、新規レコードの終了日/時刻が前のレコードの終了日/時刻に変更されます。

3. 新規レコードの「日付/時刻」を変更します。
4. 新規ルールのニーズに合うように「有効なルール・ロジック」を変更します。

### ルール・セット:

ルール・セットは if/then ステートメント (ルール) のグループです。if がルールの条件で、then がルールのアクションです。ルール・セットは、条件節が非常に少ないビジネス・ルールに最適です。

条件が満たされると、アクションが実行されます。このとき、ルール・セットによって複数のアクションが実行される場合があります。ルール処理の順序は、if/then ルール・セット内のルール・ステートメントの順序によって決定されます。したがって、ルールを変更または追加する場合は、正しい順序で配置する必要があります。

ルール・セットは、if/then ルールとアクション・ルールの 2 種類のルールを持つことができます。

- if/then ルールは、実行するアクションを着信メッセージの条件に従って決定します。
- アクション・ルールは、実行するアクションを着信メッセージに無関係に決定します。

ルール内の条件は条件式を含みます。条件式は単純なストリング、*and*、*or*、または *not* にすることができます。

ルール・セットに定義されたテンプレートを使用して、新規ルール・セットを作成したりビジネス・ルール・マネージャーにある既存のルール・セットを変更したりすることができます。テンプレートは、ルール・セットがどのように機能するかを決定する構造を提供します。ルール・テンプレートはルール・セット間で共有されません。

### ルール・セット・レコードの作成:

新規のルール・セット・レコードを作成するには、既存のルール・セット・レコードをコピーし、その値を変更します。

新規ルール・セット・レコードを作成するには、以下のステップを実行します。

1. 選択したルール・セットのルール・ロジック・レコードの横にある「コピー」をクリックします。

「編集モード:Copy\_of\_TableName- ルール・セット (Edit Mode:Copy\_of\_TableName-Ruleset)」というタイトルの新規レコードの「編集」ページが開きます。

2. 「名前」フィールドで、新規ルール・セット・レコードの名前を入力します。
3. 「説明」フィールドで、新規レコードの簡単な説明を入力します。
4. 各条件の値を変更します。

**ヒント:** 各値のパラメーター設定を表示するには、フィールドにカーソルを合わせます。変数の型および範囲を示すロールオーバー・メッセージが表示されます。

5. 上または下矢印をクリックして、ルールを正しい順序に配置します。
6. 「保管」をクリックします。

レコードが一時的に保管されたことや、変更内容を「公開および復帰」ページから公開できることを示すメッセージがメッセージ・フィールドに表示されます。

### 関連タスク

#### 166 ページの『ビジネス・ルール・レコードの削除』

既存のビジネス・ルール選択レコードを、ルール選択ロジック・テーブルから削除できます。選択レコードを削除した場合、それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

### テンプレートからのルール・セット内のルールの作成:

ルール・セットに関連するルール・テンプレートを使用して、そのルール・セット内に新規ルールを作成します。

ルール・セットを変更するには、編集モードにする必要があります。

テンプレートから新規のルールを作成するには、以下のステップを実行します。

1. 「**テンプレートからの新規ルール**」をクリックして、ルールに対して使用可能なテンプレートのリストを表示します。
2. テンプレートを選択し、以下を実行します。
  - a. 「**名前**」フィールドで、新規ルールの名前を入力します。
  - b. 入力フィールド内にルールの値を指定するか、またはドロップダウン・リストから変数を選択します。
  - c. 「**追加**」をクリックします。
3. 上または下矢印をクリックして、ルールを適切な順序に配置します。

**注:** ルール処理の順序は、if/then ルール・セット内のルール・ステートメントの順序によって決定されます。したがって、ルールを変更または追加する場合は、正しい順序で配置する必要があります。

4. 「**保管**」をクリックします。

ルール・セットを公開する準備ができました。詳しくは、167 ページの『ビジネス・ルールの公開と復帰』を参照してください。

### 関連タスク

166 ページの『ビジネス・ルール・レコードの削除』

既存のビジネス・ルール選択レコードを、ルール選択ロジック・テーブルから削除できます。選択レコードを削除した場合、それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

### テンプレートを使用したルール・セット内のルールの変更:

ルール・セットに関連するテンプレートを使用して、そのルール・セット内のルールを変更します。

ルール・セットを変更するには、編集モードにする必要があります。

既存のテンプレートを使用してルールを変更するには、以下のステップを実行します。

1. 値を編集します。入力フィールド内の既存の値に上書きして入力するか、またはフィールド内に表示される下矢印をクリックし、ドロップダウン・リストから値を選択します。
2. 必要に応じて、上または下矢印をクリックして、ルールを適切な順序に配置します。

注: ルール処理の順序は、if/then ルール・セット内のルール・ステートメントの順序によって決定されます。したがって、ルールを変更または追加する場合は、正しい順序で配置する必要があります。

3. 「保管」をクリックします。

変更されたルール・セットを公開する準備ができました。詳しくは、167 ページの『ビジネス・ルールの公開と復帰』を参照してください。

**関連タスク**

166 ページの『ビジネス・ルール・レコードの削除』

既存のビジネス・ルール選択レコードを、ルール選択ロジック・テーブルから削除できます。選択レコードを削除した場合、それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

**デシジョン・テーブル:**

デシジョン・テーブルは表形式のビジネス・ルール・レコードです。テーブルでは、条件が行見出しおよび列見出しとして表され、アクションがそれら条件ケースの交点として表されます。デシジョン・テーブルは、複数の条件を持つビジネス・ルールに最適です。新しい条件を追加する場合も、新しい行または列を追加するだけで済みます。

if/then ルール・セットと同様に、デシジョン・テーブルは条件とアクションの相互作用によって駆動されます。主な違いは、デシジョン・テーブルでは、アクションが複数の条件によって決定され、条件の各セットに複数のアクションを関連付けることができる点です。条件が満たされると、対応するアクション (複数可) が実行されます。

以下の図はデシジョン・テーブルの例で、条件 は一番上の行の Purchase amount (購入金額) と左側の列の Member Type (メンバー・タイプ) です。アクション Discount (値引き) が中央と右側の列にあります。条件とアクションの方向を矢印によって示します。この例では、Gold の顧客の消費額が \$1000 を超える場合に、その顧客に 10% の値引きを行うことを示しています。アクションは、購入金額条件とメンバー・タイプ条件の交点です。

Purchase amount →	<= 1000 dollars	> 1000 dollars
Member Type ↓	Discount ↓	Discount ↓
Gold	8%	10%
Silver	3%	5%

図 9. デシジョン・テーブル

**テンプレート**

ビジネス・ルール・マネージャーでデシジョン・テーブル値を変更するには、テンプレートを使用します。テンプレートは WebSphere Integration Developer で設計されており、ビジネス・ルール定義に含まれています。テンプレートは、変更可能な

デシジョン・テーブルの側面を決定し、選択可能な有効な値のリストを提供します。デシジョン・テーブルに定義されたテンプレートに基づいて、テーブルに新規の行/列、または新規のアクションを作成し、テンプレートで作成した既存の条件またはアクションを変更します。デシジョン・テーブル・テンプレートはデシジョン・テーブル間で共有されません。

### 初期化アクション・ルール

デシジョン・テーブルは初期化アクション・ルールの使用をサポートします。初期化アクション・ルールはデシジョン・テーブルの実行前に実行され、ビジネス・オブジェクトの作成や初期値の設定などの前処理を行うことができます。ビジネス・ルール定義が初期化アクション付きで WebSphere Integration Developer で設計されている場合、ビジネス・ルール・マネージャーで初期化アクション・ルールを変更できます。

単一のテンプレートから作成できる初期化アクション・ルールは 1 つだけですが、アクション・ルールは複数のアクション式を持つことができるため、複数のアクションを実行できます。初期化ルール・テンプレートが特定のデシジョン・テーブルに定義されている場合は、そのテーブルのみで使用できます。

#### 「その他の場合」条件

「その他の場合」条件は、デシジョン・テーブル内の他の条件が適用されない場合にデフォルトで入力される特別な条件です。

WebSphere Integration Developer 内で設計されたデシジョン・テーブル定義に「その他の場合」条件がある場合にのみ、この条件がビジネス・ルール・マネージャー内に表示されます。ビジネス・ルール・マネージャーでは、ルールを動的に追加したり除去したりできません。

ただし、「その他の場合」条件に関連付けられたアクションをテンプレート化できません。「その他の場合」条件は、検査する条件 1 つにつき 0 回または 1 回使用できます。

#### デシジョン・テーブル・レコードの作成:

新規のデシジョン・テーブル・レコードを作成するには、既存のデシジョン・テーブル・レコードをコピーし、その値を変更します。

デシジョン・テーブル・レコードを作成するには、以下のステップを実行します。

1. 選択したデシジョン・テーブルの選択したルール・ロジック・レコードの横にある「コピー」をクリックします。

「編集モード:Copy\_of\_TableName- デシジョン・テーブル (Edit Mode:Copy\_of\_TableName-Decision Table)」というタイトルの新規レコードの「編集」ページが開きます。

2. 「名前」フィールドで、新規デシジョン・テーブル・レコードの名前を入力します。
3. 「説明」フィールドで、新規レコードの簡単な説明を入力します。
4. 各条件の値を変更します。

**ヒント:** 各値のパラメーター設定を表示するには、フィールドにカーソルを合わせます。変数の型および範囲を示すロールオーバー・メッセージが表示されます。

5. 「保管」をクリックします。

レコードが一時的に保管されたことや、変更内容を「公開および復帰」ページから公開できることを示すメッセージがメッセージ・フィールドに表示されます。詳しくは、167 ページの『ビジネス・ルールの公開と復帰』を参照してください。

**関連タスク**

166 ページの『ビジネス・ルール・レコードの削除』

既存のビジネス・ルール選択レコードを、ルール選択ロジック・テーブルから削除できます。選択レコードを削除した場合、それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

**「特別なアクション (Special actions)」メニュー:**

「デシジョン・テーブル」ページには、デシジョン・テーブルの値を編集したりテンプレート構造および変数を変更したりするための「特別なアクション (Special actions)」メニューがあります。

「特別なアクション (Special actions)」メニューは、デシジョン・テーブルが編集モードのときに、横に「特別なアクション (Special Actions)」アイコンが表示されるすべてのフィールドで使用可能です。フィールドの「特別なアクション (Special actions)」アイコンをクリックすると、そのフィールドで使用可能なオプションのリストが開きます。有効なオプションを以下の表にリストします。

**注:** 列または行を再配列しても、表のビジュアル表示にのみ影響し、条件およびアクションが処理される順序には影響しません。

メニュー・オプション	説明	条件の変更	アクションの変更
下に追加	新規の条件値 (行) を現行セルの下に追加します (方向は垂直)	可能	
右に追加	新規の条件値をセルの右側に追加します (水平方向に追加)	可能	
テンプレートの変更	セル値を変更できます	可能	可能
上へ移動	条件値または変数を上の行に移動します (方向は垂直)	可能	
下へ移動	条件値または変数を下の行に移動します (方向は水平)	可能	
左へ移動	条件値または変数を左に移動します (方向は水平)	可能	



メニュー・オプション	説明	条件の変更	アクションの変更
右へ移動	条件値または変数を右に移動します (方向は垂直)	可能	
削除	条件値または変数を削除します	可能	
メニューを閉じる	メニューを閉じます	可能	可能

### デシジョン・テーブル・レコードの変更:

デシジョン・テーブルを編集するには、該当する入力フィールドに新規の値を直接入力するか、またはフィールドのリスト・ボックス・オプションから値を選択します。

デシジョン・テーブルを変更するには、編集モードにする必要があります。

デシジョン・テーブルの値を変更するには、以下のステップを実行します。

1. 値を編集します。入力フィールド内の既存の値に上書きして入力するか、またはフィールド内に表示される下矢印をクリックし、ドロップダウン・リストから値を選択します。

#### 制約事項:

- 初期設定のルールは、WebSphere Integration Developer 内で設計されたビジネス・ルール定義に初期設定のルールがある場合にのみ、デシジョン・テーブルに表示されます。単一のテンプレートに関連付けることができる初期化アクション・ルールは 1 つのみですが、アクション・ルールの内部には複数のアクション式を含めることができます。
  - 「その他の場合」条件は、WebSphere Integration Developer 内で設計されたビジネス・ルール定義に「その他の場合」条件がある場合にのみ、デシジョン・テーブルに表示されます。「その他の場合」条件をビジネス・ルール・マネージャー内で追加または除去することはできません。ただし、「その他の場合」条件に関連付けられたアクションをテンプレート化することはできます。
2. フィールドの横にある「特別なアクション (Special actions)」アイコンをクリックして、フィールドで使用可能なオプションのリストを開き、必要なアクションを選択します。

注: 列または行を再配列するオプションの選択は、テーブルのビジュアル表示にのみ影響し、条件およびアクションが処理される順序には影響しません。

3. 「保管」をクリックします。

ルールがローカル側で変更されます。ルールをサーバーに公開する準備ができました。詳しくは、167 ページの『ビジネス・ルールの公開と復帰』を参照してください。

#### 関連タスク

### 『ビジネス・ルール・レコードの削除』

既存のビジネス・ルール選択レコードを、ルール選択ロジック・テーブルから削除できます。選択レコードを削除した場合、それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

### デシジョン・テーブルのテンプレート値の変更:

デシジョン・テーブル・テンプレートの構造と値を変更するには、「**特別なアクション (Special actions)**」メニューを使用し、該当する入力フィールドに値を直接入力します。

デシジョン・テーブルを変更するには、編集モードにする必要があります。

デシジョン・テーブル・テンプレートを変更するには、以下のステップを実行します。

1. 変更するデシジョン・テーブル・フィールドの横にある「**特別なアクション (Special actions)**」アイコンをクリックして、フィールドのオプションのリスト・ボックスを開き、「**テンプレートの変更**」を選択します。
2. 入力フィールド内の既存の値にテンプレートの新規の値を上書きして入力します。
3. 「**アクション**」列の「**変更**」をクリックします。
4. 「**保管**」をクリックします。

デシジョン・テーブル・テンプレートは変更され、公開する準備ができました。詳しくは、167 ページの『ビジネス・ルールの公開と復帰』を参照してください。

### ビジネス・ルール・レコードの削除:

既存のビジネス・ルール選択レコードを、ルール選択ロジック・テーブルから削除できます。選択レコードを削除した場合、それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

ルールを削除するには、編集モードにする必要があります。

ビジネス・ルール・レコードを削除するには、以下のステップを実行します。

1. 「ルール」ページで、ルールを選択し、「**削除**」をクリックします。

選択レコードは削除されます。それに関連付けられているルール・セットまたはデシジョン・テーブル定義はルール・グループに残り、そのページの「使用可能なルール・ロジック」セクション内にリストされます。そのルールは、デフォルトのルールとして、または特定の日時を指定して再度追加できます。

**注:** ビジネス・ルール・グループに対する各操作には、選択レコードとして、またはデフォルトの宛先として関連付けられた 1 つ以上のアクティブなビジ

ネス・ルールが存在する必要があります。すべてのビジネス・ルール・レコードを削除しようとする、エラーになります。

## 2. 「保管」をクリックします。

レコードが一時的に保管され、サーバーに公開する準備が整います。

### ビジネス・ルールの公開と復帰:

ビジネス・ルール・グループの一部を保管するときには、変更内容はローカル側で保管されます。アプリケーション・サーバー上にあるデータベースに変更内容を格納するには、ルールの変更内容を公開する必要があります。また、ビジネス・ルールをその元の状態に復帰させることにより、ルールに対してローカル側で保管された変更内容を取り消すこともできます。

ナビゲーション・ペインを持つ任意のビジネス・ルール・マネージャー・ページ上で作業する必要があります。

サーバーは「ルール」ページ・レベルで変更を公開します。

ビジネス・ルールへの変更内容を公開するには、以下のステップを実行します。

1. 「公開および復帰」をクリックします。
2. 「公開および復帰」ページで、コンテンツ領域の左端の列にあるチェック・ボックスをクリックして、データベースに送信するページを選択します。すべての「ルール」ページをまとめて単一トランザクションとして公開するか、またはページのサブセットのみを公開するかを選択できます。

**注:** ビジネス・ルールに対してローカル側で保管されたすべての変更内容を取り消し、データベース上にあるルールの元のコピーで変更済みのルールを置き換えるには、「ルール」ページのチェック・ボックスを選択し、「復帰」をクリックします。ビジネス・ルールを公開すると、データベース上にある元のコピーが変更されるため、公開の後でビジネス・ルールを復帰させることはできません。

## 3. 「公開」をクリックします。

「公開および復帰」ページを最新表示すると、選択した「ルール」ページは表示されなくなります。これは、選択した「ルール」ページがデータベースに書き込まれたことを示します。

ビジネス・ルールをサーバーにエクスポートする準備ができました。

### ビジネス・ルール・マネージャーのトラブルシューティング:

ビジネス・ルール・マネージャーの使用中に発生する可能性のある問題には、ログイン・エラー、ログイン競合、アクセス競合などがあります。

これらの問題のトラブルシューティングには、さまざまな対策を取ることができます。

### ログイン・エラー:

ログイン・エラーは、ログイン時に発生します。

ログイン・エラー・メッセージは、以下のとおりです。

ログインを処理できません。ユーザー ID とパスワードを確認して、再試行してください。

**注:** ログイン・エラーは、グローバル・セキュリティーが使用可能になっていて、ユーザー ID とパスワードのいずれかまたは両方に誤りがある場合にのみ発生します。

ログイン・エラーを解決するには、以下のステップを実行します。

1. エラー・メッセージで「OK」をクリックして、「ログイン」ページに戻ります。
2. 「ユーザー ID」と「パスワード」に有効な値を入力します。
  - パスワードで大/小文字を区別する必要がある場合は、Caps Lock キーがオンになっていないことを確認してください。
  - ユーザー ID とパスワードのスペルが正しいことを確認してください。
  - システム管理者に問い合わせ、ユーザー ID とパスワードが正しいことを確認してください。
3. 「ログイン」をクリックします。

ログイン・エラーが解決された場合は、ビジネス・ルール・マネージャーにログインできるようになります。エラーが解決されない場合は、システム管理者に連絡してください。

#### **ログイン競合エラー:**

ログイン競合エラーは、同じユーザー ID を持つ別のユーザーが既にアプリケーションにログインしている場合に発生します。

ログイン競合メッセージは、以下のとおりです。

同じユーザー ID を使って、別のユーザーが現在ログインしています。次の中から選択してください。

通常、ユーザーがログアウトせずにブラウザを閉じると、このエラーが発生します。この状態が発生した場合、セッションがタイムアウトする前に次のログインが試行されると、ログイン競合になります。

**注:** ログイン競合エラーは、グローバル・セキュリティーが使用可能になっている場合にのみ発生します。

以下の 3 つのオプションから選択します。

- ログイン・ページに戻る。

別のユーザー ID を使用してアプリケーションを開く場合は、このオプションを使用します。

- 同じユーザー ID を持つ別のユーザーをログアウトする。

別のユーザーをログアウトし、新しいセッションを開始する場合は、このオプションを使用します。

注: 他のセッションで行われた未公開のローカル変更は、すべて失われます。

- 同じユーザー ID を持つ別のユーザーのコンテキストを継承し、そのユーザーをログアウトする。

進行中の作業を続行する場合は、このオプションを使用します。前のセッションで保管された未公開のローカル変更は、すべて保持されます。ビジネス・ルール・マネージャーは、前のセッションで表示された最後のページを開きます。

#### アクセス競合エラー:

1 人のユーザーがデータベース内のビジネス・ルールを更新しているときに、別のユーザーが同時に同じルールを更新しようとする、アクセス競合エラーが発生します。

ローカル側の変更をデータベースに公開すると、このエラーが報告されます。

アクセス競合エラーを解決するには、以下のアクションを実行します。

- 「ルール」ページを公開します。
- エラーの原因となっているビジネス・ルールのソースを検索し、ローカル・マシンでの変更がまだ有効かどうかチェックします。別のユーザーが変更を行った後では、ローカル側で変更する必要がない場合があります。
- ビジネス・ルール・マネージャーで作業を続行する場合は、エラーが発生した「ルール」ページをデータベースから再ロードする必要があります。これは、エラーが発生した「ルール」ページのローカル変更は使用できなくなるからです。エラー状態でない他の「ルール」ページのローカル変更は引き続き使用できます。
- エラーが報告されたルールの「公開および復帰」ページで、「再ロード」をクリックして、「ルール」ページを再ロードします。

#### セレクター・コンポーネントの概要

ビジネスの変更に応じて、ビジネスを駆動するビジネス・プロセスも変更する必要があります。プロセスの設計は変更することなく、特定のプロセスが元の設計とは異なる結果を戻すよう変更しなければならない場合もあります。セレクター・コンポーネントは、このような状況に柔軟に対応するためのフレームワークを提供します。

セレクター・コンポーネントは、特定の基準に基づいて結果を変更するサービスに対する単一のインターフェースを提供します。セレクター・コンポーネントは、インターフェースおよびセレクター・テーブルで構成されます。セレクター・テーブルは、要求を処理するコンポーネント (ターゲット・コンポーネントと呼ばれます) を一定の基準に基づいて決定します。サーバーは、ターゲット・コンポーネントによって提供された処理結果をクライアントに戻します。

ビジネス・プロセスの構築時に、ソリューション設計者はセレクター・コンポーネントの要件を識別し、セレクター・コンポーネントが処理を完了するために使用するインターフェースおよびセレクター・テーブルを定義します。セレクター・コンポーネントの開発に関係する作業については、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。

セレクター・コンポーネントの管理は、セレクター・コンポーネントに関連した作業またはセレクター・テーブルに関連した作業で構成されます。

**制約事項:** セレクター・コンポーネント・パネルにアクセスするには、管理コンソールにログインする際にユーザー ID を指定する必要があります。ユーザー ID を指定せずにログインすると、ログアウトして有効なユーザー ID でもう一度ログインするよう、警告を受信します。

#### セレクター・コンポーネントの表示:

セレクター・コンポーネントの表示は、セレクター・コンポーネントを管理する最初のステップです。この表示から、セレクター・コンポーネントの一部またはすべてをエクスポートしたり、セレクター・コンポーネントから構成されるセレクター・テーブルを表示したりできます。

このタスクを実行するには、WebSphere Process Server の管理コンソールで作業する必要があります。

このタスクを実行して、サーバー上に存在するセレクター・コンポーネントを判別します。

**制約事項:** セレクター・コンポーネント・パネルにアクセスするには、管理コンソールにログインする際にユーザー ID を指定する必要があります。ユーザー ID を指定せずにログインすると、ログアウトして有効なユーザー ID でもう一度ログインするよう、警告を受信します。

1. 「サーバー」をクリックして、さまざまなサーバー・タイプを表示します。
2. 「アプリケーション・サーバー」をクリックして、アプリケーション・サーバー・リストを展開します。
3. 「*servername*」をクリックして、表示するセレクターを持つサーバーをサーバー・リストから選択します。
4. 「ビジネス・インテグレーション」の下にある「セレクター」をクリックします。

コンソールに、定義されたすべてのセレクター・コンポーネントのリストが、各コンポーネントの説明と共に表示されます。

#### セレクター・テーブルの表示:

セレクター・テーブルの表示は、テーブルを管理する最初のステップです。ターゲット・コンポーネントのリストが表示されるので、処理基準の変更、特定の基準で実行されるターゲット・コンポーネントの変更、新規ターゲット・コンポーネントの追加、テーブルからのターゲット・コンポーネントの削除 (基準の削除) などをこの画面から実行できます。

このタスクを実行するには、WebSphere Process Server の管理コンソールで作業する必要があります。

他のセレクター・テーブル関連タスクを実行するには、セレクター・テーブルを表示し、テーブルを構成する項目を選択します。このタスクは、セレクター・コンポーネントを表示した後に開始されます。

1. セレクター・コンポーネント画面からセレクター・コンポーネントをクリックします。 ブラウザーに、選択したコンポーネント内のセレクター・テーブルが表示されます。
2. 画面上のセレクター・テーブルの 1 つをクリックします。 ブラウザーに、セレクター・テーブルを構成するターゲット・コンポーネントが表示されます。

### ターゲット・コンポーネントの変更:

ターゲット・コンポーネントを変更すると、セレクター・コンポーネント処理を変更できます。そのためには、特定のターゲット・コンポーネントの選択基準を変更するか、選択基準に対してターゲット・コンポーネントを変更するか、または選択基準とターゲット・コンポーネントの両方を変更します。

このタスクを実行するには、セレクター・テーブルが存在している必要があります。

ターゲット・コンポーネントを変更して、選択基準を変更したり、セレクター・テーブルの項目に別のターゲット・コンポーネントを使用したりします。

**重要:** 長期実行アプリケーションのターゲット・コンポーネントを変更する場合は、その前にアプリケーションを停止します。長期実行アプリケーションの処理中には、ターゲット・コンポーネントを変更しないでください。

1. 『セレクター・テーブルの表示』で説明されているように、セレクター・テーブルを表示します。
2. 変更するターゲット・コンポーネントの横にある「**ターゲット ID**」をクリックします。 ターゲット・コンポーネントの詳細パネルが表示されます。
3. 項目を変更します。

### 項目の変更する部分 ターゲット宛先

### 変更の手順

1. ターゲット・コンポーネント・リストの横にある矢印をクリックします。すると、適切なターゲット・コンポーネントが表示されます。
2. リストからターゲット・コンポーネントを選択します。

### 選択基準

1. 「**開始日**」、「**終了日**」、またはその両方を上書きします。入力する日付はシステムのロケールによって異なり、ロケール形式に応じて表示されます。米国英語ロケールの場合、表示される形式は、以下のとおりです。
  - 月
  - 日
  - 年 (YYYY 形式)
  - 時刻 (HH:MM:SS 形式)
  - 時間帯

**重要:** 「**開始日**」は、「**終了日**」より前の日付を指定する必要があります。そうしないと、この変更をコミットできません。

## 項目の変更する部分 ターゲット宛先および選択基準

## 変更の手順

1. ターゲット・コンポーネント・リストの横にある矢印をクリックします。すると、適切なターゲット・コンポーネントが表示されます。
2. リストからターゲット・コンポーネントを選択します。
3. 「開始日」、「終了日」、またはその両方を上書きします。入力する日付はシステムのロケールによって異なり、ロケール形式に応じて表示されます。米国英語ロケールの場合、表示される形式は、以下のとおりです。
  - 月
  - 日
  - 年 (YYYY 形式)
  - 時刻 (HH:MM:SS 形式)
  - 時間帯

**重要:** 「開始日」は、「終了日」より前の日付を指定する必要があります。そうしないと、この変更をコミットできません。

4. **オプション:** 「デフォルト」チェック・ボックスをクリックして、これをデフォルトのターゲット・コンポーネントにします。

選択基準が他のターゲット・コンポーネントの範囲に含まれない場合、セレクター・コンポーネントはこのターゲット・コンポーネントを使用します。

5. 「適用」または「OK」をクリックします。

この画面で作業を継続するには、「適用」をクリックします。ターゲット・コンポーネント画面に戻るには、「OK」をクリックします。

6. ターゲット・コンポーネント画面で「保管」をクリックして、変更内容をセレクター・テーブルに保管します。

セレクター・テーブル・ファイルには、更新された選択基準とターゲット・コンポーネントが格納されます。セレクター・コンポーネントは、更新されたセレクター・テーブルを使用して、受信した次の要求を処理します。

### ターゲット・コンポーネントの追加:

現在セレクター・テーブルに存在するものと異なる選択基準に合う追加処理が必要な場合、ターゲット・コンポーネントを追加します。

このタスクを実行するには、セレクター・テーブルが存在している必要があります。

ビジネス・プロセスにさらに柔軟性が必要な場合、ターゲット・コンポーネントを追加します。セレクター・コンポーネントがアクティブである間、新規コンポーネントを追加できます。



1. 『セレクトター・コンポーネントの表示』で説明されているように、セレクトター・コンポーネントを表示します。
2. 『セレクトター・テーブルの表示』で説明されているように、セレクトター・テーブルを表示します。
3. セレクトター・テーブル画面で「新規」をクリックします。 ブラウザーに、既にデータが記入されたターゲット・コンポーネント詳細パネルが表示されます。
4. 『ターゲット・コンポーネントの変更』で説明されているように、アプリケーション要件に合わせてターゲット宛先情報を編集します。
5. 「OK」をクリックしてターゲット・コンポーネントを保管し、ターゲット・コンポーネント画面に戻ります。

セレクトター・テーブルに新規ターゲット・コンポーネントが組み込まれます。セレクトター・コンポーネントは、更新されたセレクトター・テーブルを使用して、受信した次の要求を処理します。

#### ターゲット・コンポーネントの削除:

セレクトター・テーブル内で特定の選択基準に合う項目を削除することによってターゲット・コンポーネントを削除すると、セレクトター・コンポーネントの処理が変更されます。

このタスクを実行するには、セレクトター・テーブルが存在している必要があります。

ビジネス・プロセスでターゲット・コンポーネントの処理が不要になったら、ターゲット・コンポーネントを削除します。ターゲット・コンポーネントの削除後は、要求が入力されてそれが他のどの特定の選択基準とも一致しない場合、デフォルトの基準によって要求が処理されます。

1. 『セレクトター・テーブルの表示』で説明されているように、ターゲット・コンポーネントを表示します。 パネルにセレクトター・テーブル画面が表示されます。
2. 削除するターゲット・コンポーネントの横にあるチェック・ボックスをクリックし、「削除」をクリックします。 システムによってパネルが更新され、残りのターゲット・コンポーネントが表示されます。
3. 「保管」をクリックします。 残りのターゲット・コンポーネントの項目を表示している更新されたセレクトター・テーブルが、システムによって保管されます。

セレクトター・テーブル・ファイルには、残りのターゲット・コンポーネントのみが含まれるようになります。セレクトター・コンポーネントは、更新されたセレクトター・テーブルを使用して、受信した次の要求を処理します。

#### 管理コンソールを使用したセレクトター・コンポーネントのエクスポート:

セレクトター・コンポーネントをエクスポートすると、開発環境または他の実稼働環境にインポートできるファイルが作成されます。これにより、開発の成果物を実動システムの成果物と同期させることができます。

このタスクを開始するには、『セレクトター・コンポーネントの表示』に説明されているように、セレクトター・コンポーネントが既に表示されている必要があります。

セレクター・テーブルを変更した後で開発環境と実稼働環境を同期する必要がある場合、セレクター・コンポーネントをエクスポートします。このタスクは、セレクター・コンポーネント画面から開始されます。

1. エクスポートするセレクターを選択します。

セレクターの横にあるチェック・ボックスをクリックし、「エクスポート」をクリックします。ブラウザーには、選択したセレクター・コンポーネントへのHTML リンクのリストが表示されます。(これは「エクスポートするセレクター」パネルです。) 各セレクターにはファイル拡張子として .zip が付いています。

2. ファイルをダウンロードします。

各ファイル名をクリックすると、プロンプトが出てファイルを保管するように要求されます。プロンプトが出たら、「OK」をクリックして、ファイル・システムにファイルを保管します。

**注:** ダウンロードするときファイル名を変更することもできます。

3. セレクター表示に戻ります。

「戻る」をクリックして、セレクターのリストに戻ります。

指定した場所にファイルが保管されます。

#### 管理コンソールを使用したセレクター・コンポーネントのインポート:

セレクター・コンポーネントをインポートすると、アプリケーションを再インストールしなくても、インストール済みのセレクター・コンポーネントを更新できます。

管理コンソールで作業する必要があり、エクスポート機能によって作成された圧縮ファイルのロケーションを知っている必要があります。

インストール済みのアプリケーションが使用しているセレクターを変更した後で、それらの変更内容を別のクラスターまたはサーバーにも適用したい場合に、セレクターをインポートします。ご使用の開発環境と実稼働環境内の変更内容を同期させるときにも、この機能を使用できます。

**ヒント:** セレクター・コンポーネントは、コマンド行を使用してインポートすることもできます。

1. 『セレクター・コンポーネントの表示』で説明されているように、セレクター・コンポーネントのインポート先のサーバー上のセレクターを表示します。
2. 「インポート」をクリックします。
3. 「**セレクターのインポートの準備**」パネル上でファイルへのパスを指定します。

更新されたセレクターのセレクター・テーブルを表示して、変更内容を確認します。

## ターゲットの概要

ターゲットには、参照用に構成されたターゲットを変更することによって処理を変更できる機能があるため、柔軟性がさらに向上します。

コンポーネントは、他のモジュールのコンポーネントを呼び出すことができます。このように既存のロジックを再利用することにより、アプリケーションを作成するための時間とコストを最小限に抑えることができます。WebSphere Process Server は、ターゲットによって、柔軟性を追加します。ターゲットにより、インストール済みアプリケーションの処理が向上します。管理コンソールを使用すれば、アプリケーションを再作成または再デプロイすることなく、アプリケーションでモジュール間呼び出しのエンドポイントを変更できるためです。

このような柔軟性を十分に生かすには、システムによるターゲットの命名方法について理解しておく必要があります。呼び出しモジュールからのリンクは、正しいターゲットに接続されている必要があります。

## ターゲット名

ターゲット名は、呼び出しコンポーネントがターゲットを呼び出す方法が元になっています。名前の形式は以下のとおりです。

### 呼び出しタイプ

#### 名前の形式

**同期** Java Naming and Directory Interface (JNDI) 形式に準拠する名前。例えば、以下のような形式です。

```
folder/export/fullpath_to_target/target_component_name
```

**非同期** 次の形式の名前。

```
folder/calling_component_name/  
full_path_to_target_component/target_component_name
```

### 複数の宛先

この名前は、非同期呼び出しと同じです。ただし、ターゲットは、実際にメッセージを複数の宛先コンポーネントに送信します。

### 関連タスク

#### 『ターゲットの変更』

参照のターゲットを変更することにより、アプリケーションの再コンパイルや再インストールを行わなくとも、更新されたコンポーネントをアプリケーションが柔軟に利用できるようになります。

## ターゲットの変更

参照のターゲットを変更することにより、アプリケーションの再コンパイルや再インストールを行わなくとも、更新されたコンポーネントをアプリケーションが柔軟に利用できるようになります。

参照のターゲットを変更する前に、以下を実行する必要があります。

- 新規ターゲットが同じデータ・オブジェクト・タイプを使用することを確認する。
- モジュールがターゲットを同期的に起動しているか非同期的に起動しているかを確認する。
- 参照が単一のサービスをターゲットにしているか、複数のサービスをターゲットにしているかを確認する。

元のターゲットと同じインターフェースを持つ別のサービスが、モジュールで使用可能な新機能または改良機能を提供する場合、モジュールからのインポートのターゲットを変更します。

1. 変更する参照を含むモジュールを停止します。
  - a. 管理コンソールを使用して、Service Component Architecture (SCA) モジュールを表示します。  
  
「アプリケーション」 > 「SCA モジュール」を使用してこのパネルにナビゲートします。
  - b. モジュールを選択し、「停止」を押します。表示は更新され、アプリケーションが停止済みとして表示されます。
2. 参照のターゲット宛先を変更します。

変更を行う方法は、モジュールがターゲットを呼び出す方法によって異なります。

呼び出しのタイプ	変更方法
単一ターゲット・サービス	<ol style="list-style-type: none"> <li>1. 管理コンソールを使用して「SCA モジュール」を表示します。「アプリケーション」 &gt; 「SCA モジュール」を使用してこのパネルにナビゲートします。</li> <li>2. 表示されたリストから、変更するターゲットを参照するインポートを含むモジュールを選択します。</li> <li>3. 「インポート」の横の正符号 (+) をクリックして、インポートのリストを展開します。</li> <li>4. 変更するインポートをリストから選択します。</li> <li>5. 「ターゲット」領域で、リストから「モジュール」を選択します。</li> <li>6. 「エクスポート」リストが最新の内容に更新されたら、新規ターゲット用のエクスポートを選択します。</li> <li>7. 「OK」をクリックして変更を保管します。</li> </ol>

呼び出しのタイプ	変更方法
複数のターゲット・サービス	<ol style="list-style-type: none"> <li>モジュールが存在するシステムのバスを表示します。「サービス統合」&gt;「バス」を使用してパネルにナビゲートします。</li> <li><b>SCA.System.cellname.Bus</b> を選択します。</li> <li>「宛先」をクリックしてバスの宛先ターゲットを表示します。</li> <li>呼び出し側モジュールをターゲットに接続するインポートを表す宛先を選択します。この ID は import という語を含みます。</li> <li>「コンテキスト・プロパティ (Context properties)」をクリックしてプロパティのリストを表示します。</li> <li>リスト内のターゲット・プロパティをクリックして、変更するプロパティを選択します。</li> <li>「コンテキスト値 (Context value)」フィールドを新規宛先ターゲットに変更します。</li> <li>「OK」をクリックして、「コンテキスト・プロパティ (Context properties)」パネルに戻ります。</li> <li>「OK」をクリックして変更を保管します。</li> </ol>

3. 変更を保管します。プロンプトが出されたら、「保管」をクリックします。

モジュールを始動し、モジュールが予期される結果を受け取ることを確認します。

## J2C アクティベーション・スペックの削除

サービスを含むアプリケーションをインストールすると、システムによって J2C アプリケーションの仕様が作成されます。アプリケーションを再インストールする前に、この仕様が削除する必要がある場合があります。

アプリケーションのインストールに失敗したために仕様が削除する場合、Java Naming and Directory Interface (JNDI) 名の中のモジュールとインストールできなかったモジュールの名前とが一致するようにしてください。JNDI 名の 2 番目の部分が、宛先をインプリメントしたモジュールの名前に相当します。例えば、sca/SimpleBOCrsmA/ActivationSpec の場合、**SimpleBOCrsmA** がモジュール名です。

サービスを含むアプリケーションをインストールした後で間違って構成を保管したが、J2C アクティベーション・スペックが不要な場合は、その仕様が削除します。

1. 削除するアクティベーション・スペックを見つけます。

仕様は「リソース・アダプター」パネルに表示されます。「リソース」>「リソース・アダプター」をクリックして、このパネルにナビゲートします。

a. 「Platform Messaging Component SPI Resource Adapter」を見つけます。

このアダプターを見つけるには、スタンドアロン・サーバーの「ノード」スコープまたは Network Deployment 環境の「サーバー」スコープで作業する必要があります。

2. Platform Messaging Component SPI Resource Adapter に関連した J2C アクティベーション・スペックを表示します。

リソース・アダプター名をクリックすると、次のパネルが表示され、関連した仕様が表示されます。

3. 削除するモジュール名に一致した「JNDI 名」の仕様をすべて削除します。
  - a. 該当する仕様の横にあるチェック・ボックスをクリックします。
  - b. 「削除」をクリックします。

システムは、選択された仕様を表示から削除します。

変更を保管します。

## SIBus 宛先の削除

SIBus 宛先とは、アプリケーションでサービスを使用可能にするための関連付けのことです。宛先の削除が必要になる場合があります。

アプリケーションのインストールに失敗したために宛先を削除する場合、宛先名の中のモジュールとインストールできなかったモジュールの名前とが一致するようにしてください。宛先の 2 番目の部分が、宛先をインプリメントしたモジュールの名前に相当します。例えば、sca/SimpleBOCrsmA/component/test/sca/cros/simple/cust/Customer の場合、SimpleBOCrsmA がモジュール名です。

サービスを含むアプリケーションのインストール後に不注意で構成を保管した場合、または SIBus 宛先を必要としなくなった場合、その宛先を削除します。

**注:** このタスクは、SCA システム・バスからのみ宛先を削除します。サービスを含むアプリケーションを再インストールする前に、アプリケーション・バスからもその項目を削除する必要があります (このインフォメーション・センターの管理セクションの『J2C アクティベーション・スペックの削除』を参照してください)。

1. 管理コンソールにログインします。
2. SCA システム・バスの宛先を表示します。

「サービス統合」>「バス」をクリックして、パネルにナビゲートします。

3. SCA システム・バスの宛先を選択します。

画面上で、「SCA.SYSTEM.cellname.Bus」をクリックします。ここで、cellname は、削除しようとしている宛先を持つモジュールが含まれているセルの名前です。

4. 削除するモジュールと一致するモジュール名を含む宛先を削除します。
  - a. 該当する宛先の横にあるチェック・ボックスをクリックします。
  - b. 「削除」をクリックします。

パネルには残りの宛先のみが表示されます。

これらの宛先を作成したモジュールに関連した J2C アクティベーション・スペックを削除します。

## 拡張メッセージング・リソースの管理: 概要

拡張メッセージングは、コンテナ管理メッセージングを使用可能にします。基本の Java Message Service (JMS) サポート、Enterprise Java Bean (EJB) コンポーネント・モデル、および EJB 2.0 メッセージ駆動型 Bean のサポートを拡張することで、既存のコンテナ管理パーシスタンスおよびトランザクション動作が使用可能になります。

拡張メッセージングは Bean 管理メッセージングのインプリメンテーションを使用して、JMS インターフェースを提供します。これにより、Bean 管理メッセージングおよび拡張メッセージングの両方が、一貫性のある JMS サポートを確実に使用できます。JMS のサポートが拡張メッセージング・サービスによって管理されるため、JMS の使用は単純化されます。

管理コンソール・インターフェースにより、拡張メッセージング・サービスとそのサービスを使用するアプリケーションで必要なリソースを構成できます。

拡張メッセージングの完全な説明については、WebSphere Business Integration Server Foundation インフォメーション・センターで以下の項目を参照してください。

- 『拡張メッセージング: 概説』
- 『アプリケーションにおける拡張メッセージングの使用』

注: 拡張メッセージング・サービス機能は、WebSphere Process Server 6.0.1 では推奨されません。このリリースでは、新規アプリケーションと既存のアプリケーションで拡張メッセージングの使用を継続できますが、これらのアプリケーションを、標準の JMS API を使用するアプリケーションに置き換えるか、同等のメッセージング・テクノロジーに置き換える必要があります。

## 拡張メッセージング・サービスの使用可能化

拡張メッセージング・サービスでは、コンテナ管理メッセージング (拡張メッセージング) をサポートするランタイム・サービスが提供されます。このサービスは、アプリケーション・サーバーの始動時に自動的に開始することも、手動で開始することもできます。

**必要なセキュリティ・ロール:** このタスクを実行するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. 「サーバー」>「アプリケーション・サーバー」> *server\_name* > 「拡張メッセージング・サービス」をクリックして、「拡張メッセージング・サービス」ページを表示します。
3. サーバーの始動時に拡張メッセージング・サービスが自動的に開始されるようにするには、「サーバー起動時にサービスを使用可能にする」チェック・ボックスを選択します。サービスを手動で開始する場合、このチェック・ボックスがクリアされていることを確認してください。
4. 「OK」をクリックします。

5. プロンプトが出されたら、コンソール・タスクバーで「保管」をクリックして、変更内容をマスター・リポジトリに保管します。
6. 変更内容を有効にするために、アプリケーション・サーバーを停止して再始動します。

なお、『遅延応答を処理するリスナー・ポート拡張の構成』に説明されているように、「拡張メッセージング・サービス」ページを使用して、遅延応答を処理するリスナー・ポート拡張を構成することもできます。

## 遅延応答を処理するリスナー・ポート拡張の構成

アプリケーションのリスナー・ポートを、拡張メッセージング環境で遅延応答を処理する拡張機能を持つように構成できます。

**必要なセキュリティ・ロール:** このタスクを実行するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. リスナー・ポートが定義および構成されていることを確認します。
2. 管理コンソールから、「サーバー」>「アプリケーション・サーバー」> *server\_name* >「拡張メッセージング・サービス」>「リスナー・ポートの拡張」の順にクリックします。「リスナー・ポートの拡張」ページが開きます。
3. 「新規」をクリックして、新規リスナー・ポート拡張を作成します。「新規リスナー・ポート拡張 (New Listener Port Extension)」ページが開きます。
4. 「使用可能」チェック・ボックスを選択して、遅延応答処理を使用可能にします。
5. 「要求間隔」フィールドで、デフォルト値を受け入れるか、新規の値を指定します。要求間隔は、リスナー・ポートが遅延応答をチェックする頻度を指定します。
6. 「要求タイムアウト」フィールドで、デフォルト値を受け入れるか、新規の値を指定します。要求タイムアウト値は、リスナー・ポートが遅延応答を待機する時間を指定します。その時間が経過した後に受信した応答は廃棄されます。
7. 「リスナー・ポート」ドロップダウン・メニューを使用して、この拡張に使用するリスナー・ポートを指定します。
8. 「OK」をクリックします。
9. プロンプトが出されたら、コンソール・タスクバーで「保管」をクリックして、変更内容をマスター・リポジトリに保管します。
10. 変更内容を有効にするために、アプリケーション・サーバーを停止して再始動します。

## 拡張メッセージング・プロバイダーの管理

拡張メッセージング・プロバイダーは、コンテナ管理メッセージング (拡張メッセージング) で使用するために定義されたリソースを管理します。管理コンソールの「拡張メッセージング・プロバイダー」ページを使用して、各リソースの一般プロパティを表示し、編集するリソースを選択します。

拡張メッセージング・リソースは、以下のように、セル・スコープ、ノード・スコープ、またはサーバー・スコープで定義できます。



- セル・スコープ - 最も一般的なスコープ。セル・スコープで定義された拡張メッセージング・リソースは、オーバーライドされていない場合、すべてのノードおよびサーバーから参照できます。
- ノード・スコープ - ノード・スコープで定義された拡張メッセージング・リソースは、セル・スコープで定義された重複する定義をオーバーライドします。これらのリソースは、同一ノード内のサーバー・スコープでオーバーライドされていない場合、そのノード内のすべてのサーバーから参照できます。
- サーバー・スコープ - サーバー・スコープで定義された拡張メッセージング・リソースは、セル・スコープまたは親ノード・スコープで定義された重複する定義をオーバーライドします。これらのリソースは特定のサーバーからのみ参照できます。

スコープの詳細については、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

**必要なセキュリティ・ロール:** このタスクを実行するには、管理者、オペレーター、コンフィギュレーター、またはモニターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダー」ページが開きます。
2. 適切なスコープの横にあるラジオ・ボタンをクリックします。
3. 「適用」をクリックします。ページの下部の「有効範囲」、「名前」、および「説明」フィールドは更新され、選択されたリソース・プロバイダー用の値を反映します。

これで、選択した拡張メッセージング・プロバイダーの入力ポート、出力ポート、またはその他のカスタム・プロパティを作成、変更、または削除できるようになります。

#### 関連タスク

『新規入力ポートの追加』

184 ページの『新規出力ポートの追加』

#### 新規入力ポートの追加:

セッション Bean から構成された受信側 Bean は、受信用の Java Message Service (JMS) 宛先のプロパティを定義するために入力ポートを必要とします。また、入力ポートは、メッセージの選択および処理の詳細や、応答宛先の詳細を提供することもできます (必要に応じて)。

**注:** メッセージ駆動型 Bean から構成された受信側 Bean には入力ポートは必要ありません。そこに含まれる詳細は、デプロイ済みのメッセージ駆動型 Bean およびメッセージ・リスナー・サービスに関連付けられます。

このタスクでは、入力ポートの初期プロパティを構成します。必要に応じて、後からポートのプロパティを変更できます。

**必要なセキュリティ・ロール:** このタスクを実行するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダー」ページが開きます。
2. 新規ポートを追加するリソース・プロバイダーのスコープを選択します。
3. 「適用」をクリックします。
4. 「追加プロパティ」テーブルから「入力ポート」をクリックします。「入力ポート」ページが開きます。
5. 「新規」をクリックします。「入力ポート」構成ページが開きます。
6. 新規入力ポートに適切なプロパティを指定します。プロパティの設定について詳しくは、『入力ポート設定』を参照してください。
7. 「OK」をクリックします。
8. プロンプトが出されたら、コンソール・タスクバーで「保管」をクリックして、変更内容をマスター・リポジトリに保管します。
9. 変更内容を有効にするために、アプリケーション・サーバーを停止して再始動します。

#### 関連タスク

『入力ポートの構成の変更』

#### 入力ポートの構成の変更:

必要に応じて既存の入力ポートの構成を変更できます。入力ポートは、受信用の Java Message Service (JMS) 宛先のプロパティを定義します。また、メッセージの選択および処理の詳細や、応答宛先の詳細を提供します (必要に応じて)。

**必要なセキュリティ・ロール:** このタスクを実行するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダー」ページが開きます。
2. 新規ポートを追加するリソース・プロバイダーのスコープを選択します。
3. 「適用」をクリックします。
4. 「追加プロパティ」テーブルから「入力ポート」をクリックします。「入力ポート」ページが開きます。
5. 変更する入力ポートを選択します。「入力ポート」構成ページが開きます。このページに、選択したポートの現在の構成プロパティが表示されます。
6. 入力ポートのプロパティを変更します。プロパティの設定について詳しくは、『入力ポート設定』を参照してください。
7. 「OK」をクリックします。
8. プロンプトが出されたら、コンソール・タスクバーで「保管」をクリックして、変更内容をマスター・リポジトリに保管します。
9. 変更内容を有効にするために、アプリケーション・サーバーを停止して再始動します。

**入力ポート設定:** 入力ポートには以下の構成プロパティがあります。

### 有効範囲

拡張メッセージング・プロバイダーが定義される範囲。値は構成ファイルのロケーションを表します。

**名前** 管理目的で使用する入力ポートの名前。このフィールドの値はストリングでなければなりません。

### JNDI 名

リソースの Java Naming and Directory Interface (JNDI) 名。このフィールドの値はストリングでなければなりません。

**説明** 管理目的で使用する入力ポートの説明。このフィールドの値はストリングでなければなりません。

このフィールドはオプションです。

### カテゴリ

リソースの分類またはグループ化の際に使用するカテゴリ・ストリング。このフィールドの値は、長さが 1 から 30 までの ASCII 文字のストリングでなければなりません。

このフィールドはオプションです。

### JMS 接続ファクトリー JNDI 名

入力ポートで使用される Java Message Service (JMS) 接続ファクトリーの JNDI 名 (例えば、jms/connFactory1)。このフィールドの値はストリングでなければなりません。

### JMS 宛先 JNDI 名

入力ポートで使用される JMS 宛先の JNDI 名 (例えば、jms/destn1)。このフィールドの値はストリングでなければなりません。

### JMS 肯定応答モード

メッセージの肯定応答で使用される JMS モード。このプロパティは、Bean 管理トランザクション区分を使用するメッセージ駆動型 Bean の場合 (すなわちトランザクション・タイプを Bean に設定した場合) にのみ使用されます。

このフィールドの有効な値は以下のとおりです。

- 自動応答 (Auto Acknowledge) - 以下のいずれかの場合、セッションは自動的にメッセージに応答します。
  - セッションが、メッセージを受信する呼び出しから正常に戻った場合
  - セッションがメッセージ・リスナーを呼び出してメッセージを処理し、そのリスナーから正常な応答を受け取った場合
- 重複可の応答 (Dups OK Acknowledge) - セッションはメッセージの送達にのみ応答します。そのため、JMS が失敗すると、重複メッセージが送達されることがあります。

デフォルト・モードは「自動応答 (Auto Acknowledge)」です。

### 宛先タイプ

JMS リソース・タイプ。このフィールドの有効な値は以下のとおりです。

- キュー (Queue) - 受信側 Bean はキュー宛先からメッセージを受信します。

- トピック (Topic) - 受信側 Bean はトピック宛先からメッセージを受信します。

デフォルト値は「キュー (Queue)」です。

#### サブスクリプション永続性

JMS トピック・サブスクリプションが永続的であるかどうかを指定します。このフィールドの有効な値は以下のとおりです。

- 永続 - サブスクライバーは、JMS によって保持される固有の ID を持つ永続的サブスクリプションを登録します。ID が同じである後続のサブスクライバー・オブジェクトは、前のサブスクライバーが残した状態でサブスクリプションを再開します。永続的サブスクリプションのアクティブ・サブスクライバーがない場合、JMS は、サブスクリプションのメッセージが受信されるか期限切れになるまで、そのメッセージを保存します。
- 非永続 - 非永続的サブスクリプションは、それらのサブスクライバーの存続期間中だけ存続します。クライアントは、サブスクライバーがアクティブである期間のみ、トピックで公開されたメッセージを見ることができます。サブスクライバーが非アクティブである場合、クライアントはそのトピックで公開されたメッセージを見ることができません。

デフォルト値は「永続」です。

このフィールドは、JMS 宛先タイプがトピックの場合にのみ必要です。

#### 応答 JMS 接続ファクトリー JNDI 名

応答に使用する JMS 接続ファクトリーの JNDI 名。このフィールドの値はストリングでなければなりません。

#### 応答 JMS 宛先 JNDI 名

応答に使用する JMS 宛先の JNDI 名。このフィールドの値はストリングでなければなりません。

#### 新規出力ポートの追加:

出力ポートは、送信されるメッセージの宛先を送信側 Bean で定義するために必要なプロパティを指定します。また、応答が予期される場合は、オプションのプロパティも指定します。出力ポートは、デプロイメント時に送信側 Bean に関連付けられます。

このタスクでは、出力ポートの初期プロパティを構成します。必要に応じて、後からこのプロパティを変更できます。

**必要なセキュリティ・ロール:** このタスクを実行するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダー」ページが開きます。
2. 新規ポートを追加するリソース・プロバイダーのスコープを選択します。
3. 「適用」をクリックします。
4. 「追加プロパティ」テーブルから「出力ポート」をクリックします。「出力ポート」ページが開きます。
5. 「新規」をクリックします。「出力ポート」構成ページが開きます。

6. 新規出力ポートに適切なプロパティを指定します。プロパティの設定について詳しくは、『出力ポート設定』を参照してください。
7. 「OK」をクリックします。
8. プロンプトが出されたら、コンソール・タスクバーで「保管」をクリックして、変更内容をマスター・リポジトリに保管します。
9. 変更内容を有効にするために、アプリケーション・サーバーを停止して再始動します。

#### 関連タスク

『出力ポートの構成の変更』

#### 出力ポートの構成の変更:

必要に応じて既存の出力ポートの構成を変更できます。出力ポートは、送信されるメッセージの宛先を定義し、応答が予期される場合は宛先を指定します。

**必要なセキュリティ・ロール:** このタスクを実行するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダー」ページが開きます。
2. 新規ポートを追加するリソース・プロバイダーの範囲を選択します。
3. 「適用」をクリックします。
4. 「追加プロパティ」テーブルから「出力ポート」をクリックします。「出力ポート」ページが開きます。
5. 変更する出力ポートを選択します。「出力ポート」構成ページが開きます。このページに、選択したポートの現在の構成プロパティが表示されます。
6. 出力ポートのプロパティを変更します。プロパティの設定について詳しくは、『出力ポート設定』を参照してください。
7. 「OK」をクリックします。
8. プロンプトが出されたら、コンソール・タスクバーで「保管」をクリックして、変更内容をマスター・リポジトリに保管します。
9. 変更内容を有効にするために、アプリケーション・サーバーを停止して再始動します。

**出力ポート設定:** 出力ポートには以下の構成プロパティがあります。

#### 有効範囲

このフィールドでは、拡張メッセージング・プロバイダーの範囲を指定します。値は構成ファイルのロケーションを表します。

このフィールドは編集できません。

**名前** 管理目的で使用する出力ポートの名前。このフィールドの値はストリングでなければなりません。

#### JNDI 名

出力ポートの Java Naming and Directory Interface (JNDI) 名。このフィールドの値はストリングでなければなりません。

**説明** 管理目的で使用する出力ポートの説明。このフィールドの値はストリングでなければなりません。

このフィールドはオプションです。

#### **カテゴリ**

リソースの分類またはグループ化の際に使用するカテゴリ・ストリング。このフィールドの値は、最長 30 文字の ASCII 文字のストリングでなければなりません。

このフィールドはオプションです。

#### **JMS 接続ファクトリー JNDI 名**

出力ポートで使用される Java Message Service (JMS) 接続ファクトリーの JNDI 名 (例えば、jms/connFactory1)。このフィールドの値はストリングでなければなりません。

#### **JMS 宛先 JNDI 名**

出力ポートで使用される JMS 宛先の JNDI 名。このフィールドの値はストリングでなければなりません (例えば、jms/destn1)。

#### **JMS デリバリー・モード**

メッセージの送達で使用される JMS モード。値は、以下のいずれかでなければなりません。

- 永続的 - 宛先に書き込まれるメッセージは永続的
- 非永続的 - 宛先に書き込まれるメッセージは非永続的

デフォルト値は「永続的」です。

#### **JMS 優先順位**

キュー宛先のメッセージ優先順位。値は 0 から 9 までの整数でなければなりません。デフォルト値は 4 です。

#### **JMS 存続時間**

メッセージがキュー内に存続する時間 (ミリ秒単位)。指定された時間が経過すると、メッセージの有効期限が切れます。

値は 0 から  $n$  までの整数でなければなりません。

- 0 - メッセージはタイムアウトにならない
- $n$  - メッセージは  $n$  ミリ秒後にタイムアウトになる

デフォルト値は 0 です。

#### **JMS メッセージ ID 使用不可**

システムが JMS メッセージ ID を生成するかどうかを指定します。値は、以下のいずれかでなければなりません。

- 選択済み - システムは JMS メッセージ ID を生成しない
- クリア - システムは JMS メッセージ ID を自動的に生成する

デフォルトでは、JMS メッセージ ID は生成されます。

#### **JMS メッセージ・タイム・スタンプ使用不可**

システムが JMS メッセージ・タイム・スタンプを生成するかどうかを指定します。値は、以下のいずれかでなければなりません。

- 選択済み - メッセージ・タイム・スタンプを送信済みメッセージに追加しない
- クリア - メッセージ・タイム・スタンプを送信済みメッセージに自動的に追加する

デフォルトでは、メッセージ・タイム・スタンプは追加されます。

#### 応答 JMS 接続ファクトリー JNDI 名

出力ポートによって処理される応答に使用する JMS 接続ファクトリーの JNDI 名 (例えば、jms/connFactory1)。このフィールドの値はストリングでなければなりません。

#### 応答 JMS 宛先 JNDI 名

出力ポートによって処理される応答に使用する JMS 宛先の JNDI 名 (例えば、jms/destn1)。このフィールドの値はストリングでなければなりません。

---

## Enterprise Service Bus の管理

Enterprise Service Bus の管理には、サービス・バス環境の管理、メディエーション・モジュールのデプロイと管理、サービス統合およびメディエーション・モジュールで使用されるリソースの管理、およびメディエーション・モジュールと対話するクライアントおよびアダプターの管理が関係しています。管理者は、基盤となる WebSphere Application Server の全機能を管理することもできます。

管理者は、Enterprise Service Bus により、サービス・アプリケーションとしてのメディエーション・モジュールのデプロイメントをサポートする、ESB サーバーおよびサービス統合バスの環境を作成します。WebSphere Process Server をインストールすると、ESB にデプロイされたサービス・アプリケーションで使用する 2 つのサービス統合バスが得られます。手始めにバス内のサーバーを 1 台にして、その後、複数のサーバーまたはサーバー・クラスターを構築することにより、容量や可用性を拡張することもできます。必要に応じて他のバスを追加して、別の ESB の作成、ESB に接続するアプリケーションのデプロイ、WebSphere MQ との統合を可能にできます。

管理者は、サーバーおよびバス環境にメディエーション・モジュールを (SCA モジュールとして) デプロイできます。それらのモジュールにより、ユーザーがデプロイしたモジュールを表示し、要求が正常に処理されているかどうかをモニターできます。管理者は、メディエーション・モジュールを開始または停止できます。また、他の方法でモジュールを管理して、例えば、モジュールの構成を変更したり、モジュールを停止または更新したり、あるいはそのアクティビティを管理したりできます。また管理者は、メディエーション・モジュールの SCA インポートに変更して、例えば、あるモジュールを別のモジュールに宛先変更できます。これによりモジュールは、異なるサービス・プロバイダーを呼び出すことができ、モジュールの再ビルドと再デプロイを行わなくても、サービス要求と応答をさまざまな方法で処理できます。

管理者は、WebSphere 管理コンソール、WebSphere 管理 (wsadmin) スクリプト・プログラム、コマンド行ツール、および管理プログラムなど、さまざまなツールを使用して Enterprise Service Bus を管理できます。

管理用タスクについては、主に管理コンソールを使用する場合を基にして説明します。管理コンソール内の各タスクは、1 つ以上のパネルでサポートされます。タスク・フィルターを使用して、実行するタスクに最適なパネルのセットを表示し、それらのパネルでのみ作業できます。

**すべて** これにより、管理コンソールのすべてのパネルが表示されます。これは、管理者が WebSphere Process Server のすべてのパーツ、および基盤となる WebSphere Application Server を管理する場合に最適です。

エンタープライズ・サービス・バス機能の管理について詳しくは、以下にリストする関連タスクを参照してください。

### アプリケーションの統合

これにより、以下のタスク領域に適したパネルが表示されます。

- メディエーション・モジュールとサービス・アプリケーションのデプロイメントをサポートするために必要とされるサービス統合バス、サーバー、サーバー・クラスター、メッセージング・エンジン、およびネットワーク・トポロジーの構成の調整
- デプロイされたサービス・アプリケーションおよびメディエーション・モジュールに必要なリソース (例えば、JMS 接続ファクトリーや Common Event Infrastructure プロファイル) の作成
- メディエーション・モジュールとサービス・アプリケーションの操作制御

これは、242 ページの『サービス・アプリケーションの管理』の説明にあるように、管理者がメディエーション・モジュールをサービス・アプリケーションとしてデプロイして管理する場合に最適です。

### サーバーおよびバス

これにより、以下のタスク領域に適したパネルが表示されます。

- メディエーション・モジュールとサービス・アプリケーションのデプロイメントをサポートするために必要とされるサービス統合バス、サーバー、サーバー・クラスター、メッセージング・エンジン、およびネットワーク・トポロジーの定義
- インフラストラクチャー・サービスを使用可能および使用不可に設定
- アプリケーションおよびメディエーション・モジュールのインストール
- デプロイされたサービス・アプリケーションおよびメディエーション・モジュールに必要なリソース (例えば、JMS 接続ファクトリーや Common Event Infrastructure プロファイル) の作成
- バスおよびサーバー環境の操作制御

これは、管理者がサービス・アプリケーションおよびメディエーション・モジュールのデプロイメントをサポートするために必要なサーバーおよびバス環境を管理する場合に最適です。管理には、ネットワークおよびバス・トポロジーの定義、適切なリソースの定義、ランタイム・システムのモニター、およびランタイム・エラーが発生した場合はそのトラブルシューティングが含まれます。バスおよびサービス環境の管理について詳しくは、189 ページの『バス環境の管理』を参照してください。



## バス環境の管理

サービスの統合担当者は、エンタープライズ・サービス・バスを稼働するサーバーやバス環境について意識することなく、SCA モジュールをデプロイすることができます。その一方で、管理者は、サーバーやバス環境の管理が必要となる場合があります。例えば、サーバーの始動や停止、SCA モジュールへのサービス品質の変更、サーバー容量の追加、より分散化されたバス環境の作成などが必要になる場合があります。

スタンドアロン・サーバーでは、デフォルトのバス環境は、SCA モジュールをデプロイするために使用される 2 つのサービス統合バスから構成されます。このサーバーを使用して、サーブレット、Enterprise Java Bean、JMS 宛先など、J2EE のコンポーネントおよびリソースをデプロイすることもできます。

デフォルトのバス環境が、ご使用の要件に適切な場合もあります。また一方では、サーバー容量を増やしたバス環境や、さまざまな部門に対応する、より分散化されたバス環境を作成したり、テスト用設備と実動用設備を切り離す別個のバス環境を作成することもできます。

バス環境の管理には、インストールおよびカスタマイズの作業、バス・トポロジーの計画、および製品構成の作成などの作業が含まれます。重点を置いているのは、サービス統合バスの管理、サーバーの管理、サーバーのリソースの管理、セルとノードの論理管理ドメインのセットアップと管理、クラスタリングと高可用性構成を利用したワークロードのバランス方法です。

WebSphere Process Server でのバス環境は、WebSphere Application Server により提供されるサービス統合テクノロジーを使用します。ここでの各トピックでは、バス環境の管理に必要なサービス統合テクノロジーに関連するタスクについて説明します。

- サーバーおよびバス環境の管理の概要
- SCA.SYSTEM バス・トポロジーの管理
- SCA ランタイム用サーバーまたはクラスタの構成
- サーバー管理
- サーバー・クラスタを使用したワークロード・バランシング
- バス・トポロジーのその他の用途

### バス環境の管理の概要

Service Integrator では、サーバーおよび、エンタープライズ・サービス・バスを実行するバス環境を考慮することなく SCA モジュールをデプロイできます。ただし、サーバーおよびバス環境の管理が必要となる場合があるため、管理者はその管理について理解しておく必要があります。

この一連のトピックでは、バス環境を管理する背景について説明します。初めてバス環境の管理を始める場合に役立ちます。バス環境の管理について十分な知識がある場合は、『バス環境の管理』のサブトピックである他の各タスクの説明に直接進んでも構いません。

**バス環境の概要:**

バス環境は、1 つ以上のサービス統合バス、ESB サーバー、およびそのリソースで構成されており、セルとノードからなる論理的な管理可能ドメインに編成されます。

サービスの統合担当者は、バス環境を意識することなく SCA モジュールをデプロイすることができますが、管理者はサーバーとバスを管理する必要があるため、その管理作業の内容を理解する必要があります。

- SCA ランタイム (メディエーション・モジュールによって利用される) は、コンポーネントとモジュール間の非同期対話をサポートするための堅固なインフラストラクチャーとして、SCA.SYSTEM サービス統合バス上のキューを使用します。このキューは、SCA.SYSTEM バスのメンバーとして、サーバーによりホストされます。
- ESB サーバーは、WebSphere Process Server でメディエーション・モジュールおよびサービス・アプリケーションを実行するのに必要な統合テクノロジー、インフラストラクチャー・サービス、構成、およびランタイム管理を提供します。バス・メンバーとして、サーバーには、SCA.SYSTEM バスのコア・メッセージング機能を提供するメッセージング・エンジンがあります。

サーバーと SCA.SYSTEM バスは両方とも、SCA モジュールのデプロイおよび実行に適したデフォルト・プロパティで構成されています。

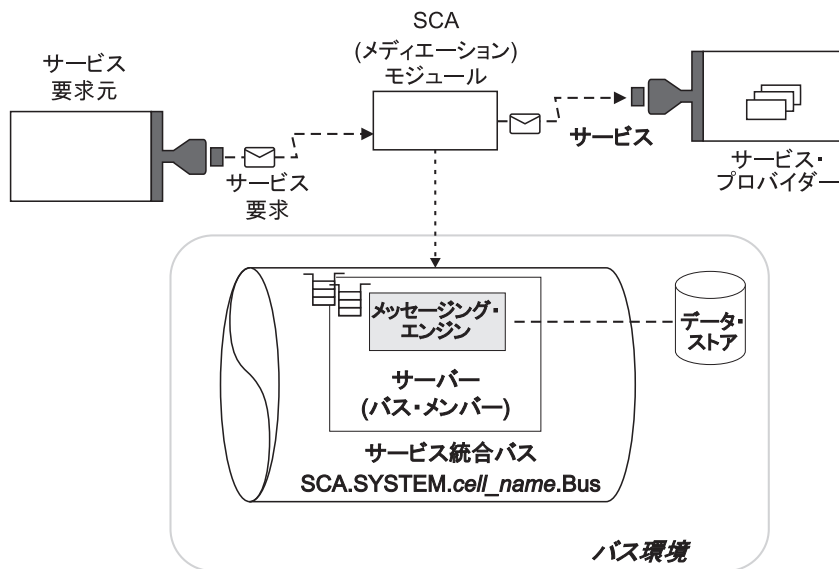


図 10. 1 つのサーバーが SCA.SYSTEM サービス統合バスに割り当てられたバス環境： バス・メンバーとして、サーバーには 1 つのメッセージング・エンジンが割り当てられており、このエンジンには状態データおよびメッセージを保管するためのデータ・ストアがあります。この図は、メディエーション・モジュールがバス環境にデプロイされていて、バス・メンバーに割り当てられていることも示しています。

より高度な使用方法として、スタンドアロン・サーバーのバス環境の構成を変更したり、ネットワーク・デプロイメント・セルのバス環境を作成したりできます。以下に例を挙げます。

- セキュアで確実な配信 (メッセージが失われないことが保証され、安全に移送される) からベストエフォート (システム障害が発生した場合はメッセージが失われる可能性がある) まで、さまざまなサービス品質を構成できます。

- メディエーション・モジュールをホストするサーバーをいくつか提供するようにネットワーク・デプロイメント・セルをセットアップすることもできます。このようにすると、スケーラビリティの確保、処理可能なクライアント接続数の拡大、およびメッセージ・スループットの向上という利点が得られます。サーバー・クラスターを作成することもできます。これにより、一群のサーバーを一括管理できるようになり、これらのサーバーをワークロード管理に関与させることができます。
- 全バス環境は、各部門ごとに別個の管理可能ドメインを用意したり、テスト用設備と実動用設備を区別するために、いくつかのスタンドアロンおよびデプロイメント・マネージャーのプロファイルで構成されている場合があります。各プロファイルには独自の SCA.SYSTEM サービス統合バスがあります。

SCA モジュールに使用される SCA.SYSTEM バスの他に、別のサービス統合バスを作成し、それを使用して、モジュールが提供するサービス統合ロジックをサポートすることもできます。例えば、SCA.APPLICATION バスが用意されており、JMS バインディングによってデプロイされたモジュールの JMS キュー宛先やその他の JMS リソースを定義するために使用できます。

他のバスを作成して、WebSphere Application Server の場合と同様に使用できます。例えば、WebSphere Process Server 内でサービス要求元やサービス・プロバイダーとして動作するアプリケーション用、あるいは WebSphere MQ へのリンク用などです。WebSphere Process Server デプロイメント・マネージャーを使用して、WebSphere Application Server 上にデプロイされたアプリケーションおよびモジュール用の個々のアプリケーション・サーバーを管理することもできます。

#### アプリケーション・サーバー:

WebSphere Process Server には、WebSphere Application Server が組み込まれています。これにより、J2EE アプリケーション・コンポーネント (エンタープライズ Bean、Web サービス、サーブレット、JCA コネクタ、および非同期 Bean など) をサポートする、高性能で、セキュアかつ管理可能なミドルウェア・コンテナが提供されます。

WebSphere Process Server は、サービス・アプリケーションのデプロイメントおよび管理で使用する機能を提供します。例えば、ブラウザ・ベースの管理コンソール、コマンド・スクリプト、(サービス・バス、サーバー、およびクラスターの) トポロジー管理、JMX と呼ばれる標準管理プログラミング・インターフェース・セット、すべてのホストされるアプリケーション・コンポーネントおよびインフラストラクチャーを包含する管理モデルなどがあります。

サーバー・インフラストラクチャーは、ほとんどのミドルウェアに共通のネットワーク I/O、キューイング、スケジューリング、スレッド化、およびディスパッチング機能を提供します。また、さまざまなトランスポートとプロトコルをサポートすることにより、プロトコル・セマンティクスおよびメッセージ・エンコードのコーディングを担当する開発者の負担を軽減します。

WebSphere Process Server は、通常はインプリメント困難な高品質のサービスをランタイム機能で提供することにより、アプリケーションを拡張および強化できます。例えば、アトミック・トランザクションや呼び出し元の特定のセキュリティ・ロールなどのホスティング・コンテナの要件は、サービスのメタデータ中で

宣言として表現されます。パフォーマンスのモニターと測定、ワークロード・ルーティング、およびワークロード管理などの運用機能は、管理上使用可能にできます。

WebSphere Process Server は、J2EE プログラミング・モデルの標準プログラミング・インターフェースをインプリメントする、アプリケーション・ランタイム・サービスを提供します。例えば、アプリケーションのトレースおよびロギングのインターフェースは、サーバー・ランタイムのインターフェースと統合され、問題診断についての一貫性のあるビューを提供します。セキュリティー・インターフェースは WebSphere Process Server セキュリティーを拡張してアプリケーション固有のビジネス・ニーズに対処します。ネーム・スペース・サービスは、サービスのディスカバリーおよびバインディングを可能にし、管理ドメイン全体で持続できます。

WebSphere Process Server でデプロイされるサービス・インプリメンテーションでは、明示的にコード化したりサーバー・インフラストラクチャーを認識したりしなくても、これらの利点をすべて享受できます。

### サービス統合バス:

サービス統合バスは、WebSphere Process Server でデプロイされるメディエーション・モジュールと対話エンドポイントのリソースを構成する際の有効範囲を規定します。

**要確認:** エンタープライズ・サービス・バスは、SCA で記述された対話エンドポイントの統合ロジックおよび接続に使用される、アーキテクチャーです。サービス統合バスは、WebSphere Process Server を構築するときの基盤である WebSphere Application Server によって提供されるテクノロジーの一部です。ご使用のエンタープライズ・サービス・バスには、複数のサービス統合バス、WebSphere MQ、および他のテクノロジーが組み込まれている場合があります。

エンタープライズ・サービス・バスの略称として、*ESB* という用語を使用します。サービス統合バスの略称として、*バス* という用語を使用します。

WebSphere Process Server では、バスにより、特定品質の対話サービスを提供するエンドポイント間でのメッセージ・ルーティングが可能になり、必要であれば一時的にメッセージを持続させることができます。セキュアで確実な配信 (メッセージが失われないことが保証され、安全に移送される) からベストエフォート (システム障害が発生した場合はメッセージが失われる可能性がある) まで、さまざまなサービス品質を構成できます。エンドポイントのインプリメント担当者は、SCA モジュールのエクスポートとインポートで注釈を宣言することにより、望ましいサービス品質を選択します。サービス品質を指定しない場合、WebSphere Process Server によってデフォルト設定が適用されます。

WebSphere Process Server をインストールすると、SCA モジュールをデプロイするための *SCA.SYSTEM* バスが作成されます。このバスの構成が SCA モジュールにとって不適切な場合、または他の用途のためにサービス統合バスが必要な場合は、マルチ・サーバー・バス、別々のサーバーを使用するいくつかのシングル・サーバー・バス、および WebSphere MQ にリンクされたバスなどのさまざまなバス環境から選択することができます。

多くのシナリオでは、単純なバス・トポロジー (例えば、単一サーバーなど) が使用されます。複数のサーバーを単一のバスに追加することにより、接続ポイントの数を増やすことができます。サーバー・クラスターをバスのメンバーとして追加すれば、スケーラビリティを向上させ、高可用性を実現できます。ただし、サーバーは、バス・メンバーになっていなくても、バスに接続できます。より複雑なバス・トポロジーでは、複数のバスが構成されており、相互に接続されて複雑なネットワークを形成している場合があります。企業は、組織上の理由で複数のバスをデプロイすることもあります。例えば、独立部門をいくつか持つ企業では、部署ごとに別々に管理されるバスを持つ場合があります。

### **サーバー・クラスター:**

サーバー・クラスターは、一括管理され、ワークロード管理に関与する一群のサーバーです。クラスターのメンバーになっているサーバーは、別々のホスト・マシンに存在していても構いません。

クラスターに属するサーバーは、そのクラスター・セットのメンバーであり、それらのサーバーすべてに同一のアプリケーション・コンポーネントをデプロイする必要があります。これらのサーバーで稼働するよう構成されたアプリケーションを除いて、クラスター・メンバー間で構成データを共有する必要はありません。同一クラスターのメンバーには、大規模なマルチプロセッサ・エンタープライズ・サーバー・システムで稼働するメンバーもあれば、より小規模なシステムで稼働するメンバーもあります。この 2 つのクラスター・メンバーのサーバー構成の設定はまったく異なります。ただし、割り当てられたアプリケーション・コンポーネントの領域は例外です。この構成領域に関しては、同じ設定です。これにより、すべてのワークロードを単一アプリケーション・サーバーで処理するのではなく、クライアント作業をクラスターのすべてのメンバー間で分散させることができます。

WebSphere Process Server では、サーバー・クラスターは、基盤となる WebSphere Application Server によってサポートされます。

サーバー・クラスターについて詳しくは、関連するトピックを参照してください。

### **バス環境の選択**

スタンドアロン・サーバーでは、デフォルトのバス環境は、SCA モジュールをデプロイするために使用される 2 つのサービス統合バスから構成されます。ご使用のサービス・アプリケーションにこのバスの構成が適さない場合は、デプロイメント・マネージャー・セル内に 1 つの複数サーバー用のバス、異なるサーバーを使用する複数の単一サーバー用バス、アプリケーション用や WebSphere MQ へのリンク用などのその他のバスといったバス環境から、自由に選択することができます。

SCA モジュールに提供されている SCA.SYSTEM バスを使用する以外にも、ほかのアプリケーション・サーバーおよびサービス統合バスを作成して、別のアプリケーションやモジュールをサポートしたり、WebSphere MQ ネットワークに接続したりすることができます。別のサービス統合バスを WebSphere Application Server Network Deployment バージョン 6 などで使用する場合についての説明は、WebSphere Application Server トピックへのリンクにより参照できます。この一連のトピックでは主に、SCA.SYSTEM バスを使用した SCA モジュールのサポートに重点を置いています。

バス環境は、インストールと構成の間に作成されるため、SCA モジュールをデプロイする前に、バス環境に対する要件を検討する必要があります。バス環境を選択する場合は、以下の点を検討してから、このトピックの最後に示されている関連するサブトピックをお読みください。

- バスにデプロイされたモジュールに必要なクライアント接続数およびスループットを考慮します。

その目的は、クライアントが感じるモジュールのパフォーマンスが低下し始める以下のポイントを特定することです。

- 新たにこれ以上クライアント接続を行うとパフォーマンスが低下し始めるときの、バスに同時接続しているクライアントの接続数。
- 新たにこれ以上の要求をメッセージング・エンジンを介して送信しようとするときパフォーマンスが低下し始めるときの、メッセージング・エンジンで送受信されている要求および応答の数。

サーバーが実行しているホストの特性やデプロイするモジュールの特質、その他の要因によって方式は異なるため、すべての環境に適用できる 1 つの特定の方式はありません。

単一サーバー・バスを使用している場合に、クライアント接続数に起因してパフォーマンスが低下したり、またはスループットが低下し始めたときは、以下のいくつかの方法によりバス環境の容量を増大できます。

- スタンドアロン・プロファイルで、同じサーバーを使用する複数の単一サーバー・バスを作成できます。こうすることで、クライアント接続を複数のバスに分散できます。しかし、この状態では、要求のスループットを担っているのは相変わらず 1 つのサーバーのみです。
  - さらに多くのクライアント接続および要求のスループットを実現するには、複数のバスに分散された複数のサーバーを使用します。(1 つ以上のバスに分散された複数のサーバーを使用するには、ネットワーク・デプロイメント・セル内の管理対象ノードに関するサーバー・プロファイルが必要です。)
- メッセージング・エンジンを介して送信される要求のサイズを考慮します。

各メッセージング・エンジンは、要求および要求関連データを含む 2 つのメモリー・バッファーを管理します。十分なスペースがない場合、メッセージング・エンジンはバッファーにデータを追加するときに、バッファーに含まれる既存のデータを破棄してスペースを確保することがあります。

実行中のメッセージング・エンジンが必要以上にバッファーからデータを破棄する場合があります。その場合は、別のサーバーをバスに追加して、別のメッセージング・エンジンを作成できます。または、異なるサーバーをそれぞれのバス・メンバーとして使用する複数の単一サーバー・バスを作成することもできます。各サーバー内のメッセージング・エンジンは各自のメモリー・バッファー・セットおよびデータ・ストアを使用します。(1 つ以上のバスに分散された複数のサーバーを使用するには、ネットワーク・デプロイメント・セル内のノードに関するサーバー・プロファイルが必要です。)

- ご使用の各サービス・アプリケーションに異なるサービス品質を適用するかどうかを考慮します。

各バスのサービス品質などのプロパティは、固有に構成できます。複数のバスを作成して、それぞれ異なるサービス品質で構成し、各モジュールを適切な構成のバスにデプロイできます。

- バス内で複数のサーバーを使用するその他の理由を考慮します。

アプリケーションによっては、1つのサーバーのみで構成されるサービス統合バスが適切な場合もあります。しかし、1つのバス内で複数のサーバー（各サーバーにはメッセージング・エンジンが作成されている）を使用することで、次のような利点が得られます。

- メッセージング・ワークロードを複数のサーバーに分散します。
- 要求処理を要求元アプリケーションに近い場所で行い、ネットワーク・トラフィックを削減します。例えば、送信アプリケーションと受信アプリケーションが同じサーバー・プロセス内で実行されている場合は、それらのアプリケーション間で送受信されるすべての要求を、リモート・サーバー内で実行されているメッセージング・エンジンを介して発信する方法は非効率です。
- システム障害やリンク障害時の可用性が向上します。これにより、Single Point of Failure の発生をなくし、必要な場合には、2つのサーバー間のストア・アンド・フォワードが可能になります。
- スケーラビリティが向上します。
- すべてのネットワーク・ホストが1つのメッセージング・エンジンに接続するのを制限するファイアウォールまたはその他のネットワーク制限に対応できます。

- 複数の SCA.SYSTEM バス環境を使用するその他の理由を考慮します。

各サービス統合バスは個別に構成できるため、バスごとの構成を変えて各モジュール用に適切な形にすることができます。例えば、いくつかのバスをセキュリティを備えた実稼働環境用に構成し、それ以外のバスをセキュリティを備えていないテスト用に構成できます。

また、モジュールの管理ごとに複数のバスを作成することもできます。例えば、各管理セルとその SCA.SYSTEM バスを組織内の異なる部門用に使用したり、テスト機能と実動機能に分けて使用したりすることもできます。

SCA.SYSTEM バス以外にも、別のバスを他のアプリケーションで使用するように作成したり、バスを接続してバス間のメッセージングを可能にしたりできます。異なる組織内のバスにも接続できます。バスを相互に接続すると、あるアプリケーションから別のバスに存在するアプリケーションへのメッセージ送信や、他のバスに存在しているリソースの使用が可能になります。バス間のリンクの構成で許可されていれば、公開されたメッセージを複数のバスに送信することもできます。

- SCA 以外のサービス統合バスを使用する理由を考慮します。

SCA モジュールに使用される SCA.SYSTEM バスの他に、別のサービス統合バスを作成し、それを使用して、モジュールが提供するサービス統合ロジックをサポートすることもできます。例えば、SCA.APPLICATION バスが用意されており、JMS バインディングによってデプロイされたモジュールの JMS キュー宛先やその他の JMS リソースを定義するために使用できます。

他のバスを作成して、WebSphere Application Server の場合と同様に、WebSphere Process Server 内でサービス要求元やプロバイダーとして動作するアプリケーション用に使用したり、WebSphere MQ にリンクしたりできます。WebSphere Process Server デプロイメント・マネージャーを使用して、WebSphere Application Server 上にデプロイされたアプリケーションおよびモジュール用に使用する各アプリケーション・サーバーを管理することもできます。

- SCA モジュールをサポートしないアプリケーション・サーバーが必要かどうかを考慮します。

WebSphere Process Server のデプロイメント・マネージャー・セルには、WebSphere Application Server のサーバーを実行するアプリケーション・サーバー・ノードを組み込むことができます。これらのアプリケーション・サーバーを、WebSphere Application Server によりサポートされるアプリケーションおよびモジュールに使用できます。WebSphere Application Server のサービス統合テクノロジーを利用しない場合は、アプリケーション・サーバーをサービス統合バスに追加する必要はありません。

#### 提供される SCA\* バス:

SCA モジュールのデプロイメントをサポートするために、1 つまたは 2 つのサービス統合バスが作成されます。これらのバスは、名前が SCA で始まり、SCA\* バスと呼ばれることもあります。

WebSphere Process Server スタンドアロン・プロファイルを作成すると、1 つのアプリケーション・サーバーが自動的に SCA サポート用に構成され、そのサーバーは SCA.SYSTEM バスおよび SCA.APPLICATION バスのメンバーとなります。

ローカル・メッセージングで非同期 SCA サポート用の最初のクラスターまたはサーバーを構成すると、新しい Network Deployment セル内に、SCA.SYSTEM バスが自動的に作成されます。構成されたサーバーまたはクラスターは、バスのメンバーとなります。同様に、最初のクラスターまたはサーバーを JMS SCA サポート用に構成すると、SCA.APPLICATION バスが作成されます。

#### SCA.SYSTEM バス

SCA モジュール (エンタープライズ・サービス・バスでサービス統合ロジックを提供するメディエーション・モジュールなど) のキュー宛先をホストするために使用されるバスです。SCA ランタイム (メディエーション・モジュールによって利用される) は、SCA.SYSTEM バスのキュー宛先を堅固なインフラストラクチャーとして使用して、コンポーネントとモジュール間の非同期対話をサポートします。

Network Deployment セルでは、非同期 SCA のサポート用の最初のサーバーまたはクラスターを構成するときにこのバスが作成されます。

#### SCA.APPLICATION バス

このバスを使用して、JMS バインディングを使用してデプロイされるモジュールのリソースを作成します。このバスは、SCA モジュールのデプロイ以外の目的でサービス統合バスを作成する場合の一例です。

Network Deployment セルでは、JMS SCA のサポート用の最初のサーバーまたはクラスターを構成するときにこのバスが作成されます。



SCA.SYSTEM バスは、リソース (キュー宛先など) をメディエーション・モジュールおよび対話エンドポイント用に構成する有効範囲を提供します。バスにより、特定品質の対話サービスを提供するエンドポイント間でのメッセージ・ルーティングが可能になり、必要であれば一時的にメッセージを持続させることができます。セキュアで確実な配信 (メッセージが失われないことが保証され、安全に移送される) からベストエフォート (システム障害が発生した場合はメッセージが失われる可能性がある) まで、さまざまなサービス品質を構成できます。エンドポイントのインプリメント担当者は、SCA モジュールのエクスポートとインポートで注釈を宣言することにより、望ましいサービス品質を選択します。サービス品質を指定しない場合、WebSphere Process Server によってデフォルト設定が適用されます。

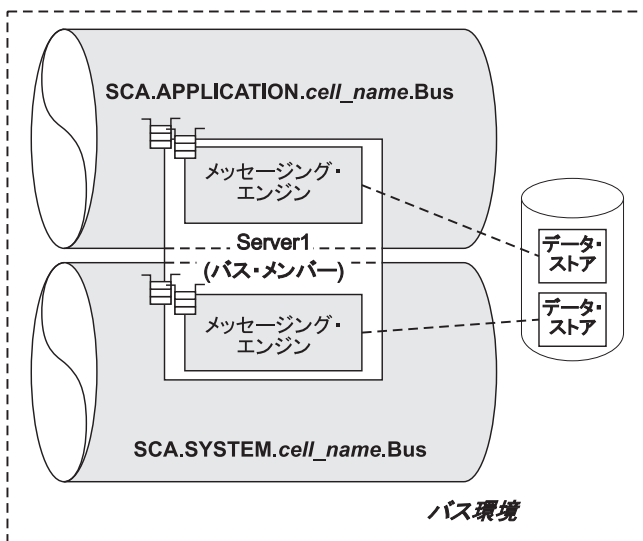


図 11. スタンドアロン・サーバー・プロファイルの SCA.\* バス： WebSphere Process Server のインストール時に作成されるサービス統合バス。

スタンドアロン・プロファイルには、バスで 1 つのメッセージング・エンジンとなるバス・メンバーが 1 つあり、このトポロジーは、一部のアプリケーションにとっては適切です。

単一のメッセージング・エンジンは、クライアント接続数が多くなり過ぎる場合、1 つのメッセージング・エンジンでメッセージ・スループット率を維持できない場合、またはメッセージのサイズがメッセージング・エンジンで 사용되는メッセージ・バッファに有害な影響を与える場合には適切ではありません。

サービス統合バスに複数のメッセージング・エンジンを追加するには、Network Deployment セル内に管理対象ノードが存在する必要があります。

#### 単一サーバー・エンタープライズ・サービス・バスの作成:

最もシンプルなトポロジーは、単一スタンドアロン・サーバーから構成されるエンタープライズ・サービス・バスです。

単一サーバーのエンタープライズ・サービス・バスでは、SCA.SYSTEM バスには、自身の唯一のバス・メンバーである 1 台のサーバーが配置されます。WebSphere Process Server にメディエーション・モジュールをインストールすると、モジュール

で使用されるキュー宛先がバス・メンバーに定義されます。これらのキュー宛先は SCA ランタイムによって使用され、このランタイムは、コンポーネントとモジュールの間の非同期対話をサポートする堅固なインフラストラクチャーとして、メディエーション・モジュールによって活用されます。

サーバーは、キュー宛先や要求の処理で利用するサービス統合テクノロジーを提供するメッセージング・エンジンをホストします。

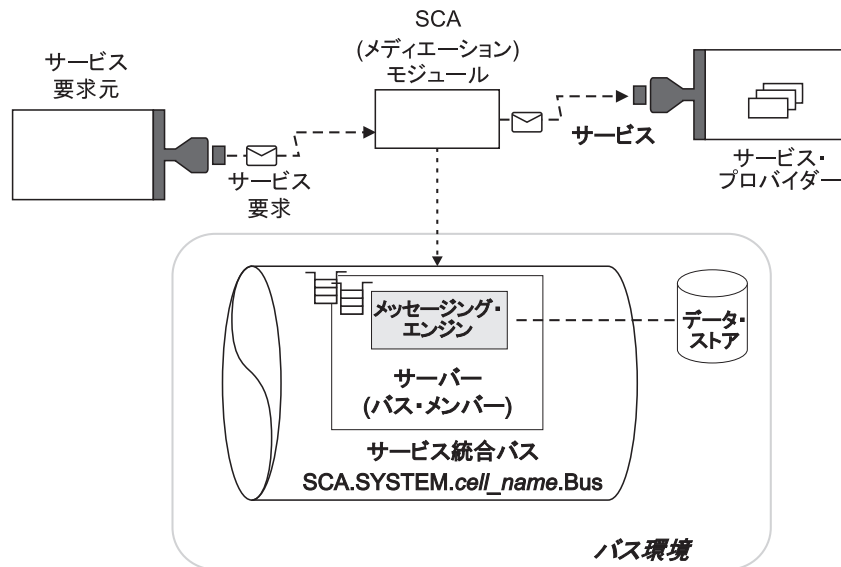


図 12. 単一サーバー・バス：バス・メンバーとして 1 台の ESB サーバーを持つ、1 つのサービス統合バスからなるバス環境を表しています。この図は、SCA モジュールがバス環境にデプロイされ、必要な宛先がバス・メンバーに割り当てられていることも示しています。

スタンドアロンのサーバーでは、バスに配置可能なメッセージング・エンジンは 1 台のみで、一部のシナリオにはこのトポロジーが適しています。

単一のメッセージング・エンジンが適さないシナリオとしては、クライアント接続数が極端に多くなる場合、メッセージのスループット速度が単一のメッセージング・エンジンでは維持できない場合、または、メッセージのサイズが単一のメッセージング・エンジンで提供できる以上のスペースをデータ・ストアに必要とする場合などがあります。

サービス統合バスに複数のメッセージング・エンジンを追加するには、Network Deployment セル内に管理対象ノードを作成する必要があります。

1. WebSphere Process Server の完全インストールを実行します。コア・プロダクト・ファイルがインストールされ、最初のスタンドアロン・サーバー・プロファイルが作成されます。
2. 最初のステップのコンソールを使用するか、startServer server1 コマンドを実行して server1 を始動します。

これで、WebSphere Process Server サンプルを実行し、エンタープライズ・サービス・バスにサービス・アプリケーションをデプロイできます。

## クラスタリングが存在しない複数サーバーのエンタープライズ・サービス・バスの作成:

複数のサーバーで構成されるエンタープライズ・サービス・バスでは、スケーラビリティの確保、処理可能なクライアント接続数の拡大、およびメッセージ・スループットの向上、という利点が得られます。また、SCA モジュールを異なるサーバーにデプロイして、サービスのさまざまなリソースや品質を提供したり、組織内の異なる部門を何らかの形で分離したり、テスト機能と実動機能を分離したりすることもできます。

バス環境で複数のサーバーを作成するには、Network Deployment セル内に管理対象ノードが存在している必要があります。

メディアエーション・モジュールに必要な SCA ランタイムに対して、各サーバーを構成します。この拡張構成によって、SCA ランタイムで使用されるキューの宛先をこのサーバー上でローカルにホストするか、リモート・サーバー上でホストするかが定義されます。このサーバー上でキューの宛先をホストするように指定した場合、サーバーは SCA.SYSTEM バスのメンバーとなり、キューの宛先が割り当てられるメッセージング・エンジンを取得します。このサーバー上でキューの宛先をホストしないように指定した場合、サーバーは SCA.SYSTEM バスのメンバーとなる必要がなく、メッセージング・エンジンも必要ありません。

200 ページの図 13 に示すシナリオを参考にしてください。

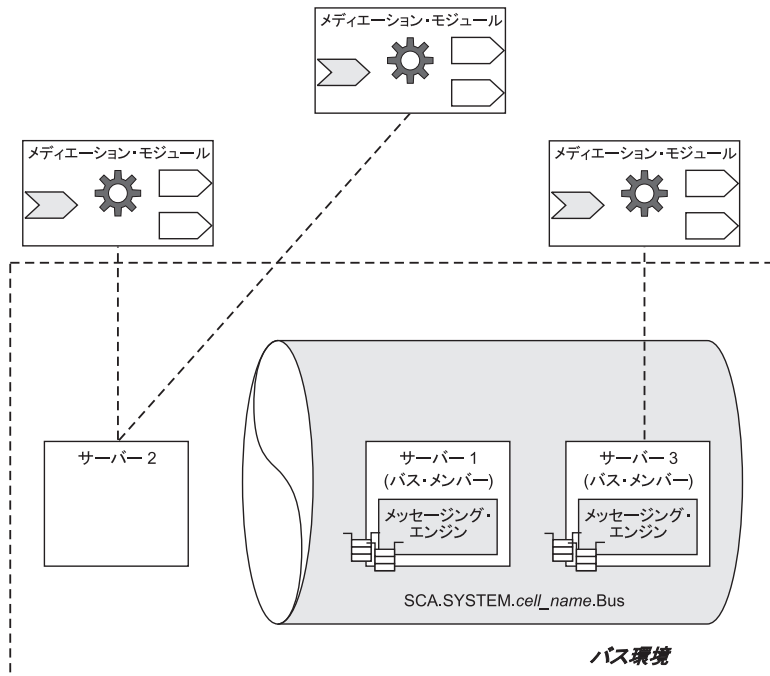


図 13. クラスタリングが存在しない複数サーバーのバス：サーバー 1 は、任意のサーバーにデプロイされるメディエーション・モジュールのキューの宛先をホストし、メディエーション・モジュールやその他のサービス・アプリケーションをホストしないように構成されます。サーバー 2 は、メディエーション・モジュールやその他のサービス・アプリケーションをホストし、別のサーバー（この図のサーバー 2 またはサーバー 3）上でホストされるキューの宛先を使用するように構成されます。サーバー 3 は、メディエーション・モジュールやその他のサービス・アプリケーションをホストし、また任意のサーバーにデプロイされるメディエーション・モジュールのキューの宛先もホストするように構成されます。

SCA.SYSTEM バスに存在するすべてのメッセージング・エンジンは暗黙的に接続されているため、要求はバス内のいずれかのメッセージング・エンジンによって処理されます。バス内のそれぞれのメッセージング・エンジンに割り当てられたリソースの情報は、バス内のすべてのメッセージング・エンジン間で共有されます。

バス内のすべてのメッセージング・エンジンが同時に実行している必要はありません。1 つのメッセージング・エンジンが停止した場合でも、残りのメッセージング・エンジンが稼働し続けます。ただし、メッセージング・エンジンに割り当てられたリソース、特にメディエーション・モジュールのキュー・ポイントは、エンジンが停止すると使用不可になります。また、あるサーバーで使用するために作成されたメッセージング・エンジンはそのサーバーでのみ実行できます。したがって、そのサーバーは Single Point of Failure であり、サーバーが稼働できない場合、メッセージング・エンジンは使用できません。サーバー・クラスターをバスのメンバーとして構成することにより、メッセージング・エンジンはクラスター内のいずれかのサーバーで実行可能となるため、サーバーに障害が起こった場合でも、メッセージング・エンジンはクラスター内の代替サーバー上で実行できます。

任意の Network Deployment セル内に、複数サーバーのエンタープライズ・サービス・バスを作成できます。一般に、以下のステップを実行する必要があります (Network Deployment の構成を参照)。

1. 作成する Network Deployment シナリオを決定します。

2. Network Deployment セルを 1 つ以上の管理対象ノードで構成します。
3. デプロイメント・マネージャー管理コンソールから、管理対象ノード上にサーバーを作成します。
4. SCA ランタイムのサーバーを構成します。

これで、WebSphere Process Server サンプルを実行し、エンタープライズ・サービス・バスにサービス・アプリケーションをデプロイできます。

#### **クラスタリングが存在する複数サーバーのエンタープライズ・サービス・バス:**

Network Deployment セルは、複数のサーバー (その一部またはすべてがサーバー・クラスタのメンバーである) で構成されるエンタープライズ・サービス・バスに使用できます。

SCA モジュールのキューの宛先をホストするサーバーは、SCA.SYSTEM バス内に 1 つのメッセージング・エンジンを保有します。多くの用途においてこれで十分ですが、このメッセージング・エンジンを実行できるのは、そのエンジンを作成したサーバー上だけです。したがって、そのサーバーは Single Point of Failure であり、サーバーが稼働できない場合、メッセージング・エンジンは使用できません。サーバー・クラスタをバスのメンバーとして構成することにより、メッセージング・エンジンはクラスタ内のいずれかのサーバーで実行可能となるため、そのサーバーに障害が起こった場合でも、メッセージング・エンジンはクラスタ内の代替サーバー上で実行できます。これについては、202 ページの図 14 で説明します。

クラスタ・バス・メンバーを構成するもう 1 つの利点は、SCA モジュールに関連したワークロードを複数のサーバー間で共有できることです。あるクラスタ・バス・メンバーにデプロイされた SCA モジュールの場合、使用するキューの宛先はクラスタ・サーバーにより実行される一連のメッセージング・エンジン間で分割されます。SCA モジュールを介して渡されるメッセージの共有については、クラスタ内のメッセージング・エンジンがそれぞれ処理します。

つまり、クラスタ・バス・メンバーを使用することにより、構成可能なポリシーに応じてフェイルオーバーまたはワークロードの共有、あるいはその両方を実現できます。

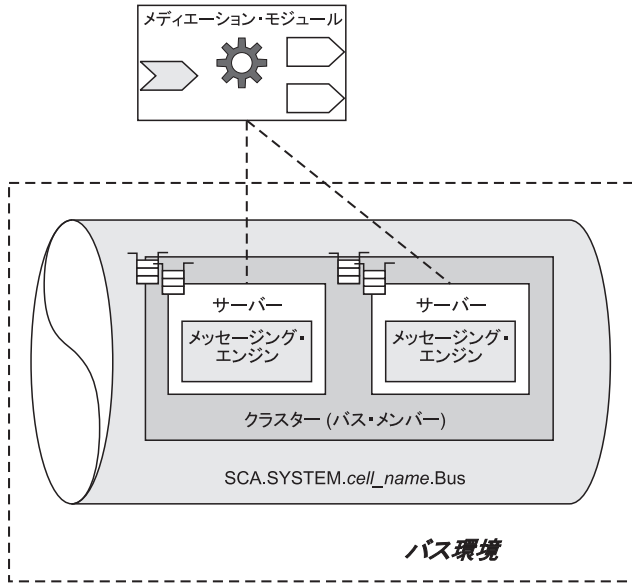


図 14. フェイルオーバー用クラスター・サーバーが存在する複数サーバーのバス

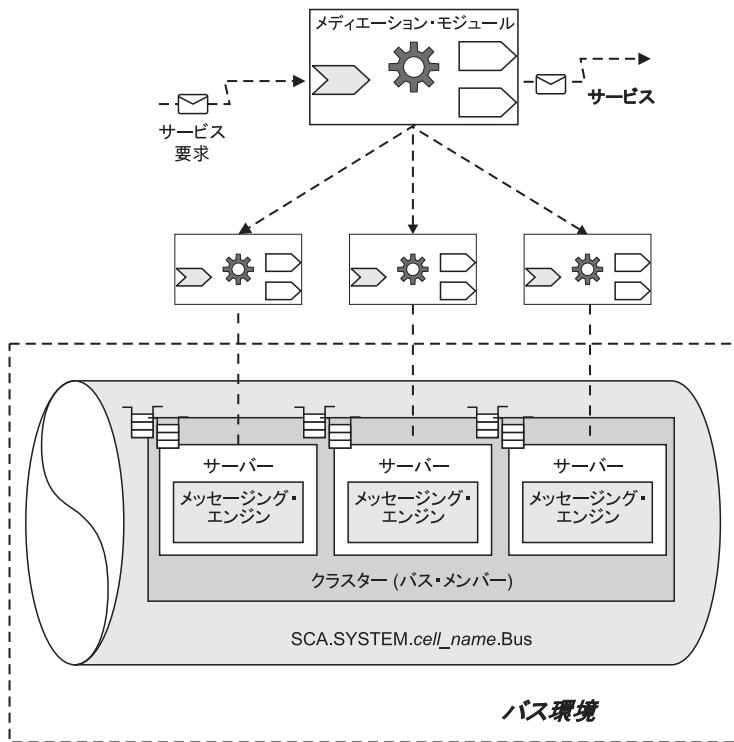


図 15. ワークロード共有用のクラスター・サーバーが存在する複数サーバーのバス

任意の Network Deployment セル内に、クラスタリングが存在する複数サーバーのエンタープライズ・サービス・バスを作成できます。一般に、以下のステップを実行する必要があります (『203 ページの『Network Deployment の構成』』を参照)。

1. 作成する Network Deployment シナリオを決定します。
2. Network Deployment セルを 1 つ以上の管理対象ノードで構成します。

3. デプロイメント・マネージャー管理コンソールから、管理対象ノード上にクラスタを作成します。
4. SCA ランタイムのクラスタを構成します。

これで、WebSphere Process Server サンプルを実行し、エンタープライズ・サービス・バスにサービス・アプリケーションをデプロイできます。

### Network Deployment の構成:

Network Deployment によって、ユーザーはサーバーのグループをデプロイメント・マネージャーから集中的に管理できます。デプロイメント・マネージャーは、そのセル内のすべての管理対象ノードの構成と状態を管理し、セル内の任意の管理対象ノードにアプリケーションをデプロイします。

Network Deployment を構成する前に、以下の手順を完了してください。

- 作成する Network Deployment シナリオを決定します。を参照してください。
- 同じセル内で分散オペレーティング・システムと z/OS オペレーティング・システムの両方のノードを使用するかどうかを決定します。
- 複数のセルを使用するかどうかを決定します。
- サーバー・クラスタを使用するかどうかを決定します。

ソリューションに Network Deployment が必要と判断した場合は、デプロイメント・マネージャーと管理対象ノードが必要です。

Network Deployment を構成するには、次のステップを実行します。

1. デプロイメント・マネージャーを作成し、始動して、セルに管理対象ノードを追加する準備をします。
  - a. 1 つのシステム上で、デプロイメント・マネージャーを構成します。
  - b. デプロイメント・マネージャーを開始します。
  - c. デプロイメント・マネージャーで、管理コンソールを始動します。
2. 管理対象ノードをセルに追加します。
  - a. **オプション:** デプロイメント・マネージャーに、最初の管理対象ノードとしてスタンドアロン・サーバー・ノードを統合します。スタンドアロン・サーバーを使用して、サービス・アプリケーションの使用状況を検証した後、そのノードを新しい Network Deployment セルに統合できます。これにより、このサービス・アプリケーションの利用を、1 つ以上のノード上で分散サーバー・グループやサーバー・クラスタに発展させることができます。

このステップでは、スタンドアロン・サーバーの SCA\* バスをデプロイメント・マネージャー・セルの SCA\* バスに適合させます。

- b. デプロイメント・マネージャーに対して別のシステム上で、カスタム・プロファイルとして管理対象ノードを構成します。デプロイメント・マネージャーにカスタム・ノードを統合し、空の管理対象ノードを作成することができます。
- c. プロファイルの作成時にカスタム・ノードを統合しなかった場合は、ここで、そのカスタム・ノードをデプロイメント・マネージャーに統合する必要があります。

3. **オプション:** SCA\* バス上に、メッセージング・エンジン用のユーザー指定のデータ・ストア構成を構成します。

サーバー・クラスターを SCA ランタイムの宛先をホストするように構成すると、SCA\* バス上に新しいメッセージング・エンジンが作成されます。データ・ストアと、メッセージング・エンジンがそのデータ・ストアとの対話に使用する JDBC データ・ソースの両方を明示的に構成する必要があります。

同様に、SCA ランタイム用にサーバーを構成すると、SCA\* バス上に新しいメッセージング・エンジンが作成され、デフォルトで Cloudscape データ・ソースを使用するように構成されます。このデフォルトの構成では、管理者による追加構成を必要とせずに、メッセージング・エンジンを稼働させることができます。デフォルトのデータ・ソース構成を使用しない場合は、構成パラメーターを変更できます。例えば、データ・ソースを変更するか、またはデータ・ストアを別の JDBC プロバイダーを使用するように構成できます。

4. 各管理対象ノード用のサーバー・クラスターを構成します。
  - a. サーバー・クラスターを作成します。この操作方法については、「Administering the Bus」の資料の『サーバー・クラスターの作成』を参照してください。
  - b. メディエーション・モジュールに必要な SCA ランタイムのクラスターを構成します。詳細は、「Administering the Bus」の資料の『SCA ランタイムのサーバーまたはクラスターの構成』を参照してください。新しいクラスターにメディエーション・モジュールをデプロイする前に、そのクラスターを構成して、モジュールの SCA ランタイムで使用されるキュー宛先のホスト先を指定します。
  - c. クラスターを開始します。この操作方法については、「Administering the Bus」の資料の『使用可能なサーバー・クラスターの管理』を参照してください。

サーバー・クラスターをサポートする管理対象ノードごとに、このステップを繰り返し実行します。

5. クラスター外部で稼働するサーバーを構成します。Network Deployment 環境に単一サーバーを追加するのは、一般的ではありません。通常は、高可用性を実現するために、Network Deployment 環境内にクラスターを構成することをお勧めします。
  - a. 管理対象ノードで、サーバーを作成します。この操作方法については詳しくは、「Administering the Bus」の資料の『サーバーの作成』を参照してください。
  - b. メディエーション・モジュールに必要な SCA ランタイムのサーバーを構成します。詳細は、「Administering the Bus」の資料の『SCA ランタイムのサーバーまたはクラスターの構成』を参照してください。新しいサーバーにメディエーション・モジュールをデプロイする前に、そのサーバーを構成して、モジュールの SCA ランタイムで使用されるキュー宛先のホスト先を指定します。

サーバーをサポートする管理対象ノードごとに、このステップを繰り返し実行します。

#### 複数のエンタープライズ・サービス・バス・トポロジーの作成:



分散バス環境に SCA モジュールをデプロイおよび管理することが必要になる場合があります。例えば、異なる部門間で個別のエンタープライズ・サービス・バスを使用したり、テスト用設備と実動用設備を区別したりする場合があります。

作成する各スタンドアロンのサーバーまたはデプロイメント・マネージャーには、自身の管理セルがあり、これらは個別のエンタープライズ・サービス・バスの管理可能ドメインとみなせます。

完成したバス環境は、いくつかのスタンドアロン・サーバーとネットワーク・デプロイメント・セルで構成されている場合があります、それぞれのセルは SCA モジュール用に使用する独自の SCA.SYSTEM バスを持つ個別のエンタープライズ・サービス・バスを表しています。

SCA モジュールに使用される SCA.SYSTEM バスの他に、別のサービス統合バスを作成し、それを使用して、モジュールが提供するサービス統合ロジックをサポートすることもできます。例えば、SCA.APPLICATION バスは、JMS SCA サポート用に用意されています。これは、JMS バインディングによってデプロイされたモジュールの JMS キュー宛先やその他の JMS リソースを定義するために使用できます。

他のバスを作成して、WebSphere Application Server の場合と同様に使用したり (例えば、WebSphere Process Server 内でサービス要求元やプロバイダーとして動作するアプリケーション用)、あるいは WebSphere MQ にリンクしたりできます。

WebSphere Process Server デプロイメント・マネージャーを使用して、WebSphere Application Server 上にデプロイされたアプリケーションおよびモジュール用に使用する各アプリケーション・サーバーを管理することもできます。

これらの他のバスは別々に使用したり、バス同士を接続してバス間でメッセージの受け渡しができるようにすることも可能です。また、別の組織内のバスと相互に接続することもできます。バスを相互に接続すると、あるアプリケーションから別のバスに存在するアプリケーションへのメッセージ送信や、他のバスに存在しているリソースの使用が可能になります。公開されたメッセージは、バス間のリンクがこれを許可するように構成されている場合は、複数のバス間をスパンできます。

サービス統合バスは、単一セルに含まれている必要があります。つまり、1 つのバスは複数のセルをスパンできません。ただし、1 つのセルに複数のバスを含めることができます。この場合、セルの各バスは、セルの内部で互いに「外部」となります。1 つのセル内または異なるセル間でバスを相互接続できます。

あるバスを別のバスにリンクするプロセスは、バス同士が同じセル内であっても、異なるセル内であっても同じです。

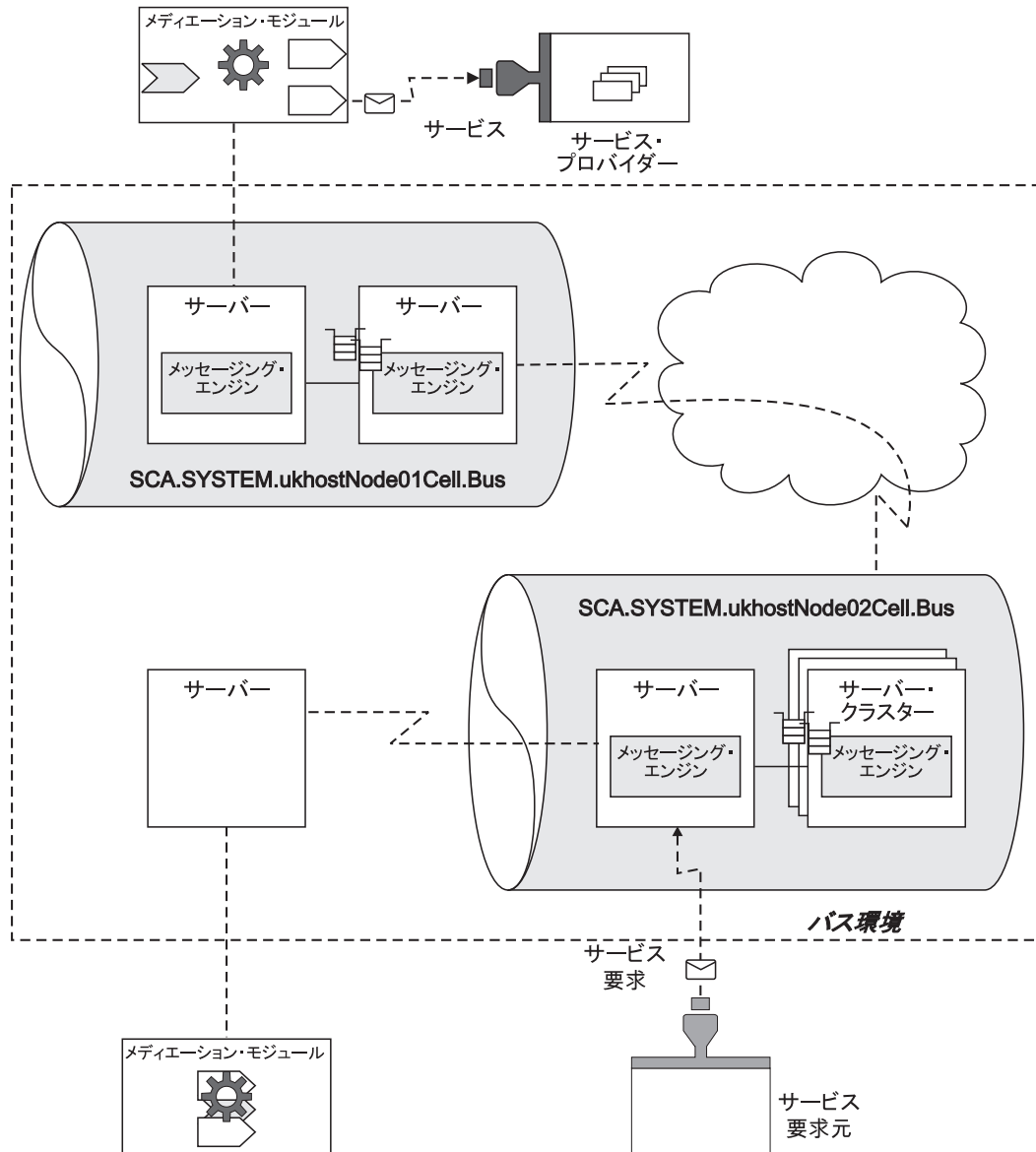


図 16. 複数のエンタープライズ・サービス・バス・トポロジー

複数のエンタープライズ・サービス・バス・トポロジーを作成するには、いくつかのスタンドアロン・サーバーおよび Network Deployment シナリオを構成します。

- 別のスタンドアロン構成をいくつか作成します。
- ネットワーク・デプロイメント構成をいくつか作成します。

これで、WebSphere Process Server サンプルを実行し、エンタープライズ・サービス・バスにサービス・アプリケーションをデプロイできます。さまざまなセル内でバスを接続していると、1つのセル内のモジュールから別のセル内のモジュールへメッセージを送信できるため、ほかのセル内のバスによって提供されるリソースを利用することができます。

**WebSphere MQ ネットワークへのリンクを備えたエンタープライズ・サービス・バス (ESB):**

ESB には、WebSphere MQ ネットワークへのリンクを含めることができます。これにより、WebSphere MQ キュー・マネージャーに接続されたアプリケーションは、サービス統合バスに接続されているアプリケーションにメッセージを送信することができます。そのようなアプリケーションは WebSphere MQ にメッセージを送信できません。

サービス統合バスが WebSphere MQ ネットワークに接続されている場合は、以下のサポートを追加することによってネットワークを拡張します。

- サービス統合バスにデプロイされたサービス・アプリケーション
- サービス統合バスに接続された JMS アプリケーション
- Web サービス要求元またはプロバイダー

WebSphere MQ ネットワークは、メッセージング・エンジン上で構成される外部バスによって表されます。WebSphere MQ リンクと呼ばれる構成体は、送信側と受信側のチャンネルを使用して MQ キュー・マネージャーにメッセージング・エンジンを接続し、これにより、バスと WebSphere MQ ネットワークのいわゆるゲートウェイ・キュー・マネージャーとの間の橋渡しを行うことができます。

WebSphere MQ リンクは、リンクをホストするメッセージング・エンジンだけでなく、バスの他のメッセージング・エンジンとの接続も提供します。バスのメッセージング・エンジン全体は、WebSphere MQ ネットワークにとって、単一のキュー・マネージャーのように見えます (メッセージング・エンジンは WebSphere MQ リンクからキュー・マネージャー名を継承します)。

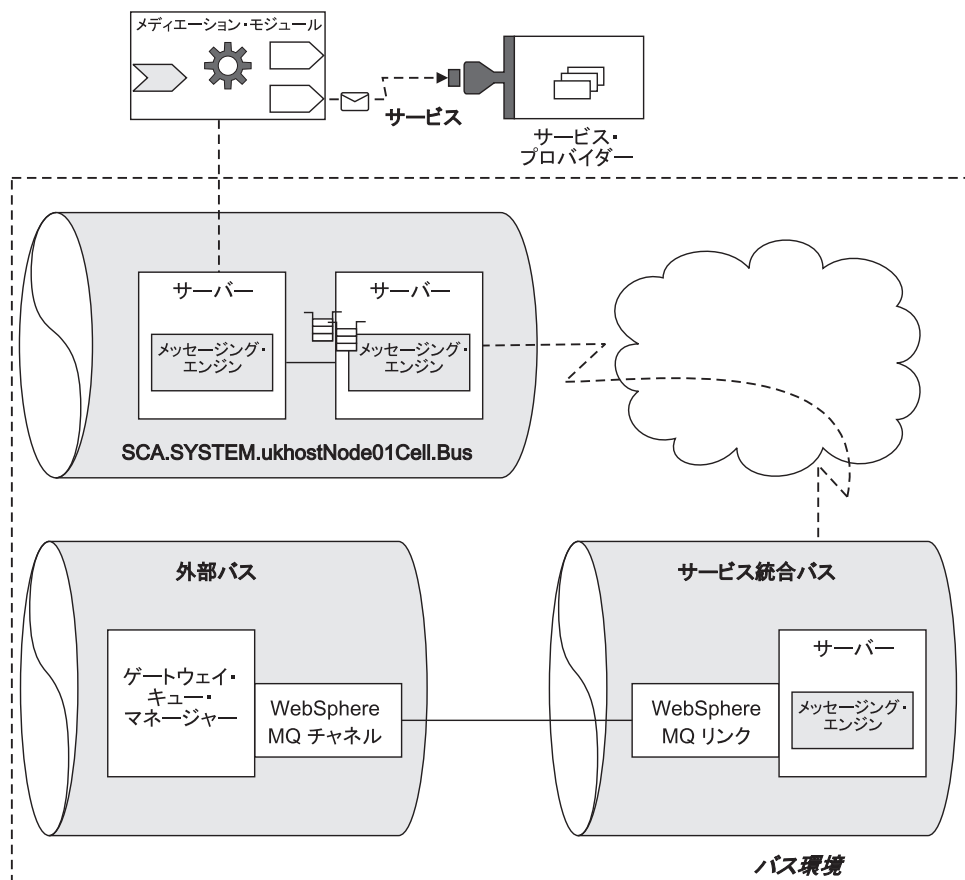


図 17. WebSphere MQ ネットワークへのリンクを備えたサービス統合バス

WebSphere MQ リンクは、さまざまな構成で使用できます。メッセージング・エンジンは、異なるゲートウェイ・キュー・マネージャーへの複数の WebSphere MQ リンクを含むことができます。

WebSphere MQ ネットワークへのリンクは、WebSphere Application Server のサービス統合テクノロジーによってインプリメントされます。WebSphere MQ ネットワークへのリンクの使用について詳しくは、WebSphere Application Server の資料の『WebSphere MQ との相互協調処理 (Interoperating with WebSphere MQ)』を参照してください。

### 管理インターフェースのサーバーおよびバス機能

管理コンソールまたはコマンド行から、サーバーとバス環境の局面を表示および変更することができます。

#### 「サーバー」 → 「アプリケーション・サーバー」

このページ・セットでは、ESB サーバーにアクセスします。このサーバーは、WebSphere Process Server によって提供される特殊なテンプレートから作成されたアプリケーション・サーバーです。

ESB サーバーは、メディエーション・モジュールを実行するために必要な、統合テクノロジー、インフラストラクチャー・サービス、構成、およびランタイムの管理を提供します。

サーバーの詳細設定ページの「**サーバー・メッセージング (Server messaging)**」カテゴリ・ラベルの下からは、メッセージング・エンジン、メッセージング・エンジンのインバウンド・トランスポート、および WebSphere MQ リンクのインバウンド・トランスポートなどのサービス統合プロパティにアクセスできます。

#### 「サービス統合」 → 「バス」

このページ・セットにより、メディエーション・モジュール (および他の SCA モジュール) のデプロイメントで使用する、WebSphere Application Server のサービス統合テクノロジーにアクセスできます。

各サービス統合バスには有効範囲があり、その範囲内で、WebSphere Process Server にデプロイされているメディエーション・モジュールと対話エンドポイントのリソースを構成できます。バスにより、特定品質の対話サービスを提供するエンドポイント間でのメッセージ・ルーティングが可能になり、必要であれば一時的にメッセージを持続させることができます。

WebSphere Process Server スタンドアロン・プロファイルを作成すると、1つのアプリケーション・サーバーが自動的に SCA サポート用に構成され、そのサーバーは SCA.SYSTEM バスおよび SCA.APPLICATION バスのメンバーとなります。

ローカル・メッセージングで非同期 SCA サポート用の最初のクラスターまたはサーバーを構成すると、新しい Network Deployment セル内に、SCA.SYSTEM バスが自動的に作成されます。構成されたサーバーまたはクラスターは、バスのメンバーとなります。同様に、最初のクラスターまたはサーバーを JMS SCA サポート用に構成すると、SCA.APPLICATION バスが作成されます。

WebSphere Process Server にメディエーション・モジュールをインストールすると、SCA\* バスに必要なバスの宛先が作成されます。

バスの詳細設定ページでは、バス宛先など、多くのカテゴリ・ラベルの下からサービス統合リソースにアクセスできます。

## SCA.SYSTEM バス・トポロジーの管理

サービス統合バスである SCA.SYSTEM バスは、SCA モジュールのデプロイに使用します。バスのローカル・トポロジーの表示、バスのプロパティの変更、バス・メンバーとしてのサーバーおよびクラスターの追加、トポロジー全体を管理するその他の管理タスクの実行が可能です。

独自の管理セルを持つスタンドアロン・サーバーは、そのセル用の SCA.SYSTEM サービス統合バスに割り当てられます。このバス・トポロジーは、いくつかのアプリケーション統合シナリオで SCA モジュールをデプロイする場合に適していますが、異なるバス・トポロジーを作成して、バス内で複数のメンバーを使用したり、複数のサービス統合バスや場合によっては WebSphere MQ へのリンクを含むより高度な環境を使用して、その利点を得ることができます。

デプロイメント・マネージャーを作成すれば、ネットワーク・デプロイメント・セルに管理対象ノードを追加し、それらのノードに 1 つ以上のサーバーあるいはサーバー・クラスターを SCA.SYSTEM バスのメンバーとして作成して、SCA モジュール

ルをデプロイできます。SCA.SYSTEM バスとサーバーを作成すると、いくつかの重要なプロパティにのみ値が割り当てられ、その他のプロパティはデフォルト値のままです。

以下のサブトピックの情報をもとに、SCA.SYSTEM バスおよびサーバーのトポロジーとプロパティを表示し、必要に応じて構成し、それらのランタイム状態を管理します。これらのサブトピックでは、メディエーション・モジュールを SCA.SYSTEM バスにデプロイする際に頻繁に行うタスクについて説明しています。サーバーおよびバス環境の一般的な管理に関するこれらのタスクやその他のタスクは、関連するタスクのリストおよび、241 ページの『バス・トポロジーのその他の用途』に説明されています。

SCA.SYSTEM バスおよびサーバーを管理する場合、以下のことを考慮してください。

- サーバー間での SCA モジュールの分散方法
- サーバーをまとめて管理する場合、またはワークロード管理を利用する場合に必要なサーバー・クラスターの種類
- バス、サーバー、その他のリソースの命名規則
- 指定するメッセージのパーシスタンス (永続性) およびその他のサービス品質
- メッセージング・エンジンでメッセージの保持に使用するデータベース

#### 初期 SCA.SYSTEM バス・トポロジーの作成:

WebSphere Process Server スタンドアロン・プロファイルを作成すると、1 つのアプリケーション・サーバーが自動的に SCA サポート用に構成され、そのサーバーは SCA.SYSTEM バスおよび SCA.APPLICATION バスのメンバーとなります。Network Deployment セルでは、非同期 SCA サポート用の最初のクラスターまたはサーバーを構成すると、SCA.SYSTEM バスが作成されます。

このタスクを続行する前に、バス環境において複数のサーバーまたはサーバー・クラスターが必要かどうかを考慮します。スタンドアロン・サーバー・プロファイルで存在できるサーバーは 1 つだけです。同じバス環境で複数のサーバーまたはサーバー・クラスターが必要となる可能性がある場合は、管理対象ノードを持つネットワーク・デプロイメント・セルの作成を考慮します。1 つまたは複数のマシン上で複数のスタンドアロン・サーバーを作成することもできます。

それぞれのスタンドアロン・ノードまたはネットワーク・デプロイメント・セルは個別の管理セルを構成します。管理セルには SCA.SYSTEM バスと呼ばれる固有のサービス統合バスが存在し、これはセル内で SCA モジュールをデプロイするのに使用されます。

このタスクを実行して、セル用に SCA.SYSTEM バスを作成し、1 つの ESB サーバーをバス・メンバーとします。SCA.SYSTEM バスとサーバーを作成するときは、いくつかの重要なプロパティの値のみを入力し、その他のプロパティはデフォルト値のままにします。

バスの構成は変更できます。例えば、バス・プロパティを変更したり、Network Deployment セルでは、サーバー・クラスターまたはサーバーをバス・メンバーとして追加したりします。

1 つのサーバーをバス・メンバーとして SCA.SYSTEM バスを作成するには、次のいずれかの手順を実行します。

- スタンドアロン・サーバーを作成します。

WebSphere Process Server スタンドアロン・プロファイルを作成すると、1 つのアプリケーション・サーバーが自動的に SCA サポート用に構成され、そのサーバーは SCA.SYSTEM バスおよび SCA.APPLICATION バスのメンバーとなります。

このトポロジーではサーバーの数は 1 つに限られます。

- Network Deployment セルを作成してから、管理対象ノード上にクラスターまたはサーバーを追加します。ローカル・メッセージングで非同期 SCA サポート用の最初のクラスターまたはサーバーを構成すると、新しい Network Deployment セル内に、SCA.SYSTEM バスが自動的に作成されます。構成されたサーバーまたはクラスターは、バスのメンバーとなります。同様に、最初のクラスターまたはサーバーを JMS SCA サポート用に構成すると、SCA.APPLICATION バスが作成されます。

ネットワーク・デプロイメントにより、集中管理が行えるという利点、およびサーバー・クラスター間のワークロード・バランシングが可能になるという利点があります。まずは、カスタム・プロファイル用の 1 つの管理対象ノードを持つセルを作成して、その後、必要に応じてそのノードにクラスターまたはサーバーを追加したり、管理対象ノードを新たに追加したりできます。

セル用の管理コンソールにログインすると、「サービス統合 (Service integration)」→「バス (Buses)」ページにリストされた SCA.SYSTEM バスを参照できます。新規サーバーも「サーバー」→「アプリケーション・サーバー」ページにリストされています。

新規サーバーの始動後、SCA.SYSTEM バスを使用して SCA モジュールをデプロイできます。

バスの構成は変更できます。例えば、ネットワーク・デプロイメント・セルにサーバーまたはサーバー・クラスターを追加できます。

#### サービス統合バスのトポロジーの表示:

管理コンソールのバス・トポロジー・ビューは、選択したバスのバス・メンバーおよびメッセージング・エンジンを表示します。各メッセージング・エンジンに関して、ランタイム状況が表示されます。

このタスクを使用して、バス・メンバーおよびサービス統合バス内のメッセージング・エンジンをツリー表示します。このローカル・トポロジー・ビューを使用して、バスのメンバーとしてサーバーおよびサーバー・クラスターを追加することもできます。

サービス統合バスのローカル・トポロジー・ビューを表示するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サービス統合 (Service integration)」 → 「バス (Buses)」をクリックします。コンテンツ・ペインにバスのリストが表示されません。
2. コンテンツ・ペインで、バスの名前をクリックします。例えば `SCA.SYSTEM.localhostCell01.Bus` をクリックします。
3. 「ローカル・トポロジー (Local Topology)」タブをクリックします。

バスのトポロジーが展開可能なツリー形式で表示されます。

ツリーのノードを展開すると、バス・メンバーとそのメッセージング・エンジンが表示されます。

バスのメンバーとしてサーバーおよびサーバー・クラスターを追加することもできます。

#### バス・プロパティーの変更:

サービス統合バスは、各プロパティーをデフォルト値に設定したままで作成できます。必要に応じてバスの使用方法を変更する場合、その変更に応じてプロパティーの値も後で変更できます。

サービス統合バスの用途を変更する必要がある場合、どのプロパティーを変更するかを判断するのにこのタスクが役立ちます。このタスクでは、バス・プロパティーを変更する際の考慮事項についての概要を解説し、基盤となる **WebSphere Application Server** のサービス統合テクノロジー用に提供されている各タスクの説明にリンクしています。

#### • SCA.SYSTEM バスのメッセージング・セキュリティの構成

メッセージング・セキュリティを有効にすると、バスおよびバスに存在するすべての宛先へのアクセスには許可が必要になります。つまり、バスに接続しようとするすべてのユーザーは、バス・リソースを使用するための直接の許可、またはユーザー・グループの一員としての許可が必要です。

**SCA.SYSTEM** バスを作成すると、デフォルトでバスのメッセージング・セキュリティが有効になります。許可されたメッセージング・エンジンのみが、保護されたバスへの接続を作成できます。

**SCA.SYSTEM** バスは、**SCA\_Auth\_Alias** という認証別名、デフォルト権限許可の初期セットで構成されます。このデフォルト権限許可により、**SCA** は保護されたバスに接続でき、バス上のすべてのローカル宛先への全アクセス権限が与えられます。デフォルト権限許可を変更して、バスへのアクセスを特定のユーザーに限定したり、異なる認証別名を作成して使用したりできます。

デフォルトで、同じ **SCA\_Auth\_Alias** がエンジン間認証別名として使用され、バス上のメッセージング・エンジン間の通信が許可されます。異なるエンジン間認証別名を作成して使用することもできます。

さらにセキュリティを構成する場合は、セキュア・トランスポート接続 (SSL または **HTTPS**) を使用して、アプリケーション・クライアントとメッセージング・エンジン間の伝送、およびメッセージング・エンジン間の伝送におけるメッセージの機密性と保全性を確保できます。**WebSphere Process Server** が備えるト



ランスポート・チェーン InboundSecureMessaging (JFAP over SSL over TCP/IP) を指定できます。または、別のランスポート・チェーンを作成して指定することもできます。

- 削除されたメッセージ・ポイント上のメッセージの保存または破棄の設定

SCA モジュール用のアプリケーションをアンインストールすると、SCA ランタイムで使用されるそのモジュールのキュー・ポイントが削除されます。デフォルトでは、「破棄」メッセージ・プロパティがクリアされ、これらのキュー・ポイント上に存在するすべてのメッセージはシステムの例外宛先に保存されるため、後でメッセージを処理できます。「破棄」メッセージ・プロパティを選択すると、削除されたキュー・ポイント上のすべてのメッセージを破棄できます。

- 構成情報のすべての更新をメッセージング・エンジンに動的に渡すかどうかの設定

デフォルトでは、SCA.SYSTEM バスの Configuration reload enabled プロパティが選択されます。つまり、構成情報へのすべての更新が動的にサーバーに渡されるため、メッセージング・エンジンが開始済みかどうかにかかわらず、更新がメッセージング・エンジンで有効になります。(メッセージング・エンジンは開始されると、実行しているサーバー内の情報を使用します。)

- メッセージング・エンジンのメッセージ・ポイントに保存できるメッセージの最大総数の設定

デフォルトでは、SCA.SYSTEM バスの High message threshold プロパティが 50000 メッセージに設定されます。これにより、このプロパティの初期値がバス内の各メッセージング・エンジンで設定されます。キュー・ポイント上の現在のメッセージの深さがメッセージ数の上限しきい値に等しい場合、メッセージング・エンジンは、キューに入れられたメッセージが消費されるまでは新たなメッセージを受け入れません。メッセージのスループットの問題を解決する 1 つの方法として、(バス上またはメッセージング・エンジン上の) High message threshold プロパティを変更できます。

## メッセージング・エンジン用のデータ・ストアのセットアップ

データベース管理者および WebSphere Process Server 管理者は、SCA ランタイムの各メッセージング・エンジン用のデータ・ストアをセットアップする必要があります。

WebSphere Process Server にメディエーション・モジュールをデプロイするためには、事前にサーバーまたはサーバー・クラスターを、必要な SCA ランタイムの宛先をホストするように構成しておく必要があります。この拡張構成では、サーバーまたはクラスターを SCA.SYSTEM バスのメンバーとして追加し、そのバス上でサーバーまたはクラスター用のメッセージング・エンジンを構成します。サーバーまたはクラスターの SCA 構成の JMS エlementを実行すると、SCA.APPLICATION バスの 2 番目のメッセージング・エンジンが作成されます。このメッセージング・エンジンごとに、別個のデータベースまたはスキーマを作成する必要があります。

各メッセージング・エンジンは、それぞれ固有のデータ・ストアを持ち、データベースに永続的データを保管します。データ・ストア内のすべてのテーブルは、同じデータベース・スキーマ内に収容されます。各データ・ストアを別個のデータベー

スに作成できます。あるいは、同じデータベースに複数のデータ・ストアを作成して、各データ・ストアが異なるスキーマを使用することも可能です。

メッセージング・エンジンは、JDBC データ・ソースのインスタンスを使用して、そのメッセージング・エンジン用のデータ・ストアを収容するデータベースと対話します。

サーバー・クラスターを SCA ランタイムの宛先をホストするように構成するためには、データ・ストアと、メッセージング・エンジンがそのデータ・ストアとの対話に使用する JDBC データ・ソースの両方を事前に構成しておく必要があります。

サーバーを SCA ランタイムの宛先をホストするように構成すると、メッセージング・エンジンがデフォルトで Cloudscape データ・ソースを使用するように構成されます。このデフォルトの構成では、管理者による追加構成を必要とせずに、メッセージング・エンジンを稼働させることができます。デフォルトのデータ・ソース構成を使用しない場合は、構成パラメーターを変更できます。例えば、データ・ソースを変更するか、またはデータ・ストアを別の JDBC プロバイダーを使用するように構成できます。

**注:** z/OS プラットフォームでのセットアップ手順については、レッドブック「Architecting High Availability Using WebSphere V6 on z/OS」を参照してください。

メッセージング・エンジン用のデータ・ストアをセットアップするには、以下のステップを実行します。以下のステップは、基盤となる WebSphere Application Server の場合と同じです。詳しくは、WebSphere Application Server のトピックを参照してください。

1. 以下のステップを実行するように、データベース管理者に依頼します。
  - a. データベースを作成します。データ・ストアに使用するリレーショナル・データベース管理システム (RDBMS) を選択します。サーバーでデフォルトの Cloudscape プロバイダーを使用させる場合を除いて、サーバーを SCA ランタイムの宛先をホストするように構成する前にデータベースを作成します。データベースの作成方法については、選択した RDBMS の資料を参照してください。

SCA メッセージング・エンジンのデフォルトのデータベース名は SCADB です。

- b. ユーザーおよびスキーマを作成します。デフォルトのスキーマ名は IBMWSSIB です。

使用するデータベース・タイプが 1 つのデータベース・インスタンスで複数のスキーマをサポートしない場合は、メッセージング・エンジンごとに別個のデータベース・インスタンスを構成する必要があります。

ユーザーおよびスキーマの作成方法については、WebSphere Application Server の資料の『データベース内でのユーザーおよびスキーマの作成 (Creating users and schemas in the database)』を参照してください。

- c. データ・ストア・テーブルを作成します。データベース管理者にテーブルの作成を依頼するか、または WebSphere が自動的にテーブルを作成できるよう

にするかを決定します。テーブルを自動的に作成する場合は、WebSphere Process Server にテーブルと索引を作成するための十分な権限があることを確認します。後でメッセージング・エンジンを構成するときに、テーブルを自動的に作成する必要があることを指示します。

メッセージング・エンジン用のデータ・ストア・テーブルの作成方法について詳しくは、WebSphere Application Server の資料の『テーブルの作成 (Creating the tables)』を参照してください。

- d. データ・ストアにアクセスするための十分な権限を持つデータベース・ユーザー ID を作成します。メッセージング・エンジンは、データベース・ユーザー ID を使用して、データ・ストア・テーブルへのアクセスを許可します。ユーザー ID には、データ・ストア用のテーブル・スペース、テーブル、索引、ストアード・プロシージャを作成できる権限が必要です。

選択したデータベースに必要な特権について詳しくは、WebSphere Application Server の資料の『データベース特権 (Database privileges)』を参照してください。

2. 以下のステップを実行してください。

- a. データベース管理者からデータベース・ユーザー ID を入手します。
- b. JDBC データ・ソースを構成します。メッセージング・エンジンがデータベースとの対話に使用する JDBC データ・ソースには、以下の命名規則が使用されます。

- SCA.SYSTEM バス: *nodeservercluster-SCA.SYSTEM\_bus\_name*
- SCA.APPLICATION バス: *nodeservercluster-SCA.APPLICATION\_bus\_name*

サーバー・クラスタのメッセージング・エンジンの場合、データ・ストア構成はクラスタの用途によって以下のように異なります。

- ワークロードの共有のための構成の場合は、すべてのメッセージング・エンジンにそれぞれ固有のデータ・ストアが必要です。
- フェイルオーバー用の構成の場合は、フェイルオーバー・クラスタ内のすべてのメッセージング・エンジン用にデータ・ストアを 1 つ用意する必要があります。メッセージング・エンジンがフェイルオーバー・クラスタで稼働する場合、そのデータ・ストアは、データ・ストアを所有するメッセージング・エンジンを稼働する可能性のあるすべてのサーバーからアクセスできなければなりません。

メッセージング・エンジンをクラスタのメンバーとして作成する場合は、データ・ストアと、メッセージング・エンジンがそのデータ・ストアとの対話に使用する JDBC データ・ソースの両方を明示的に構成する必要があります。

- 各メッセージング・エンジンがそれぞれ固有の JDBC データ・ソースを持つ場合、データ・ソースをノード・スコープで作成できます。
- いくつかのメッセージング・エンジンが 1 つの JDBC データ・ソースを共有する場合は、データ・ソースをノード・スコープではなく、クラスタ・スコープで作成する必要があります。

また、クラスタ・メンバーが稼働するすべてのノードに、JDBC ドライバー jar を必ずインストールしておく必要があります。

データ・ソースの構成方法について詳しくは、WebSphere Application Server の資料の『JDBCデータ・ソースの構成 (Configuring a JDBC data source)』を参照してください。

## SCA ランタイム用サーバーまたはクラスタの構成

サーバーまたはサーバー・クラスタ上にメディエーション・モジュールをデプロイするには、事前にそのサーバーまたはクラスタに必要な SCA ランタイム用に構成しておく必要があります。この拡張構成で、SCA ランタイムで使用されるキュー宛先をローカルにホストするか、リモート・サーバーまたはクラスタ上でホストするかを定義します。

スタンドアロン・サーバー・プロファイルを作成している場合は、そのサーバーは既に SCA ランタイム用に構成されているため、そのサーバーに対してこの作業は行わないでください。

Network Deployment で SCA モジュールをサポートするためのサーバーまたはサーバー・クラスタを作成している場合は、この作業を実行して、必要な SCA ランタイム用にそのサーバーまたはクラスタを構成する必要があります。

この作業を開始する前に、SCA ランタイムに必要な以下のような 1 つ以上のサーバーまたはサーバー・クラスタが既に作成済みであることが必要です。

- メディエーション・モジュールをデプロイするサーバーまたはクラスタ。
- SCA ランタイムで使用するキュー宛先を作成するサーバーまたはクラスタ。

両方に同じサーバーまたはクラスタを使用できます。

サーバーまたはサーバー・クラスタの作成に関する詳細は、以下のタスクの記述を参照してください。

- サーバーの作成
- サーバー・クラスタの作成

WebSphere Process Server にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、メディエーション・モジュールで処理されるメッセージを保持するために SCA ランタイムによって使用されます。

キュー宛先を作成する場所を指定するには、サーバーまたはクラスタ構成用の以下の選択肢のいずれかを実行します。

- サーバーまたはクラスタの拡張構成を設定して、宛先をホストします。この構成では、メディエーション・モジュール (ローカルにデプロイされている、あるいは別のサーバーまたはクラスタにデプロイされている) のキュー宛先を、このサーバーまたはクラスタでホストできるように定義します。また、この構成では、サーバーまたはクラスタを SCA.SYSTEM バスのメンバーとして追加し、サーバーまたはクラスタのメッセージング・エンジンを構成します。

この構成タスクを実行する場合は、217 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスタの構成』を参照してください。

- リモート宛先を使用するように、サーバーまたはクラスターの拡張構成を設定します。この構成では、サーバーまたはクラスターがリモート宛先ロケーションを宛先として使用するよう定義します。

このサーバーまたはクラスターは SCA.SYSTEM バスのメンバーである必要はありません。

このサーバーまたはクラスターにメディエーション・モジュールをデプロイすると、キュー宛先は SCA.SYSTEM バスのメンバーである別のサーバーまたはクラスター上に構成されます。

この構成タスクを実行する場合は、221 ページの『メディエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスターの構成』を参照してください。

サーバーまたはクラスターは、メディエーション・モジュールをデプロイできるように構成されます。

これで、サーバーまたはクラスター上にメディエーションをデプロイしたり、サーバーまたはクラスターを管理するその他のアクションを実行できるようになります。

#### **メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成:**

メディエーション・モジュールで、SCA ランタイムにより使用されるキュー宛先をホストするように、クラスターまたはサーバーを構成できます。サーバーまたはクラスターは、管理セル内の任意のサーバーまたはクラスターにデプロイされたメディエーション・モジュールのキュー宛先をホストできます。

この作業を開始する前に、以下の操作が完了していることが必要です。

- クラスター内にメッセージング・エンジン用のデータ・ストアをセットアップします (213 ページの『メッセージング・エンジン用のデータ・ストアのセットアップ』を参照)。
- 以下の説明のいずれかに従って、サーバーまたはクラスターを作成します。
  - サーバーの作成
  - サーバー・クラスターの作成

WebSphere Process Server にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、メディエーション・モジュールで処理されるメッセージを保持するために SCA ランタイムによって使用されます。

メディエーション・モジュールがデプロイされているサーバーまたはサーバー・クラスター上、あるいは別のサーバーまたはクラスター上にキュー宛先が作成されるように、サーバーまたはクラスターを構成できます。

**注:** サーバーまたはクラスターを使用してメディエーション・モジュールをデプロイしたりメディエーション・モジュールのキュー宛先をホストしたりするには、事前にこのタスクを完了する必要があります。または、221 ページの『メ

ディエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスタの構成』で説明しているように、リモート宛先ロケーションを宛先として使用するようサーバーまたはクラスタを構成できます。

メディエーション・モジュールのキュー宛先をホストするサーバーまたはクラスタはキュー処理用に最適化でき、メディエーション・モジュールがデプロイされた別の複数のサーバーおよびクラスタにこれらのキューを提供できます。

この構成タスクでは、ローカルにデプロイされたメディエーション・モジュールまたは別のサーバーまたはクラスタにデプロイされたメディエーション・モジュールのキュー宛先を、サーバーまたはクラスタがホストできるように定義します。また、この構成では、サーバーまたはクラスタを `SCA.SYSTEM` バスのメンバーとして追加し、サーバーまたはクラスタのメッセージング・エンジンを構成します。

- サーバーを構成すると、WebSphere はデフォルトのプロパティでメッセージング・エンジンを作成できます。デフォルトのメッセージング・エンジンの構成では、デフォルトの JDBC データ・ソースを使用し、そのデータ・ソースのデータ・ストアとして Cloudscape JDBC プロバイダーを使用します。この場合、追加の構成を行うことなくメッセージング・エンジンを実行できます。

デフォルトのデータ・ソース構成を使用しない場合は、異なるデータ・ソースを使用するか、異なる JDBC プロバイダーを使用するようにデータ・ストアを構成できます。

- クラスタを構成する場合、データ・ストア、およびメッセージング・エンジンがデータ・ストアとの対話に使用する JDBC データ・ソースを明示的に作成する必要があります。また、接続プールの「ページ・ポリシー (Purge policy)」を「EntirePool」に設定する必要があります。

メディエーション・モジュールのキュー宛先をホストするようにサーバーまたはクラスタを構成するには、管理コンソールを使用して以下のステップを実行します。

1. 「Network Deployment 環境の構成」のガイド付きアクティビティの「拡張構成」ページを表示します。
  - a. ナビゲーション・ペインで、「ガイド付きアクティビティ (Guided Activities)」 → 「Network Deployment 環境の構成」をクリックします。
  - b. コンテンツ・ペインで、「アプリケーション・サーバーの構成」 → 「クリックして実行 (Click to perform)」をクリックします。
2. 構成するクラスタおよびサーバーを選択します。
  - 「構成するクラスタの選択」の下にあるリストからクラスタ名を選択します。
  - 「構成するサーバーの選択」の下にあるリストからサーバー名を選択します。

いくつかのクラスタおよびサーバーを同時に構成できるように、このステップを数回繰り返すことができます。

3. 「追加」をクリックします。これにより、表示されるリストにクラスタまたはサーバーが追加されます。「削除」ボタンを使用して、リストから項目を削除できます。

4. 「SCA 宛先のセットアップ」を選択します。
5. 「次へ」をクリックします。これにより、「共通データベース・テンプレート設定の選択」ページが表示されます。
6. メッセージング・エンジンのデータ・ストアとして使用するデータベースの詳細を指定します。

データベース・プロバイダー・タイプ、データベースにアクセスする際に使用するユーザー名とパスワードを、前提条件となる作業で行ったデータ・ストア用の構成に合わせて指定します。

これらの設定は後で変更できますが、ここで最も一般的なデータベース・タイプを選択しておき、他のクラスターやサーバーの構成時に同じ情報を再度入力する手間を省くこともできます。セットアップに多数の異なるデータベース・タイプを使用する場合は、このステップを省略してください。

7. 「宛先ロケーションの構成」オプションが選択されていること、および「SCA アプリケーションをホストしない」オプションがクリアされていることを確認します。
8. 「データベース・セットアップ」の下で、メッセージング・エンジンのデータ・ストアとして使用するデータベースの詳細を指定します。サーバー・クラスターの場合は、このステップを実行する必要があります。サーバーの場合は、そのサーバーがデフォルトの Cloudscape データベースを使用しない場合のみ、このステップを実行する必要があります。

データベース・プロバイダー・タイプ、データベースにアクセスする際に使用するユーザー名とパスワードを、前提条件となる作業で行ったデータ・ストア用の構成に合わせて指定します。

これらの設定は後で変更できますが、ここで最も一般的なデータベース・タイプを選択しておき、他のクラスターやサーバーの構成時に同じ情報を再度入力する手間を省くこともできます。セットアップに多数の異なるデータベース・タイプを使用する場合は、このステップを省略してください。

9. 「システム・バス」の下で、SCA.SYSTEM バス上のメッセージング・エンジン用のデータ・ストアの詳細を指定します。
  - a. **オプション:** 既存のデータ・ソースを使用する場合は、「既存のデータ・ソースの使用」チェック・ボックスを選択し、ドロップダウン・リストから名前を選択します。
  - b. 「スキーマ名」フィールドに、SCA.SYSTEM バスのデータ・ソース用のテーブルを入れるために使用するデータベースのスキーマ名を入力します。デフォルトのスキーマ名は **IBMWSSIB** です。
10. 「アプリケーション・バス」の下で、SCA.APPLICATION バス上のメッセージング・エンジン用のデータ・ストアの詳細を指定します。
  - a. **オプション:** 既存のデータ・ソースを使用する場合は、「既存のデータ・ソースの使用」チェック・ボックスを選択し、ドロップダウン・リストから名前を選択します。
  - b. 「スキーマ名」フィールドに、SCA.SYSTEM バスのデータ・ソース用のテーブルを入れるために使用するデータベースのスキーマ名を入力します。デフォルトのスキーマ名は **IBMWSSIB** です。

11. メッセージング・エンジンにデータ・ソース用のデータベース・テーブルを自動作成させる場合は、「**テーブルの作成**」チェック・ボックスを選択します。メッセージング・エンジンをホストするサーバーの始動時に、テーブルが作成されます。それ以外の場合は、データベース管理者がデータベース・テーブルを作成する必要があります。
12. 「**次へ**」をクリックします。
13. 「エミッター・ファクトリー・プロファイル JNDI 名」フィールドの値を検証します。このフィールドでは、イベントをイベント・インフラストラクチャーにサブミットするために使用されるイベント・エミッター・プロファイル・ファクトリーの JNDI 名を指定します。

CEI サーバーへのイベント出力のためのデフォルト構成を変更する場合は、ドロップダウン・リストから該当するエミッター・ファクトリー・プロファイルの JNDI 名を選択するか、または表示されるテキスト・ボックスに別の値を入力します。クラスター環境では、選択した値がクラスター内のすべてのサーバーに使用されます。

14. 「**次へ**」をクリックします。
15. 「**要約**」ページで詳細を確認します。クラスターまたはサーバーを構成するには、「**終了**」をクリックします。

選択した詳細情報に誤りがある場合は、「**戻る**」をクリックして前のページに戻り、詳細の内容を変更できます。

16. 「**終了**」をクリックします。

クラスターまたはサーバーが SCA ランタイムおよび CEI イベント・エミッター用に構成されます。

クラスターまたはサーバーがデプロイメント・マネージャー・セル用に構成された最初のクラスターまたはサーバーである場合は、このアクティビティーによって、セルの SCA.SYSTEM バスおよび SCA.APPLICATION バスが作成され、これらのバス上にメッセージング・エンジンが構成されます。

正常に終了したら、以下のようなメッセージが表示されます。

```
以下のターゲットの SCA のインストールが正常に完了しました:  
cells:localhostCell101:clusters:ESBcluster1  
以下のターゲットの CEI イベント・エミッター構成が正常に完了しました:  
cells:localhostCell101:clusters:ESBcluster1
```

17. 変更内容をマスター構成に保管します。

クラスターの SCA 構成を表示するために、「Service Component Architecture」パネル（「**サーバー**」 → 「**クラスター**」 → *name* → 「**[追加プロパティ] Service Component Architecture**」）を使用できます。このパネルから、SCA.SYSTEM バスおよび SCA.APPLICATION バスに追加されたメッセージング・エンジンにアクセスできます。

リモート宛先ロケーションを使用するように構成されている別のサーバーまたはクラスターにメディアエーション・モジュールをデプロイできます。その別のサーバーまたはクラスターの「リモート宛先ロケーション (Remote Destination Location)」ブ



ロパティにこのサーバーまたはクラスターが設定されている場合は、デプロイするメディエーション・モジュールのキュー宛先をこのサーバーまたはクラスターでも構成できます。

パフォーマンスに応じてメッセージング・エンジンの JDBC データ・ソースを調整できます。

データ・ストアに指定したユーザー名とパスワードは、名前がデータ・ストア構成 (「バス」 → *SCA.SYSTEM\_bus\_name* → 「メッセージング・エンジン」 → *me\_name* → 「データ・ストア」) で指定されている認証別名を作成する際に使用されます。このパネルを使用して、データ・ソース JNDI 名、スキーマ名、および認証別名を変更できます。

#### メディエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスターの構成:

メディエーション・モジュールのキュー宛先を別のクラスターまたはサーバーに作成するように、クラスターまたはサーバーを構成できます。

このタスクを開始する前に、以下のサーバーまたはサーバー・クラスターを 1 つ以上作成しておく必要があります。

- メディエーション・モジュールをデプロイするサーバーまたはクラスター。これは、このタスクで構成するサーバーまたはクラスターです。
- SCA ランタイムで使用するキュー宛先を作成するサーバーまたはクラスター。

**注:** 217 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』で説明しているように、このサーバーまたはクラスターはメディエーション・モジュールのキュー宛先をホストするように構成しておく必要があります。

両方に同じサーバーまたはクラスターを使用できます。

WebSphere Process Server にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、メディエーション・モジュールで処理されるメッセージを保持するために SCA ランタイムによって使用されます。

メディエーション・モジュールがデプロイされているサーバーまたはサーバー・クラスター上、あるいは別のサーバーまたはクラスター上にキュー宛先が作成されるように、サーバーまたはクラスターを構成できます。

**注:** このサーバーまたはクラスターにデプロイされるメディエーション・モジュールのキュー宛先を、別のサーバーまたはクラスターでホストするように構成するには、事前にこのタスクを完了しておく必要があります。または、217 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』で説明しているように、キュー宛先をホストするようサーバーまたはクラスターを構成できます。

メディエーション・モジュールのキュー宛先をホストしないサーバーまたはクラスターは、他のコンポーネントを実行するように最適化でき、SCA.SYSTEM バスのメンバーである必要はありません。

この構成タスクでは、別のサーバーまたはクラスター上に作成されたキュー宛先を使用するようにサーバーまたはクラスターを定義します。この構成では、サーバーまたはクラスターを SCA.SYSTEM バスのメンバーとして追加しません。

別のサーバーまたはクラスター上に作成されたキュー宛先を使用するようにサーバーまたはクラスターを構成するには、管理コンソールを使用して以下のステップを実行します。

1. 「Network Deployment 環境の構成」のガイド付きアクティビティの「拡張構成」ページを表示します。
  - a. ナビゲーション・ペインで、「ガイド付きアクティビティ (Guided Activities)」 → 「Network Deployment 環境の構成」をクリックします。
  - b. コンテンツ・ペインで、「アプリケーション・サーバーの構成」 → 「クリックして実行 (Click to perform)」をクリックします。

2. 構成するクラスターおよびサーバーを選択します。
  - 「構成するクラスターの選択」の下にあるリストからクラスター名を選択します。
  - 「構成するサーバーの選択」の下にあるリストからサーバー名を選択します。

いくつかのクラスターおよびサーバーを同時に構成できるように、このステップを数回繰り返すことができます。

3. 「追加」をクリックします。これにより、表示されるリストにクラスターまたはサーバーが追加されます。「削除」ボタンを使用して、リストから項目を削除できます。
4. 「リモート SCA 宛先の使用」を選択します。このオプションはデフォルトで選択されています。
5. 「次へ」をクリックします。これにより、「共通データベース・テンプレート設定の選択」ページが表示されます。
6. メッセージング・エンジンのデータ・ストアとして使用するデータベースの詳細を指定します。

データベース・プロバイダー・タイプ、データベースにアクセスする際に使用するユーザー名とパスワードを、前提条件となる作業で行ったデータ・ストア用の構成に合わせて指定します。

これらの設定は後で変更できますが、ここで最も一般的なデータベース・タイプを選択しておき、他のクラスターやサーバーの構成時に同じ情報を再度入力する手間を省くこともできます。セットアップに多数の異なるデータベース・タイプを使用する場合は、このステップを省略してください。

7. 「次へ」をクリックします。これにより、「Service Component Architecture のリモート宛先ロケーションの使用」ページが表示されます。
8. SCA の宛先をホストすることになるクラスターまたはサーバーを選択します。
  - a. 「リモート宛先ロケーションの使用」が選択されていること、およびその他のオプションがクリアされていることを確認します。
  - b. 関連するメニューを使用して、SCA 宛先をホストするリモート・クラスターまたはサーバーの名前を選択します。

9. 「次へ」をクリックします。
10. 「エミッター・ファクトリー・プロファイル JNDI 名」フィールドの値を検証します。このフィールドでは、イベントをイベント・インフラストラクチャーにサブミットするために使用されるイベント・エミッター・プロファイル・ファクトリーの JNDI 名を指定します。

CEI サーバーへのイベント出力のためのデフォルト構成を変更する場合は、ドロップダウン・リストから該当するエミッター・ファクトリー・プロファイルの JNDI 名を選択するか、または表示されるテキスト・ボックスに別の値を入力します。クラスター環境では、選択した値がクラスター内のすべてのサーバーに使用されます。

11. 「次へ」をクリックします。
12. 「要約」ページで詳細を確認します。クラスターまたはサーバーを構成するには、「終了」をクリックします。

選択した詳細情報に誤りがある場合は、「戻る」をクリックして前のページに戻り、詳細の内容を変更できます。

13. 「終了」をクリックします。

クラスターまたはサーバーが SCA ランタイムおよび CEI イベント・エミッター用に構成されます。

正常に終了したら、以下のようなメッセージが表示されます。

以下のターゲットの SCA のインストールが正常に完了しました：  
cells:localhostCell101:clusters:ESBcluster2

以下のターゲットの CEI イベント・エミッター構成が正常に完了しました：  
cells:localhostCell101:clusters:ESBcluster2

14. 変更内容をマスター構成に保管します。

クラスターの SCA 構成を表示するために、「Service Component Architecture」パネル（「サーバー」 → 「クラスター」 → *name* → 「[追加プロパティ] Service Component Architecture」）を使用できます。

このサーバーまたはクラスターにメディエーション・モジュールをデプロイすると、メディエーション・モジュールのキュー宛先は「リモート宛先ロケーションの使用」プロパティで指定されたサーバーまたはクラスター上に構成されます。

サーバーまたはクラスターの構成を変更して、サービス統合バスのメンバーとして追加したり、サーバーまたはクラスターを管理するその他のアクションを実行したりできます。

## ESB サーバーの管理

サーバー構成によって、アプリケーションとそのコンポーネントを実行するためのサービスを、サーバーが提供する方法を制御する設定が定義されます。管理者は、既存のサーバーおよびバス環境で、サーバーを作成および構成できます。

- サーバーを新規作成するには、ネットワーク・デプロイメント・セルに管理対象ノードが配置されている必要があります。

WebSphere Process Server の完全インストール (デフォルト) を選択すると、デフォルトのバス環境が作成されます。このバス環境は、単一のサーバーが、SCA モジュールのデプロイに使用される 2 つのサービス統合バスに割り当てられる構成になっています。このサーバーを使用して、サーブレット、Enterprise Java Bean、JMS 宛先など、J2EE のコンポーネントおよびリソースをデプロイすることもできます。

ご使用の SCA モジュールによっては、単一サーバーの構成で十分な場合もあります。しかし、サーバー容量の追加、ネットワーク・サービスを提供するトランスポート・チェーンの変更、または、ユーザー独自のカスタム・サービスの提供などを行う場合は、この構成を変更できます。

サーバーの管理について詳しくは、以下のサブトピックを参照してください。

- サーバーを作成します。

管理コンソールの「新規アプリケーション・サーバーの作成」ウィザード、または `createApplicationServer wsadmin` コマンドのいずれかを使用して、サーバーを作成できます。

- 使用可能なサーバーを管理します。

サーバーの作成後、ランタイム状態のモニター、構成の変更、サーバーの始動および停止、その他のさまざまな管理用タスクの実行などを通じて、サーバーを管理できます。

- トランスポート・チェーンを構成します。

サービス統合テクノロジー、WebSphere Secure Caching Proxy、および高可用性マネージャーによるコア・グループ・ブリッジ・サービスなどの機能に対し、ネットワーク・サービスを提供するように、トランスポート・チェーンを構成する必要があります。

- カスタム・サービスを作成します。

サーバーまたはノード・エージェントの始動時およびシャットダウン時にフック・ポイントが実行されるように定義するには、カスタム・サービス・クラスを作成してカスタム・サービス・インスタンスを構成します。アプリケーション・サーバーまたはノード・エージェントの始動時に、カスタム・サービスが開始され、初期化されます。

- サーバーのプロセスを定義します。

サーバーの操作性を向上させるには、アプリケーション・サーバー・プロセスを開始または初期化するためのコマンド行情報を定義します。この設定では、実行するプログラム、プログラムを実行する引数、および作業ディレクトリーなどのランタイム・プロパティを定義します。

- サーバーによる Java 仮想マシンの使用を管理します。

サーバー構成の一部として、システムによる Java 仮想マシン (JVM) の使用率を高める設定を定義できます。

#### サーバー・テンプレート:

サーバー・テンプレートは、新規アプリケーション・サーバーの初期構成設定を定義するために使用します。新規アプリケーション・サーバーを作成するときには、

サーバーのタイプに応じたサーバー・テンプレートを選択します。サーバーの作成時に異なるテンプレートを指定しない場合は、デフォルト・テンプレートが使用されます。

サーバー・クラスター内のサーバーは、すべて同じサーバー・テンプレートで作成されます。

異機種混合セルでは、管理対象ノードを分散プラットフォームおよび z/OS オペレーティング・システム・プラットフォームに混在させることができます。サーバーを作成する管理対象ノードを選択すると、そのノードのオペレーティング・システム・プラットフォームに使用できるテンプレートを管理コンソールで選択できます。

括弧内に示すオペレーティング・システム・プラットフォームの管理対象ノードに以下のデフォルト・サーバー・テンプレートが提供されます。

**defaultWPSServer (分散プラットフォーム) または defaultWPSServerZOS (z/OS プラットフォーム)**

拡張統合機能のサービス・アプリケーション (エンドポイント・オーケストレーション用のビジネス・プロセスと状態マシン、動的意思決定のためのビジネス・ルールなど) をサーバーにサポートさせる場合は、このテンプレートを使用します。

サーバーは以下のものもサポートできます。

- Enterprise Service Bus にあるエンタープライズ・サービス・バス上のメディアエーション・モジュールに対するサービス・アプリケーション。
- WebSphere Application Server にあるエンタープライズ・アプリケーション。

**defaultESBServer (分散プラットフォーム) または defaultESBServerZOS (z/OS プラットフォーム)**

エンタープライズ・サービス・バス上のメディアエーション・モジュールに対するサービス・アプリケーションをサーバーにサポートさせる場合は、このテンプレートを使用します。

サーバーは、WebSphere Application Server にあるエンタープライズ・アプリケーションもサポートできます。

**default (分散プラットフォーム) または defaultZOS (z/OS プラットフォーム)**

WebSphere Application Server にあるエンタープライズ・アプリケーションをサーバーにサポートさせる場合は、このテンプレートを使用します。

これらのデフォルト・サーバー・テンプレートの代わりに、既存の別のアプリケーション・サーバーに基づく独自のテンプレートを作成できます。これにより、既存のサーバーと同じ構成プロパティを持つテンプレートを作成できます。

**サーバーの作成:**

管理コンソールの「新規アプリケーション・サーバーの作成」ウィザード、または createApplicationServer wsadmin コマンドのいずれかを使用して、サーバーを作成できます。

サーバーを作成する前に、カスタム・プロファイルを作成し、そのノードを Network Deployment セル内に統合しておく必要があります。

SCA モジュールをデプロイする前に、SCA モジュールを実行するサーバーを作成しておく必要があります。

複数の SCA モジュールを同じサーバー上にデプロイ可能なため、新規サーバーを作成する場合に実行する必要があるのはこのタスクのみです。例えば、1 台のサーバーでは処理しきれない数のクライアント接続を処理したり、より高レベルなメッセージ・スルーput を提供したり、SCA モジュールの管理を分割して異なるサーバー上で実行する場合などに、サーバーを作成します。

サーバーを作成するには、管理コンソールを使用して以下のステップを実行します。

1. サーバーのリストを表示します。

ナビゲーション・ペインで、「サーバー」 → 「アプリケーション・サーバー」を展開します。

2. 「新規」をクリックします。
3. 「新規アプリケーション・サーバーの作成」ページで、指示に従ってサーバーを定義します。
  - a. サーバーを実行するノードを選択します。

このノードは、WebSphere Process Server 管理対象ノードでなければなりません。

- b. サーバーの固有の名前を入力します。

この名前はノード内で固有でなければなりません。

- c. 「次へ」をクリックします。
- d. サーバーを作成するために使用する defaultESBServer テンプレート (または既存のサーバー) を選択します。選択したテンプレートまたはサーバーが、新しいサーバーの初期プロパティを定義します。
  - テンプレートを選択する代わりに、既存の ESB サーバーをテンプレートとして選択できます。新規サーバーは、選択したサーバーのすべてのプロパティを継承します。
  - z/OS ノードを選択した場合、提供されるテンプレート名はそのプラットフォームを示します。
  - 使用可能なサーバー・テンプレートについて詳しくは、224 ページの『サーバー・テンプレート』を参照してください。
- e. 「次へ」をクリックします。
- f. 「固有の HTTP ポートの生成 (Generate Unique Http Ports)」オプションでは、新しいサーバーに HTTP トランスポート用の固有のポートを持たせるかどうかを選択します。

デフォルトでは、「固有の HTTP ポートの生成 (Generate Unique Http Ports)」オプションが選択されています。このオプションを選択すると、このサーバーと共に使用する予定の仮想ホストの別名リストを、上記の新しいポ

ート値を含むように更新しなければならない可能性があります。このオプションを選択解除する場合は、同じ物理マシン上の他のサーバーとデフォルトのポート値が競合しないようにしてください。

固有の HTTP ポートの使用については、『HTTP トランスポートの構成 (Configuring HTTP transports)』を参照してください。

- g. 「次へ」をクリックします。
- h. 既存のサーバーをテンプレートとして使用して新規サーバーを作成する場合、アプリケーションを既存のサーバーから新規サーバーにマップする選択は行わないでください。デフォルトでは、このオプションは無効になっています。
- i. 「次へ」をクリックします。
- j. 「新規サーバーの確認」ページで、要約をチェックします。サーバーの作成を完了するには、「終了」をクリックします。

設定を変更したい場合は、「前へ」をクリックしてサーバー設定を検討または変更してください。

- 4. **オプション:** 複数言語のエンコード・サポートを管理コンソールで使用するには、UTF-8 エンコードが使用可能になるようにアプリケーション・サーバーを構成してください。
- 5. 変更内容をマスター構成に保管します。

サーバーが作成され、「アプリケーション・サーバー」ページのサーバーのリストに表示されます。

これで、サーバーの開始や SCA モジュールのデプロイなどの、サーバーの管理を実行できるようになります。

**注:** サーバーは、いくつかのプロパティでデフォルト値を使用して作成されています。サーバーには、設定可能な多くのプロパティがあります。「新規アプリケーション・サーバーの作成」ページでサーバーを作成する際に値が指定されるのは、重要なプロパティの一部のみです。ご使用のサーバーのすべてのプロパティを表示し、さらにサーバーをカスタマイズするには、「アプリケーション・サーバー」ページでサーバーの名前をクリックします。サーバーの詳細設定のページが表示されるので、必要に応じてプロパティを参照し変更できます。

### 新しい WebSphere Process Server カスタム・プロファイルの作成:

カスタム・プロファイルは空のノードで、利用可能にするためにはネットワーク・デプロイメント・セルに統合する必要があります。スタンドアロン・サーバー・プロファイルとは異なり、カスタム・プロファイルのノードにはデフォルトのサーバーがありません。カスタム・ノードには、デフォルト・アプリケーションもありません。カスタム・プロファイルは、統合すると管理対象ノードに変更されます。統合後、カスタム・プロファイルは `nodeagent` プロセスには対応していますが、サーバー・プロセスには対応していません。実動またはその他の使用のために空のノードをカスタマイズするには、デプロイメント・マネージャーの管理コンソールを使用する必要があります。`nodeagent` を開始すると、デプロイメント・マネージャーからのコマンドに回答できるようになります。

この手順を実行する前に、以下を確認します。

- WebSphere Process Server を既にインストールしていることを確認します。
- カスタム・サーバー・プロファイルを作成することと、それを対話式に実行することを確認します。計画がこれらの手順と異なる場合は、プロファイル・ウィザードを使用したプロファイルの作成と拡張を確認して、プロファイル作成または拡張の手順に関するその他の文書を参照してください。
- 新規プロファイルを作成するために十分なディスクと一時スペースがあることを確認します。

プロファイル	必要なディスク・スペース	必要な一時スペース
デプロイメント・マネージャー・プロファイル	30 MB	40 MB
カスタム・プロファイル	10 MB	40 MB
アプリケーション・サーバー・プロファイル	200 MB	40 MB

- WebSphere Process Server のインストール時にグローバル・セキュリティーを使用可能にした場合は、プロファイルを作成する前にそれを使用不可にしてください。グローバル・セキュリティーの使用可能化と使用不可化については、『WebSphere Process Server セキュリティーの設定』を参照してください。

WebSphere Process Server のフルインストール手順を実行すると、server1 という名前のサーバーを持つ default という名前のスタンドアロン・サーバー・プロファイルが作成されます。ただし、製品のネットワーキングの利点を生かすため、プロファイル・ウィザードを使用して追加プロファイルを作成することもできます。

次の手順に従い、新規カスタム・プロファイルを作成します。

1. Linux または UNIX システムでは root として、Windows システムでは管理者グループのメンバーとしてログオンします。
2. WebSphere Process Server プロファイル・ウィザードを開始して、新規ランタイム環境を作成します。

各種プラットフォームでさまざまな方法を使用してプロファイル・ウィザードを開始できます。詳細については、『プロファイル・ウィザードの開始』を参照してください。

システムが InstallShield Multiplatform を開始してプロファイル・ウィザードを実行し、「ようこそ」パネルが表示されます。

3. 「次へ」をクリックします。以下のパネルのいずれかが表示されます。
  - WebSphere Application Server または WebSphere Application Server Network Deployment プロファイルが存在する場合は、「既存プロファイルの検出」パネルが表示されます。この手順では、既存のプロファイルを WebSphere Process Server プロファイルに拡張するのではなく、新規プロファイルを作成することを想定しています。新規プロファイルを作成するには、「**新しい WebSphere Process Server プロファイルの作成**」を選択して「次へ」をクリックします。「プロファイル・タイプの選択」パネルが表示されます。



- プロファイルが存在しない場合は、「プロファイル・タイプの選択」パネルが表示されます。
- 4. 「**WebSphere Process Server のカスタム**」を選択して、「次へ」をクリックします。「統合」パネルが表示されます。
- 5. デプロイメント・マネージャーにカスタム・ノードを統合するかどうかを選択します。
  - ノードを統合するよう選択した場合は、デプロイメント・マネージャーのホスト名または IP アドレス、および SOAP ポートを指定し、「次へ」をクリックして続行します。「プロファイル名」パネルが表示されます。

**重要:**

以下の項目すべてが該当する場合は、この時点でカスタム・ノードを統合できます。

- デプロイメント・マネージャーが稼働している。
- デプロイメント・マネージャーが、WebSphere Process Server デプロイメント・マネージャー・プロファイルである。
- デプロイメント・マネージャー・ノードでセキュリティが有効になっていない。
- デプロイメント・マネージャーがデフォルトの SOAP JMX コネクター・タイプを使用し、コネクターが有効になっている。

以下の項目のいずれかが該当する場合は、この時点ではカスタム・ノードを統合しないでください。

- デプロイメント・マネージャーが実行中でないか、または実行中かどうか不明である。
- デプロイメント・マネージャーが、まだ WebSphere Process Server プロファイルに拡張されていない WebSphere Application Server Network Deployment プロファイルである。
- デプロイメント・マネージャー・ノードでセキュリティが有効になっている。
- SOAP コネクターが無効になっている。
- デプロイメント・マネージャーが、デフォルトではないリモート・メソッド呼び出し (RMI) を優先 Java Management Extensions (JMX) コネクターとして使用するよう再構成されている。(デプロイメント・マネージャーの管理コンソールで、「システム管理 (System Administration)」>「デプロイメント・マネージャー」>「管理サービス (Administration services)」を選択して、優先コネクター・タイプを確認できます。)

デプロイメント・マネージャーが稼働していないか、その他の理由で使用不可である場合に、カスタム・ノードを統合すると、「プロファイル作成が失敗しました。」パネルが表示されます。「完了」をクリックしてこのパネルを閉じます。

- ノードを統合しない場合は、「**addNode コマンドを使用してこのノードを後に統合 (Federate this node later using the addNode command)**」の横にあるチェック・ボックスを選択し、「次へ」をクリックして続行します。「プロファイル名」パネルが表示されます。管理コンソールまたはこのコマンドで

ノードを統合する方法については、『カスタム・ノードのデプロイメント・マネージャーへの統合』を参照してください。

6. プロファイルに固有の名前を指定するか、デフォルトの名前を受け入れます。その他のプロファイルが存在する場合は、「このプロファイルをデフォルトにする (Make this profile the default)」の横にあるチェック・ボックスを選択して、このプロファイルをデフォルト・プロファイルにすることができます。「次へ」をクリックします。

作成する各プロファイルに固有の名前を付ける必要があります。複数のプロファイルがある場合は、最上位のレベルで、この名前によってそれらのプロファイルを区別できます。プロファイル命名時の考慮事項については、『プロファイル、ノード、ホスト、およびセルの命名に関する考慮事項』を参照してください。「プロファイル・ディレクトリー」パネルが表示されます。

7. プロファイルのデフォルト・ディレクトリー・ロケーションを受け入れるか、「参照」ボタンを使用して別のロケーションを指定し、「次へ」をクリックします。

このディレクトリーには、ランタイム環境を定義するファイル (コマンド、構成ファイル、ログ・ファイルなど) が入ります。デフォルトでは、このディレクトリーは以下のロケーションにあります。

-   **Linux および UNIX プラットフォームの場合:**

`install_root/profiles/profile_name`

-  **Windows プラットフォームの場合:**

`install_root¥profiles¥profile_name`

ここで、*profile\_name* は指定したプロファイル名です。このディレクトリー・ロケーションは、システム上で有効であればどのディレクトリー・ロケーションにでも変更できます。ディレクトリーが既に存在している場合は、空でなければなりません。そうでない場合は、プロファイル・ウィザードによってエラーが表示されます。

「ノード名およびホスト名」パネルが表示されます。

8. カスタム・プロファイルのノード名およびホスト名を指定するか、デフォルトを受け入れて、「次へ」をクリックします。

作成するノードごとに固有の名前を使用します。ノードおよびホストの命名時に考慮する必要がある予約済みの用語やその他の問題については、『プロファイル、ノード、ホスト、およびセルの命名に関する考慮事項』を参照してください。

ステップ 5 の「統合」パネルでノードを統合することを選択したかどうかに応じて、次のパネルのいずれかが表示されます。

- ノードを後で統合することを選択した場合は、「プロファイルの要約」パネルが表示されます。この場合は、ステップ 10 に進みます。
- ノードをすぐに統合することを選択した場合は、「ポート値割り当て」パネルが表示されます。この場合は、ステップ 9 に進みます。

9. プロファイル・ウィザードは、その他の WebSphere 製品が現在使用しているポートを検査します。ただし、WebSphere 以外のアプリケーションにポートを指定して使用している場合は、ポートが競合しないことを確認してください。「次へ」をクリックします。

カスタム・プロファイルを統合するとき、**addNode** コマンドを使用すると、競合しないポートが使用されます。つまり、プロファイルを作成する際にはデフォルトのポート割り当てを使用し、**addNode** コマンドでノードを統合する際に、競合しないポートを指定できます。ポート割り当ては、マシン上で固有でなければなりません。マシンが異なれば、サーバー・プロセスで同じポート割り当てを使用しても競合は起きません。「プロファイルの要約」パネルが表示されます。

10. プロファイルの特性を検討し、このプロファイルを作成する場合は「次へ」、プロファイルの特性を変更する場合は「戻る」をクリックします。

プロファイル・ウィザードでは、プロファイルが作成中であることを示す進行状況パネルが表示されます。エラーが検出されない場合は、プロセスの最後に「プロファイル作成が完了しました」パネルが表示されます。

**重要:** プロファイルの作成中にエラーが検出された場合は、「プロファイル作成が完了しました」パネルではなく、例えば次のような別のパネルが表示されることがあります。

- 「プロファイル作成が警告で完了しました。」パネル。プロファイル作成は完了されたが、警告が生成されたことを示します。
- 「プロファイル作成は不完全です (Profile Creation is Incomplete)」パネル。プロファイルは作成されたが、使用不可の状態であることを示します。
- 「プロファイル作成が失敗しました。」パネル。プロファイルが作成されなかったことを示します。

これらの各パネルが表示されたら、対応するエラーの解決に役立つ詳細が記載されたログ・ファイルが作成されます。

11. ファースト・ステップ・コンソールを起動するチェック・ボックスが選択されていることを確認し、「完了」をクリックしてプロファイル・ウィザードを閉じ、ファースト・ステップ・コンソールを起動します。

新しいカスタム・プロファイルが作成されました。プロファイル内のノードは、ノードを統合し (プロファイル作成時に統合しなかった場合)、デプロイメント・マネージャーを使用してノードをカスタマイズするまでは空です。

カスタム・ノードを作成したときに、ノードをネットワーク・デプロイメント・セルに統合していない場合は、統合します。

#### SCA ランタイム用サーバーまたはクラスターの構成:

サーバーまたはサーバー・クラスター上にメディエーション・モジュールをデプロイするには、事前にそのサーバーまたはクラスターを必要な SCA ランタイム用に構成しておく必要があります。この拡張構成で、SCA ランタイムで使用されるキュー宛先をローカルにホストするか、リモート・サーバーまたはクラスター上でホストするかを定義します。

スタンドアロン・サーバー・プロファイルを作成している場合は、そのサーバーは既に SCA ランタイム用に構成されているため、そのサーバーに対してこの作業は行わないでください。

Network Deployment で SCA モジュールをサポートするためのサーバーまたはサーバー・クラスターを作成している場合は、この作業を実行して、必要な SCA ランタイム用にそのサーバーまたはクラスターを構成する必要があります。

この作業を開始する前に、SCA ランタイムに必要な以下のような 1 つ以上のサーバーまたはサーバー・クラスターが既に作成済みであることが必要です。

- メディエーション・モジュールをデプロイするサーバーまたはクラスター。
- SCA ランタイムで使用するキュー宛先を作成するサーバーまたはクラスター。

両方に同じサーバーまたはクラスターを使用できます。

サーバーまたはサーバー・クラスターの作成に関する詳細は、以下のタスクの記述を参照してください。

- サーバーの作成
- サーバー・クラスターの作成

WebSphere Process Server にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、メディエーション・モジュールで処理されるメッセージを保持するために SCA ランタイムによって使用されます。

キュー宛先を作成する場所を指定するには、サーバーまたはクラスター構成用の以下の選択肢のいずれかを実行します。

- サーバーまたはクラスターの拡張構成を設定して、宛先をホストします。この構成では、メディエーション・モジュール (ローカルにデプロイされている、あるいは別のサーバーまたはクラスターにデプロイされている) のキュー宛先を、このサーバーまたはクラスターでホストできるように定義します。また、この構成では、サーバーまたはクラスターを SCA.SYSTEM バスのメンバーとして追加し、サーバーまたはクラスターのメッセージング・エンジンを構成します。

この構成タスクを実行する場合は、217 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』を参照してください。

- リモート宛先を使用するように、サーバーまたはクラスターの拡張構成を設定します。この構成では、サーバーまたはクラスターがリモート宛先ロケーションを宛先として使用するよう定義します。

このサーバーまたはクラスターは SCA.SYSTEM バスのメンバーである必要はありません。

このサーバーまたはクラスターにメディエーション・モジュールをデプロイすると、キュー宛先は SCA.SYSTEM バスのメンバーである別のサーバーまたはクラスター上に構成されます。

この構成タスクを実行する場合は、221 ページの『メディエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスターの構成』を参照してください。

サーバーまたはクラスターは、メディエーション・モジュールをデプロイできるように構成されます。

これで、サーバーまたはクラスター上にメディエーションをデプロイしたり、サーバーまたはクラスターを管理するその他のアクションを実行できるようになります。

#### 使用可能なサーバーの管理:

サーバーの作成後、ランタイム状態のモニター、構成の変更、サーバーの始動および停止、その他のさまざまな管理用タスクの実行などを通じて、サーバーを管理できます。

これらのタスクは、作成済みのサーバーに対して実行します。

アプリケーションやサービスを実行するには、ESB サーバーを構成し、稼働する必要があります。サーバーの作成後、他のさまざまな管理用タスクを実行してサーバーを管理できます。

サーバーの管理には、管理コンソールまたは各種の `wsadmin` コマンドのいずれかを使用できます。

このトピックでは、管理コンソールを使用して使用可能なサーバーを管理する代替的な方法について要約を説明し、関連タスクの説明へのリンクを提供します。

- サーバーのリストを表示します。

ナビゲーション・ペインで、「サーバー」 → 「アプリケーション・サーバー」をクリックします。

「アプリケーション・サーバー」コレクション・ページが表示され、サーバー、サーバーのセル、およびサーバーを保持するノードがリストされます。このページでは、サーバーが開始済みであるか、停止中であるか、または使用不可であることを示す状況が表示されます。このページを使用して、サーバーに対するいくつかのアクションを実行できます。例えば、新規サーバーの作成、サーバーの始動または停止、サーバーの詳細設定を行うページのオープンなどです。

- サーバーに関する詳細を表示します。

「アプリケーション・サーバー」ページでサーバーのリストを表示し、サーバーの名前をクリックします。

すると、このサーバーの設定ページが表示されます。このページを使用して、サーバーおよびそのリソースの詳細情報を表示したり、構成を変更したりできます。

- [../../com.ibm.websphere.nd.doc/info/ae/ae/trun\\_svr\\_start.html](http://../com.ibm.websphere.nd.doc/info/ae/ae/trun_svr_start.html)

アプリケーションおよびサービスを実行するには、サーバーが開始済みでなければなりません。サーバーが Network Deployment セル内にある場合、ノード・エージェントが既に実行中である必要があります。

- 管理対象ノードのノード・エージェントを開始するには、  
`wesb_install%profiles%profile_name%bin%startNode` コマンドを実行します。ここで、`profile_name` には、管理対象ノードのカスタム・プロファイルの名前が入ります。
- サーバーを開始するには、「アプリケーション・サーバー」コレクション・ページでサーバーのリストを表示し、サーバーのチェック・ボックスを選択してから、「開始」をクリックします。

- [../../com.ibm.websphere.nd.doc/info/ae/ae/trun\\_watch\\_app.html](http://../com.ibm.websphere.nd.doc/info/ae/ae/trun_watch_app.html).

ランタイム・コンポーネントの状況をモニターして、開始後にも必要な操作が可能な状態であることを確認する必要があります。

サーバー・ログに表示されたメッセージを参照できます。管理コンソールの「**トラブルシューティング (Troubleshooting)**」 → 「**ログとトレース (Logs and Trace)**」ページを使用して、ランタイム・コンポーネントの状況をモニターすることもできます。

- [../../com.ibm.websphere.nd.doc/info/ae/ae/trun\\_svr\\_stop.html](http://../com.ibm.websphere.nd.doc/info/ae/ae/trun_svr_stop.html).

サーバーを停止すると、アプリケーションおよびサービスを実行するためにこのサーバーを使用できなくなります。

「アプリケーション・サーバー」コレクション・ページでサーバーのリストを表示し、サーバーのチェック・ボックスを選択してから「**停止**」をクリックします。

- 『サーバーの削除』

サーバーを削除すると、サーバーの ID および構成が削除され、そのサーバーへのアプリケーションやサービスのデプロイメントも同時に削除されます。そのため、このサーバーはアプリケーションやサービスのデプロイに使用できなくなります。

「アプリケーション・サーバー」ページでサーバーのリストを表示し、サーバーのチェック・ボックスを選択してから「**削除**」をクリックします。

#### サーバーの削除:

サーバーを削除すると、サーバーの ID および構成が削除され、そのサーバーへのアプリケーションやサービスのデプロイメントも同時に削除されます。そのため、このサーバーはアプリケーションやサービスのデプロイに使用できなくなります。

アプリケーションやサービスを実行するための特定のサーバーが不要になった場合は、サーバーを削除できます。

サーバーを削除するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サーバー」 → 「アプリケーション・サーバー」をクリックします。「アプリケーション・サーバー」ページにサーバーのリストが表示され、サーバーが開始済みであるか、停止中であるか、使用不可であることを示す状況が表示されます。
2. サーバーが停止中であることを確認してください。

サーバーを停止するには、サーバー名の横のチェック・ボックスを選択してから、「**停止**」をクリックします。

3. サーバーを削除します。

サーバーを削除するには、サーバー名の横のチェック・ボックスを選択してから、「**削除**」をクリックします。

4. 変更内容をマスター構成に保管します。

これで、サーバーは「アプリケーション・サーバー」ページのリストから削除され、アプリケーションやサービスのデプロイに使用できなくなります。

## サーバー・クラスターによるワークロードの平衡化

サーバー・クラスターは、一括管理され、ワークロード管理に関与する一群のサーバーです。管理者は、既存のサーバーおよびバス環境にサーバー・クラスターを作成および構成できます。

クラスター環境を作成する前に、以下のアクションを実行してください。

1. サーバー・クラスターを使用する環境を計画して準備します。例えば、次の項目について考慮します。
  - クラスタリングを正常にインプリメントするための十分なリソースがあることを確認します。
  - サーバーをクラスター化することはアプリケーションにとって効果的かどうかを判断します。アプリケーションで大量の要求を処理する場合は、クラスターにインストールすることにより、処理スループットが改善される場合があります。クラスターへのインストールを考慮する別の理由として、定期保守および装置障害時のアプリケーションの可用性の確保などがあります。
  - いずれかのアプリケーションにサービスが含まれているかどうか判別します。
  - Network Deployment や WebSphere Application Server Network Deployment バージョン 6 で提供されるクラスタリング・サポートについての理解を深めます。クラスタリング・サポートについて詳しくは、関連するトピックを参照してください。
  - ステップを実行する前に、このタスクを概観してください。
2. デプロイメント・マネージャー・プロファイルと 1 つ以上のカスタム・プロファイルを作成し、カスタム・プロファイルをデプロイメント・マネージャー・セルに統合することにより、Network Deployment セルを作成します。

一群のサーバー間でサービス要求などのワークロードの平衡を取る場合は、サーバー・クラスターを作成して、そのクラスターのメンバーとしてサーバーを追加します。また、バックアップ・クラスターを作成し、割り当て先のサーバー・クラスターのフェイルオーバー機能をサポートすることもできます。

ワークロード管理用のクラスターを構成および使用する方法について理解するため、次のシナリオについて考えてみます。クライアント要求は、単一マシンにある複数のクラスター・メンバー間で分散されます。(クライアントとは、エンド・ユーザーとサーバーを接続する、任意のサービス要求元、サーブレット、Java アプリケーション、あるいは他のプログラムまたはコンポーネントを指します。) さらに複雑なワークロード管理のシナリオでは、クラスター・メンバーを同じシスプレックス内に分散させることができます。

サーバー・クラスターの管理方法について詳しくは、以下のサブトピックを参照してください。サブトピックでは、エンタープライズ・サービス・バスでのサーバー・クラスターの使用に関連した情報を提供し、適切な WebSphere Application Server のトピックへのリンクを掲載しています。

- サーバー・クラスターを作成します。

サーバー・クラスターの作成について詳しくは、サーバー・クラスターの作成を参照してください。

- クラスターのサーバー・メンバーをいくつか作成します。

サーバー・メンバーの作成について詳しくは、『『クラスター・メンバーの作成』』を参照してください。

- 使用可能なクラスターを管理します。

サーバー・クラスターを作成した後は、その実行時状態のモニター、構成の変更、クラスターの開始と停止を含め、広範な管理タスクを実行してクラスターを管理できます。

#### クラスター・メンバーの作成:

このタスクを使用して、サーバー・クラスターのメンバーとしてサーバーを作成します。

クラスター・メンバーを作成するには、サーバー・クラスターの作成の説明に従って、事前にクラスターを作成しておく必要があります。

サーバー・クラスターでのワークロードの分散にサーバーを関与させる場合は、そのサーバーのクラスター・メンバーを作成します。

クラスター・メンバーを作成するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サーバー」 → 「クラスター」をクリックします。「サーバー・クラスター」ページが表示され、ネットワーク・デプロイメント・セルのクラスターがリストされます。
2. コンテンツ・ペインで、クラスターの名前をクリックします。
3. 「クラスター・メンバー」をクリックします。「クラスター・メンバー」ページが表示され、クラスターのメンバーのリスト、メンバーが置かれているノードの説明、およびメンバーが開始、停止、または問題が発生しているかどうかの説明が表示されます。
4. 「新規」をクリックし、「新規クラスター・メンバーの作成 (Create new cluster members)」ページのステップに従います。



- a. 「メンバー名」では、クラスターのメンバーとして追加する新規サーバーの名前を入力します。
  - b. 「ノードの選択」では、サーバーを実行するノードを選択します。
  - c. 「固有 HTTP ポートの生成」が選択されていることを確認します。
  - d. 「テンプレートの選択:」の下で、defaultESBServer テンプレートを選択します。
    - 1) 「デフォルトのアプリケーション・サーバー・テンプレート」を選択します。
    - 2) ドロップダウン・リストから、defaultESBServer を選択します。
      - テンプレートを選択する代わりに、既存の ESB サーバーを選択できます。新規サーバーは、選択したサーバーのすべてのプロパティを継承します。
      - z/OS ノードを選択した場合、提供されるテンプレート名はそのプラットフォームを示します。
      - 使用可能なサーバー・テンプレートについて詳しくは、224 ページの『サーバー・テンプレート』を参照してください。
  - e. 「適用」をクリックしてクラスター・メンバーを終了します。クラスター・メンバーはさらに追加できます。追加するすべてのクラスター・メンバーは、同じサーバー・テンプレートを基にして作成されるので、少ないステップで追加できます。
  - f. 「次へ」をクリックします。
  - g. 新規クラスター・メンバーの要約情報を確認し、「終了」をクリックします。
5. 変更内容をマスター構成に保管します。構成の変更を保管する作業の一環として、「保管」ページで「保管」をクリックする前に、「ノードと変更を同期化」を選択できます。
  6. クラスター・メンバーのプロパティを表示するには、クラスター・メンバー・ページの「メンバー名」の下でメンバーの名前をクリックします。これにより、クラスター・メンバー・インスタンスの設定ページが表示されます。

これで、既存のサーバー・クラスターのメンバーであるアプリケーション・サーバーが作成されました。

クラスターのプロパティおよびローカル・トポロジを表示できます。「サーバー」 → 「クラスター」 → 「[コンテンツ・ペイン] *cluster\_name*」をクリックします。

クラスターの構成は、変更可能です。「保管」をクリックせず、管理構成を保管しなかった場合は、「構成」タブと「ローカル・トポロジ」タブのみが表示されます。「ランタイム」タブを表示するには、管理構成を保管する必要があります。また上述のとおり、変更がノードと同期されていることも確認してください。

この他に、さらにクラスター・メンバーを作成したり、クラスターを開始したりできます。クラスター構成オプションについて詳しくは、『クラスターによるワークロードの平衡化』を参照してください。

スクリプトを使用してクラスター・メンバーを作成するタスクを自動化できます。クラスターでスクリプトを使用する方法については、『スクリプトによるクラスターの作成』を参照してください。

クラスター・メンバーを管理するには、クラスター・メンバー・ページを使用できます。このページを使用して、クラスター・メンバーをリストし、詳細を表示したり操作したりするメンバーを選択できます。

スクリプトを使用してクラスター・メンバーを作成するタスクを自動化できます。クラスター・メンバーでスクリプトを使用する方法については、『スクリプトによるクラスターの作成』を参照してください。

### SCA ランタイム用サーバーまたはクラスターの構成:

サーバーまたはサーバー・クラスター上にメディエーション・モジュールをデプロイするには、事前にそのサーバーまたはクラスターに必要な SCA ランタイム用に構成しておく必要があります。この拡張構成で、SCA ランタイムで使用されるキュー宛先をローカルにホストするか、リモート・サーバーまたはクラスター上でホストするかを定義します。

スタンドアロン・サーバー・プロファイルを作成している場合は、そのサーバーは既に SCA ランタイム用に構成されているため、そのサーバーに対してこの作業は行わないでください。

Network Deployment で SCA モジュールをサポートするためのサーバーまたはサーバー・クラスターを作成している場合は、この作業を実行して、必要な SCA ランタイム用にそのサーバーまたはクラスターを構成する必要があります。

この作業を開始する前に、SCA ランタイムに必要な以下のような 1 つ以上のサーバーまたはサーバー・クラスターが既に作成済みであることが必要です。

- メディエーション・モジュールをデプロイするサーバーまたはクラスター。
- SCA ランタイムで使用するキュー宛先を作成するサーバーまたはクラスター。

両方に同じサーバーまたはクラスターを使用できます。

サーバーまたはサーバー・クラスターの作成に関する詳細は、以下のタスクの記述を参照してください。

- サーバーの作成
- サーバー・クラスターの作成

WebSphere Process Server にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、メディエーション・モジュールで処理されるメッセージを保持するために SCA ランタイムによって使用されます。

キュー宛先を作成する場所を指定するには、サーバーまたはクラスター構成用の以下の選択肢のいずれかを実行します。

- サーバーまたはクラスターの拡張構成を設定して、宛先をホストします。この構成では、メディエーション・モジュール (ローカルにデプロイされている、あるいは別のサーバーまたはクラスターにデプロイされている) のキュー宛先を、こ

のサーバーまたはクラスターでホストできるように定義します。また、この構成では、サーバーまたはクラスターを SCA.SYSTEM バスのメンバーとして追加し、サーバーまたはクラスターのメッセージング・エンジンを構成します。

この構成タスクを実行する場合は、217 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』を参照してください。

- リモート宛先を使用するように、サーバーまたはクラスターの拡張構成を設定します。この構成では、サーバーまたはクラスターがリモート宛先ロケーションを宛先として使用するよう定義します。

このサーバーまたはクラスターは SCA.SYSTEM バスのメンバーである必要はありません。

このサーバーまたはクラスターにメディエーション・モジュールをデプロイすると、キュー宛先は SCA.SYSTEM バスのメンバーである別のサーバーまたはクラスター上に構成されます。

この構成タスクを実行する場合は、221 ページの『メディエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスターの構成』を参照してください。

サーバーまたはクラスターは、メディエーション・モジュールをデプロイできるように構成されます。

これで、サーバーまたはクラスター上にメディエーションをデプロイしたり、サーバーまたはクラスターを管理するその他のアクションを実行できるようになります。

#### **使用可能なサーバー・クラスターの管理:**

サーバー・クラスターを作成したら、ランタイム状態のモニタリング、構成の変更、クラスターまたはクラスター・メンバーの開始と停止、およびクラスター管理のためのその他のさまざまな管理用タスクの実行が可能です。

アプリケーションおよびサービスの実行状況は、クラスターとそのクラスターのメンバーであるサーバーの構成および実行状況に応じて異なります。サーバー・クラスターを作成したら、クラスター管理のためのさまざまな管理用タスクを実行できます。

クラスターの管理は、管理コンソールまたは一連の `wsadmin` コマンドを使用して行えます。

このトピックでは、使用可能なクラスターを管理コンソールを使用して管理するさまざまな方法についての概要を説明し、関連するタスクを説明している箇所へのリンクも記載します。

- クラスターのリストを表示します。

ナビゲーション・ペインで、「サーバー」 → 「クラスター」をクリックします。

クラスターがリストされた「サーバー・クラスター (Server Cluster)」コレクション・ページが表示されます。このページでは、各クラスターの状況 (開始、停止、または使用不可) も示されます。このページを使用して、クラスターに関する一連のアクションを実行できます。例えば、新規クラスターを作成したり、クラスターを開始および停止したり、クラスターの詳細設定ページを開いたりできます。

- クラスターに関する詳細を表示します。

クラスターのリストを表示し、「サーバー・クラスター (Server Cluster)」コレクション・ページでクラスター名をクリックします。

そのクラスターの設定ページが表示されます。このページを使用して、サーバー・クラスター構成およびローカル・トポロジーの表示や変更が可能です。サーバー・クラスターの作成後に管理用の構成を保存すれば、サーバー・クラスターのランタイム情報を表示することもできます。

- クラスターのメンバーであるサーバーのリストを表示します。

クラスターに関する詳細を表示した後、「サーバー・クラスター (Server Cluster)」設定ページで「**クラスター・メンバー**」をクリックします。

「クラスター・メンバー」コレクション・ページが表示されます。このページにクラスターのメンバーがリストされ、メンバーが存在するノードが示され、メンバーの状況 (開始、停止、または問題が発生) が表示されます。このページを使用して、クラスター・メンバーに関する一連のアクションを実行できます。例えば、新規のクラスター・メンバーを作成したり、クラスター・メンバーの詳細設定ページを開いたりできます。

- [../com.ibm.websphere.nd.doc/info/ae/ae/trun\\_wlm\\_member.html](http://com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_member.html)

「クラスター・メンバー」コレクション・ページを表示し、「**クラスター・メンバー**」をクリックします。「クラスター・メンバー」ページにクラスターのメンバーがリストされ、メンバーが存在するノードが示され、メンバーの状況 (開始、停止、または問題が発生) が表示されます。

- [../com.ibm.websphere.nd.doc/info/ae/ae/trun\\_wlm\\_cluster\\_start.html](http://com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_start.html)

アプリケーションおよびサービスを実行するには、クラスターを開始している必要があります。

クラスターのリストを表示し、「サーバー・クラスター (Server Cluster)」コレクション・ページでクラスターに関するチェック・ボックスを選択し、「**開始**」または「**リップルスタート (Ripplestart)**」をクリックします。

- [../com.ibm.websphere.nd.doc/info/ae/ae/trun\\_watch\\_app.html](http://com.ibm.websphere.nd.doc/info/ae/ae/trun_watch_app.html)

ランタイム・コンポーネントの状況をモニターして、開始後にも必要な操作が可能であることを確認する必要があります。

クラスター・ログに表示されたメッセージを参照できます。管理コンソールの「**トラブルシューティング (Troubleshooting)**」 → 「**ログとトレース (Logs and Trace)**」ページを使用して、ランタイム・コンポーネントの状況をモニターすることもできます。

- ワークロード管理ランタイムの動作を調整します。

アプリケーションにタイムアウトの問題が発生した場合や、ネットワークの待ち時間が極端に長い場合は、`com.ibm.CORBA.RequestTimeout` プロパティのタイムアウト間隔を変更します。または、クライアントのワークロード管理状態の最新表示が早すぎたり遅すぎたりする場合は、

`com.ibm.websphere.wlm.unusable.interval` プロパティの間隔を変更します。

- クラスタを停止します。

サーバー・クラスタを停止すると、アプリケーションおよびサービスを実行できなくなります。

クラスタのリストを表示し、「サーバー・クラスタ (Server Cluster)」コレクション・ページでクラスタに関するチェック・ボックスを選択し、「**停止**」または「**即時停止 (Immediate Stop)**」をクリックします。

- [../com.ibm.websphere.nd.doc/info/ae/ae/trun\\_wlm\\_clustermember\\_delete.html](http://../com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_clustermember_delete.html)

クラスタ・メンバーを削除すると、クラスタからサーバーが除去され、そのサーバーは削除されます。クラスタ・メンバーを削除したら、必ずサーバーも削除されます。

クラスタ・メンバーのリストを表示し、「クラスタ・メンバー」コレクション・ページでクラスタ・メンバーに関するチェック・ボックスを選択し、「**削除**」をクリックします。

- [../com.ibm.websphere.nd.doc/info/ae/ae/trun\\_wlm\\_cluster\\_delete.html](http://../com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_delete.html)

サーバー・クラスタを削除すると、クラスタおよびすべてのクラスタ・メンバーが削除されます。クラスタを削除する場合、特定のクラスタ・メンバーのみを残したり、そのクラスタのある部分にインストールしたアプリケーションのみを残したりすることはできません。

クラスタのリストを表示し、「サーバー・クラスタ (Server Cluster)」コレクション・ページでクラスタに関するチェック・ボックスを選択し、「**削除**」をクリックします。

## バス・トポロジーのその他の用途

SCA モジュールに使用される `SCA.SYSTEM` バスは、WebSphere Application Server で使用される場合と同様、標準のサービス統合バスです。 `SCA.SYSTEM` バスのほかにも、サービス統合バスを作成してサービス統合ロジックやその他のアプリケーションをサポートできます。

例えば、`SCA.APPLICATION` バスが用意されており、JMS バインディングによってデPLOYされたモジュールの JMS キュー宛先やその他の JMS リソースを定義するために使用できます。

他のバスを作成して、WebSphere Application Server の場合と同様に使用したり (例えば、WebSphere Process Server 内でサービス要求元やプロバイダーとして動作するアプリケーション用)、あるいは WebSphere MQ にリンクしたりできます。

WebSphere Process Server デプロイメント・マネージャーを使用して、WebSphere

Application Server 上にデプロイされたアプリケーションおよびモジュール用に使用する各アプリケーション・サーバーを管理することもできます。

バス・トポロジーのその他の用途については、WebSphere Application Server 用の『サービス統合 (Service integration)』に関する一連のトピックを参照してください。

## サービス・アプリケーションの管理

サービス・アプリケーションを WebSphere Process Server の管理コンソールから管理できます。サービス・アプリケーションはサービスを提供します。また、Service Component Architecture (SCA) モジュールに関連付けられています。

WebSphere Process Server がサポートする SCA モジュールの種類は、メディエーション・モジュールと呼ばれます。メディエーション・モジュールでは、サービス要求およびサービス応答のフォーマット、内容、またはターゲットを簡単に変更できます。

### サービス・アプリケーションの使用開始

サービス・アプリケーションのデプロイ後に、そのサービス・アプリケーションのコンポーネントを表示し、管理できます。

アプリケーションおよび関連する Service Component Architecture (SCA) モジュールを表示し、管理できます。WebSphere Process Server がサポートする SCA モジュールの種類は、メディエーション・モジュールと呼ばれます。

デプロイしたすべての SCA モジュールをリストでき、SCA モジュールのサービス要求元およびサービス・プロバイダーへの接続方法の詳細を表示できます。

#### アプリケーション統合環境の概要:

サービス・アプリケーションには、Service Component Architecture (SCA) モジュールが関連付けられています。サービス・アプリケーションは、WebSphere Process Server の EAR (エンタープライズ・アーカイブ) ファイル内にデプロイします。

#### サービス・アプリケーションのデプロイ

サービス・アプリケーションを含む EAR ファイルをデプロイするプロセスは、他の EAR ファイルをデプロイするプロセスと同じです。メディエーション・パラメーターの値はデプロイメント時に変更できます。SCA モジュールを含む EAR ファイルをデプロイした後は、以下に関する詳細を表示できます。

- サービス・アプリケーション。
- サービス・アプリケーションに関連付けられた SCA モジュール。
  - SCA モジュールがサービス要求元およびサービス・プロバイダーに接続する方法を確認できます。SCA モジュールは、エクスポートを通じてサービス要求元と接続し、インポートを通じてサービス・プロバイダーと接続します。

#### SCA モジュールの詳細の表示

表示可能な SCA モジュールの詳細には、以下が含まれます。表示できる詳細の正確な内容は、SCA モジュールによって異なります。

- SCA モジュール名。
- SCA モジュールの説明。
- 関連アプリケーション名。
- SCA モジュールのインポート。
  - インターフェース。
    - インポート・インターフェースは、SCA モジュールがサービスにアクセスする方法を記述した抽象定義です。
  - バインディング。
    - インポート・バインディングは具象定義です。SCA モジュールがサービスにアクセスするときの物理メカニズムを指定します。例えば、SOAP/HTTP を使用します。
- SCA モジュールのエクスポート。
  - インターフェース。
    - エクスポート・インターフェースは、サービス要求元が SCA モジュールにアクセスする方法を記述した抽象定義です。
  - バインディング。
    - エクスポート・バインディングは具象定義です。サービス要求元が SCA モジュールにアクセスするとき、間接的にサービスにアクセスするときの物理メカニズムを指定します。
- SCA モジュール・プロパティ。

#### サービス・アプリケーションの理解:

サービス・アプリケーションはサービスを提供します。また、Service Component Architecture (SCA) モジュールに関連付けられています。SCA モジュールによってサービスはカプセル化されるため、サービス・ユーザーに影響を与えることなくサービスを変更できます。WebSphere Process Server がサポートする SCA モジュールの種類は、メディエーション・モジュールと呼ばれます。

#### メディエーション・モジュール:

メディエーション・モジュールは、Service Component Architecture (SCA) モジュールであり、サービス要求のフォーマット、内容、またはターゲットを変更できます。

メディエーション・モジュールは、サービス要求元とサービス・プロバイダー間の送信途中にあるメッセージを操作します。これにより、メッセージをさまざまなサービス・プロバイダーに送付できます。メッセージを変換することもできます。つまり、メッセージの内容または形式を変更できます。さらに、メディエーション・モジュールは、メッセージ・ロギングや要件に応じたエラー処理などの機能を提供します。

メディエーション・モジュールの一定の側面を、モジュールを再デプロイすることなく、WebSphere Process Server 管理コンソールから動的に変更できます。

## メディエーション・モジュールのコンポーネント

メディエーション・モジュールに含まれる項目には、以下のものがあります。

- インポート
  - インポートは、SCA モジュールとサービス・プロバイダー間の対話を定義します。
  - インポートによって、SCA モジュールは外部サービスをローカル・サービスのよう呼び出すことができます。
  - メディエーション・モジュールのインポートは WebSphere Process Server から表示できます。インポート・バインディングが Web サービスまたは SCA バインディングである場合は変更できます。
- エクスポート
  - エクスポートは、SCA モジュールとサービス要求元間の対話を定義します。
  - エクスポートによって、SCA モジュールはサービスを提供できます。エクスポートは、SCA モジュールの外部インターフェース (アクセス・ポイント) を定義します。
  - メディエーション・モジュールのエクスポートは WebSphere Process Server から表示できます。
- SCA コンポーネント
  - SCA コンポーネント、つまりサービス・コンポーネントは SCA ビルディング・ブロックです。SCA コンポーネントを使用して、メディエーション・モジュールなどの SCA モジュールを作成します。WebSphere Integration Developer を使用して、SCA モジュールおよびコンポーネントをグラフィカルに作成しカスタマイズできます。また、SCA モジュールの一部のプロパティを、モジュールを再デプロイすることなく WebSphere Process Server 管理コンソールから動的にカスタマイズできます。
  - 一般に、メディエーション・モジュールには、メディエーション・フロー・コンポーネントと呼ばれる特定のタイプの SCA コンポーネントが含まれます。メディエーション・フロー・コンポーネントは、メディエーション・フローを定義します。メディエーション・モジュールに含めることができるメディエーション・フロー・コンポーネントは、最大で 1 つです。
  - メディエーション・フロー・コンポーネントに含めることができるメディエーション・プリミティブは、1 つでも複数でも構いません。また、メディエーション・プリミティブをまったく含めなくても構いません。WebSphere Process Server は、提供されているメディエーション・プリミティブのセットをサポートします。これにより、メッセージ・ルーティングおよび変換の機能が提供されます。WebSphere Process Server がサポートするメディエーション・プリミティブの中には、カスタム・ロジックを呼び出すことができるものがあります。
  - メディエーション・モジュールには、メディエーション・フロー・コンポーネントが含まれている必要はありません。メディエーション・フロー・コンポーネントが含まれていないメディエーション・モジュールには、サービス要求のあるプロトコルから別のプロトコルに変換するという目的があります。例えば、SOAP/JMS を使用して作成したサービス要求を、送信前に SOAP/HTTP に変換しなければならない場合があります。



注: メディエーション・モジュールは WebSphere Process Server から表示できます。 WebSphere Process Server から、メディエーション・モジュールに一定の変更を加えることもできます。ただし、WebSphere Process Server モジュールの内部から SCA コンポーネントを表示あるいは変更することはできません。SCA コンポーネントをカスタマイズするには、WebSphere Integration Developer を使用してください。

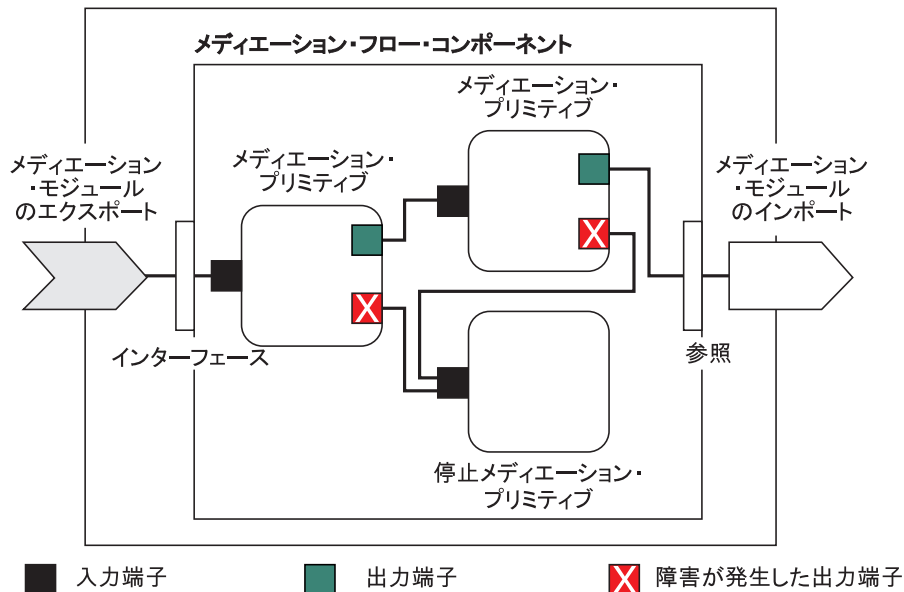


図 18. メディエーション・モジュールの簡単な例: メディエーション・モジュールには 1 つのメディエーション・フロー・コンポーネントが含まれます。メディエーション・フロー・コンポーネントにはメディエーション・プリミティブが含まれます。

- プロパティ

- メディエーション・プリミティブはプロパティを持ちます。一部のプロパティは、SCA モジュールの追加のプロパティとして管理コンソールに表示させることができます。
- メディエーション・プリミティブ・プロパティを WebSphere Process Server 管理コンソールから可視にするには、Integration Developer がプロパティにプロモート済みのフラグを立てる必要があります。一部のプロパティは管理上の構成に適しています。これらのプロパティは統合サイクルから管理サイクルへプロモート可能であるため、WebSphere Integration Developer はこれらのプロパティをプロモート可能なプロパティとして記述します。プロパティの中には、管理構成に不向きなものもあります。そのようなプロパティを変更すると、メディエーション・フローに影響が及び、メディエーション・モジュールを再デプロイしなければならない場合が多いためです。WebSphere Integration Developer では、メディエーション・プリミティブのプロモートされるプロパティの下に、プロモートするプロパティのリストが表示されます。
- WebSphere Process Server 管理コンソールを使用すると、メディエーション・モジュールの再デプロイや、サーバーまたはモジュールの再始動を行うことなく、プロモートされるプロパティの値を変更できます。メディエーション・

フローを新たに呼び出すと、プロパティーの変更内容が即時に有効になります (プロパティーの変更をデプロイメント・マネージャー・セルで行った場合を除く)。デプロイメント・マネージャー・セル内で行われた変更は、セル内のすべてのノードが同期化された後に有効になります。処理中のメディエーション・フローの呼び出しでは、前の値が引き続き使用されます。

**注:** プロパティー値ではなく、メディエーション・プリミティブのプロパティー名およびタイプを変更する場合は、WebSphere Integration Developer を使用してください。

### メディエーション・モジュールのデプロイ

メディエーション・モジュールは、WebSphere Integration Developer を使用して作成され、一般に EAR (エンタープライズ・アーカイブ) ファイル内で WebSphere Process Server にデプロイされます。

プロモートされるプロパティーの値は、デプロイメント時に変更できます。

WebSphere Integration Developer からメディエーション・モジュールをエクスポートして、WebSphere Integration Developer がメディエーション・モジュールを JAR (Java アーカイブ) ファイル内にパッケージ化し、その JAR ファイルを EAR ファイル内にパッケージ化できるようにすることができます。この後、管理コンソールから新規アプリケーションをインストールすることにより、EAR ファイルをデプロイできます。

メディエーション・モジュールは 1 つのエンティティーと考えられます。ただし、SCA モジュールは、1 つの JAR ファイルに保管されたいくつかの XML ファイルによって定義されます。

### メディエーション・モジュールが含まれる EAR ファイルの例

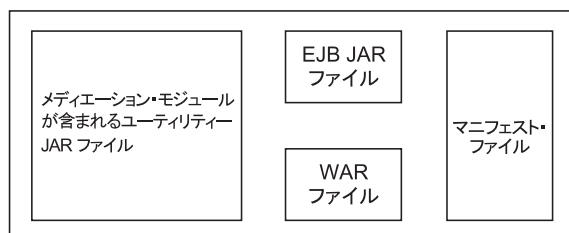


図 19. メディエーション・モジュールが含まれる EAR ファイルの簡単な例: EAR ファイルには JAR が含まれません。ユーティリティー JAR ファイルにはメディエーション・モジュールが含まれます。

### インポートとインポート・バインディング:

インポートは、Service Component Architecture (SCA) モジュールとサービス・プロバイダー間の対話を定義します。インポートにより、SCA モジュールは、ローカルのサービスに対するように外部サービス (SCA モジュールの外部にあるサービス) にアクセスできます。インポート・バインディングでは、外部サービスにアクセスする特定の方法を定義します。

外部サービスにアクセスする必要がある場合、SCA モジュールにはインポートは不要です。メディエーション・モジュールは、SCA モジュールの一種です。

### インターフェースとバインディング

1 つの SCA モジュールのインポートには、少なくとも 1 つのインターフェースが必要であり、1 つの SCA モジュールのインポートにはバインディングが 1 つだけあります。

- インターフェース
  - インポート・インターフェースは抽象定義です。アクセス・ポイントを定義します。
  - インポート・インターフェースは、Web サービスを記述する際に使用する XML 言語である、WSDL (Web サービス記述言語) によって定義されます。
  - 1 つの SCA モジュールは、多くのインポート・インターフェースを持つことができます。
- バインディング
  - インポート・バインディングは具象定義です。SCA モジュールが外部サービスにアクセスするために使用する物理的なメカニズムを指定します。

### サポートされるインポート・バインディング

エンタープライズ・サービス・バスは、次のインポート・バインディングをサポートします。

- Web サービス・バインディング
  - Web サービス・バインディングを使用すると、Web サービスにアクセスできます。
  - サポートされるプロトコルは SOAP/HTTP および SOAP/JMS です。
- SCA バインディング
  - SCA バインディングでは、SCA モジュールを他の SCA モジュールに接続します。
  - SCA バインディングはデフォルト・バインディングとも呼ばれます。
- Java Message Service (JMS) 1.1 のバインディング
  - JMS バインディングにより、WebSphere Application Server のデフォルト・メッセージング・プロバイダーとの相互運用が可能になります。
  - JMS は、さまざまなトランスポート・タイプ (TCP/IP や HTTP(S) など) を活用します。
  - JMS Message クラスおよびその 5 つのサブタイプ (Text、Bytes、Object、Stream、および Map) が自動的にサポートされます。
- WebSphere MQ JMS バインディング
  - WebSphere MQ JMS バインディングにより、WebSphere MQ ベースの JMS プロバイダーとの相互運用が可能になります。
  - WebSphere MQ を JMS プロバイダーとして使用する場合は、WebSphere MQ JMS バインディングが必要です。
  - JMS Message クラスおよびその 5 つのサブタイプ (Text、Bytes、Object、Stream、および Map) が自動的にサポートされます。

- WebSphere MQ バインディング
  - WebSphere MQ バインディングにより、WebSphere MQ との相互運用が可能になります。
  - ネイティブの WebSphere MQ アプリケーションと通信する場合は、WebSphere MQ バインディングが必要です。
  - WebSphere MQ バインディングは、WebSphere MQ クライアント接続を介したりモート・キュー・マネージャーとの間でのみ使用できます。ローカル・キュー・マネージャーとの間では使用できません。
- WebSphere Adapter バインディング
  - WebSphere Adapter によって、エンタープライズ情報システム (EIS) との対話が可能になります。

### エクスポートとエクスポート・バインディング:

エクスポートは、Service Component Architecture (SCA) モジュールとサービス要求元間の対話を定義します。エクスポートにより、SCA モジュールがサービスを提供できるようになります。エクスポート・バインディングは、SCA モジュールがサービス要求元によってアクセスされる際の特定の方法を定義します。

メディエーション・モジュールは、SCA モジュールの一種です。

### インターフェースとバインディング

SCA モジュールのエクスポートには、少なくとも 1 つのインターフェースが必要です。

- インターフェース
  - エクスポート・インターフェースは抽象定義です。アクセス・ポイントを定義します。
  - エクスポート・インターフェースは、Web サービスを記述する際に使用する XML 言語である、WSDL (Web サービス記述言語) を使用して定義されます。
  - 1 つの SCA モジュールは、多くのエクスポート・インターフェースを持つことができます。
- バインディング
  - エクスポート・バインディングは具象定義です。サービス要求元がサービスにアクセスするために使用する物理的なメカニズムを指定します。
  - 通常、1 つの SCA モジュール・エクスポートには、1 つのバインディングが指定されています。バインディングが指定されていないエクスポートは、ランタイムにより SCA バインディングを持つエクスポートと解釈されます。

### サポートされるエクスポート・バインディング

エンタープライズ・サービス・バスは、次のエクスポート・バインディングをサポートします。

- Web サービス・バインディング
  - Web サービス・バインディングを使用すると、Web サービスにアクセスできます。

- サポートされるプロトコルは SOAP/HTTP および SOAP/JMS です。
- SCA バインディング
  - SCA バインディングでは、SCA モジュールを他の SCA モジュールに接続します。
  - SCA バインディングはデフォルト・バインディングとも呼ばれます。
- Java Message Service (JMS) 1.1 のバインディング
  - JMS バインディングにより、WebSphere Application Server のデフォルト・メッセージング・プロバイダーとの相互運用が可能になります。
  - JMS は、さまざまなトランスポート・タイプ (TCP/IP や HTTP(S) など) を活用します。
  - JMS Message クラスおよびその 5 つのサブタイプ (Text、Bytes、Object、Stream、および Map) が自動的にサポートされます。
- WebSphere MQ JMS バインディング
  - WebSphere MQ JMS バインディングにより、WebSphere MQ ベースの JMS プロバイダーとの相互運用が可能になります。
  - WebSphere MQ を JMS プロバイダーとして使用する場合は、WebSphere MQ JMS バインディングが必要です。
  - JMS Message クラスおよびその 5 つのサブタイプ (Text、Bytes、Object、Stream、および Map) が自動的にサポートされます。
- WebSphere MQ バインディング
  - WebSphere MQ バインディングにより、WebSphere MQ との相互運用が可能になります。
  - ネイティブの WebSphere MQ アプリケーションと通信する場合は、WebSphere MQ バインディングが必要です。
  - WebSphere MQ バインディングは、WebSphere MQ クライアント接続を介したりモート・キュー・マネージャーとの間でのみ使用できます。ローカル・キュー・マネージャーとの間では使用できません。
- WebSphere Adapter バインディング
  - WebSphere Adapter によって、エンタープライズ情報システム (EIS) との対話が可能になります。

#### 管理インターフェースのサービス・アプリケーション機能:

WebSphere Process Server では、管理コンソールを使用してサービス・アプリケーションの各側面を表示および変更できます。

サービス・アプリケーションはサービスを提供します。また、Service Component Architecture (SCA) モジュールに関連付けられています。エンタープライズ・サービス・バスでサポートされる SCA モジュールのタイプはメディエーション・モジュールです。

## 表示可能な SCA モジュールの詳細

Service Component Architecture (SCA) モジュールを含む EAR (Enterprise ARchive) ファイルをデプロイした後は、SCA モジュールの詳細を表示できます。すべての SCA モジュールと関連アプリケーションをリストし、特定の SCA モジュールの詳細を表示できます。

表示可能な SCA モジュールの詳細には、以下のような情報があります。

- SCA モジュール名
- 関連アプリケーション
- SCA モジュール インポート。
  - インターフェース
  - バインディング
- SCA モジュール エクスポート。
  - インターフェース
  - バインディング
- SCA モジュール・プロパティー

## 変更可能な SCA モジュールの詳細

SCA モジュールを含む EAR ファイルをデプロイ済みの場合は、EAR ファイルを再デプロイしなくても、管理コンソールを使用して、以下の SCA モジュールの詳細を変更できます。

- タイプ SCA のインポート・バインディング。
  - インポート・バインディングを変更することにより、サービス対話を変更できます。
  - SCA バインディングでは、SCA モジュールを他の SCA モジュールに接続します。1 つの SCA モジュールは、2 番目の SCA モジュールと相互作用でき、別の SCA モジュールと相互作用するように変更することもできます。
  - Web サービス・バインディングは、SOAP を使用して SCA モジュールを外部サービスに接続します。
- タイプが Web サービス (WS) のバインディングをインポートします。
  - インポート・バインディングを変更すると、サービス対話を変更できるようになります。
  - WS インポート・バインディングによって SCA モジュールは Web サービスにアクセスできます。WS インポート・バインディングは、指定されたエンドポイントにあるサービスを呼び出します。エンドポイントを変更してバインディングが別のエンドポイントにあるサービスを呼び出すようにすることができます。ほか、インターフェースに互換性があるまったく別のサービスを呼び出すようにすることもできます。
- メディエーション・モジュール・プロパティー。
  - メディエーション・モジュールのプロパティーは、関連付けられているメディエーション・プリミティブに属します。ただし、WebSphere Process Server 管理コンソールでは、これらのプロパティーの一部は SCA モジュールの追加プロパティーとして表示されます。WebSphere Process Server からメディエーション

オン・プリミティブ・プロパティを表示できるようにするには、Integration Developer がプロパティにプロモート済みというフラグを立てる必要があります。

- メディエーション・モジュール・プロパティを変更することにより、メディエーションの動作を変更できます。行うことができるメディエーションの変更は、プロモートされたプロパティによって異なります。

注: バインディングが指定されていないエクスポートは、ランタイムにより SCA バインディングを持つエクスポートと解釈されます。

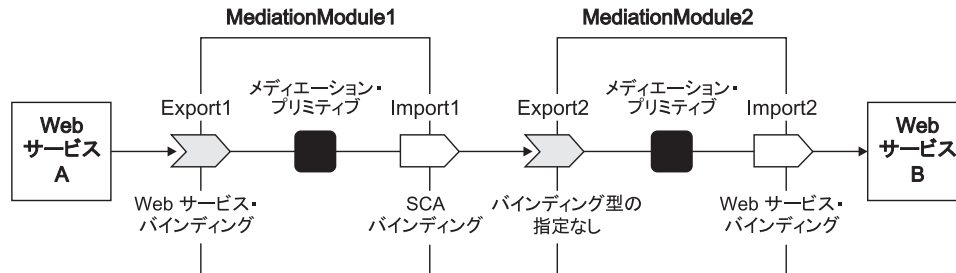


図 20. 別のメディエーション・モジュールと対話するメディエーション・モジュールの例: MediationModule1 は MediationModule2 に接続

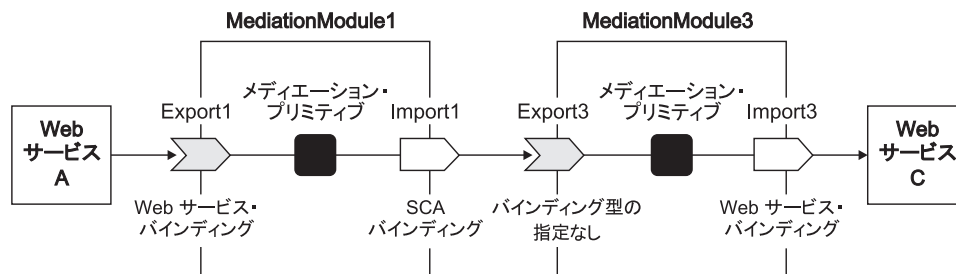


図 21. 別のメディエーション・モジュールと対話するメディエーション・モジュールの例: MediationModule1 は MediationModule3 に接続

## 実動サーバーへのモジュールのインストール

このトピックでは、テスト・サーバーからアプリケーションを取り出して実稼働環境にデプロイするステップについて説明します。

実動サーバーにサービス・アプリケーションをデプロイする前に、テスト・サーバーでアプリケーションをアSEMBルおよびテストします。テストが終了したら、「モジュールの開発とデプロイ」の PDF の『サーバーへのデプロイの準備』の説明に従って関連するファイルをエクスポートし、そのファイルをデプロイ先の実動システムに移動します。詳しくは、WebSphere Integration Developer および WebSphere Application Server for z/OS のインフォメーション・センターを参照してください。

1. モジュールおよび他のファイルを実動サーバーにコピーします。

アプリケーションに必要なモジュールおよびリソース (EAR、JAR、RAR、および WAR の各ファイル) が実稼働環境に移動します。

2. `serviceDeploy` コマンドを実行して、インストール可能な EAR ファイルを作成します。

このステップでは、アプリケーションを実動サーバーにインストールする準備として、サーバーに対してモジュールを定義します。

- a. デプロイするモジュールを含む JAR ファイルを探します。
  - b. 前のステップで見つかった JAR ファイルを入力として使用して、コマンドを実行します。
3. ステップ 2 の EAR ファイルをインストールします。アプリケーションのインストール方法は、アプリケーションをスタンドアロンのサーバーにインストールするか、またはセル内のサーバーにインストールするかによって異なります。

**注:** アプリケーションのインストールには、管理コンソールとスクリプトのどちらかを使用できます。詳しくは、WebSphere Application Server インフォメーション・センターを参照してください。

4. 構成を保管します。これで、モジュールがアプリケーションとしてインストールされます。
5. アプリケーションを開始します。

これで、アプリケーションはアクティブになり、モジュールを介して処理フローが実行されます。

アプリケーションをモニターして、サーバーが要求を正しく処理しているか確認してください。

#### コンソールによるメディエーション・モジュール EAR ファイルのインストール:

メディエーション・モジュールのインストールは、そのメディエーション・モジュールのインストール可能 EAR ファイルを、メディエーション・モジュールをホストするサーバーまたはクラスターに移動する操作で構成されます。インストールされたメディエーション・モジュールが正常に開始および実行されると、デプロイされたと見なすことができます。

メディエーション・モジュールを JAR ファイルにエクスポートしてある場合は、`serviceDeploy` コマンドを使用して、メディエーション・モジュールの JAR ファイルからインストール可能 EAR ファイルを作成します。メディエーション・モジュールのインストール可能 EAR ファイルの作成方法について詳しくは、251 ページの『実動サーバーへのモジュールのインストール』を参照してください。

WebSphere Process Server がメディエーション・モジュールによって提供される機能を使用してアプリケーションとサービスを統合できるようにするには、そのモジュールの EAR ファイルをサーバーまたはクラスターにインストールしてから、デプロイされたモジュールを開始する必要があります。

このトピックでは、管理コンソールを使用してメディエーション・モジュール EAR ファイルをインストールする方法について説明します。あるいは、他の方法でインストールすることも可能です。例えば、エンタープライズ・アプリケーション・フ



ファイルを WebSphere Application Server にインストールする場合のように、wsadmin ツールの `install` コマンドや `installinteractive` コマンドを使用する方法などです。

**重要:** 以下のステップを実行し始めてからアプリケーションのインストールを中止する場合は、「キャンセル」をクリックして終了します。管理コンソールの別のページに移動する場合は、その前に必ず、アプリケーションのインストール・ページで「キャンセル」をクリックしてください。

管理コンソールを使用してメディアエーション・モジュール EAR ファイルをインストールするには、以下のステップを実行します。

1. コンソールのナビゲーション・ツリーで、「アプリケーション」 → 「新規アプリケーションのインストール」をクリックします。 2 ページある「アプリケーション・インストールの準備」ページの最初のページが表示されます。
2. 最初の「アプリケーション・インストールの準備」ページで、以下のサブステップを実行します。

- a. メディアエーション・モジュール EAR ファイル (.ear ファイル) の絶対パス名を指定します。 インストールする EAR ファイルは、クライアント・マシン (管理コンソール Web ブラウザーを実行するマシン) またはサーバー・マシン (クライアントの接続先マシン) にあります。クライアント・マシンの EAR ファイルを指定すると、管理コンソールは管理コンソールが動作しているマシンにその EAR ファイルをアップロードし、アプリケーションのインストールを続行します。

- b. 「次へ」をクリックします。

3. 2 番目の「アプリケーション・インストールの準備」ページで、以下のサブステップを実行します。

- a. デフォルト・バインディングを生成するかどうかを選択します。

デフォルト・バインディングを使用すると、アプリケーション内の不完全なバインディングすべてにデフォルト値が入力されます。既存のバインディングは変更されません。

デフォルト・バインディングの生成で使用されるデフォルト値は、カスタマイズできます。

- b. 「次へ」をクリックします。 セキュリティー警告が表示されたら、「継続」をクリックします。「新規アプリケーションのインストール」ページが表示されます。デフォルト・バインディングを生成することを選択し、アプリケーションでリソース・アダプターを使用しない場合は、要約ステップ (以下の最後のステップ) に進んでください。

4. 「ステップ: インストール・オプションの選択 (Step: Select installation options)」パネルで、以下の設定の値を入力します。 このページの設定値について詳しくは、「Select installation options」設定を参照してください。 値を指定しない場合は、デフォルト値が使用されます。

- a. 「アプリケーションをインストールするディレクトリー」では、アプリケーション EAR ファイルをインストールするディレクトリーを指定します。デフォルト値は `APP_INSTALL_ROOT/cell_name` の値です。`APP_INSTALL_ROOT` 変数は、`install_root/installedApps` です。例えば、WebSphere Process Server が Windows の `C:\Program Files\WESB\` にイン

ストールされている場合、デフォルト・ロケーションは `C:\Program Files\WESB\profile_name\installedApps\cell_name` になります。

**注:** スタンドアロン・プロファイルへのアプリケーションのインストール時にインストール・ディレクトリが指定されていない場合、アプリケーションは `APP_INSTALL_ROOT/base_cell_name` にインストールされます。スタンドアロン・サーバーをネットワーク・デプロイメント・セルに追加すると、新規サーバー構成のセル名は、デプロイメント・マネージャー・ノードのセル名になります。addNode ユーティリティで `-includeapps` オプションを使用すると、addNode 操作より先にインストールされていたアプリケーションは、引き続き `APP_INSTALL_ROOT/base_cell_name` をインストール・ディレクトリとして使用します。ただし、スタンドアロン・サーバーがネットワーク構成に追加された後にインストールされるアプリケーションは、デフォルトのインストール・ディレクトリ `APP_INSTALL_ROOT/network_cell_name` を使用します。addNode 操作の実行時にアプリケーションを `APP_INSTALL_ROOT/network_cell_name` ロケーションに移動するには、インストール時にインストール・ディレクトリを `${APP_INSTALL_ROOT}/${CELL}` として明示的に指定する必要があります。このような場合、アプリケーション・ファイルは常に `APP_INSTALL_ROOT/current_cell_name` 下に置かれません。

- b. 「アプリケーションの配布 (Distribute application)」では、WebSphere Process Server によってインストール先のアプリケーション・バイナリーを展開/削除するかどうかを指定します。デフォルトでは、アプリケーションの配布が使用可能になっています。したがって、コンソールで変更を保管すると、新規にインストールされたアプリケーションのアプリケーション・バイナリーは指定されたディレクトリに展開されます。またアプリケーション・バイナリーは、アンインストールして構成の変更を保管すると削除されます。このオプションを使用不可に設定する場合は、アプリケーションを実行するすべてのノードの宛先ディレクトリ内でアプリケーション・バイナリーが適切に展開されるようにする必要があります。

**重要:** このオプションを使用不可に設定し、アプリケーション・バイナリーをノードにコピーおよび展開しない場合は、後で構成を保管したり、手動で同期化しても、アプリケーション・バイナリーはノードに移動しません。

- c. 「バイナリー構成の使用」では、サーバーまたはクラスターが、バインディング、拡張子、またはデプロイメント記述子に関して、アプリケーション・デプロイメント文書 `deployment.xml` ファイルにあるものを使用する (デフォルト) か、EAR ファイルにあるものを使用するかを指定します。
- d. 「アプリケーション名」には、アプリケーションの名前を入力します。アプリケーション名はセル内で固有のものにする必要があります。オブジェクト名での使用が許可されていない文字は使用できません。オブジェクト名での使用が許可されていない文字のリストについては、オブジェクト名を参照してください。
- e. 「リソース用の MBean の作成 (Create MBeans for resources)」では、アプリケーションの開始時に、アプリケーション内のさまざまなリソース (サ

ープレットや JSP ファイルなど) 用に MBean を作成するかどうかを指定します。デフォルトでは、MBean インスタンスが作成されます。

- f. 「**クラスの再ロードを使用可能にする**」では、アプリケーション・ファイルの更新時にクラスの再ロードを使用可能にするかどうかを指定します。デフォルトでは、クラスの再ロードは使用不可です。クラスの再ロードを可能にすると、メディエーション・モジュールの `deployment.xml` ファイルで `reloadEnabled` が `true` に設定されます。メディエーション・モジュールのクラス定義が変更されると、サーバー・ランタイムはアプリケーションを停止して開始し、アプリケーション・クラスを再ロードします。
- g. 「**再ロード間隔 (秒) (Reload interval in seconds)**」では、更新済みファイルを探してアプリケーションのファイル・システムをスキャンするときの秒数を指定します。デフォルトは、EAR ファイルの `IBM 拡張 (META-INF/ibm-application-ext.xml)` ファイルで指定されている再ロード間隔属性の値です。再ロードを使用可能にするには、ゼロより大きい値 (例えば、1 から 2147483647 の範囲) を指定します。再ロードを使用不可にするには、ゼロ (0) を指定します。

ここで指定する再ロード間隔は、クラスの再ロードが使用可能になっている場合にのみ有効になります。

- h. 「**Web サービスのデプロイ (Deploy Web services)**」では、アプリケーションのインストール時に Web サービスのデプロイ・ツール `wsdeploy` を実行するかどうかを指定します。このツールは、Web サービスを使用してアプリケーションを実行するために必要なコードを生成します。デフォルトでは、`wsdeploy` ツールは実行されません。EAR ファイルに Web サービスを使用するモジュールが含まれており、このファイルで `wsdeploy` ツールが以前に実行されたことがない場合は、アセンブリー・ツールの「**デプロイ**」メニュー選択またはコマンド行から、この設定を使用可能にする必要があります。
- i. 「**入力のオフ/警告/失敗の妥当性検査 (Validate Input off/warn/fail)**」では、WebSphere Process Server により、アプリケーションのインストール時または更新時に指定したアプリケーション参照を検査するかどうかを指定します。また、妥当性検査を使用可能にする場合は、参照に誤りがあることを警告するか、操作を失敗させるかどうかを指定します。アプリケーションは通常、コンテナー管理パーシスタンス (CMP) Bean のデータ・ソースを使用して、またはデプロイメント記述子で定義されるリソース参照またはリソース環境参照を使用して、リソースを参照します。妥当性検査では、アプリケーションによって参照されるリソースが、そのアプリケーションのデプロイメント・ターゲットの有効範囲内で定義されているかどうかを検査されます。リソースの妥当性検査を行わない場合は「**オフ**」、誤りのあるリソース参照について警告メッセージを出す場合は「**警告**」、誤りのあるリソース参照の結果、失敗するオペレーションを停止する場合は「**失敗**」を選択します。
- j. 「**プロセス組み込み構成 (Process embedded configuration)**」では、組み込み構成を処理するかどうかを指定します。組み込み構成は、`resource.xml` や `variables.xml` などのファイルで構成されています。選択する場合、または `true` の場合、組み込み構成は `.ear` ファイルからアプリケーション有効

範囲にロードされます。`.ear` ファイルに組み込み構成が含まれていない場合のデフォルトは `false` です。`.ear` ファイルに組み込み構成が含まれている場合のデフォルトは `true` です。

5. 「ステップ: モジュールをサーバーにマップ (Step: Map modules to servers)」パネルで、すべてのモジュールに対し、「クラスターおよびサーバー」リストからターゲットのサーバーまたはクラスターを選択します。「モジュール」の横にあるチェック・ボックスを選択して、メディエーション・モジュールを選択します。

アプリケーションが `WebSphere Adapter` を使用する場合は、`RAR` ファイルごとにターゲットのサーバーまたはクラスターを指定します。また、`RAR` モジュールに定義されているリソース・アダプターを使用するその他のすべてのモジュールを同じターゲットにマップしてください。

注: サーバーに `RAR` ファイルをインストールする場合、`WebSphere Process Server` はコネクタ・モジュールのマニフェスト (`MANIFEST.MF`) を検索します。まず、`RAR` ファイルの `connectorModule.jar` ファイル内を検索し、`_connectorModule.jar` ファイルからマニフェストをロードします。クラスパス・エントリーが `connectorModule.jar` ファイルのマニフェストにある場合、`RAR` はそのクラスパスを使用します。インストール済みコネクタ・モジュールが必要なクラスおよびリソースを検出できるようにするには、コンソールを使用して `RAR` の「クラスパス」設定を確認します。クラスパス設定について詳しくは、管理コンソールの「リソース・アダプター」設定および `WebSphere` 関連のリソース・アダプターの設定を参照してください。

`Web` サーバーは、アプリケーションへの要求を送付するターゲットとして指定できます。各 `Web` サーバーのプラグイン構成ファイル (`plugin-cfg.xml`) は、`Web` サーバーを経由して送付されるアプリケーションに基づいて生成されます。`Web` サーバーによってアプリケーションにサービスを提供する場合は、**Ctrl** キーを使用して、アプリケーション・サーバーまたはクラスターを `Web` サーバーと一緒に選択します。これは、その `Web` サーバーのプラグイン構成ファイル `plugin-cfg.xml` を、`Web` サーバーを経由して送付されるアプリケーションに基づいて生成するためです。

6. アプリケーションがリソース参照を定義する場合は、「ステップ: リソース参照をリソースにマップ」で、リソース参照で定義される論理名を表すリソースの `JNDI` 名を指定します。アプリケーションで定義される各リソース参照は、「要約」パネルで「終了」をクリックする前に、`WebSphere Process Server` 構成で定義されているリソースにバインドする必要があります。
  - a. **オプション:** ログイン構成名とリソースの認証プロパティを指定します。
  - b. 「**OK**」をクリックして値を保管し、マッピング・ステップに戻ります。
7. アプリケーションで `Web` モジュールを使用する場合は、「ステップ: `Web` モジュールの仮想ホストをマップ」で、アプリケーションで定義されている `Web` モジュールにマップする仮想ホストをリストから選択します。

仮想ホスト定義で指定されたポート番号は、`Web` モジュールのサーブレットや `JSP` ファイルなどの成果物にアクセスするために使用する `URL` で使用されます。各 `Web` モジュールには、マップ先となる仮想ホストが必要です。必要と

されるすべての仮想ホストを指定しない場合、「要約」パネルで「終了」をクリックすると、検証エラーが表示されます。

8. アプリケーションのデプロイメント記述子でセキュリティー・ロールが定義されている場合は、「**ステップ: セキュリティー・ロールをユーザー/グループにマップ**」で、セキュリティー・ロールのそれぞれにマップするユーザーとグループを指定します。

「**ロール**」を選択して、すべてのロールまたは個々のロールを選択します。以下の選択項目から、ロールごとに 1 つセキュリティーの適用方法を選択します。

オプション	説明
全員	これは、セキュリティーなしと同等です。
全認証者	正当なユーザー名とパスワードで認証するユーザーが、このロールのメンバーです。
マップされたユーザー	個々のユーザーがこのロールのメンバーとしてリストされます。
マップされたグループ	グループは、ユーザーを追加するための最も便利な方法です。指定されたグループのメンバーすべてがこのロールのメンバーになります。

「**マップされたユーザー**」または「**マップされたグループ**」では、ユーザー・レジストリーから特定のユーザーまたはグループを選択し、以下のサブステップを実行します。

- a. ロールを選択し、「**ユーザーのルックアップ**」または「**グループのルックアップ**」をクリックします。
  - b. 表示される「**ユーザー/グループのルックアップ**」パネルで検索条件を入力し、ユーザー・レジストリーからユーザーまたはグループのリストを抽出します。
  - c. 表示される結果から、個々のユーザーまたはグループを選択します。
  - d. 「**OK**」をクリックして、選択したユーザーまたはグループを、「**ステップ: セキュリティー・ロールをユーザー/グループにマップ**」パネルで選択したロールにマップします。
9. アプリケーションのデプロイメント記述子で **Run As** ロールが定義されている場合は、「**ステップ: RunAs ロールをユーザーにマップ**」で、すべての **Run As** ロールに対して **Run As** ユーザー名とパスワードを指定します。 **Run As** ロールは、別のエンタープライズ Bean との対話中に、特定のロールとして実行する必要のあるエンタープライズ Bean によって使用されます。「**ロール**」を選択して、すべてのロールまたは個々のロールを選択します。ロールを選択した後、ユーザー名、パスワード、および確認パスワードの値を入力し、「**適用**」をクリックします。
  10. アプリケーションにリソース環境参照が含まれている場合は、「**ステップ: リソース環境参照をリソースにマップ**」で、リソース環境参照で定義される論理

名にマップするリソースの JNDI 名を指定します。各リソース環境参照にリソースが関連付けられていない場合は、「終了」をクリックした後に検証エラーが表示されます。

11. アプリケーションが **Run-As ID** をシステム ID として定義する場合は、「**ステップ: RunAs システムを RunAs ロールに置換**」において、オプションでその ID を *Run-As* ロールに変更し、指定された Run As ロールのユーザー名とパスワードを指定できます。システム ID を選択すると、WebSphere Application Server のセキュリティー・サーバー ID を使用して呼び出しが行われることが暗黙指定されます。この ID には多くの特権が付与されるので、慎重に使用してください。
12. アプリケーションに、Oracle データベースがバックエンド処理を実行しているリソースにマップするリソース参照がある場合は、「**ステップ: Oracle タイプ・プロバイダーの分離レベルの指定 (Step: Specify the isolation level for Oracle type provider)**」で、アプリケーションによって使用されるときにそれらのリソースで使用される分離レベルを指定または訂正します。Oracle データベースは、ReadCommitted と Serializable の分離レベルのみをサポートします。
13. アプリケーションが、プロモート済みのプロパティを持つモジュールを含む場合は、それらが現行値と共に表示されます。必要な場合は、この段階で現行値を変更できます。変更された値は、プロパティのタイプおよびその他の関連制約に照らして検証されます。
14. 「要約」パネルで、アプリケーション・モジュールをインストールするセル、ノード、およびサーバーを検証します。
  - a. 「セル/ノード/サーバー」の横で、「**ここをクリックする**」をクリックします。
  - b. 設定を確認します。
  - c. 「終了」をクリックします。

いくつかのメッセージが表示され、アプリケーション・ファイルが正常にインストールされたかどうかを示されます。

OutOfMemory 例外が発生し、ソース・アプリケーション・ファイルがインストールされない場合は、システムのメモリーが不足しているか、アプリケーションに含まれるモジュールが多すぎてサーバーに正常にインストールできない可能性があります。システム・メモリーの不足が例外の原因でない場合は、.ear ファイル内のモジュールを減らして再度アプリケーションをパッケージしてください。システム・メモリーの不足とモジュールの数が例外の原因ではない場合は、管理コンソールを実行しているアプリケーション・サーバーの Java 仮想マシン・ページで指定したオプションを確認します。その後、アプリケーション・ファイルを再度インストールしてください。

**Windows** インストール時には、特定のアプリケーション・ファイルが、構成セッションによって表されるディレクトリーに展開されます。構成が保管されると、これらのファイルは、WebSphere Application Server の構成リポジトリーに保管されます。Windows マシンでは、ファイル・パスの長さが 256 文字に制限されています。したがって、構成セッションまたは構成リポジトリーのアプリケーション・ファイルのパスが 256 文字より長くなると、アプリケーションのインストールが失敗します。FileNotFoundException 例外が発生し、「*path name too long*」という文字列を含むメッセ

ージが表示される場合もあります。この問題を解決するには、アプリケーション名とモジュール URI 名の長さを短くして、ファイル・パスの長さが短くなるようにします。その後、アプリケーション・ファイルを再度インストールしてください。

アプリケーション・ファイルを正常にインストールしたら、以下のアクションを実行します。

1. アプリケーションが必要とする共有ライブラリーすべてをアプリケーションに関連付けます。
2. 変更を構成に保管します。アプリケーションは管理構成に登録され、アプリケーション・ファイルは、デフォルトでターゲット・ディレクトリー `install_root/installedApps/cell_name` に、またはユーザーが指定するディレクトリーにコピーされます。Network Deployment プロファイルにインストールされると、デプロイメント・マネージャーの構成と個々のノードの構成が同期化されるときに、ファイルがリモート・ノードにコピーされます。
3. モジュールをサーバー・クラスターでデプロイする場合は、「エンタープライズ・アプリケーション」ページの「**ロールアウト更新 (Rollout Update)**」をクリックして、変更された構成をクラスターのメンバーすべてに伝搬します。ロールアウト更新では、クラスター・メンバーを含むノードの構成を順次更新します。

WebSphere Process Server がメディエーション・モジュールによって提供される機能を使用してアプリケーションとサービスを統合できるようにするには、デプロイされたモジュールを開始する必要があります。モジュールは、手動で開始するか、自動的に開始するように構成できます。また、他の方法でモジュールを管理して、例えば、モジュールの構成を変更したり、モジュールを停止または更新したり、あるいはそのアクティビティーを管理できます。

#### ANT タスクを使用したアプリケーションのデプロイ:

このトピックでは、ANT タスクを使用して、WebSphere Process Server に対するアプリケーションのデプロイメントを自動化する方法について説明します。ANT タスクを使用すると、複数のアプリケーションのデプロイメントを定義し、サーバーへのそれらのアプリケーションの不在デプロイを実行できます。

このタスクでは、以下のことが前提になります。

- デプロイされるアプリケーションの開発およびテストが完了している。
- 同じサーバー (1 つまたは複数) にアプリケーションをインストールする。
- ANT タスクに関する基本的な知識がある。
- デプロイメント・プロセスについて理解している。

アプリケーションの開発およびテストについて詳しくは、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。

WebSphere Application Server for z/OS インフォメーション・センターのリファレンス部分には、アプリケーション・プログラミング・インターフェースに関するセクションがあります。ANT タスクについては、パッケージ `com.ibm.websphere.ant.tasks` に説明があります。このトピックの目的に関連するタスクは、ServiceDeploy と InstallApplication です。

複数のアプリケーションを並行してインストールする必要がある場合は、デプロイメントの前に ANT タスクを作成します。こうすると、プロセスに手動で介入しなくても、ANT タスクによってアプリケーションをサーバーにデプロイおよびインストールできます。

1. デプロイするアプリケーションを識別します。
2. 各アプリケーションの JAR ファイルを作成します。
3. ターゲット・サーバーに JAR ファイルをコピーします。
4. 各サーバーの EAR ファイルを作成するために ServiceDeploy コマンドを実行する ANT タスクを作成します。
5. ステップ 4 で作成した各 EAR ファイルに対して、InstallApplication コマンドをアプリケーション・サーバー上で実行する ANT タスクを作成します。
6. ServiceDeploy ANT タスクを実行して、アプリケーションの EAR ファイルを作成します。
7. InstallApplication ANT タスクを実行して、ステップ 6 で作成した EAR ファイルをインストールします。

アプリケーションがターゲット・サーバーに正しくデプロイされます。

### アプリケーションの不在デプロイの例

次に、myBuildScript.xml ファイルに含まれる ANT タスクの例を示します。

```
<?xml version="1.0">

<project name="OwnTaskExample" default="main" basedir=".">
  <taskdef name="servicedeploy"
    classname="com.ibm.websphere.ant.tasks.ServiceDeployTask" />
  <target name="main" depends="main2">
    <servicedeploy scaModule="c:/synctest/SyncTargetJAR"
      ignoreErrors="true"
      outputApplication="c:/synctest/SyncTargetEAREAR"
      workingDirectory="c:/synctest"
      noJ2eeDeploy="true"
      cleanStagingModules="true"/>
  </target>
</project>
```

このステートメントは、ANT タスクの呼び出し方法を指定します。

```
${WAS}/bin/ws_ant -f myBuildScript.xml
```

**ヒント:** このファイルにさらに別のプロジェクト・ステートメントを追加して、複数のアプリケーションを無人デプロイすることができます。

管理コンソールを使用して、新しくインストールされたアプリケーションが正しく開始してワークフローを処理していることを確認します。

### メディエーション・モジュールの管理

WebSphere Process Server にデプロイされたメディエーション・モジュールをリストできます。各メディエーション・モジュールに関連する情報を表示し、一部のインポート・バインディングを変更することもできます。



サービス・アプリケーションをデプロイした後で、メディエーション・モジュールに関連する Service Component Architecture (SCA) モジュールを参照および管理できます。メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

#### メディエーション・モジュールの使用:

WebSphere Process Server にデプロイされたメディエーション・モジュールをリストできます。また、メディエーション・モジュールの開始と停止や、メディエーション・モジュールまたはそのアプリケーションの詳細も表示できます。

デプロイしたメディエーション・モジュールを表示するには、管理コンソールを使用して以下のステップを実行します。

**注:** メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

SCA モジュールをリストします。 ナビゲーション・ペインで「アプリケーション」 → 「SCA モジュール」を展開します。

コンテンツ・ペインに、WebSphere Process Server にデプロイされているメディエーション・モジュールが表示されます。メディエーション・モジュールに関連したアプリケーションが表示され、そのアプリケーションが実行中かどうかも表示されます。

#### メディエーション・モジュール詳細の表示:

WebSphere Process Server にデプロイされたメディエーション・モジュールに関する情報を表示できます。

デプロイされているメディエーション・モジュールに関する詳細を表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで「アプリケーション」 → 「SCA モジュール」を展開します。
2. 参照したい特定の SCA モジュールを選択します。 コンテンツ・ペインで、SCA モジュールをクリックします。

コンテンツ・ペインに SCA モジュール名および説明、ならびに関連するエンタープライズ・アプリケーションの名前が表示されます。また、インポートとエクスポートの拡張可能形式のリストおよびモジュール・プロパティ・リンクも表示されます。

#### メディエーション・モジュール用アプリケーション詳細の表示:

メディエーション・モジュールを WebSphere Process Server にデプロイするために使用するアプリケーションに関する詳細を表示できます。

メディエーション・モジュールをデプロイするのに使用するアプリケーションによって、一連の構成プロパティが定義されます。メディエーション・モジュールとその関連コンポーネントは、使用する際にこのプロパティの影響を受けます。このアプリケーションをインストールした際に、そのプロパティの値のすべてではないがほとんどを指定しています。

アプリケーションのインストール後にプロパティを検討して、必要があればいくつかの値を変更します。

メディエーション・モジュールのデプロイに使用するアプリケーションの詳細を表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで「アプリケーション」 → 「SCA モジュール」を展開します。
2. 参照したい SCA モジュールのアプリケーション名をクリックします。

アプリケーション名は「アプリケーション」というラベルの列にリストされています。

コンテンツ・ペインに「アプリケーション詳細」ページが表示され、アプリケーションの構成プロパティ、およびローカル・トポロジー (該当する場合) が表示されます。このページからプロパティ値を変更して、アプリケーションの構成を検討し、変更するための追加コンソール・ページにリンクできます。

『アプリケーションの構成 (Configuring an application)』で説明されているように、アプリケーションの構成プロパティを検討し、必要に応じて変更できます。

#### **メディエーション・モジュールの開始と停止:**

実行していない (停止状態の) メディエーション・モジュールを開始したり、実行中の (開始済み状態の) メディエーション・モジュールを停止したりできます。メディエーション・モジュールの状態を変更するには、モジュールのデプロイに使用するアプリケーションを開始または停止します。

メディエーション・モジュールに使用するアプリケーションを開始または停止するには、251 ページの『実動サーバーへのモジュールのインストール』で説明しているように、事前にそのメディエーション・モジュールを WebSphere Process Server にデプロイしておく必要があります。デプロイすることにより、アプリケーションがアプリケーション・サーバー (またはサーバー・クラスター) にインストールされます。

メディエーション・モジュールおよび関わりのあるコンポーネントのサービスを使用するには、関連するアプリケーションを開始します。デフォルトでは、サーバーの始動時にアプリケーションが自動的に開始します。

以下の機能を使用して、アプリケーションを手動で開始または停止できます。

- 管理コンソール
- wsadmin startApplication コマンドおよび stopApplication コマンド
- ApplicationManager または AppManagement MBeans を使用する Java プログラム

メディエーション・モジュールを開始または停止するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで「アプリケーション」 → 「SCA モジュール」を展開します。
2. 開始または停止する SCA モジュールのチェック・ボックスを選択します。

3. 「開始」ボタンまたは「停止」ボタンをクリックします。

オプション	説明
開始	アプリケーションを実行して、アプリケーションの状態を「開始済み」に変更します。アプリケーションをデプロイしているサーバーのうち、一部のサーバーが稼働していない場合は、状態が「一部開始済み (partially started)」に変更されます。
停止	アプリケーションの処理を停止して、アプリケーションの状態を「停止済み」に変更します。

アプリケーションを再始動するには、再始動するアプリケーションを選択して「停止」をクリックした後、「開始」をクリックします。

アプリケーションの状態が変更され、アプリケーションが開始済みまたは停止済みであることを示すメッセージがページの上部に表示されます。

アプリケーションが存在するサーバーの始動時に自動的にアプリケーションを開始するかどうかを変更できます。WebSphere アプリケーションの開始および停止の詳細については、『アプリケーションの始動と停止』を参照してください。

#### メディエーション・モジュール・プロパティの表示:

WebSphere Process Server にデプロイされたメディエーション・モジュールのプロパティを表示できます。

サービス・アプリケーションを実行する前に、適切なプロパティ値が設定されているかどうかチェックできます。

デプロイしたメディエーション・モジュールのプロパティを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。ナビゲーション・ペインで「アプリケーション」→「SCA モジュール」を展開します。
2. SCA モジュール を選択します。コンテンツ・ペインで、SCA モジュールをクリックします。
3. SCA モジュールのプロパティをリストします。コンテンツ・ペインで、「追加プロパティ」の下にある「モジュール・プロパティ」をクリックします。

コンテンツ・ペインに、SCA モジュールのプロパティが表示されます。これらのプロパティは更新できます。テーブル内にプロパティが表示され、プロパティ名、タイプ、および値が示されます。管理コンソールから更新できるのは、プロパティ値のみです。プロパティ名とタイプを更新するには、WebSphere Integration Developer を使用します。更新できるプロパティが存在しない場合は、メッセージが表示されます。

#### メディエーション・モジュール・プロパティの変更:

一部のメディエーション・モジュール・プロパティの値を変更できます。

ランタイム環境の変更に合わせて、プロパティ値の変更も必要なことがあります。

メディエーション・モジュール・プロパティの値を変更するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで「アプリケーション」→「SCA モジュール」を展開します。
2. SCA モジュールを選択します。 コンテンツ・ペインで、SCA モジュールをクリックします。
3. SCA モジュール・プロパティを表示します。 コンテンツ・ペインで、「追加プロパティ」の下にある「モジュール・プロパティ」を選択します。これにより、更新できるモジュール・プロパティが表示されます。プロパティ名、タイプ、および値が表示されます。ただし、管理コンソールから更新できるのは、プロパティ値のみです。プロパティ名とタイプを変更するには WebSphere Integration Developer を使用します。
4. プロパティ値を選択します。「プロパティ」テーブルからプロパティ値をクリックします。
5. プロパティ値を変更します。 プロパティ・タイプに適合する値を入力します。
6. 変更を保管します。「OK」をクリックします。次に、変更をマスター構成に保管します。

プロパティ値が変更されました。メディエーション・フローを新たに呼び出すと、プロパティの変更内容が即時に有効になります (プロパティの変更をデプロイメント・マネージャー・セルで行った場合を除く)。デプロイメント・マネージャー・セル内で変更した場合は、セル内のすべてのノードが同期化された後に変更内容が有効になります。処理中にメディエーション・フローを呼び出した場合は、前の値が引き続き使用されます。

注: プロパティ値ではなく、プロパティ名やタイプを変更したい場合は、WebSphere Integration Developer を使用してください。

#### インポートの使用:

WebSphere Process Server にデプロイされたメディエーション・モジュールのインポートをリストできます。また、インポート・インターフェースを表示したり、SCA インポート・バインディングおよび Web サービス (WS) インポート・バインディングを変更したりすることができます。

デプロイしたメディエーション・モジュールのインポートをリストするには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、「アプリケーション」→「SCA モジュール」を展開します。
2. 参照したい特定の SCA モジュールを選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「インポート」を展開します。

コンテンツ・ペインにインポートのリストが表示されます。インポートが存在しない場合は、情報メッセージが表示されます。

#### インポート・インターフェースの表示:

WebSphere Process Server にデプロイされたメディエーション・モジュールのインポート・インターフェースを表示できます。

デプロイしたメディエーション・モジュールのインポート・インターフェースを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、「アプリケーション」→「SCA モジュール」 を展開します。
2. SCA モジュールを選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「インポート」を展開します。
4. インポート詳細を表示します。 コンテンツ・ペインでモジュール・コンポーネントの下にある対象のインポートを展開します。
5. インポート・インターフェースを表示します。「インターフェース」を展開します。
6. インターフェースを選択します。

コンテンツ・ペインに WSDL (Web サービス記述言語) インターフェースが表示されます。

#### インポート・バインディングの表示:

メディエーション・モジュールを WebSphere Process Server にデプロイした後、一部のタイプのインポート・バインディングを表示できます。

メディエーション・モジュールのインポート・バインディングを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、「アプリケーション」→「SCA モジュール」 を展開します。
2. SCA モジュールを選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「インポート」を展開します。
4. インポート詳細を表示します。 コンテンツ・ペインでモジュール・コンポーネントの下にある対象のインポートを展開します。
5. インポート・バインディングを表示します。「バインディング」を展開します。 バインディングのタイプが Web サービス、SCA、JMS、またはアダプターのいずれかである場合、そのバインディング・タイプが表示されます。その他のバインディング・タイプは表示されません。
6. バインディングを選択します。 バインディングが Web サービス・インポート・バインディング、または SCA インポート・バインディングである場合は、バインディングを選択して、インポート・バインディングの詳細を取得できます。

コンテンツ・ペインにインポート・バインディングの詳細が表示されます。表示される詳細は、バインディングのタイプによって異なります。

バインディングが Web サービス・インポート・バインディングである場合は、サービス名、ポート名、およびエンドポイント URL が表示されます。

バインディングが SCA インポート・バインディングである場合は、表示される詳細に現在のターゲット・モジュールが含まれます。また、現在のターゲット・エクスポートとそのインターフェースも表示されます。ドロップダウン・メニューに、WebSphere Process Server にデプロイされているその他のターゲット・モジュールが表示されます。別のターゲット・モジュールを選択すると、ターゲット・エクスポートおよびエクスポート・インターフェースのリストが変わります。別のターゲット・エクスポートを選択すると、エクスポート・インターフェースのリストが変わります。

### SCA インポート・バインディングの変更:

場合によって、WebSphere Process Server のメディエーション・モジュールのインポート・バインディングを変更できます。インポート・バインディングを変更できるのは、それが SCA インポート・バインディングまたは Web サービス・インポート・バインディングである場合のみで、その他のタイプのバインディングの場合は変更できません。

メディエーション・モジュールの SCA インポート・バインディングを変更するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、「アプリケーション」→「SCA モジュール」 を展開します。
2. SCA モジュールを選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「インポート」を展開します。
4. インポート詳細を表示します。 コンテンツ・ペインでモジュール・コンポーネントの下にある対象のインポートを展開します。
5. インポート・バインディングを表示します。「バインディング」を展開します。
6. SCA インポート・バインディングを選択します。 SCA インポート・バインディングは、ID [SCA] を使用して示されます。
7. 新規ターゲットの SCA モジュールを選択します。「ターゲット」ドロップダウン・メニューからモジュールを選択します。異なる SCA モジュールを選択すると、表示されるエクスポートおよびエクスポート・インターフェースが変更されます。
8. エクスポートを選択します。「エクスポート」ドロップダウン・メニューからエクスポートを選択します。
9. 変更内容をマスター構成に保管します。

選択した SCA モジュールのインポート用に、SCA インポート・バインディングが変更されました。

WebSphere Process Server では、エクスポート・インターフェースに対応しない各インポート・インターフェースに対して警告が出されます。WebSphere Process Server では、インポートとエクスポートの WSDL (Web サービス記述言語) ポート・タイプ名が比較されます。ポート・タイプ名が異なる場合は警告が出されますが、この警告は無視してかまいません。ただし、ポート・タイプ名が一致する場合、WebSphere Process Server では、行われた操作を等価とみなし警告は出されません。

### Web サービス・インポート・バインディングの変更:

場合によって、WebSphere Process Server のメディエーション・モジュールのインポート・バインディングを変更できます。インポート・バインディングを変更できるのは、それが SCA インポート・バインディングまたは Web サービス・インポート・バインディングである場合のみで、その他のタイプのバインディングの場合は変更できません。

メディエーション・モジュールの Web サービス・インポート・バインディングを変更するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、「アプリケーション」→「SCA モジュール」 を展開します。
2. SCA モジュールを選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「インポート」を展開します。
4. インポート詳細を表示します。 コンテンツ・ペインでモジュール・コンポーネントの下にある対象のインポートを展開します。
5. インポート・バインディングを表示します。「バインディング」を展開します。
6. Web サービス・インポート・バインディングを選択します。 Web サービス・インポート・バインディングは、ID [Web service] によって示されます。
7. エンドポイント URL を変更します。 エンドポイントが正しい形式の URL であることを確認します。
8. 変更内容をマスター構成に保管します。

選択した SCA モジュールのインポート用に Web サービス・インポート・バインディングが変更されました。

マスター構成を更新し、SCA モジュールを再開した後で、変更が有効になります。

### エクスポートの使用:

WebSphere Process Server にデプロイされたメディエーション・モジュールのエクスポートをリストできます。エクスポート・インターフェースおよびエクスポート・バインディングも表示できます。

デプロイしたメディエーション・モジュールのエクスポートをリストするには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、「アプリケーション」→「SCA モジュール」 を展開します。
2. 参照したい特定の SCA モジュールを選択します。
3. SCA モジュールのエクスポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「エクスポート」を展開します。

コンテンツ・ペインにエクスポートのリストが表示されます。エクスポートが存在しない場合は、情報メッセージが表示されます。

### エクスポート・インターフェースの表示:

WebSphere Process Server にデプロイされたメディエーション・モジュールのエクスポート・インターフェースを表示できます。

デプロイされているメディエーション・モジュールのエクスポート・インターフェースを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで「アプリケーション」 → 「SCA モジュール」を展開します。
2. SCA モジュール を選択します。
3. SCA モジュールのエクスポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「エクスポート」を展開します。
4. エクスポートの詳細を表示します。 コンテンツ・ペインで、モジュール・コンポーネントの下にある対象のエクスポートを展開します。
5. エクスポート・インターフェースを表示します。「インターフェース」を展開します。
6. インターフェースを選択します。

コンテンツ・ペインに WSDL (Web サービス記述言語) インターフェースが表示されます。

#### **エクスポート・バインディングの表示:**

メディエーション・モジュールを WebSphere Process Server にデプロイした後、一部のタイプのエクスポート・バインディングを表示できます。

メディエーション・モジュールのエクスポート・バインディングを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、「アプリケーション」 → 「SCA モジュール」 を展開します。
2. SCA モジュール を選択します。
3. SCA モジュールのエクスポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「エクスポート」を展開します。
4. エクスポートの詳細を表示します。 コンテンツ・ペインで、モジュール・コンポーネントの下にある対象のエクスポートを展開します。
5. エクスポート・バインディングを表示します。「バインディング」を展開します。 バインディング・タイプが Web サービス、SCA、JMS、またはアダプターのいずれかである場合、そのバインディング・タイプは大括弧に囲まれて表示されます。バインディング・タイプが MQ または MQ JMS である場合、空の大括弧が表示されます。エクスポートにバインディングが指定されていない場合、バインディングが存在しないことを示すメッセージが管理コンソールに表示されます。この場合、ランタイムはバインディングがデフォルトの SCA タイプであるものと見なします。
6. バインディングを選択します。 バインディングが Web サービス・エクスポート・バインディングである場合、バインディングを選択して、エクスポート・バインディングの詳細を取得できます。



コンテンツ・ペインにエクスポート・バインディングの詳細が表示されます。Web サービス・エクスポート・バインディングの場合は、サービスおよびポート名が表示されます。

## WSRR へのアクセスの管理

管理コンソールを使用して、WebSphere Service Registry and Repository (WSRR) のすべてのアクセス定義を作成、構成、表示できます。

WSRR のインストールおよび構成は、WSRR インフォメーション・センター ([http://publib.boulder.ibm.com/infocenter/sr/v6r0/topic/com.ibm.sr.doc/cwsr\\_overview\\_overview.html](http://publib.boulder.ibm.com/infocenter/sr/v6r0/topic/com.ibm.sr.doc/cwsr_overview_overview.html)) にある説明に従って行ってください。

WebSphere Process Server は、WSRR 製品をサポートしているため、サービスの保管および検索が可能です。WSRR は、エンタープライズ・アプリケーションとしてインストールされ、Web サービス・インターフェースとなります。このインターフェースを使用すると、エンドポイント・ルックアップ・メディエーション・プリミティブをレジストリー・インスタンスに (メディエーション・フローの一環として) 接続し、サービスを検索して呼び出すことができます。WSRR 定義とその接続プロパティーが、レジストリー・インスタンスに接続し、呼び出す Web サービスを検索するための手段となります。すべての WSRR 定義のリストを表示するには、管理コンソールを使用して以下のステップを実行します。

ナビゲーション・ペインで、「サービス統合 (Service integration)」 → 「WSRR 定義」を展開します。

これにより、コンテンツ・ペインにすべての WSRR 定義のリストが表示されます。各定義には、WSRR インスタンスを識別し、これにアクセスする方法を指定するプロパティー・セットが収められます。

WSRR 定義をクリックして、その定義の一般的なプロパティーを表示します。コンテンツ・ペインから、詳細な接続プロパティーを表示することもできます。

### 新規 WSRR 定義の作成:

管理コンソールを使用して、新しい WSRR 定義を作成できます。

各定義には、WSRR インスタンスを識別し、これにアクセスする方法を指定するプロパティー・セットが収められます。新しい WSRR 定義を作成するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サービス統合 (Service integration)」 → 「WSRR 定義」を展開します。コンテンツ・ペインに「WSRR 定義」ページが表示されます。このページにすべての WSRR 定義のリストが表示されます。
2. 「新規」ボタンをクリックします。「WSRR 定義の構成 (WSRR definitions configuration)」ページが表示されます。
3. 以下のプロパティー・フィールドに入力します。

#### WSRR 定義名

これは必須のフィールドであり、セル内で固有でなければなりません。この WSRR 定義の管理名です。

**説明** 必要であれば、定義の説明を入力できます。

### デフォルトの WSRR 定義

これは、この定義がデフォルトであるかどうかを示します。この定義が作成した最初の定義である場合は、自動的にデフォルトとして設定されます。このページからデフォルトを更新することはできません。

### キャッシュのタイムアウト

それ以上経過したら WSRR キャッシュ内の照会結果の有効期限が切れ、リフレッシュされる時間 (秒単位)。このフィールドは、デフォルト値からの変更が可能です。値 0 (ゼロ) を指定した場合、照会結果はキャッシュされません。

### 接続タイプ

現在選択できるのは、「**Web サービス**」のみです。

4. 「適用」をクリックして、これらのプロパティを保管します。
5. コンテンツ・ペインの「追加プロパティ」の下で、「**接続プロパティ**」をクリックします。「**接続プロパティの構成 (Connection properties configuration)**」ページが表示されます。
6. 以下のプロパティ・フィールドに入力します。

### 接続タイプ

これは、レジストリー定義が作成されるときに設定されます。変更はできません。

### レジストリー URL

WSRR インスタンスへの接続に使用される URL。デフォルトは `http://localhost:9080/WSRRCoreSDO/services/WSRRCoreSDOPort` です。

### 認証別名

WSRR インスタンスでの認証に使用される別名。

7. 「適用」をクリックして、これらのプロパティを保管します。
8. 「保管」をクリックして、マスター構成に変更を適用します。「保管」をクリックします。

新しい WSRR 定義が作成されます。

### WSRR 定義の削除:

管理コンソールを使用して、WSRR 定義を削除できます。

WSRR 定義を削除するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「**サービス統合 (Service integration)**」 → 「**WSRR 定義**」を展開します。コンテンツ・ペインに「**WSRR 定義**」ページが表示されます。このページにすべての WSRR 定義のリストが表示されます。
2. 削除する WSRR 定義の横にあるチェック・ボックスを選択します。「**削除**」をクリックします。定義がコレクション・パネルからなくなります。
3. 「保管」をクリックして、マスター構成に変更を適用します。「保管」をクリックします。

WSRR 定義は削除されます。

注: デフォルトの WSRR 定義は削除できません (唯一の定義である場合を除く)。現在のデフォルトの定義を削除したい場合は、最初に別の定義をデフォルトとして設定してください。

### デフォルトの WSRR 定義の設定:

管理コンソールから、WSRR 定義をデフォルトの定義として設定できます。

WSRR 定義をデフォルトとして設定するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サービス統合 (Service integration)」 → 「WSRR 定義」を展開します。コンテンツ・ペインに「WSRR 定義」ページが表示されます。このページにすべての WSRR 定義のリストが表示されます。
2. デフォルトとして設定する WSRR 定義の横にあるチェック・ボックスを選択します。「デフォルトとして設定」をクリックします。選択した WSRR 定義の「デフォルト」列に「はい」が表示されます。以前に別の WSRR 定義をデフォルトとして設定していた場合は、その WSRR 定義の「デフォルト」列に「いいえ」が表示されます。
3. 「保管」をクリックして、マスター構成に変更を適用します。「保管」をクリックします。

選択した WSRR 定義がデフォルトの定義として設定されます。

注: 「デフォルト」列は、デフォルトの定義が常にリストの一番上または一番下に表示されるようにソートできます。

## SCA 要求のスループット管理

WebSphere Process Server にデプロイされた各 SCA モジュールでは、処理中の要求はキュー・ポイントおよびメッセージング・エンジンのデータ・ストアに保持されます。SCA 要求のデータを表示し、必要に応じて SCA 要求のスループット管理アクションを実行できます。

SCA モジュールをエンタープライズ・サービス・バスで実行している場合は、通常、そのエンタープライズ・サービス・バスを介して要求が送信されるため、管理の必要はありません。必要に応じて、要求のスループットの確認、要求内容の確認、または何らかの問題が発生した場合は要求の削除を行うことができます。要求のスループット全体のモニター、要求の信頼性設定の変更などのアクションを実行しなければならない場合もあります。

要求は、基盤となる WebSphere Application Server のサービス統合テクノロジーにより、メッセージとして処理されます。そのため、要求管理アクションは、サービス統合メッセージを処理する WebSphere Application Server タスクを使用して管理されます。

このトピックでは、必要に応じて実行するメインタスクの概要を説明し、詳細を参照するための WebSphere Application Server タスクへのリンクが記載されています。

- メッセージ・ポイントにあるメッセージのリスト表示 (Listing messages on a message point)

処理中の SCA 要求は SCA.SYSTEM.bus のキュー・ポイントに保持されます。SCA 要求は、SCA モジュールのコンポーネントのキュー宛先、またはキュー・ポイントをホストするメッセージング・エンジンを使用してリストできます。例:

「サービス統合 (Service integration)」 → 「バス」 →  
「SCA.SYSTEM.localhostNode01Cell.Bus」 → 「宛先」 →  
「StockQuoteService\_Export」 → 「キュー・ポイント (Queue points)」 →  
「StockQuoteService\_Export@localhostNode01.server1-SCA.SYSTEM.  
localhostNode01Cell.Bus」 → 「ランタイム」 → 「メッセージ」

- メッセージ・ポイントでロックされたメッセージの解決 (Resolving locked messages on a message point)

問題が発生した場合は、SCA 要求が処理中のキュー・ポイントにロックされたままになる場合があります。メッセージの **State** プロパティを表示して、要求がロックされているかどうかを確認できます。必要があればアクションを実行して問題を解決します。

- メッセージ・ポイントにあるメッセージの削除 (Deleting messages on a message point)

例外的に、選択したバス宛先またはメッセージング・エンジンのメッセージ・ポイントに存在する 1 つ以上のメッセージを削除しなければならないことがあります。通常は、メッセージ・ポイントにあるメッセージを削除する必要はありません。このタスクはトラブルシューティング手順の一部として実行するものです。

- メッセージング・エンジンのデータ・ストア内のデータの表示 (Viewing data in the data store for a messaging engine)

メッセージング・エンジンでは、要求を揮発性 (非永続) データおよび耐久 (永続) データとしてデータ・ストアに保持します。

データ・ストアのデータベース・ツールを使用して、メッセージング・エンジンのデータ・ストア内の要求データを表示できます。例えば、メッセージング・エンジンがデフォルトの Cloudscape データベースを使用している場合は、CloudView ツールを使用して要求データを表示できます。

- 宛先に対するメッセージ信頼性の変更 (Changing message reliability for a destination)

要求メッセージには、メッセージ配信の信頼性を定めるサービス品質属性が指定されています。配信保証の要件およびシステム性能に応じて、信頼性を選択できます。管理者がバス宛先上で信頼性設定を指定するか、個々の作成者により (通常、API 呼び出しによるアプリケーション制御のもとで) 信頼性を指定できます。

#### データ・ストア内のデータの表示:

メッセージング・エンジンはメッセージ、トランザクション状態、通信チャンネル状態など、非永続データと永続データの両方をデータ・ストアに保持します。データベース・ツールを使用して、メッセージング・エンジンに関するデータ・ストア内のデータを表示できます。

CloudView ツールを使用して、メッセージング・エンジンに関する Cloudscape データ・ストア内のデータを表示するには、その前にメッセージング・エンジンを停止する必要があります。

メッセージング・エンジンが停止すると、非永続データは失われます。データは、制御されながら失われる場合もあれば、制御されずに失われる場合もあります。永続データは、サーバーの再始動後に使用可能になります。

場合によっては、メッセージング・エンジンにより処理中のメッセージを検査するために、データ・ストア内のデータを表示する必要が生じることがあります。

データ・ストア用のデータベース・ツールを使用して、メッセージング・エンジンに関するデータ・ストア内のデータを表示できます。例えば、メッセージング・エンジンがデフォルトの Cloudscape データベースを使用している場合は、CloudView ツールを使用して要求メッセージを表示できます。

1. CloudView ツールを開始します。例えば、Windows では次のサブステップを実行します。
  - a. コマンド・ウィンドウを開きます。
  - b. ディレクトリーを `install_root/cloudscape/bin/embedded` に変更します。
  - c. `cview` と入力します。
2. メッセージング・エンジンのデータ・ストアを開きます。CloudView ツールを使用して、次のサブステップを実行します。
  - a. 「ファイル (File)」 → 「開く (Open)」をクリックします。
  - b. データベース・ファイルを参照して選択します。

メッセージング・エンジンの場合、データベースはディレクトリー `install_root/profiles/profile_name/databases/com.ibm.ws.sib` にメッセージング・エンジンの名前で格納されます。例えば、Windows でのデフォルトのスタンドアロン・プロファイルの場合、メッセージング・エンジン `localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus` (SCA.SYSTEM バス上のサーバー 1) のデータベース・ファイルのディレクトリーは次のとおりです。

`install_root/profiles/default/databases/com.ibm.ws.sib/localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus`

- c. 「開く (Open)」をクリックします。
3. CloudView 制御を使用してデータを表示します。
    - a. ナビゲーション・ペインのテーブルを展開します。
    - b. テーブル名をクリックします。
    - c. コンテンツ・ペインで「データ」タブをクリックします。
  4. オプション: CloudView ツールの使用方法に関するヘルプを表示します。

CloudView の使用方法に関するヘルプを表示するには、「ヘルプ」ボタンをクリックするか、メニュー・バーのオプション「ヘルプ」 → 「Cview ヘルプ... (Cview help...)」をクリックします。

バス宛先のメッセージ信頼性の変更:

メッセージには、メッセージ配信の信頼性を指定するサービス品質属性が備わっています。配信保証の要件およびシステム性能に応じて、信頼性を選択できます。

管理者がバス宛先上で信頼性設定を指定するか、個々の作成者により (通常、API 呼び出しによるアプリケーション制御のもとで) 信頼性を指定できます。管理者は、宛先に設定されたデフォルトの信頼性を作成者がオーバーライド可能かどうか、および接続している作成者が要求できる信頼性の上限を指定できます。

宛先のメッセージの信頼性設定を参照または変更するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、以下をクリックします 「サービス統合」 → 「バス」
2. コンテンツ・ペインで、宛先が存在するバスの名前をクリックします。
3. 「宛先」をクリックします。
4. 宛先名をクリックします。宛先の詳細ページが表示されます。
5. 信頼性プロパティを調べます。次のプロパティによって、宛先のメッセージの信頼性が制御されます。

#### **Default reliability**

製作者が信頼性を明示的に設定していないときに、この宛先に生成されるメッセージに割り当てる信頼性。

#### **Maximum reliability**

この宛先が受け入れるメッセージの最大の信頼性。

これらのプロパティには、以下にリストされた値を指定できます。

#### **Best effort nonpersistent**

メッセージング・エンジンが停止または障害が発生すると、メッセージが破棄されます。また、メッセージ送信用の接続が使用不可になった場合、およびシステム・リソースに余裕のない場合も、メッセージが破棄されることがあります。

#### **Express nonpersistent**

メッセージング・エンジンが停止または障害が発生すると、メッセージが破棄されます。また、メッセージ送信用の接続が使用不可になった場合も、メッセージが破棄されることがあります。

#### **Reliable nonpersistent**

メッセージング・エンジンが停止または障害が発生すると、メッセージが破棄されます。

#### **Reliable persistent**

メッセージング・エンジンに障害が発生すると、メッセージが破棄される場合があります。

#### **Assured persistent**

メッセージは破棄されません。

メッセージの信頼性を制御するこれらのプロパティの使用については、『メッセージ信頼性レベル (Message reliability levels)』を参照してください。

6. デフォルトの信頼性設定を作成者がオーバーライドできるようにするかどうかを検討します。

#### デフォルトの信頼性を作成者がオーバーライドできるようにする

このオプションを選択すると、宛先で設定されたデフォルトの信頼性を製作者がオーバーライドできるようになります。

7. **オプション:** 必要に応じて、宛先プロパティを変更します。

『バス宛先の構成 (Configuring bus destinations)』の説明に従って、必要に応じてその他のプロパティを設定することにより、宛先をより詳細に構成できます。

8. 「**OK**」をクリックします。
9. 変更内容をマスター構成に保管します。

## メディエーション・モジュールのその他の用途

WebSphere 管理コンソールを使用して、メディエーション・モジュールそのものを管理する以外にも、メディエーション・モジュールで使用されるリソースの管理、メディエーション・モジュールが含まれるアプリケーションの管理が可能です。また、これらの作業をコマンドを使用して実行することもできます。

メディエーション・モジュールを管理するために頻繁に行うその他のタスクについては、260 ページの『メディエーション・モジュールの管理』で説明しています。

これらのトピックには、WebSphere Application Server のトピックへのリンクが用意されているので、メディエーション・モジュールのデプロイに使用するアプリケーション関連のタスクについては、これらのトピックを参照してください。

メディエーション・モジュールのその他の用途については、以下のサブトピックを参照してください。

### メディエーション・モジュールのリソースの管理:

メディエーション・モジュールは、WebSphere Application Server のサービス統合テクノロジーによって提供されるリソースを利用します。またメディエーション・モジュールは、Java Message Service (JMS) および Common Event Infrastructure によって提供されるリソースを含む、広範なリソースも利用します。メディエーション・モジュールのリソースの管理には、WebSphere 管理コンソール、コマンド、およびスクリプト・ツールを使用できます。

メディエーション・モジュールのリソースの管理について詳しくは、関連するトピックを参照してください。

### サービス統合テクノロジー

バス宛先などのサービス統合リソースにより、メディエーション・モジュールでサービス統合テクノロジーを使用できます。メディエーション・モジュールによって利用される SCA ランタイムは、コンポーネントとモジュール間の非同期対話をサポートする堅固なインフラストラクチャーとして、キュー宛先を使用します。メディエーション・モジュールを WebSphere Process Server にインストールすると、モジュールによって使用される宛先が、SCA.SYSTEM.bus のメンバーで定義されます。これらのバス宛先は、非同期対話を使用するメディエーション・モジュールのコンポーネント用に処理されるメッセージを保持するために使用します。

**Queue** `sca/module_name`

モジュール `module_name` に送信される非同期要求をバッファーに入れるために使用する宛先です。

**Queue** `sca/module_name/export/export_name`

モジュール・エクスポート `export_name` に送付される非同期要求をバッファーに入れるために使用する宛先です。

**Queue** `sca/module_name/exportlink/export_name`

非同期要求をモジュールに送信するエクスポートによって使用される宛先です。要求は、そのエクスポートにリンクされたコンポーネント・ターゲットに送付されます。

**Queue** `sca/module_name/component/component_name`

コンポーネント `component_name` に送信される非同期要求をバッファーに入れるために使用する宛先です。

**Queue** `sca/module_name/component/component_name/source/source_name`

コンポーネントのソース・インポート `source_name` に送付される非同期要求をバッファーに入れるために使用する宛先です。

**Queue** `sca/module_name/component/component_name/target/target_name`

コンポーネントのターゲット・エクスポート `target_name` に送付される非同期要求をバッファーに入れるために使用する宛先です。

**Queue** `sca/module_name/import/import_name`

インポート `import_name` に送信される非同期要求をバッファーに入れるために使用する宛先です。

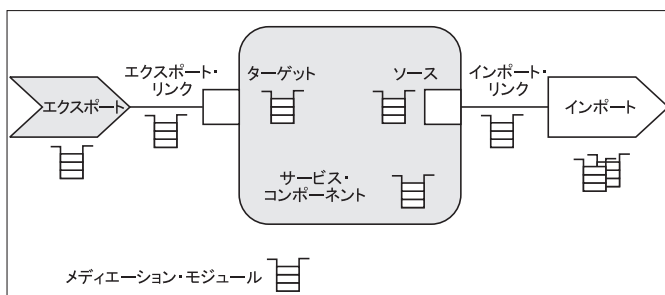
**Queue** `sca/module_name/importlink/import_name`

モジュールからの非同期要求を送信するインポートによって使用される宛先です。要求は、そのインポートにリンクされたモジュール・エクスポートに送付されます。

**Queue** `sca/module_name/import/sca/dynamic/import/scaimport` [for SCA binding]

**Queue** `sca/module_name/import/sca/dynamic/import/wsimport` [for Web service binding]

**Queue** `sca/contextStore/module_name`



宛先ごとに、キュー・ポイントも作成され、バス・メンバーのメッセージング・エンジンで定義されます。



メディエーション・モジュールは、これらのリソースを管理しなくても、デプロイおよび使用できます。ただし、リソースの構成を調整したり (例えば、使用されるサービスの最大メッセージ品質を変更する場合)、トラブルシューティングでメッセージを見つける際にリソースを使用することもできます。

### **Java Message Service (JMS)**

JMS リソースによりメディエーション・モジュールでは、Java Message Service (JMS) プログラミング・インターフェースに基づく通信方式として、非同期メッセージングを使用できます。メディエーション・モジュールの場合は、WebSphere Application Server のデフォルトのメッセージング・プロバイダーまたは MQ JMS バインディングを介した WebSphere MQ によって JMS がサポートされます。Java Message Service の使用状態を管理するため、次のリソースを管理できます。

#### **JMS 接続ファクトリー**

JMS 接続ファクトリーは、Point-to-Point メッセージングとパブリッシュ/サブスクライブ・メッセージングの両方において、対応する JMS 宛先の JMS プロバイダーとの接続を作成するために使用します。接続ファクトリー管理オブジェクトを使用して、プロバイダーの JMS 接続ファクトリーを管理します。

#### **JMS キュー**

JMS キューは、Point-to-Point メッセージングの宛先として使用されます。JMS キュー宛先管理オブジェクトを使用して、プロバイダーの JMS キューを管理します。

#### **JMS トピック**

JMS トピックは、パブリッシュ/サブスクライブ・メッセージングの宛先として使用されます。トピック宛先管理オブジェクトを使用して、プロバイダーの JMS トピックを管理します。

#### **JMS アクティベーション・スペック**

JMS アクティベーション・スペックは、1 つ以上のメッセージ駆動型 Bean に関連付けられており、これらの Bean がメッセージを受信するのに必要な構成を提供します。

#### **JMS リスナー・ポート**

JMS リスナー・ポートは、接続ファクトリー、宛先、およびメッセージ駆動型 Bean の間の関連を定義します。これにより、ポートに関連付けられているデプロイ済みのメッセージ駆動型 Bean が、宛先からメッセージを取得できます。

### **Common Event Infrastructure (CEI)**

CEI リソースにより、メディエーション・モジュールでは、イベント・データを管理するための標準の形式と機構を使用できます。Common Event Infrastructure の使用を管理するため、次のリソースを管理できます。

#### **データ・ストア・プロファイル**

デフォルトのデータ・ストアが使用するプロパティを定義します。デフォルトのデータ・ストアは、Common Event Infrastructure によって提供されるデータ・ストアです。

### エミッター・ファクトリー・プロファイル

このプロファイルは、イベント・エミッターのオプションを定義します。

### イベント・バス伝送プロファイル

このプロファイルは、イベント・バスへの EJB エントリーを定義します。

### イベント・グループ・プロファイル

このプロファイルは、セレクター式で判別されるイベントのリストを定義します。JMS キューおよび JMS トピックを各イベント・グループに関連付けることができます。イベント・サーバー配布サービスが使用可能で、イベントがイベント・グループに一致する場合、イベントはそのイベント・グループ用に構成されたトピックまたはキューに配布されます。

### イベント・サーバー・プロファイル

このプロファイルは、イベント・サーバーのプロパティを定義します。

### フィルター・ファクトリー・プロファイル

このプロファイルは、フィルターのプロパティを定義します。フィルターはフィルター構成ストリングを使用して、イベントがバスに渡されるかどうかを判別します。

### JMS 伝送プロファイル

イベント・テーブルを含むデータベース・スキーマ。

### 非同期メッセージングの使用:

これらのトピックでは、WebSphere Process Server で非同期メッセージングを使用して、エンタープライズ・アプリケーションが JMS リソースおよびメッセージ駆動型 Bean を使用できるようにする方法を説明します。

WebSphere Process Server では、Java Message Service (JMS) プログラミング・インターフェースに基づく通信方式として、非同期メッセージングをサポートしています。この JMS サポートは、エンタープライズ・アプリケーションで使用できるように構成した、1 つ以上の JMS プロバイダーおよび関連したサービスとリソースによって提供されます。JMS 1.1 インターフェースを使用する EJB 2.1 アプリケーションと、JMS 1.0.2 インターフェースを使用する EJB 2.0 アプリケーションをデプロイできます。このサポートは、WebSphere Process Server を構築するときの基盤である WebSphere Application Server によって提供されます。

WebSphere 管理コンソールを使用して、非同期メッセージングのための WebSphere Process Server サポートを管理できます。例えば、メッセージング・プロバイダーおよびそのリソースを構成して、メッセージング・サービスのアクティビティを制御できます。

非同期メッセージングを使用する WebSphere エンタープライズ・アプリケーションのインプリメントについて詳しくは、インフォメーション・センターの以下のサブトピックを参照してください。または、<http://www-306.ibm.com/software/webservers/appserv/was/library/>の WebSphere Application Server Network Deployment バージョン 6 ライブラリー Web ページから入手できる「アプリケーションとその環境の管理」およびその他の PDF 資料を参照することもできます。

- WebSphere Application Server によるメッセージングの学習

このトピックでは、WebSphere Process Server を用いて、エンタープライズ・アプリケーションで非同期メッセージングを使用する方法について学習します。

- **JMS プロバイダーのインストールおよび構成**

このトピックでは、WebSphere Process Server で JMS プロバイダーを使用するさまざまな方法について説明します。JMS プロバイダーにより、Java Message Service (JMS) および WebSphere Process Server で、Java Message Service (JMS) および他のメッセージ・リソースを使用できるようになります。

- **デフォルトのメッセージング・プロバイダーの使用**

このトピックは、デフォルトのメッセージング・プロバイダーによって提供されるメッセージング・リソースを WebSphere アプリケーションで使用できるようにする方法に関する一連のトピックへの入り口です。デフォルトのメッセージング・プロバイダーは、WebSphere Application Server の一部としてインストールおよび実行されており、サービス統合テクノロジーを基礎としています。

- **バージョン 5 デフォルト・メッセージング・リソースのメンテナンス**

このトピックは、デフォルト・メッセージング・プロバイダーによって、WebSphere Application Server バージョン 5 アプリケーション用に提供されるメッセージング・リソースの保守に関する一連のトピックへの入り口です。

- **WebSphere MQ の JMS リソースの使用**

このトピックは、WebSphere MQ によって提供される JMS リソースを WebSphere アプリケーションで使用できるようにする方法に関する一連のトピックへの入り口です。

- **一般プロバイダーの JMS リソースの使用**

このトピックは、WebSphere のデフォルトのメッセージング・プロバイダーや WebSphere MQ 以外の一般メッセージング・プロバイダーによって提供される JMS リソースを WebSphere アプリケーションで使用できるようにする方法に関する一連のトピックへの入り口です。

- **メッセージ駆動型 Bean の管理サポート**

これらのタスクを使用して、メッセージ駆動型 Bean をサポートするためのリソースを管理します。これらは、リソース・アダプター、JMS プロバイダー、およびそれらが提供するリソースを管理するためのタスクとは別のタスクです。

- **WebSphere メッセージングのトラブルシューティング**

この概説タスクを使用して、WebSphere Messaging に関連すると考えられる問題を解決してください。WebSphere Messaging に関連すると考えられる問題を特定して解決するには、標準の WebSphere Process Server トラブルシューティング機能を使用します。

### **コマンドを使用したサービス・アプリケーション管理:**

コマンドを使用してサービス・アプリケーションを管理できます。コマンドはスク립ト内で使用できます。

サービス・アプリケーションのコマンドを実行するには、wsadmin ツールを使用する必要があります。

wsadmin ツールはさまざまな方法で使用できます。個々のコマンドとして、またはスクリプト内で、このツールを対話式に使用できます。複数のマシンを管理する場合には、スクリプト内で複数のコマンドを実行すると便利です。

WebSphere Process Server には、SCA モジュールおよびそのインポートとエクスポートを表示できるコマンドがあります。Web サービスおよび SCA インポート・バインディングに変更を加えることもできます。

1. SCA 管理コマンドをリストします。 `$AdminTask help SCAAdminCommands`
2. 特定のコマンドの詳細ヘルプを表示します。 `$AdminTask help command_name`

```
$AdminTask help listSCAModules
```

### メディエーション・モジュールの管理:

コマンド行により WebSphere Process Server にデプロイされたメディエーション・モジュールをリストできます。各メディエーション・モジュールに関連する情報を表示し、一部のインポート・バインディングを変更することもできます。

WebSphere Process Server コマンドを実行するには、wsadmin ツールを使用する必要があります。

コマンドは個別にも実行できますし、スクリプトでも実行できます。複数のマシンを管理する場合や、定期的にレポートを作成する場合には、スクリプトで複数のコマンドを実行すると便利です。

### メディエーション・モジュールのリスト表示:

コマンドを使用して、WebSphere Process Server にデプロイされたメディエーション・モジュールをリストできます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

デプロイしたメディエーション・モジュールをリストするには、wsadmin ツールを使用して以下のステップを実行します。

メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。したがって、デプロイしたメディエーション・モジュールをリストするには、SCA モジュールをリストします。

```
デプロイされている SCA モジュールをリストします。 $AdminTask  
listSCAModules
```

WebSphere Process Server にデプロイされている SCA モジュールおよびそれに関連したアプリケーションをリストします。出力は、`module name:application name` のフォーマットで戻されます。このようにリストすることにより、以降のコマンドでスクリプトを使用する際に、出力の解析処理と名前の抽出処理が容易になります。

### メディエーション・モジュール詳細の表示:

コマンドを使用して、メディエーション・モジュールの属性を表示できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

メディエーション・モジュールの説明を表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュールの説明を表示するには、メディエーション・モジュール名がわかっている必要があります。

**注:** メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされている SCA モジュールがリストされます。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールの詳細を表示します。 `$AdminTask showSCAModule {-moduleName moduleName}`

**注:** *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュールの名前および説明を表示します。

```
$AdminTask showSCAModule {-moduleName myModule -applicationName myApplication}
```

#### メディエーション・モジュールのプロパティの表示:

コマンドを使用して、指定したメディエーション・モジュールのプロパティを表示できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

メディエーション・モジュールのプロパティを表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュールのプロパティを表示するには、メディエーション・モジュール名がわかっている必要があります。

**注:** メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされている SCA モジュールがリストされます。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのプロパティを表示します。 `$AdminTask showSCAModuleProperties {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュールのプロパティを表示します。

```
$AdminTask showSCAModuleProperties {-moduleName myModule -applicationName myApplication}
```

#### メディエーション・モジュール・プロパティの変更:

コマンドを使用して、指定したメディエーション・モジュールのプロパティ値を変更できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストできます。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのプロパティをリストします。 `$AdminTask showSCAModuleProperties {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

3. SCA モジュールのモジュール・プロパティを変更します。 `$AdminTask modifySCAModuleProperty -moduleName moduleName -propertyName propertyName -newPropertyValue newPropertyValue`

注: オプションで、*applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・プロパティのプロパティ値が変更されます。

```
$AdminTask modifySCAModuleProperty {-moduleName myModule -applicationName myApplication -propertyName myPropertyName -newPropertyValue myNewPropertyValue}
```

#### インポートのリスト表示:

コマンドを使用して、WebSphere Process Server にデプロイされているメディエーション・モジュールの任意のインポートをリストできます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

メディエーション・モジュールのインポートをリストするには、wsadmin ツールを使用して以下のステップを実行します。

メディエーション・モジュールのインポートをリストするには、メディエーション・モジュール名がわかっている必要があります。

注: メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストし、listSCAImports コマンドにより、特定の SCA モジュールのインポートをリストします。SCA モジュールにインポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのインポートをリストします。 `$AdminTask listSCAImports {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュールのインポートをリストします。

```
$AdminTask listSCAImports {-moduleName myModule -applicationName myApplication}
```

### インポート詳細の表示:

コマンドを使用して、WebSphere Process Server にデプロイされているメディアエーション・モジュールのインポートの詳細を表示できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

メディアエーション・モジュールのインポートの詳細を表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディアエーション・モジュール・インポートの詳細を表示するには、メディアエーション・モジュール名およびインポート名がわかっている必要があります。

注: メディアエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストし、listSCAImports コマンドにより、特定の SCA モジュールのインポートをリストします。SCA モジュールにインポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのインポートをリストします。 `$AdminTask listSCAImports {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

3. 特定の SCA モジュール・インポートの詳細を表示します。 `$AdminTask showSCAImport {-moduleName moduleName -import importName}`

注: *moduleName* および *importName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・インポートのインポート詳細を表示します。

```
$AdminTask showSCAImport {-moduleName myModule -applicationName  
myApplication -import myImport}
```

#### **インポート・バインディングの表示:**

コマンドを使用して、WebSphere Process Server にデプロイされているメディエーション・モジュールのインポート・バインディングを表示できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

特定のメディエーション・モジュールのインポート・バインディングを表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュール・インポートのインポート・バインディングを表示するには、メディエーション・モジュール名およびインポート名がわかっている必要があります。

**注:** メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストし、listSCAImports コマンドにより、特定の SCA モジュールのインポートをリストします。SCA モジュールにインポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 **\$AdminTask listSCAModules**
2. 特定の SCA モジュールのインポートをリストします。 **\$AdminTask listSCAImports {-moduleName moduleName}**

**注:** *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

3. 特定のインポートのインポート・バインディングを表示します。 **\$AdminTask showSCAImportBinding {-moduleName moduleName -import importName}**

**注:** *moduleName* および *importName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・インポートのインポート・バインディングを表示します。

```
$AdminTask showSCAImportBinding {-moduleName myModule -applicationName  
myApplication -import myImport}
```

#### **SCA インポート・バインディングの変更:**

コマンドを使用して、WebSphere Process Server にデプロイされているメディエーション・モジュールの SCA インポート・バインディングを変更できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。



SCA バインディングによって、ある Service Component Architecture (SCA) モジュールが別の SCA モジュールに接続されます。SCA モジュールには、各種のメディアエーション・モジュールがあります。

特定のメディアエーション・モジュールによって別のメディアエーション・モジュールが起動されるようにしたい場合、SCA インポート・バインディングを変更します。インポート・バインディングを変更する場合は、インポートとエクスポートが一致するように、つまり、同等の操作を行うようにしなければなりません。それには、WSDL の検討が必要となる場合があります。

特定のメディアエーション・モジュールのインポート・バインディングを変更するには、ソースおよびターゲットのメディアエーション・モジュールの名前、およびその特定のインポートとエクスポートの名前がわかっている必要があります。

`listSCAModules` コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストできます。`listSCAImports` コマンドにより、特定の SCA モジュールのすべてのインポートが、`listSCAExports` コマンドにより、特定の SCA モジュールのすべてのエクスポートがリストされます。

1. デプロイされている SCA モジュールをリストします。

```
$AdminTask listSCAModules
```
2. 特定の SCA モジュールのインポートをリストします。

```
$AdminTask listSCAImports {-moduleName moduleName}
```

**注:** `moduleName` の指定に加えて、オプションで `applicationName` も指定できます。`applicationName` を指定することにより、パフォーマンスが向上します。

3. SCA タイプの SCA モジュール・インポート・バインディングを変更します。

```
$AdminTask modifySCAImportSCABinding -moduleName moduleName -import importName -targetModule targetModuleName -targetExport targetExportName
```

**注:** オプションで、`applicationName` および `targetApplicationName` も指定できます。`applicationName` および `targetApplicationName` を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・インポートに対する SCA インポート・バインディングが変更されました。

WebSphere ESB では、エクスポート・インターフェースに対応しない各インポート・インターフェースに対して警告が出されます。WebSphere Process Server では、インポートとエクスポートの WSDL ポート・タイプ名が比較され、同じでない場合は警告が出されます。ただし、ポート・タイプ名が一致する場合、WebSphere Process Server では、行われた操作を等価とみなし警告は出されません。

```
$AdminTask modifySCAImportSCABinding {-moduleName myModule -applicationName myApplication -import myImport -targetModule myTargetModule -targetApplicationName myTargetApplication -targetExport myTargetExport}
```

**インポート Web サービス・バインディングの変更:**

コマンドを使用して、WebSphere Process Server にデプロイされているメディエーション・モジュールの Web サービス・インポート・バインディングを変更できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

特定のメディエーション・モジュールによって別の Web サービスが起動されるようにしたい場合、インポート Web サービス・バインディングを変更します。インポート・バインディングを変更する場合は、インポートとエクスポートが一致するように、つまり、同等の操作を行うようにしなければなりません。それには、WSDL の検討が必要となる場合があります。wsadmin ツールを使用して、以下のステップを実行します。

特定のメディエーション・モジュールのインポート Web サービス・バインディングを変更するには、ターゲット・エンドポイントの特定の URL がわかっている必要があります。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストできます。listSCAImports コマンドにより、特定の SCA モジュールのすべてのインポートが、listSCAExports コマンドにより、特定の SCA モジュールのすべてのエクスポートがリストされます。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのインポートをリストします。 `$AdminTask listSCAImports {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

3. WS タイプのインポート・バインディングを変更します。 `$AdminTask modifySCAImportWSBinding -moduleName moduleName -import importName -endpoint targetEndpointName`

特定の Web サービス・インポートのインポート Web サービス・バインディングが変更されました。

WebSphere ESB では、エクスポート・インターフェースに対応しない各インポート・インターフェースに対して警告が出されます。WebSphere Process Server では、インポートとエクスポートの WSDL ポート・タイプ名が比較され、同じでない場合は警告が出されます。ただし、ポート・タイプ名が一致する場合、WebSphere Process Server では、行われた操作を等価とみなし警告は出されません。

```
$AdminTask modifySCAImportWSBinding {-moduleName myModule -applicationName myApplication -import myImport -endpoint http://myTargetEndpoint}
```

### エクスポートのリスト表示:

コマンドを使用して、WebSphere Process Server にデプロイされているメディエーション・モジュールの任意のエクスポートをリストできます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

メディエーション・モジュールのエクスポートをリストするには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュールのエクスポートをリストするには、メディエーション・モジュール名がわかっている必要があります。

**注:** メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストし、listSCAExports コマンドにより、特定の SCA モジュールのエクスポートをリストします。SCA モジュールにエクスポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのエクスポートをリストします。 `$AdminTask listSCAExports {-moduleName moduleName}`

**注:** *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュールのエクスポートをリストします。

```
$AdminTask listSCAExports {-moduleName myModule -applicationName myApplication}
```

#### **エクスポート詳細の表示:**

コマンドを使用して、WebSphere Process Server にデプロイされているメディエーション・モジュールのエクスポートの詳細を表示できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

メディエーション・モジュールのエクスポートの詳細を表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュールのエクスポート詳細を表示するには、メディエーション・モジュール名およびエクスポート名がわかっている必要があります。

**注:** メディエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストし、listSCAExports コマンドにより、特定の SCA モジュールのエクスポートをリストします。SCA モジュールにエクスポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのエクスポートをリストします。 `$AdminTask listSCAExports {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

- 特定の SCA モジュール・エクスポートの詳細を表示します。 `showSCAExport -moduleName moduleName -export exportName`

注: *moduleName* および *exportName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・エクスポートのエクスポート詳細を表示します。

```
$AdminTask showSCAExport {-moduleName myModule -applicationName  
myApplication -export myExport}
```

### エクスポート・バインディングの表示:

コマンドを使用して、WebSphere Process Server にデプロイされているメディアエーション・モジュールのエクスポート・バインディングを表示できます。

次のコマンドを実行するには、wsadmin ツールを使用する必要があります。

特定のメディアエーション・モジュールのエクスポート・バインディングを表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディアエーション・モジュール・エクスポートのエクスポート・バインディングを表示するには、そのメディアエーション・モジュール名およびエクスポート名がわかっている必要があります。

注: メディアエーション・モジュールは、Service Component Architecture (SCA) モジュールの一種です。

`listSCAModules` コマンドにより、WebSphere Process Server にデプロイされているすべての SCA モジュールをリストし、`listSCAExports` コマンドにより、特定の SCA モジュールのエクスポートをリストします。

- デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
- 特定の SCA モジュールのエクスポートをリストします。 `$AdminTask listSCAExports {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

- 特定のエクスポートのエクスポート・バインディングを表示します。  
`$AdminTask showSCAExportBinding {-moduleName moduleName -export  
exportName}`

注: *moduleName* および *export* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・エクスポートのエクスポート・バインディングを表示します。表示される情報は、バインディングのタイプによって異なります。指定され

たバインディングがエクスポートに存在しない場合は、そのバインディングが SCA タイプであるとランタイムによりみなされます。

```
$AdminTask showSCAExportBinding {-moduleName myModule -applicationName myApplication -export myExport}
```

### エンタープライズ・アプリケーションの管理:

コンソールの「エンタープライズ・アプリケーション」ページ（「アプリケーション」>「エンタープライズ・アプリケーション」とクリックして表示）を使用して、サーバーにインストールされているエンタープライズ・アプリケーションの表示および管理を行います。

アプリケーションの構成で指定した値を表示するには、リストからアプリケーション名をクリックします。「アプリケーション詳細」ページが開き、アプリケーションの構成プロパティ、および該当する場合はローカル・トポロジーが表示されます。このページから既存値を変更して、アプリケーションを構成する追加コンソール・ページにリンクできます。

エンタープライズ・アプリケーションを管理するには、その名前の横にあるチェック・ボックスをクリックしてから、以下のいずれかのボタンを使用します。

表 15. エンタープライズ・アプリケーションを管理するためのボタン

ボタン	結果のアクション
開始	<p>アプリケーションの実行を試みます。アプリケーションが正常に開始すると、アプリケーションの状態は以下のいずれかに変わります。</p> <ul style="list-style-type: none"> <li>• 開始済み - アプリケーションは、すべてのデプロイメント・ターゲットで開始しました。</li> <li>• 部分開始 - アプリケーションは、1 つ以上のデプロイメント・ターゲットでまだ開始中です。</li> </ul>
停止	<p>アプリケーションの処理の停止を試みます。アプリケーションが正常に停止すると、アプリケーションの状態は以下のいずれかに変わります。</p> <ul style="list-style-type: none"> <li>• 停止済み - アプリケーションは、すべてのデプロイメント・ターゲットで停止しました。</li> <li>• 部分停止 - アプリケーションは、1 つ以上のデプロイメント・ターゲットでまだ停止中です。</li> </ul>
インストール	<p>ウィザードが開き、エンタープライズ・アプリケーションまたはモジュール (.jar, .war, または .ear ファイルなど) をサーバーにデプロイできるようにします。</p>
アンインストール	<p>WebSphere Application Server の構成リポジトリからアプリケーションを削除し、さらに、構成の保管後にアプリケーション・モジュールのインストール先のすべてのノードのファイル・システムから、そのアプリケーション・バイナリーを削除します。</p>

表 15. エンタープライズ・アプリケーションを管理するためのボタン (続き)

ボタン	結果のアクション
更新	ウィザードが開き、サーバー上にデプロイされているアプリケーション・ファイルを更新できるようにします。アプリケーション全体を更新することも、単一モジュール、単一ファイル、あるいはアプリケーションの一部のみを更新することもできます。新規ファイルまたはモジュールの名前が、サーバー上に既に存在するファイルまたはモジュールの名前と同じである場合は、既存のファイルまたはモジュールが、その新規ファイルまたはモジュールに置き換えられます。同じ名前のファイルまたはモジュールがない場合は、デプロイされているアプリケーションに追加されます。
ファイルの除去	デプロイされたアプリケーションまたはモジュールから、ファイルを削除します。このボタンは、構成リポジトリ、およびファイルがインストールされているすべてのノードのファイル・システムから、ファイルを削除します。
エクスポート	「アプリケーション EAR ファイルのエクスポート」ページが開きます。このページでエンタープライズ・アプリケーションを EAR ファイルにエクスポートできます。デプロイされているアプリケーションをバックアップしたり、そのバインディング情報を保存したりする場合にも、「エクスポート」アクションを使用します。
DDL のエクスポート	「アプリケーション DDL ファイルのエクスポート」ページが開きます。このページでエンタープライズ・アプリケーションの EJB モジュール内の DDL ファイルをエクスポートできます。

アプリケーションの管理について詳しくは、WebSphere Application Server for z/OS の資料を参照してください。

## リレーションシップの管理

Relationship Manager は、手動でリレーションシップ・データを制御および操作するためのツールです。このツールで、自動化されたリレーションシップ管理で検出されたエラーを訂正し、より完全なリレーションシップ情報を提供します。特に、このツールには、リレーションシップ・インスタンス・データの検索および変更を行う機能があります。

Relationship Manager によって、リレーションシップ・ランタイム・データ (ロールおよびそれらのデータを含む) を構成、照会、表示、および操作できます。

Relationship Editor を使用して、リレーションシップの定義を作成します。実行時に、リレーションシップのインスタンスには、別のアプリケーションの情報を関連付けるデータが取り込まれます。このリレーションシップ・インスタンス・データは、マップまたは WebSphere Process Server のその他のコンポーネントの実行時にリレーションシップ・インスタンスが必要になるときに作成されます。リレーションシップ・サービスは、一連のアプリケーション・プログラミング・インターフェース (API) を公開して、リレーションシップ・メタデータの検索、インスタンス・データの作成、検索、および操作を行います。データは、リレーションシップ定義

で指定されているリレーションシップ・テーブルに保管されます。Relationship Manager には、リレーションシップおよびリレーションシップ・インスタンスと対話するためのグラフィカル・ユーザー・インターフェースがあります。

Relationship Manager は、リレーションシップ・インスタンスごとに、ロールの階層リストを表示できます。リレーションシップの各ロールには、インスタンス・データ、プロパティ、およびキー属性があります。リレーションシップ・ツリーは、エンティティのタイプ、その値、最終変更日などの、リレーションシップ・インスタンス内の各ロールに関する詳細な情報も提供します。リレーションシップ・インスタンス ID は、リレーションシップ・インスタンスがリレーションシップ・テーブル内に保管されるときに、自動的に生成されます。Relationship Manager は、リレーションシップ・ツリーの最上位にこのインスタンス ID を表示します。

Relationship Manager を使用すると、全レベルでエンティティを管理できます。つまり、リレーションシップ・インスタンス、ロール・インスタンス、属性データ、プロパティ・データの各レベルです。例えば、Relationship Manager を使用して、以下を行うことができます。

- 既存のリレーションシップの値の表示および検査。
- リレーションシップ・インスタンスの作成および削除。
- リレーションシップ・インスタンスの内容の変更 (ロール・インスタンスの追加および削除など)。
- ロール・プロパティや論理状態などのリレーションシップ・ロール・インスタンスのデータの編集。
- ロール・インスタンスのアクティブ化および非アクティブ化。
- キー属性、開始日と終了日、およびプロパティ値を指定したロール・インスタンスの取得。
- 問題発生時の状態の修復。例えば、ソース・アプリケーションの破損データや不整合データが汎用および宛先アプリケーション・リレーションシップ・テーブルに送信された場合、Relationship Manager を使用して、データの信頼性が確信できる時点でデータをロールバックできます。

リレーションシップについて詳しくは、WebSphere Integration Developer インフォメーション・センターおよび WebSphere Process Server インフォメーション・センターのリレーションシップ・サービスに関するトピックを参照してください。

## リレーションシップ・タイプの表示

リレーションシップ名、表示名、および静的属性と識別属性などの、リレーションシップ・タイプに関連した情報を表示するには、このタスクを実行します。

**必要なセキュリティ・ロール:** このタスクを実行するには、モニター、オペレーター、コンフィギュレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。

情報は表形式で表示されます。各リレーションシップ・タイプはリンクです。

**ヒント:** 一度に表示される行数をカスタマイズできます。「設定」をクリックし、「最大行数」フィールド値を変更します。デフォルトは 25 です。

## リレーションシップ詳細の表示

リレーションシップ名、表示名、関連するロールとそれらの属性、プロパティ値、および静的属性と識別属性などの、リレーションシップ・タイプに関する詳細情報を表示するには、このタスクを実行します。

**必要なセキュリティ・ロール:** このタスクを実行するには、モニター、オペレーター、コンフィギュレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 次の 2 つの方法で、リレーションシップの詳細を表示できます。
  - a. リレーションシップ名をクリックします。
  - b. 「**選択**」列で、リレーションシップ名を選択し、「**詳細**」をクリックします。

リレーションシップの詳細には、表形式で表示されるロール属性が含まれます。ロール属性には、ロールの表示名、オブジェクト名、および管理対象属性設定が含まれます。

リレーションシップ・タイプのリストに戻るには、ページの上部のパスから「リレーションシップ」をクリックするか、または「戻る」をクリックします。

## ロール詳細の表示

リレーションシップ名、ロール名、表示名、プロパティ値、キー、ロール・オブジェクト・タイプ、および管理対象属性設定などの、ロールに関する詳細情報を表示するには、このタスクを実行します。

**必要なセキュリティ・ロール:** このタスクを実行するには、モニター、オペレーター、コンフィギュレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. リレーションシップ名をクリックして、「リレーションシップの詳細」ページを開きます。



5. 「**ロール・スキーマ情報**」で、関連するロール名をクリックして、「**ロールの詳細**」ページを開きます。

「**リレーションシップの詳細**」ページに戻るには、ページの上部のパスから「**リレーションシップの詳細**」をクリックするか、または「**戻る**」をクリックします。

## リレーションシップ・インスタンスの照会

リレーションシップ・ベースのインスタンス照会を実行するには、このタスクを使用します。

**必要なセキュリティー・ロール:** このタスクを実行するには、モニター、オペレーター、コンフィギュレーター、または管理者としてログインしている必要があります。

照会オプション (すべて、ID ごと、プロパティごと、またはロールごと) を選択して、リレーションシップ用のすべてのインスタンス・データまたはそのサブセットを取得します。その照会の結果セットが戻され、各行が 1 つのリレーションシップ・インスタンスを表す表形式で表示されます。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「**統合アプリケーション**」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「**リレーションシップ**」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「**選択**」列で、リレーションシップ名を選択し、「**照会**」をクリックします。
5. 照会オプション・タブのいずれかをクリックし、検索基準を指定します。

オプション	説明
「すべて」タブ	リレーションシップのすべてのインスタンスのリストを取得します。表示する内容を選択できます (すべてのアクティブ化されたリレーションシップ・インスタンス・データ、すべての非アクティブ化されたリレーションシップ・インスタンス・データ、またはすべてのアクティブ化されたおよび非アクティブ化されたリレーションシップ・インスタンス・データ)。
「ID ごと」タブ	開始インスタンス ID と終了インスタンス ID の間の範囲のリレーションシップ・インスタンスを取得します。どちらかのフィールドが空白になっている場合は、単一のインスタンスのみが戻されます。照会によって、検出されたインスタンス用のすべてのロールが戻されます。
「プロパティごと」タブ	特定のプロパティ値に基づいてリレーションシップ・インスタンスを取得します。

オプション	説明
「ロールごと」タブ	ロール名、キー属性値、ロールが作成または変更された日付範囲、または特定のプロパティ値に基づいてリレーションシップ・インスタンスを取得します。

6. 照会パラメーターを指定した後に行うオプションとして、以下があります。
- 「OK」をクリックして、照会からの結果データを表示します。
  - 「キャンセル」をクリックして、すべての変更内容を破棄し、リレーションシップ・タイプのリストに戻ります。

## リレーションシップ・インスタンスの結果の表示

リレーションシップ・インスタンス照会に一致するインスタンスのリストを表示するには、このタスクを実行します。テーブル・ビューに表示される結果には、リレーションシップ・インスタンス ID とインスタンスに関連付けられたプロパティ値が含まれています。

**必要なセキュリティ・ロール:** このタスクを実行するには、モニター、オペレーター、コンフィギュレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名を選択し、「照会」をクリックします。
5. 照会オプション・タブのいずれか（「すべて」、「ID ごと」、「プロパティごと」、または「ロールごと」）をクリックし、検索基準を指定します。照会オプションの説明については、293 ページの『リレーションシップ・インスタンスの照会』を参照してください。
6. 「OK」をクリックして、「リレーションシップ・インスタンスの結果」ページを開きます。

照会と一致するリレーションシップ・インスタンスのリストがテーブル・ビューに表示されます。各リレーションシップ・インスタンスが 1 つの行に対応します。ページと戻されたインスタンスの合計カウントが、ページの下部に表示されます。

**ヒント:** 一度に表示される行数をカスタマイズできます。「設定」をクリックし、「行」フィールド値を変更し、「適用」をクリックします。デフォルトは 25 です。一度に表示されるレコードの最小数は 1 であり、最大数はすべてのレコードです。

- インスタンスの次のセットを表示するには、進む矢印をクリックします。
- 前のページのインスタンスを表示するには、戻る矢印をクリックします。

**制約事項:** ソートを実行するにはサーバーから全照会結果セットを取得する必要があるため、多数のリレーションシップ・インスタンスをフィルタリングまたはソートすると、パフォーマンスの問題が発生する可能性があります。

す。例えば、20,000 個のリレーションシップ・インスタンスを戻す照会のリレーションシップ・インスタンス・データをソートするには、その20,000 個のインスタンスをソートする必要があります。合計カウント (ページの下部) は、予期されるリレーションシップ・インスタンス数の推定値を示すため、大量のデータ・セットのソートまたはフィルタリングによって長い待ち時間が発生するかどうかを判断できます。

一度にサーバーから読み取るインスタンス数をカスタマイズできるようにキャッシュ・サイズのヒント・パラメーターを設定する方法については、リレーションシップ・サービスの構成についてのヘルプ・トピックを参照してください。

## リレーションシップ・インスタンス詳細の表示

リレーションシップ名、リレーションシップ・インスタンス ID、プロパティー値、参加しているロール、およびロール・インスタンス値 (ロール・インスタンス ID、論理状態、キー属性、およびプロパティー値) などの、選択されたリレーションシップ・インスタンスに関する詳細情報を表示するには、このタスクを実行します。複数のロールを同時に表示できます。

**必要なセキュリティー・ロール:** このタスクを実行するには、モニター、オペレーター、コンフィギュレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「**選択**」列で、リレーションシップ名を選択し、「**照会**」をクリックします。
5. 照会オプション・タブのいずれか (「すべて」、「**ID ごと**」、「**プロパティーごと**」、または「**ロールごと**」) をクリックし、検索基準を指定し、「**OK**」をクリックします。

「リレーションシップ・インスタンスの結果」ページが開きます。

6. 次の 2 つの方法で、リレーションシップ・インスタンスの詳細を表示できます。
  - リレーションシップ・インスタンス ID をクリックします。
  - 「**選択**」列で、リレーションシップ・インスタンス ID を選択し、「**詳細**」をクリックします。

リレーションシップ・インスタンスの結果のリストに戻るには、ページの上部のパスから「リレーションシップ・インスタンス」をクリックします。

## リレーションシップ・インスタンスの詳細の編集

選択されたリレーションシップ・インスタンスに関するプロパティー値を編集するには、このタスクを実行します。

**必要なセキュリティー・ロール:** このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「**選択**」列で、リレーションシップ名を選択し、「**照会**」をクリックします。
5. 照会オプション・タブのいずれか（「**すべて**」、「**ID ごと**」、「**プロパティごと**」、または「**ロールごと**」）をクリックし、検索基準を指定し、「**OK**」をクリックします。

「リレーションシップ・インスタンスの結果」ページが開きます。

6. 次の 2 つの方法のいずれかで、リレーションシップ・インスタンスの詳細を表示します。
  - リレーションシップ・インスタンス ID をクリックします。
  - 「**選択**」列で、リレーションシップ・インスタンス ID を選択し、「**詳細**」をクリックします。
7. 必要に応じて、リレーションシップ・インスタンス・プロパティ値を変更します。

**制約事項:** 以前にリレーションシップ・インスタンスに対してプロパティ値が定義された場合のみ、プロパティ値を編集できます。

リレーションシップ・インスタンスを削除するには、ページ下部の「**削除**」をクリックします。

このページから、新規ロール・インスタンスを作成したり、既存のロール・インスタンスを削除したりすることもできます。作成または削除するには、そのロール・インスタンスを選択し、ロール・テーブルの下にある「**作成**」または「**削除**」をそれぞれクリックします。「**作成**」をクリックすると、新規ロール・インスタンスのキー属性値とプロパティ値を入力するための「新規ロール・インスタンス」ページが開きます。選択されたロール・インスタンス ID をクリックすると、ロール・インスタンスのプロパティ値を編集できます。

8. インスタンスおよびインスタンスのロールでの変更が終了した後に行うオプションとして、以下があります。
  - 「**OK**」をクリックして、変更内容をシステムに即時に保管します。
  - 「**キャンセル**」をクリックして、すべての変更内容を破棄し、「リレーションシップ・インスタンスの結果」ページに戻ります。

## リレーションシップ・インスタンスの新規作成

新しいリレーションシップ・インスタンスを作成するには、このタスクを実行します。

**必要なセキュリティ・ロール:** このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。

3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。

「リレーションシップ・タイプ」ページが開きます。

4. 「選択」列で、リレーションシップ名を選択し、「作成」をクリックします。

「新規リレーションシップ・インスタンス」ページが開きます。

5. デフォルト値以外の値が必要な場合は、「値」フィールドにプロパティ値の情報を追加し、「OK」をクリックして、新規リレーションシップ・インスタンスをローカル側で保管します。

**注:** ロール・インスタンスがないとリレーションシップ・インスタンスを持つことはできないので、リレーションシップ・インスタンス用のロール・インスタンスを作成する必要もあります。

## リレーションシップ・インスタンスの削除

選択したリレーションシップ・インスタンスを削除するには、このタスクを実行します。

**必要なセキュリティ・ロール:** このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」>「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名を選択し、「照会」をクリックします。
5. 照会オプション・タブのいずれか（「すべて」、「ID ごと」、「プロパティごと」、または「ロールごと」）をクリックし、検索基準を指定し、「OK」をクリックします。

「リレーションシップ・インスタンスの結果」ページが開きます。

6. 「選択」列で、削除するリレーションシップ・インスタンスの ID を選択します。
7. 「削除」をクリックします。

リレーションシップ・インスタンスは、システムから即時に削除されます。

## リレーションシップ・インスタンス・データのロールバック

指定した日時のリレーションシップのインスタンス・データにロールバックするには、このタスクを実行します。

**必要なセキュリティ・ロール:** このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」>「Relationship Manager」をクリックします。

3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名を選択し、「ロールバック」をクリックします。
5. 「開始日」および「終了日」フィールドにロールバックの期間を入力します。
6. 「OK」をクリックします。

リレーションシップ内のインスタンス・データのうち、指定された日時より後に作成されたすべてのインスタンス・データは、非アクティブ状態としてマークされます。

## ロール・インスタンス詳細の表示

ロール名、ロール・エレメント、キー属性とプロパティ値、状況、および論理状態などの、選択されたロール・インスタンスに関する詳細情報を表示するには、このタスクを実行します。

**必要なセキュリティ・ロール:** このタスクを実行するには、モニター、オペレーター、コンフィギュレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名を選択し、「照会」をクリックします。
5. 照会オプション・タブのいずれか（「すべて」、「ID ごと」、「プロパティごと」、または「ロールごと」）をクリックし、検索基準を指定し、「OK」をクリックします。

「リレーションシップ・インスタンスの結果」ページが開きます。

6. 次の 2 つの方法のいずれかで、リレーションシップ・インスタンスの詳細を表示します。
  - リレーションシップ・インスタンス ID をクリックします。
  - 「選択」列で、リレーションシップ・インスタンス ID を選択し、「詳細」をクリックします。
7. ロール・インスタンスに関する詳細を表示するには、ロール・インスタンス・テーブルで、関連する ID をクリックします。

## ロール・インスタンス・プロパティの編集

選択されたロール・インスタンスに関するプロパティ値を編集するには、このタスクを実行します。

**必要なセキュリティ・ロール:** このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。

2. ナビゲーション・ペインで、「統合アプリケーション」>「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「**選択**」列で、リレーションシップ名を選択し、「**照会**」をクリックします。
5. 照会オプション・タブのいずれか（「すべて」、「**ID** ごと」、「プロパティごと」、または「**ロール**ごと」）をクリックし、検索基準を指定し、「**OK**」をクリックします。

「リレーションシップ・インスタンスの結果」ページが開きます。

6. 次の 2 つの方法のいずれかで、リレーションシップ・インスタンスの詳細を表示します。
  - リレーションシップ・インスタンス ID をクリックします。
  - 「**選択**」列で、リレーションシップ・インスタンス ID を選択し、「**詳細**」をクリックします。
7. ロール・インスタンス・テーブルで、ロール・インスタンス ID をクリックして、ロール・インスタンスの詳細を表示します。
8. 必要に応じてロール・インスタンス・プロパティ情報を編集し、「**OK**」をクリックして、この変更をローカル側で保管します。

**制約事項:** 以前にリレーションシップ・インスタンスに対してプロパティ値が定義された場合のみ、プロパティ値を編集できます。

## ロール・インスタンスの新規作成

リレーションシップ用の新しいロール・インスタンスを作成するには、このタスクを実行します。

**必要なセキュリティー・ロール:** このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」>「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「**選択**」列で、リレーションシップ名を選択し、「**照会**」をクリックします。
5. 照会オプション・タブのいずれか（「すべて」、「**ID** ごと」、「プロパティごと」、または「**ロール**ごと」）をクリックし、検索基準を指定し、「**OK**」をクリックします。

「リレーションシップ・インスタンスの結果」ページが開きます。

6. 次の 2 つの方法のいずれかで、リレーションシップ・インスタンスの詳細を表示します。
  - リレーションシップ・インスタンス ID をクリックします。
  - 「**選択**」列で、リレーションシップ・インスタンス ID を選択し、「**詳細**」をクリックします。

7. 新規インスタンスを作成する対象のロールを指定し、ロール・テーブルの下にある「作成」をクリックします。

「新規ロール・インスタンス」ページが開きます。

8. キー属性とロール・プロパティの値をそれぞれの「値」フィールドに入力し、「OK」をクリックして、新規ロール・インスタンスをローカル側で保管します。

**制約事項:** キー属性値は、ロール・インスタンスの作成時にのみ設定できます。変更内容をデータベースに適用した後では、この情報は変更できません。ただし、プロパティ値は後で編集できます。

## ロール・インスタンスの削除

リレーションシップの選択したロール・インスタンスを削除するには、このタスクを実行します。

**必要なセキュリティー・ロール:** このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名を選択し、「照会」をクリックします。
5. 照会オプション・タブのいずれか（「すべて」、「ID ごと」、「プロパティごと」、または「ロールごと」）をクリックし、検索基準を指定し、「OK」をクリックします。

「リレーションシップ・インスタンスの結果」ページが開きます。

6. 次の 2 つの方法のいずれかで、リレーションシップ・インスタンスの詳細を表示します。
  - リレーションシップ・インスタンス ID をクリックします。
  - 「選択」列で、リレーションシップ・インスタンス ID を選択し、「詳細」をクリックします。
7. ロール・インスタンスを削除する対象のロールを指定します。
8. 「選択」列で、削除するロール・インスタンスをクリックし、ロール・テーブルの下にある「削除」をクリックします。

ロール・インスタンスは、ローカル側で削除されます。

## チュートリアル: Relationship Manager の管理

このチュートリアルでは、WebSphere Process Server の Relationship Manager の基本機能を説明します。リレーションシップは、同じデータ項目に対して複数の異なる環境から取得された ID を相関させるために使用されます。例えば、1 つの環境では、州名は 2 文字の略語 (AZ, TX) として識別されます。別の環境では、異なる



る略語 (Ariz、Tex.) が使用されます。リレーションシップは、最初の環境の「AZ」を 2 番目の環境の「Ariz」と関連させるために作成されます。

ここで説明するサンプル・リレーションシップは、カスタマー ID を関連させるリレーションシップです。多くのビジネス・アプリケーションでは、カスタマーのデータベースを管理し、各カスタマーにそのアプリケーション独自の ID を割り当てています。エンタープライズ環境では、同じカスタマーがビジネス・アプリケーションごとに異なる ID を持つ可能性があります。このチュートリアルでは、カスタマー ID を関連させるリレーションシップが定義されます。リレーションシップ名は「SampleCustID」です。このリレーションシップには、2 つのロールが定義されています。1 つのロールは Customer Information System (CIS) 用で、もう 1 つのロールは General Ledger (GL) アプリケーション用です。このリレーションシップは、リレーションシップ・サービス・サンプルに基づいて作成され、ロールと少量のサンプル・データも提供されています。

Relationship Manager は、リレーションシップ・インスタンスを追加、変更、および除去できるだけでなく、リレーションシップ・インスタンスのロール・インスタンスも追加、変更、および除去できるように設計されています。新規のリレーションシップ定義を作成およびデプロイするには、WebSphere Integration Developer を使用する必要があります。定義は XML ファイルとして保管され、J2EE アプリケーションの一部として特定のサーバーにデプロイされます。

## このチュートリアルの目的

このチュートリアルを完了すると、リレーションシップ・インスタンスの値を変更できるようになります。

## このチュートリアルを完了するのに必要な時間

このチュートリアルを完了するのに必要な時間は、約 10 分です。

## 前提条件

このチュートリアルでは、リレーションシップ・サービス・テクニカル・サンプルに基づいて作成されたリレーションシップを使用します。このチュートリアルを手順に従って実行する前に、サンプル・ギャラリーにアクセスし、リレーションシップ・サービス・サンプルで説明されている手順を実行して、必要なリレーションシップおよびロールを作成します。

## リレーションシップ・インスタンスの値の変更

あるカスタマーの CIS アプリケーション内でのカスタマー ID が A004 であるとしてします。このカスタマーの GL アプリケーション内でのカスタマー ID は 801 です。しかし、データ入力エラーがあったため、このカスタマーのカスタマー ID を関連させるリレーションシップ・インスタンスでは、現時点で GL カスタマー ID の値が 801 ではなく 901 になっています。このチュートリアルでは、リレーションシップ内のこの入力を訂正する手順を説明します。

1. WebSphere Process Server 管理コンソールを開きます。
2. セキュリティーが使用可能に設定されている場合は、管理者特権を持ったユーザーとしてログインします。

3. ナビゲーション・ペインで、「統合アプリケーション」>「**Relationship Manager**」をクリックします。
4. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。

SampleCustID という名前のリレーションシップが表示されます。

5. 「**選択**」列で、SampleCustID を選択し、「**照会**」をクリックします。
6. 照会の「**ロールごと**」タブをクリックします。「**ロール名**」フィールドでドロップダウン・リストから MyGLCustomer\_0 を選択し、「**キー属性**」の下にある「**値**」フィールドに 901 と入力し、「**OK**」をクリックします。

これによって、要求されたカスタマーのリレーションシップ・インスタンスが検索され、「リレーションシップ・インスタンスの結果」ページが表示されます。

7. リレーションシップ・インスタンス ID をクリックします。

これによって、GL アプリケーション内のカスタマー ID 901 のリレーションシップ・インスタンス・データ (すべての関連するロール・インスタンスを含む) が表示されます。

8. MyGLCustomer\_0 ロール・テーブルで、キー属性値 901 を持つロール・インスタンス ID を選択し、ロール・テーブルの下にある「**削除**」をクリックします。

**注:** 関連付けられたプロパティ値がないことを確認してください。他のデータが表示された場合は、ロール・インスタンスを確認して、保持したいデータをすべて記録する必要があります。

9. 「**作成**」をクリックして、このリレーションシップ・インスタンス用の新規ロール・インスタンスを作成します。

「新規ロール・インスタンス」ページが開きます。

10. 「**キー属性**」の下にある「**値**」フィールドに 801 と入力し、「**OK**」をクリックします。

新規ロール・インスタンスが保管され、テーブル内に新規ロール・インスタンスが表示されます。

リレーションシップ・インスタンス内の GL アプリケーションのカスタマー ID 値が訂正されました。

---

## イベントの順次処理

イベント順序付けにより、WebSphere Process Server コンポーネントが非同期呼び出しからのイベントを配信順に処理することが保証されます。イベントの順序は、ビジネス・インテグレーション・シナリオ全体で維持されます。

「イベント」は、1 つのコンポーネントから別のコンポーネントへ送信される要求または応答です。イベントは、データおよび呼び出しメタデータ (ターゲット・コンポーネントの名前、操作、パラメーターなど) をカプセル化します。

注: イベントの順序付けは、非同期呼び出しにより送信される要求でのみサポートされます。

## イベント順序付けを使用する理由

実装によっては、ソース・アプリケーションによって送信された順序と同じ順序でターゲット・コンポーネントがイベントを処理する必要がある場合があります。誤った順序で処理すると、エラーや例外が発生する可能性があります。例えば、ソース・アプリケーションが、ビジネス・オブジェクトを作成するイベントを生成した後、そのビジネス・オブジェクトを更新するイベントを生成した場合は、作成イベントを最初に処理する必要があります。

非同期呼び出しでは、イベントはサービス統合バスの宛先に格納され、メッセージ駆動型 Bean (MDB) の複数のインスタンスによって処理できます。その結果、正しい順序で処理されず、失敗してしまう可能性があります。この問題を避けるために、イベント順序付けを使用します。

## イベント順序付けの仕組み

イベント順序付けを有効にするには、WebSphere Integration Developer で使用可能なイベント順序付け修飾子を使用します。修飾子は、イベント順序付けが必要な各メソッドで設定する必要があります。修飾子は、それらのメソッドに対する呼び出しを順序付ける必要があることをランタイム環境に指示します。

各修飾子には、イベントの順序付けの方法を決定するイベント順序付けキーが含まれます。キーの値は、呼び出しに関連したビジネス・オブジェクトの 1 つ以上の属性から構成されます。同じキーを共有するすべてのイベントがグループ化され、順次処理されます。イベント順序付けキーを持たないイベントは、引き続き通常どおりに (順序付けされたイベントと並列に) 処理されます。

順序付けされたイベントは、ロックを獲得してから、ターゲット・コンポーネントに送信されて処理されます。そのイベントに対するビジネス・ロジックの実行が完了すると、ロックはただちに解放され、同じイベント順序付けキーを持つ次のイベントに割り当てられます。イベントが必要なロックを獲得できない場合は、ロックを獲得するまで呼び出しの実行が中断されます。

### 関連概念

#### 304 ページの『例: イベント順序付け』

イベント順序付けの仕組みを理解するために、ソース・アプリケーション (コンポーネント A) がターゲット・アプリケーション (コンポーネント B) を非同期的に呼び出して新規オーダーを作成した後、変更されたデータでそれらのオーダーを更新する場合について考えます。

#### 306 ページの『イベント順序付けを実装する場合の考慮事項』

ビジネス・インテグレーション環境でイベント順序付けの計画、実装、およびトラブルシューティングを行うには、ここに含まれるトピックに記載されている情報を使用してください。

## 例: イベント順序付け

イベント順序付けの仕組みを理解するために、ソース・アプリケーション (コンポーネント A) がターゲット・アプリケーション (コンポーネント B) を非同期的に呼び出して新規オーダーを作成した後、変更されたデータでそれらのオーダーを更新する場合について考えます。

コンポーネント A はコンポーネント B を検索し、`create` メソッドを呼び出してオーダーを作成し、その際に `Order` ビジネス・オブジェクトを使用します。`Order` ビジネス・オブジェクトは以下の属性を持ちます。

属性	タイプ
ID	string
customer	string
productName	string
quantity	integer

コンポーネント A は次に `update` メソッドを呼び出し、新規に作成したオーダーのデータを更新します。

この例では、コンポーネント A からコンポーネント B に、5 つの別個のイベントが以下に示す順序で送信されたと想定します。

- `Create1` - この呼び出しは `create` メソッドを呼び出し、ID が 1 で quantity が 10 の `Order` ビジネス・オブジェクトを渡します。
- `Create2` - この呼び出しは `create` メソッドを呼び出し、ID が 2 で quantity が 8 の `Order` ビジネス・オブジェクトを渡します。
- `Update1` - この呼び出しは `update` メソッドを呼び出し、ID が 1 で quantity が 15 の `Order` ビジネス・オブジェクトを渡します。
- `Update2` - 3 番目の呼び出しは `update` メソッドを呼び出し、ID が 1 で quantity が 12 の `Order` ビジネス・オブジェクトを渡します。
- `Update3` - この呼び出しは `update` メソッドを呼び出し、ID が 2 で quantity が 10 の `Order` ビジネス・オブジェクトを渡します。

イベントごとに、メッセージは、呼び出しと同じ順序でサービス統合バス宛先に格納されます。メッセージ駆動型 Bean (MDB) がメッセージを読み取り、ターゲット・コンポーネント (この場合はコンポーネント B) に送信して処理させます。モジュールごとの MDB の数は 1 つのみですが、その MDB のインスタンスは複数存在し、これら 5 つのメッセージは並列に処理されます。`Update2` のメッセージを処理している MDB スレッドが、`Create1` イベントのメッセージを処理しているスレッドより先に完了する可能性があります。その場合にはオーダーがまだ作成されていないため、`Update2` イベントは失敗します。

このようなエラーを防ぐために、この例ではイベント順序付けを実装しています。下記のコンポーネント定義の例では、`create` メソッドと `update` メソッドの両方にイベント順序付け修飾子を指定しています。これらのメソッドはいずれも同じイベント順序付けキー (`Order` ビジネス・オブジェクトの `ID` 属性に設定されます) を使用し、同じイベント順序付けグループに配置されます。3 番目のメソッド (`retrieve`) は順序付けしていません。

```

<interfaces>
  <interface xsi:type="wsdl:WSDLPortType" portType="ns1:ProcessOrder">
    <method name="create">
      <scdl:interfaceQualifier xsi:type="es:EventSequencingQualifier">
        <es:eventSequencing sequencingGroup="default" continueOnError="true">
          <keySpecification>
            <parameter name="Order">
              <xpath>ID</xpath>
            </parameter>
          </keySpecification>
        </es:eventSequencing>
      </scdl:interfaceQualifier>
    </method>
    <method name="update"/>
      <scdl:interfaceQualifier xsi:type="es:EventSequencingQualifier">
        <es:eventSequencing sequencingGroup="default" continueOnError="true">
          <keySpecification>
            <parameter name="Order">
              <xpath>ID</xpath>
            </parameter>
          </keySpecification>
        </es:eventSequencing>
      </scdl:interfaceQualifier>
    <method name="retrieve"/>
  </interface>
</interfaces>

```

イベント順序付けを有効にすると、この例の 5 つのイベントは以下のように処理されます。

1. コンポーネント A が Create1 要求を送信します。これは宛先に置かれ、MDB のインスタンスによって処理されます。
2. Create1 イベントがロックを獲得し、コンポーネント B に送信して処理させます。
3. コンポーネント A が Update1 要求を送信します。これは宛先に置かれ、MDB のインスタンスによって処理されます。
4. Update1 イベントがロックの獲得を試みます。Create1 イベント (Update1 と同じイベント順序付けキー値を共用) がまだロックを保持している場合は、Create1 に対するロックが解放されるまで、このイベントの処理は中断されます。
5. コンポーネント A が Create2 要求を送信します。これは宛先に置かれ、MDB のインスタンスによって処理されます。
6. Create2 要求 (イベント順序付けキーの値が異なる) がロックを獲得し、コンポーネント B に送信されて処理されます。
7. コンポーネント A が Update2 要求を送信します。これは宛先に置かれ、MDB のインスタンスによって処理されます。
8. Update2 イベントがロックの獲得を試みます。Create1 イベントまたは Update1 イベントのいずれか (Update2 と同じイベント順序付けキー値を共用) がまだロックを保持している場合は、このイベントの処理は中断されます。Update1 イベントがロックを獲得して処理され、ロックが解放されるまで処理は行われません。
9. コンポーネント A が Update3 要求を送信します。Create2 イベント (Update3 と同じイベント順序付けキー値を共用) がまだロックを保持している場合は、Create2 に対するロックが解放されるまで、このイベントの処理は中断されます。

## 関連概念

302 ページの『イベントの順次処理』

イベント順序付けにより、WebSphere Process Server コンポーネントが非同期呼び出しからのイベントを配信順に処理することが保証されます。イベントの順序は、ビジネス・インテグレーション・シナリオ全体で維持されます。

## イベント順序付けを実装する場合の考慮事項

ビジネス・インテグレーション環境でイベント順序付けの計画、実装、およびトラブルシューティングを行うには、ここに含まれるトピックに記載されている情報を使用してください。

### 関連概念

302 ページの『イベントの順次処理』

イベント順序付けにより、WebSphere Process Server コンポーネントが非同期呼び出しからのイベントを配信順に処理することが保証されます。イベントの順序は、ビジネス・インテグレーション・シナリオ全体で維持されます。

## サポートされるコンポーネントと呼び出し

イベント順序付けを実装する前に、使用する呼び出しおよびコンポーネントのタイプと、それらが順序付けをサポートするかどうかを考慮してください。

イベント順序付けは、以下の要件を満たす Service Component Architecture (SCA) コンポーネントからのすべての要求でサポートされます。

- コンポーネントが WSDL (Web サービス記述言語) インターフェースを使用する必要があります。
- コンポーネントが非同期呼び出しを使用する必要があります。

**注:** イベントが SCA 宛先に格納される前は、クライアントがイベントの順序を管理する必要があります。順序付けが必要な場合は、クライアントが単一スレッド内で SCA 呼び出しを実行する必要があります。

応答ではサポートされません。

同期が実装されているコンポーネントに対する同期呼び出し中に暗黙のうちに順序付けされるイベントに対してイベント順序付けを使用する必要はありません。クライアントが呼び出しに単一スレッドを使用する場合、呼び出しは、ターゲットがイベントの処理を完了するまで自動的に待機します。イベントが返されるまで、他の呼び出しは実行できません。

## コンポーネントに対するイベント順序付けの宣言

コンポーネント上のどのメソッドがイベント順序付けを使用する必要があるかを判別したら、WebSphere Integration Developer を使用して、それらの各メソッドにイベント順序付け修飾子を含めるよう、コンポーネント定義を更新します。

イベント順序付け修飾子は Service Component Definition Language (SCDL) で定義されているタイプを拡張し、Service Component Architecture (SCA) コンポーネントに対するサービス品質を向上させます。イベント順序付け修飾子には keySpecification エlement が含まれ、この Element によって順序付けするイベントを識別します。イベント順序付けを使用するメソッドごとに 1 つの

keySpecification エレメントが必要です。各 keySpecification では parameter エレメントを使用します。このエレメントは、イベント順序付けキーの値を提供するビジネス・オブジェクト属性 (複数可) を示します。

さらに、イベント順序付け修飾子は、sequencingGroup 属性を使用して一緒に順序を付ける必要があるメソッドをグループ化します。同じグループのメソッドによって生成されるイベントは、すべて順次処理されます。

下記のコンポーネント定義の例では、create メソッドおよび update メソッドでイベント順序付けを定義しています。両方のメソッドの keySpecification エレメントが、パラメーターを newOrder ビジネス・オブジェクトの ID 属性として定義します。

```
<interfaces>
  <interface xsi:type="wsdl:WSDLPortType" portType="ns1:ProcessOrder">
    <method name="create">
      <scdl:interfaceQualifier xsi:type="es:EventSequencingQualifier">
        <es:eventSequencing sequencingGroup="default" continueOnError="true">
          <keySpecification>
            <parameter name="newOrder">
              <xpath>ID</xpath>
            </parameter>
          </keySpecification>
        </es:eventSequencing>
      </scdl:interfaceQualifier>
    </method>
    <method name="update"/>
      <scdl:interfaceQualifier xsi:type="es:EventSequencingQualifier">
        <es:eventSequencing sequencingGroup="default" continueOnError="true">
          <keySpecification>
            <parameter name="newOrder">
              <xpath>ID</xpath>
            </parameter>
          </keySpecification>
        </es:eventSequencing>
      </scdl:interfaceQualifier>
    </interface>
  </interfaces>
```

## エクスポート・バインディングでのイベント順序付けの使用

イベント順序付けは、EIS エクスポート・バインディング、JMS エクスポート・バインディング、ネイティブ MQ、および WebSphere MQ JMS エクスポート・バインディングでサポートされます。エクスポートに正しい順序でメッセージを処理および配信させるには、エクスポート・バインディングを適切に構成する必要があります。

エクスポート・バインディングを処理するターゲット・コンポーネントでイベント順序付けを使用する場合は、以下の要件を考慮に入れてください。

- ターゲット・コンポーネントでイベント順序付けを使用する場合は、アダプター・コンポーネントが Java Message Service (JMS) エクスポートの最適化されていないパスを使用する必要があります。
- JMS エクスポート・バインディングのイベント順序付けを有効にするには、着信メッセージを処理する並行メッセージ駆動型 Bean (MDB) の数を制限する必要があります。このためには、ActivationSpec の maxConcurrency カスタム・プロパティの値を 1 に設定します。

- WebSphere MQ JMS エクスポートのイベント順序付けを有効にするには、メッセージ駆動型 Bean にメッセージを配信する並行リスナー・スレッドの数を制限する必要があります。このためには、maxSessions プロパティの値を 1 に設定します。
- ネイティブ MQ エクスポートのイベント順序付けを有効にするには、WebSphere Integration Developer を使用して eventSequencing プロパティを設定する必要があります。

#### 関連概念

127 ページの『エクスポート・バインディングとイベント順序付け』  
エクスポートがメッセージを受信順に処理および配信するように 規定できます。イベント順序付けを可能にするには、エクスポート・バインディングを適切に構成する必要があります。

#### 関連タスク

128 ページの『EIS エクスポート・バインディングでのイベントの順序付けの有効化』

EIS エクスポート・バインディングでは、イベントの順序付けがサポートされません。バインディングのアクティベーション・スペックを適切に構成する必要があります。

129 ページの『JMS エクスポート・バインディングでのイベントの順序付けの有効化』

JMS エクスポート・バインディングは、イベント順序付けをサポートしています。イベントを受信順に処理するためには、バインディングのプロパティを構成する必要があります。

129 ページの『WebSphere MQ JMS エクスポート・バインディングでのイベントの順序付けの有効化』

WebSphere MQ JMS エクスポート・バインディングは、イベント順序付けをサポートしています。イベントを受信順に処理するためには、バインディングのプロパティを構成する必要があります。

## Network Deployment 環境でのイベント順序付けの使用

イベント順序付けは、HA マネージャーの有無にかかわらず Network Deployment 環境で使用できます。このトピックの表を参照して、使用するトポロジーがサポートされることを確認してください。

イベント順序付けを使用するコンポーネントの Service Component Architecture (SCA) 宛先は分割できません。したがって、クラスターを使用している場合は、クラスターごとに 1 つだけアクティブ・メッセージング・エンジンを持つことができます。

表 16. Network Deployment 環境でのイベント順序付けのサポート

トポロジー	イベント順序付けのサポートの有無
スタンドアロン・サーバー	あり
クラスターなし	あり
アプリケーションをクラスター化している。メッセージング・エンジンと宛先はクラスター化していない。	あり



表 16. Network Deployment 環境でのイベント順序付けのサポート (続き)

トポロジー	イベント順序付けのサポートの有無
メッセージング・エンジンをクラスター化している。アプリケーションと宛先はクラスター化していない。	あり
メッセージング・エンジンと宛先をクラスター化している。アプリケーションはクラスター化していない。	なし。クラスター化された宛先は分割されており、使用の際にイベント順序付けを行うことはできません。
アプリケーションとメッセージング・エンジンをクラスター化している (同じクラスター)。宛先はクラスター化していない。	あり
アプリケーション、メッセージング・エンジン、および宛先をクラスター化している (同じクラスター)。	なし。クラスター化された宛先は分割されており、使用の際にイベント順序付けを行うことはできません。
アプリケーションとメッセージング・エンジンをクラスター化している (別のクラスター)。宛先はクラスター化していない。	あり
アプリケーション、メッセージング・エンジン、および宛先をクラスター化している (別のクラスター)。	なし。クラスター化された宛先は分割されており、使用の際にイベント順序付けを行うことはできません。

## 高可用性環境でのイベント順序付けの使用

高可用性 (HA) サポートとは、イベント順序付けランタイムなどのシステム・サブコンポーネントの可用性が高められており、ノードまたはデーモンの障害発生時にワークロードの分散が可能であることを意味します。

イベント順序付けを行う場合は、宛先でシングルトン・サービスがイベント・メッセージを処理する必要がありますが、この処理を Single Point of Failure としないために必要なサービスを HA マネージャーが提供します。その代わりに、システム異常終了の場合には、イベント順序付けランタイムはクラスター内の別のサーバーにフェイルオーバーします。

## イベント順序付けの制限

特定のタイプのコンポーネントおよび呼び出しでは、イベント順序付けのサポートが制限されます。

現行リリースのイベント順序付けには、以下の制限があります。

- 非開始 receive を行う BPEL (Business Process Execution Language) プロセスにバインドされた操作で、イベント順序付けを使用することはお勧めしません。存続期間の長いビジネス・プロセスでは、イベント順序付けは作業完了規約に基づいてロックの解放時期を判別します。この作業完了規約は、新規プロセス・インスタンスが作成されるときに必ずアクティブ化されます。しかし、非開始 receive を行う場合は新規プロセス・インスタンスが作成されません。結果として、イベント順序付けランタイムが、完了済みの作業規約を正確に検出することが困難となり、ロックの解放が早すぎたり遅すぎたりする可能性があります。
- 非開始 receive を行うビジネス・ステート・マシンにバインドされた操作でイベント順序付けを使用することはお勧めしません。

- 作業完了規約は、BPEL コンポーネントの場合にのみサポートされます。非同期呼び出しを行う他のタイプのコンポーネントでイベント順序付けを効果的に使用するには、要求応答メソッド・シグニチャーの使用をお勧めします。イベント順序付けランタイムは、応答を作業完了のシグナルとして解釈し、ロックを解放します。

**注:** メソッドを要求応答操作として宣言できない場合は、ダウストリーム・コンポーネントでイベント順序付けを指定する必要があります。その場合は、すべてのメソッドに同じイベント順序付けキーを使用していることを確認します。

- 非同期呼び出しを行うコンポーネントに対する同期呼び出しを実行するためのイベント順序付けは、このリリースではサポートされません。

## ロックの管理

ロック・マネージャーはイベント順序付けのロックを処理します。esAdmin コマンドを使用すると、ロック・マネージャー内の任意のロックをリスト、削除、またはアンロックできます。

ロック・マネージャーは、イベント・ロックに対する以下の 2 つの操作をサポートします。

- ロック - ロック操作はロックの獲得を試み、データベースにロック要求を格納します。ロックが付与されると、ロックを要求した呼び出しの処理が再開します。
- アンロック - アンロック操作は現行のロックを解放し、そのロックを次のロック要求に対して付与します。

同じロックに対する要求は、受信順にキューに入れられます。サーバーに障害が発生した場合でもロックを回復できるようにするため、ロックはデフォルトの WebSphere Process Server データベースおよびデータ・ソースに永続性をもって保管されます。

esAdmin コマンドにより、現在ロック・マネージャーでアクティブなロックおよびキューに入っているロックを管理できます。以下のセクションでは、esAdmin の使用方法を詳細に説明します。

**注:** 区画に分割されたデータベースを使用している場合は、デプロイメント・ターゲットごとに 1 回ずつ esAdmin コマンドを実行します。クラスター環境では、任意のクラスターで実行できます。ただし、デプロイメント・マネージャーで実行すると予測不能な結果になることがあります。

## ロックのリスト

esAdmin コマンドは、ロック・マネージャーのアクティブなロックとキューに入っているロックをすべてリストできるほか、特定のモジュール、コンポーネント、またはメソッドに関連したロックのみをリストすることもできます。

esAdmin では以下のいずれかのメソッドを使用します。

- listAll - ロック・マネージャーのアクティブなロックおよびキューに入っているロックをすべてリストします。

- listLocks - ロック・マネージャーのアクティブなロックおよびキューに入っているロックのサブセットをリストします。以下の 1 つ以上のパラメーターを指定すると、ロックのリストがフィルターに掛けられて返されます。

- moduleName
- componentName
- methodName

例えば、

```
esAdmin listLocks CustMod CustComp
```

を使用すると、CusMod モジュールの一部である CustComp コンポーネントについて、アクティブなロックとキューに入っているロックのリストが返されます。

このコマンドは、以下のような出力を返します。

表 17. esAdmin listLocks コマンドの出力例

ロック ID	シーケンス ID	所有者 ID	モジュール	コンポーネント	メソッド	システム・メッセージ ID
7564504	2	695376	CustMod	CustComp	createCust	A09-427BE_5002
7564504	3	232757	CustMod	CustComp	createCust	ADF-053RT_5004

上記の出力において、シーケンス ID はロック要求がキューに入った順序です。シーケンスで最小の番号が、現在ロックを保持しています。システム・メッセージ ID は、対応するサービス統合バス・メッセージの ID を指定します。この情報を使用すると、ロック要求と宛先のメッセージを相互に関連付けることができます。

## ロックの解放

esAdmin コマンドを使用して、ロックを 1 つ解放するには、以下のようにします。

```
esAdmin unlock lockId
```

ここで、lockId は esAdmin listLock または esAdmin listAll コマンドによって戻される固有のロック ID です。

このコマンドは、デッドロックが発生した場合に役立ちます。デッドロック状態のロックを解放し、そのロックをキュー内の次のロック要求に対して付与できます。

## ロックの削除

1 つ以上のロックを削除する必要がある場合は、まず、ロックに関連付けられたモジュールを停止します。次に、以下のように esAdmin コマンドを使用してデータベースからロックを削除します。

```
esAdmin deleteLocks moduleName
```

宛先にイベント・メッセージの処理を再開させるには、モジュールを再始動する必要があります。

esAdmin deleteLocks コマンドを使用するときには十分に注意してください。指定したモジュールのすべてのロックがロック・マネージャー・データベースから削除されます。

## イベント順序付けのトラブルシューティング

イベント順序付けで問題が発生した場合は、このトピックに記載されている情報を参照してください。

### イベント順序付け修飾子に関する問題

以下のようにして、コンポーネント定義が正しいことを確認してください。

- メソッドでイベント順序付け修飾子が設定されていますか？ インターフェースで修飾子が誤って設定されていると、イベント順序付けの検証に失敗します。
- パラメーター名は有効ですか？
- xpath エレメントが有効であり、正しくプリミティブに解決されますか？
- メソッドに eventSequencing エレメントが 1 つだけありますか？ 各メソッドがサポートする eventSequencing エレメントは 1 つのみです。
- メソッドに keySpecification エレメントが 1 つだけありますか？ 各メソッドがサポートする keySpecification エレメントは 1 つのみです。

### デッドロック

ロックを行う操作が呼び出され、さらにその操作が、同じコンポーネントに対し、同じイベント順序付けキーおよびグループを使用して別の操作を呼び出すと、デッドロックが発生します。デッドロックを解決するには、esAdmin コマンドを使用して現行のロックをリストし、解放します。

デッドロックを避けるため、イベント順序付けを実装するときには依存関係を十分に考慮してください。循環依存関係を持つ操作は必ず別のイベント順序付けグループに入れてください。

### BPPEL プロセスでのデッドロック

BPPEL (Business Process Execution Language) プロセスでイベント順序付けを使用する場合にデッドロックが発生することがあります。以下の両方のアクティビティーに対応する操作でイベント順序付け修飾子を設定すると、デッドロックが発生します。

- 複数インスタンス化 receive または pick アクティビティーで createInstance 属性を yes に設定している
- 相関セット指定で initiation 属性を join に設定している

このタイプのデッドロックを解決するには、esAdmin コマンドを使用して現行のロックをリストし、解放します。以後のデッドロックを防ぐため、必ずこれらのタイプの従属操作を別のイベント順序付けグループに入れてください。

## WebSphere Process Server の失敗したイベントの管理

WebSphere Process Server Recovery サービスは、Service Component Architecture (SCA) コンポーネント間で失敗した操作をモニターします。操作が失敗した場合、Recovery サービスは、イベントと障害についてのデータを収集します。その後、Failed Event Manager を使用して、失敗したイベントを表示、変更、再サブミット、または削除できます。

### 失敗したイベントとは

WebSphere Process Server のコンテキストでは、イベントとは WebSphere Process Server アプリケーションによって受け取った要求のことです。外部ソース (インバウンドのアプリケーション・アダプターなど) または Web サービスへの外部呼び出しが発生元である可能性もあります。イベントは、操作対象となるビジネス・ロジックへの参照とデータで構成されており、サービス・データ・オブジェクト (ビジネス・オブジェクト) に保管されています。イベントが受信されると、適切な WebSphere Process Server アプリケーションのビジネス・ロジックで処理されます。

単一の実行スレッドは複数のブランチ (またはスレッド) に分岐できます。個々のブランチは、同じセッション・コンテキストによってメインの起動イベントにリンクされます。

これらのいずれかのブランチのビジネス・ロジックが、システム障害、コンポーネント障害、またはコンポーネントが使用不可であったために完全に実行できない場合、イベントは失敗の状態に変わります。複数のブランチに障害が起こった場合、それぞれについて失敗したイベントが作成されます。WebSphere Process Server Recovery サービスは、以下のタイプの失敗したイベントを処理します。

- Service Component Architecture (SCA) 操作の非同期呼び出し中に発生したイベント障害
- ランタイム例外が原因のイベント障害 (すなわち、ビジネス・ロジックで使用されるメソッドで宣言されなかった例外すべて)

Recovery サービスは、同期呼び出しで発生した障害を処理しません。

一般的に、失敗したイベントには、これに関連付けられたソースおよび宛先情報が含まれます。ソースおよび宛先は、障害点 (呼び出しが失敗したロケーション) に基づくものであり、対話のタイプとは無関係です。コンポーネント A が非同期的にコンポーネント B を呼び出す例について考えます。要求メッセージは A から B に送信され、応答メッセージは B から A に送信されます。

- 最初の要求中に例外が発生した場合、Failed Event Manager にとっては、コンポーネント A がソースでコンポーネント B が宛先になります。
- 応答中に例外が発生した場合、Failed Event Manager にとっては、コンポーネント B がソースでコンポーネント A が宛先になります。

このことは、すべての非同期呼び出しで当てはまります。

### 失敗したイベントの管理方法

Recovery サービスは、失敗した非同期対話を、SCA システム・バス (SCA.SYSTEM.cell\_name.Bus) 上に作成された失敗したイベントの宛先に送信しま

す。失敗したイベントのデータは、失敗したイベントのデータベース (デフォルトでは WPCRSDB) に格納され、Failed Event Manager インターフェースを通じて管理目的で使用できます。

管理者は、Failed Event Manager を使用して、すべての WebSphere Process Server の失敗したイベントをブラウズして管理します。失敗したイベントを管理する一般的なタスクは、以下のとおりです。

- すべての失敗したイベントのブラウズ
- 特定の基準で失敗したイベントの検索
- 失敗したイベントのデータの編集
- 失敗したイベントの再サブミット
- 失敗したイベントの削除

Failed Event Manager にアクセスするには、「**統合アプリケーション**」>「**Failed Event Manager**」をクリックします。

#### 関連概念

322 ページの『失敗したイベントのデータの操作』

失敗したイベントには、それぞれに関連付けられたデータがあります。多くの場合、そのデータはイベントを再サブミットする前に編集可能です。失敗したイベントでは、2 つのデータの基本型があります。イベントに関するデータと、ビジネス・データです。

326 ページの『失敗したイベントの再サブミット』

イベントの実行を再度試行したい場合は、Failed Event Manager から再サブミットする必要があります。イベントは、変更せずに再サブミットすることも、ビジネス・データ・パラメーターを編集してから再サブミットすることもできます。

#### 関連タスク

315 ページの『失敗したイベントの検索』

失敗したイベントは、失敗したイベントのデータベースに格納され、Failed Event Manager の検索機能を使用して検索されます。セル内のすべてのサーバー上の失敗したイベントをすべて検索するか、あるいはイベントの特定のサブセットを検索するかを選択できます。

328 ページの『失敗したイベントの削除』

失敗したイベントを再サブミットしない場合、または失敗したイベントの有効期限が切れた場合、Failed Event Manager を使用してサーバーから削除します。

Failed Event Manager は、失敗したイベントを削除する 3 つのオプションを提供しています。

## リカバリーのセキュリティー考慮事項

WebSphere Process Server アプリケーションおよび環境でセキュリティーを使用可能にしている場合は、役割ベースのアクセスとユーザー ID が Recovery サブシステムにどのような影響を与えるかを理解することが重要です。

#### Failed Event Manager の役割ベースのアクセス

Failed Event Manager は、失敗したイベントのデータおよびタスクに対して、役割ベースのアクセス制御を使用します。管理者ロールおよびオペレーター・ロールのみが、Failed Event Manager 内のタスクの実行を許可され

ています。管理者またはオペレーターのいずれかとしてログインしたユーザーは、失敗したイベントに関連したすべてのデータを表示でき、すべてのタスクを実行できます。

### イベント ID とユーザー権限

失敗したイベントは、要求を発信したユーザーについての情報をカプセル化します。失敗したイベントを再サブミットする場合には、イベントを再サブミットしたユーザーを反映するように、ID 情報が更新されます。管理者またはオペレーターとしてログインした異なるユーザーがイベントを再サブミットできるため、これらのユーザーには、イベントを処理するために必要な下流のコンポーネントに対する権限を付与する必要があります。

セキュリティーの実装について詳しくは、WebSphere Process Server インフォメーション・センターの「アプリケーションとその環境の保護」のトピックを参照してください。

## 失敗したイベントの検索

失敗したイベントは、失敗したイベントのデータベースに格納され、Failed Event Manager の検索機能を使用して検索されます。セル内のすべてのサーバー上の失敗したイベントをすべて検索するか、あるいはイベントの特定のサブセットを検索するかを選択できます。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

このトピックでは、サーバー上のすべての失敗したイベントを検索する方法について説明します。また、セッション ID、ソース、宛先、日付、ビジネス・オブジェクト・タイプ、例外テキスト、またはこれらの基準の組み合わせに基づいてその他の検索を実行するためのトピックへの参照もあります。

失敗したイベントの完全なリストを取得するには、以下の手順を実行します。

1. 管理コンソールが稼働していることを確認します。
2. 「統合アプリケーション」>「Failed Event Manager」をクリックして、Failed Event Manager を始動します。
3. 「このサーバー上の失敗したイベント」ボックスで、「失敗したすべてのイベントを取得」をクリックします。

「検索結果」ページが開き、セル内にある WebSphere Process Server の失敗したイベントすべてのリストが表示されます。

失敗したイベント内のデータを表示して変更したり、失敗したイベントを再サブミットしたり、削除したりできます。

### 関連タスク

316 ページの『セッション ID による失敗したイベントの検索』

「検索」ページの「セッションによる検索」タブを使用すると、特定のセッション内で失敗したイベントのみを検索できます。

317 ページの『ソースによる失敗したイベントの検索』

「検索」ページの「ソースによる検索」タブを使用すると、失敗したイベントのうち、特定のソース・モジュール、ソース・コンポーネント、またはその両方か

ら発信されたイベントのみを検索できます。Failed Event Manager は、対話のタイプに関係なく、障害発生のポイントを基にしてソースを判断します。

317 ページの『宛先による失敗したイベントの検索』

「検索」ページの「**宛先による検索**」タブを使用すると、失敗したイベントのうち、特定の宛先モジュール、宛先コンポーネント、または宛先メソッドに関連付けられたイベントのみを検索できます。Failed Event Manager は、対話のタイプに関係なく、障害発生ポイントを基にして宛先を判断します。

318 ページの『日付による失敗したイベントの検索』

「検索」ページの「**日付ごと**」タブを使用して、特定の期間で失敗したイベントのみを検索できます。

319 ページの『ビジネス・オブジェクト・タイプによる失敗したイベントの検索』

「検索」ページの「**タイプによる検索**」タブを使用すると、失敗したイベントのうち、特定のビジネス・オブジェクトに関連付けられたイベントのみを検索できます。

320 ページの『例外による失敗したイベントの検索』

「検索」ページの「**例外による検索**」タブを使用すると、失敗したイベントのうち、特定の例外に関連付けられたイベントのみを検索できます。例外テキストの一部を指定することも、全部を指定することもできます。

321 ページの『失敗したイベントの拡張検索の実行』

「検索」ページの「**拡張**」タブを使用すれば、他の検索タブにある基準の組み合わせを使用して、失敗したイベントについてさらに詳細な検索を実行できます。

## セッション ID による失敗したイベントの検索

「検索」ページの「**セッションによる検索**」タブを使用すると、特定のセッション内で失敗したイベントのみを検索できます。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

各イベントは 1 つのセッション内で実行されます。そのイベントが失敗した場合、Failed Event Manager は、失敗した実行ブランチに固有のセッション情報をセッション ID パラメーター内にカプセル化します。あるセッションに属するすべてのリソースとプロセス (Common Base Event やビジネス・プロセスなど) には、同じセッション ID が付与されます。

セッション ID によって失敗したイベントを検索するには、以下の手順を実行します。

1. 管理コンソールが実行中であることを確認してから、「**統合アプリケーション**」 > 「**Failed Event Manager**」をクリックして、Failed Event Manager を始動します。
2. Failed Event Manager のメインページで、「**セッションで検索 (Search by session)**」をクリックします。
3. 「**セッション ID**」フィールドを使用して、検索対象のセッション ID を指定します。
4. 「**OK**」をクリックして検索を開始します。



「検索結果」ページが開き、指定したセッション中に発生した、すべての失敗したイベントのリストが表示されます。

## 宛先による失敗したイベントの検索

「検索」ページの「宛先による検索」タブを使用すると、失敗したイベントのうち、特定の宛先モジュール、宛先コンポーネント、または宛先メソッドに関連付けられたイベントのみを検索できます。Failed Event Manager は、対話のタイプに関係なく、障害発生ポイントを基にして宛先を判断します。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

検索を実行する際には、以下の事項に注意してください。

- フィールドの値では、大/小文字が区別されます。
- フィールドでは、アスタリスク (\*) ワイルドカード文字を受け入れます。
- このタブのいずれかのフィールドをブランクのままにすると、ブランク・フィールドはワイルドカードとして扱われます。Failed Event Manager はすべてのコンポーネント、モジュール、またはメソッドで検索を実行します。
- 単一の宛先基準で検索することも、複数の基準で検索することも可能です。複数の宛先基準で検索すると、リストに含める失敗したイベントを絞り込むことができます。

宛先によって失敗したイベントを検索するには、以下の手順を実行します。

1. 管理コンソールが実行中であることを確認してから、「**統合アプリケーション**」>「**Failed Event Manager**」をクリックして、Failed Event Manager を始動します。
2. Failed Event Manager のメインページで、「**宛先で検索**」をクリックします。

「検索」ページが開きます。「宛先による検索」タブが選択された状態になっています。

3. 使用する検索基準を指定します。以下のフィールドを任意に組み合わせて、検索をカスタマイズできます。
  - 「**宛先モジュール**」フィールド - このフィールドを使用して、失敗したイベントの宛先モジュールを指定します。
  - 「**宛先コンポーネント**」フィールド - このフィールドを使用して、失敗したイベントの宛先コンポーネントを指定します。
  - 「**宛先メソッド**」フィールド - このフィールドを使用して、失敗したイベントの宛先メソッドを指定します。
4. 「**OK**」をクリックして検索を開始します。

「検索結果」ページが開き、指定したモジュール、コンポーネント、またはメソッドを宛先として失敗したイベントすべてのリストが表示されます。

## ソースによる失敗したイベントの検索

「検索」ページの「ソースによる検索」タブを使用すると、失敗したイベントのうち、特定のソース・モジュール、ソース・コンポーネント、またはその両方から発

信されたイベントのみを検索できます。Failed Event Manager は、対話のタイプに関係なく、障害発生のポイントを基にしてソースを判断します。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

検索を実行する際には、以下の事項に注意してください。

- フィールドの値では、大/小文字が区別されます。
- フィールドでは、アスタリスク (\*) ワイルドカード文字を受け入れます。
- このタブのいずれかのフィールドをブランクのままにすると、ブランク・フィールドはワイルドカードとして扱われます。Failed Event Manager はすべてのコンポーネントまたはモジュールで検索を実行します。
- 失敗したイベントを最も絞り込んだリストを取得するには、「ソース・モジュール」および「ソース・コンポーネント」フィールドの両方を使用します。

ソースによって失敗したイベントを検索するには、以下の手順を実行します。

1. 管理コンソールが実行中であることを確認してから、「**統合アプリケーション**」>「**Failed Event Manager**」をクリックして、Failed Event Manager を始動します。
2. Failed Event Manager のメインページで、「**ソースで検索**」をクリックします。

「検索」ページが開きます。「**ソースによる検索**」タブが選択された状態になっています。

3. 検索基準を指定します。以下のフィールドのうちのいずれかまたは両方を使用できます。
  - 「**ソース・モジュール**」フィールド - このフィールドを使用して、失敗したイベントの発信元であるモジュールを指定します。
  - 「**ソース・コンポーネント**」フィールド - このフィールドを使用して、失敗したイベントの発信元であるコンポーネントを指定します。
4. 「**OK**」をクリックして検索を開始します。

「検索結果」ページが開き、指定したモジュール、コンポーネント、またはその両方から発信された、すべての失敗したイベントのリストが表示されます。

## 日付による失敗したイベントの検索

「検索」ページの「**日付ごと**」タブを使用して、特定の期間で失敗したイベントのみを検索できます。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

日付による検索を実行するには、以下の事項に注意します。

- 日時のフォーマットはロケール固有です。適切なフォーマットの例が、各フィールドに表示されます。

**注:** 入力する値は、要求されるフォーマットに正確に一致する必要があります。誤ったフォーマットの値を入力すると、Failed Event Manager は警告を表示して、そのフィールドのデフォルト値に置き換えます。

- 時刻は常時、サーバーのローカル時間です。管理コンソールを実行している個々のマシンのローカル時刻を反映するように、時刻が更新されることはありません。
- このタブの両方のフィールドに、値を指定する必要があります。

日付による失敗したイベントの検索を実行するには、以下の手順を実行します。

1. 管理コンソールが実行中であることを確認してから、「**統合アプリケーション**」 > 「**Failed Event Manager**」をクリックして、Failed Event Manager を始動します。
2. Failed Event Manager のメインページで、「**日付で検索**」をクリックします。
3. 「**開始日**」フィールドを使用して、開始日時を指定します。要求される値のフォーマットはロケールによって異なるため、Failed Event Manager では、このフィールドの上にロケールに合った例を表示します。入力した値が、表示された例と同じようにフォーマット設定されていることを確認します。(例えば、en\_US ロケールで要求されるフォーマットは *MM/DD/YY HH:MM Meridiem* です。したがって、このフィールド用に正しいフォーマットで値を設定すると、11/10/05 4:30 PM のようになります。)
4. 「**終了日**」フィールドを使用して、終了日時を指定します。要求される値のフォーマットはロケールによって異なるため、Failed Event Manager では、このフィールドの上にロケールに合った例を表示します。入力した値が、表示された例と同じようにフォーマット設定されていることを確認します。(例えば、en\_US ロケールで要求されるフォーマットは *MM/DD/YY HH:MM Meridiem* です。したがって、このフィールド用に正しいフォーマットで値を設定すると 11/17/05 4:30 PM となります。)
5. 「**OK**」をクリックして検索を開始します。

「検索結果」ページが開き、指定した期間に発生して失敗したイベントすべてのリストが表示されます。

## ビジネス・オブジェクト・タイプによる失敗したイベントの検索

「検索」ページの「**タイプによる検索**」タブを使用すると、失敗したイベントのうち、特定のビジネス・オブジェクトに関連付けられたイベントのみを検索できます。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

ビジネス・オブジェクト・タイプによって失敗したイベントを検索するには、以下の手順を実行します。

1. 管理コンソールが実行中であることを確認してから、「**統合アプリケーション**」 > 「**Failed Event Manager**」をクリックして、Failed Event Manager を始動します。
2. Failed Event Manager のメインページで、「**ビジネス・オブジェクト・タイプで検索**」をクリックします。

「検索」ページが開きます。「**タイプによる検索**」タブが選択された状態になっています。

3. 以下のいずれか 1 つを使用して、検索対象とするビジネス・オブジェクト・タイプを指定します。
  - 「**ビジネス・オブジェクト・タイプの選択 (Select the business object type)**」メニュー - このドロップダウン・メニューを使用して、失敗したイベントに関連付けられたビジネス・オブジェクトのタイプを選択します。このメニューには、サーバー上の失敗したイベントから検出した、すべてのビジネス・オブジェクト・タイプのリストがあります。
  - 「**別のビジネス・オブジェクト・タイプ**」フィールド - このフィールドを使用して、失敗したイベントに関連付けられたビジネス・オブジェクトのタイプを指定します。このフィールドでは、アスタリスク (\*) ワイルドカード文字を受け入れます。すべての値で、大/小文字が区別されます。
4. 「**OK**」をクリックして検索を開始します。

「検索結果」ページが開き、指定したビジネス・オブジェクト・タイプに関連付けられている、すべての失敗したイベントのリストが表示されます。

### 例外による失敗したイベントの検索

「検索」ページの「**例外による検索**」タブを使用すると、失敗したイベントのうち、特定の例外に関連付けられたイベントのみを検索できます。例外テキストの一部を指定することも、全部を指定することもできます。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

例外タイプによって失敗したイベントを検索するには、以下の手順を実行します。

1. 管理コンソールが実行中であることを確認してから、「**統合アプリケーション**」 > 「**Failed Event Manager**」をクリックして、Failed Event Manager を始動します。
2. Failed Event Manager のメインページで、「**例外テキストで検索**」をクリックします。

「検索」ページが開きます。「**例外による検索**」タブが選択された状態になっています。

3. 「**例外テキスト**」フィールドで、検索対象とする例外に関連付けられたテキストを入力します。

例外テキストのすべてまたは一部を指定できますが、アスタリスク (\*) ワイルドカード文字を指定して、検索をより容易にすることもできます。このフィールドの値では、大/小文字が区別されます。

**注:** 「**例外テキスト**」フィールドをブランクのままにすると、ワイルドカードとして扱われます。失敗したイベントがすべて戻されます。

4. 「**OK**」をクリックして検索を開始します。

「検索結果」ページが開き、指定した例外テキストに関連付けられている、すべての失敗したイベントが表示されます。

## 失敗したイベントの拡張検索の実行

「検索」ページの「拡張」タブを使用すれば、他の検索タブにある基準の組み合わせを使用して、失敗したイベントについてさらに詳細な検索を実行できます。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

以下の事項に注意してください。

- 以下で特に断りのない限り、すべてのフィールドはアスタリスク (\*) ワイルドカード文字を受け入れるものとします。
- フィールドをブランクのままにすると、ワイルドカードとして扱われます。
- 拡張検索は最適化されていません。多数の失敗したイベントのセットで拡張検索を実行すると、パフォーマンスが低下することがあります。

拡張検索を実行するには、以下の手順を実行します。

1. 管理コンソールが実行中であることを確認してから、「統合アプリケーション」>「**Failed Event Manager**」をクリックして、Failed Event Manager を始動します。
2. Failed Event Manager のメインページで、「**拡張検索**」をクリックします。

「検索」ページが開きます。「拡張」タブが選択された状態になっています。

3. 使用する検索基準を指定します。以下のフィールドを任意に組み合わせて、検索をカスタマイズできます。
  - 「宛先モジュール」フィールド - このフィールドを使用して、失敗したイベントの宛先モジュールを指定します。
  - 「宛先コンポーネント」フィールド - このフィールドを使用して、失敗したイベントの宛先コンポーネントを指定します。
  - 「セッション ID」フィールド - このフィールドを使用して、イベントを実行したセッションを指定します。このフィールドはアスタリスク (\*) ワイルドカード文字を受け入れません。
  - 「宛先メソッド」フィールド - このフィールドを使用して、失敗したイベントの宛先メソッドを指定します。
  - 「ソース・モジュール」フィールド - このフィールドを使用して、失敗したイベントの発信元であるモジュールを指定します。
  - 「ソース・コンポーネント」フィールド - このフィールドを使用して、失敗したイベントの発信元であるコンポーネントを指定します。
  - 「開始日」フィールド - 特定の期間について検索をしたい場合は、このフィールドを使用して、開始日時を指定します。このフィールドはアスタリスク (\*) ワイルドカード文字を受け入れません。
  - 「終了日」フィールド - 特定の期間について検索をしたい場合は、このフィールドを使用して、終了日時を指定します。このフィールドはアスタリスク (\*) ワイルドカード文字を受け入れません。
  - 「ビジネス・オブジェクト・タイプ」フィールド - このフィールドを使用して、失敗したイベントに関連付けられたビジネス・オブジェクトのタイプを指定します。

- ・ 「例外テキスト」フィールド - このフィールドを使用して、検索の条件とする例外に関連するテキストを指定します。
4. 「OK」をクリックして検索を開始します。

「検索結果」ページが開き、指定した基準を満たす、すべての失敗したイベントのリストが表示されます。

## 失敗したイベントのデータの操作

失敗したイベントには、それぞれに関連付けられたデータがあります。多くの場合、そのデータはイベントを再サブミットする前に編集可能です。失敗したイベントでは、2つのデータの基本型があります。イベントに関するデータと、ビジネス・データです。

### 失敗したイベントに関するデータ

失敗したイベントには、それぞれに以下のデータが関連付けられています。

- ・ イベントに固有のメッセージ ID およびセッション ID
- ・ SCA コンポーネント間のサービス呼び出しタイプ
- ・ イベントが発生したモジュールおよびコンポーネント (ソース) の名前。Failed Event Manager は、呼び出しが失敗したロケーションに基づいて、イベントのソースを判断します。
- ・ イベントの宛先モジュール、コンポーネント、およびメソッドの名前。Failed Event Manager は、呼び出しが失敗したロケーションに基づいて、イベントの宛先を判断します。
- ・ イベントが失敗した時刻
- ・ イベントが失敗したときにスローした例外

このデータは編集できません。また、失敗したイベントには関連したトレースおよび有効期限データが含まれることがあり、両方とも編集可能です。

### ビジネス・データ

イベントは一般的に、ビジネス・データを含んでいます。ビジネス・データは、ビジネス・オブジェクト内にカプセル化することも、ビジネス・オブジェクトの一部でない単純データとすることも可能です。ビジネス・データは、Failed Event Manager 内の使用可能なビジネス・データ・エディターで編集されます。

### 失敗したイベントのデータのブラウズ

失敗したイベント・データや、そのイベントに関連したビジネス・データを表示する場合は、Failed Event Manager を使用します。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

失敗したイベントには、それぞれに以下の2種類のデータが関連付けられています。

- ・ 失敗したイベントのデータ - イベントのソースと宛先、失敗した時刻、失敗したときの例外、メッセージ ID とセッション ID、およびトレースと有効期限の設定など、失敗したイベント自体に関する情報。

- ビジネス・データ - イベントに格納された情報。ビジネス・データは、ビジネス・オブジェクト内にカプセル化したものであっても、ビジネス・オブジェクトの一部ではない単純データであってかまいません。

失敗したイベント・データを表示するには、以下の手順を実行します。

1. Failed Event Manager が開いていること、および失敗したイベントのリストをシステム上で取得したことを確認します。
2. Failed Event Manager の「検索結果」ページで、ブラウズしたい失敗したイベントの ID (「メッセージ ID」列の項目) をクリックします。

「失敗したイベントの詳細」ページが開き、イベントに関する情報のすべてが表示されます。

3. 失敗したイベントに関連付けられたビジネス・データをブラウズする場合、「**ビジネス・データを編集**」をクリックします。

「ビジネス・データ・エディター・コレクション」ページが開き、失敗したイベントに関連付けられたビジネス・データを表示します。階層内の各パラメーター名はリンクです。パラメーターが単純データ型である場合、名前をクリックするとフォームが開き、パラメーターの値を編集できます。パラメーターが複合データ型の場合、名前をクリックすると、さらに階層が展開します。

## 失敗したイベントのトレースおよび有効期限データの編集

「失敗したイベントの詳細」ページでは、失敗したイベントに関連付けられたトレース制御および有効期限の値を設定または変更できます。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

**重要:** トレースまたは有効期限データに対する編集の内容は、イベントを再サブミットするまでローカルにのみ保管されます。イベントを再サブミットする前にその他の何らかのアクションを実行した場合、すべての編集は失われます。

失敗したイベントは、イベントの処理のモニターに役立つようにトレース付きで再サブミットできます。トレースは、サービスまたはコンポーネントに対して設定でき、ログまたは Common Event Infrastructure (CEI) サーバーに送ることができます。「失敗したイベントの詳細」ページで失敗したイベント・データを表示すると、そのイベントのデフォルトのトレース値 `SCA.LOG.INFO;COMP.LOG.INFO` が表示されます。このデフォルトの設定値でイベントを再サブミットした場合、セッションが SCA サービスを呼び出すか、またはコンポーネントを実行したときに、トレースは行われません。

失敗したイベントには有効期限を持つものもあります。ユーザーが、イベントを送信する非同期呼び出しについて有効期限を指定した場合、イベントが失敗してもデータは存在し続け、有効期限時刻は「失敗したイベントの詳細」ページの「**再サブミット有効期限時刻**」フィールドに表示されます。有効期限が切れた失敗したイベントは、正常に再サブミットできません。イベントを再サブミットするときに有効期限が切れないように、イベントの有効期限を編集することにより、2 度目の失敗を回避できます。

失敗したイベントのトレース・データまたは有効期限データを編集するには、以下の手順を実行します。

1. Failed Event Manager が開いていること、および失敗したイベントのリストをシステム上で取得したことを確認します。
2. Failed Event Manager の「検索結果」ページで、データを編集したい失敗したイベントの ID (「メッセージ ID」列の項目) をクリックします。

「失敗したイベントの詳細」ページが開きます。

3. イベントに有効期限があり、再サブミットする前に有効期限が切れる場合、「再サブミット有効期限時刻」フィールドの有効期限を編集します。

表示されている有効期限時刻は、サーバーのローカル時間です。このフィールドの値は、指定したロケールに応じたフォーマットにする必要があります。ロケールに応じた正しいフォーマットの例が、フィールドの上に表示されます。

4. 失敗したイベントのトレースを使用可能に設定する場合、「**トレース制御**」フィールドに新しい値を指定します。トレース値について詳しくは、WebSphere Process Server インフォメーション・センターの『モニター』のトピックを参照してください。
5. 以下のいずれかを実行します。
  - 編集したデータが正しく、イベントを再サブミットする場合、「再サブミット」をクリックして、サーバー・レベルで変更を行います。
  - 加えた変更を除去する場合、「**ローカルの変更を取り消す**」をクリックします。

編集済みの失敗したイベントは、処理されるように再サブミットされ、Failed Event Manager から除去されます。

### 関連タスク

315 ページの『失敗したイベントの検索』

失敗したイベントは、失敗したイベントのデータベースに格納され、Failed Event Manager の検索機能を使用して検索されます。セル内のすべてのサーバー上の失敗したイベントをすべて検索するか、あるいはイベントの特定のサブセットを検索するかを選択できます。

## 失敗したイベントのビジネス・データの編集

ビジネス・データは、ビジネス・オブジェクト内にカプセル化したものであっても、ビジネス・オブジェクトの一部でない単純データであってもかまいません。失敗したイベントには、単純データとそれに関連付けられたビジネス・オブジェクトの両方が含まれることもあります。失敗したイベントを再サブミットする前に、ビジネス・データ・エディターを使用して、失敗したイベントに関連したビジネス・データを編集します。

このタスクを実行するには、管理者またはオペレーターとしてログインする必要があります。

それぞれの失敗したイベントについて、エディターでは関連ビジネス・データが階層形式で表示されます。表の上部のナビゲーション・ツリーが、パラメーターをナビゲートするにつれて更新されるため、階層内のどの場所にいるかを明確に把握できます。



編集できるのは、単純データ型 (String、Long、Integer、Date、Boolean など) のみです。データ型が複合型 (配列またはビジネス・オブジェクトなど) の場合、配列またはビジネス・オブジェクトを構成する単純データ型に達するまでビジネス・データ階層をナビゲートする必要があります。複合データは、「パラメーター値」列に省略符号 (...) で示されます。

**重要:** ビジネス・データに対する編集の内容はすべて、ローカルで保管されます。失敗したイベントを再サブミットするまで、サーバー上の対応するビジネス・データは変更されません。

失敗したイベントに関連したビジネス・データを編集するには、以下の手順を実行します。

1. Failed Event Manager が開いていること、および失敗したイベントのリストをシステム上で取得したことを確認します。
2. Failed Event Manager の「検索結果」ページで、データを編集したい失敗したイベントの ID (「メッセージ ID」列の項目) をクリックします。

「失敗したイベントの詳細」ページが開きます。

3. 「失敗したイベントの詳細」ページで「**ビジネス・データを編集**」をクリックして、「ビジネス・データ・エディター・コレクション」ページにアクセスします。

このページでは、失敗したイベントに関連付けられたすべてのデータが階層図で表示されます。

4. 各パラメーターの名前をクリックして、ビジネス・データ階層をナビゲートします (これらは「パラメーター名」列でリンクとして表示されます)。値を編集したいパラメーターを見つけたら、その名前をクリックします。

パラメーターの値が編集可能な場合、「ビジネス・データ・エディター」ページが開きます。

5. 「パラメーター値」フィールドで、パラメーターの新しい値を指定します。
6. 「**OK**」をクリックします。

変更内容はローカルに保管され、「ビジネス・データ・エディター・コレクション」ページに戻ります。

7. 加えた変更を除去する場合、「**ローカル・ビジネス・データの変更を取り消す**」をクリックします。

すべての編集内容が除去され、ビジネス・データは元の状態に戻ります。

8. 編集したビジネス・データが正しければ、「**再サブミット**」をクリックして、サーバー・レベルで変更を行います。

編集済みの失敗したイベントは、処理されるように再サブミットされ、Failed Event Manager から除去されます。

## 失敗したイベントの再サブミット

イベントの実行を再度試行したい場合は、Failed Event Manager から再サブミットする必要があります。イベントは、変更せずに再サブミットすることも、ビジネス・データ・パラメーターを編集してから再サブミットすることもできます。

失敗したイベントを再サブミットすると、イベント全体ではなく、失敗したブランチに対してのみ処理が再開されます。

再サブミットしたイベントでは、イベントの処理のモニターに役立つようにトレースが使用可能になっています。トレースはサービスまたはコンポーネントに対して設定でき、その出力はログまたは Common Event Infrastructure (CEI) サーバーに送信できます。

また、イベント固有のメッセージ ID を使用して、イベントの成功または失敗を追跡できます。再サブミットしたイベントが再び失敗した場合、そのイベントは、元のメッセージ ID および更新された失敗の時刻と共に Failed Event Manager に戻されます。

### 未変更の失敗したイベントの再サブミット

1 つ以上の未変更の失敗したイベントを再サブミットして、再度処理させることができます。イベント全体ではなく、失敗したブランチに対してのみ、処理が再開されます。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. Failed Event Manager が開いていること、および失敗したイベントのリストをシステム上で取得したことを確認します。
2. 「検索結果」ページで、再サブミットしようとする失敗したイベントの横のチェック・ボックスをそれぞれ選択します。
3. 「再サブミット」をクリックします。

選択した各イベントは処理されるように再サブミットされ、Failed Event Manager から除去されます。

### 失敗したイベントのトレース付きでの再サブミット

失敗したイベントの再サブミットをモニターすることによって、失敗したイベントが正常に実行されるかどうかを判断できます。Failed Event Manager は、すべての失敗したイベント用に、トレースのオプションを提供しています。

トレースは、サービスまたはコンポーネントに対して設定でき、ログまたは Common Event Infrastructure (CEI) サーバーに出力できます。トレースの設定および表示について詳しくは、WebSphere Process Server インフォメーション・センターの『モニター』のトピックを参照してください。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. Failed Event Manager が開いていること、および失敗したイベントのリストをシステム上で取得したことを確認します。

2. 「検索結果」ページで、再サブミットしようとする失敗したイベントの横のチェック・ボックスをそれぞれ選択します。
3. 「トレース付き再サブミット」をクリックします。
4. 「トレース付き再サブミット」ページで、使用するトレースのレベルを「トレース制御」フィールドに指定します。

デフォルトでは、この値は `SCA.LOG.INFO;COMP.LOG.INFO` です。この設定値では、セッションが SCA サービスを呼び出すか、またはコンポーネントを実行したときに、トレースは行われません。

5. 「OK」をクリックして失敗したイベントを再サブミットし、「検索結果」ページに戻ります。

再サブミットしたイベントのトレース・ログを表示するには、対応するコンポーネントのロガーを開くか、または CEI ログ・ビューアーを使用します。

## 失敗したイベントに関連する Common Base Event の検索

失敗したイベントは、1 つ以上の Common Base Event に関連していることがあります。Failed Event Manager は、Common Base Event ブラウザー内に関連 Common Base Event を表示するためのリンクを提供します。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

関連 Common Base Event を調べることによって、元のイベントが失敗した状況や原因に関する追加情報が得られます。失敗したイベントと関連 Common Base Event は、同じセッション ID によってリンクされます。

関連 Common Base Event を見つけて表示するには、以下の手順を実行します。

1. 管理コンソールの内部から、Failed Event Manager を使用して、調べようとする失敗したイベントを検索します。失敗したイベントの検索方法の説明については、315 ページの『失敗したイベントの検索』を参照してください。
2. そのイベントの「失敗したイベントの詳細」ページから、「**関連の Common Base Event の参照**」をクリックします。

新規ブラウザー・ウィンドウで Common Base Event ブラウザーが開き、失敗した元のイベントに関連するすべての Common Base Event がリストされます。

## 失敗したイベントに関連するビジネス・プロセス・インスタンスの検索

失敗したイベントがビジネス・プロセスから生成された場合、Failed Event Manager は、Business Process Choreographer Explorer にそのビジネス・プロセス・インスタンスを表示するためのリンクを提供します。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

失敗したイベントを生成したビジネス・プロセス・インスタンスを調べることによって、イベントが失敗した状況や原因に関する追加情報が得られます。ビジネス・プロセス・インスタンスと失敗したイベントは、共通のセッション ID によってリンクされます。

注: すべての失敗したイベントがビジネス・プロセス・インスタンスから生成されるわけではありません。

失敗したイベントに関連するビジネス・プロセス・インスタンスを検索して調べるには、以下の手順を実行します。

1. 管理コンソールの内部から、Failed Event Manager を使用して、調べようとする失敗したイベントを検索します。失敗したイベントの検索方法の説明については、315 ページの『失敗したイベントの検索』を参照してください。
2. そのイベントの「失敗したイベントの詳細」ページから、「呼び出しプロセスを Business Process Explorer で開きます」をクリックします。

新規ブラウザ・ウィンドウで Business Process Choreographer Explorer が開き、関連プロセス・インスタンスについての情報が表示されます。

## 失敗したイベントの削除

失敗したイベントを再サブミットしない場合、または失敗したイベントの有効期限が切れた場合、Failed Event Manager を使用してサーバーから削除します。Failed Event Manager は、失敗したイベントを削除する 3 つのオプションを提供しています。

このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1 つ以上の失敗したイベントを削除するには、以下の手順を実行します。

1. Failed Event Manager が開いていること、および失敗したイベントのリストをシステム上で取得したことを確認します。
2. Failed Event Manager の「検索結果」ページから、以下のいずれかを実行します。
  - 1 つ以上の特定の失敗したイベントを削除する場合、各イベントの横のチェック・ボックスを選択してから、「削除」をクリックします。
  - 失敗したイベントのうち有効期限が切れたものだけを削除する場合、「期限切れイベントを削除」をクリックします。このようにすると、現在の一連の検索結果から、有効期限が切れたイベントのみが削除されます。
  - サーバー上のすべての失敗したイベントを削除するには、「サーバー上のすべてをクリア」をクリックします。

## Failed Event Manager のトラブルシューティング

このトピックでは、Failed Event Manager の使用中に発生する可能性がある問題について説明します。

注: このトピックでは、システムで失敗したイベントを検出、変更、再サブミット、または削除するために Failed Event Manager を使用する方法については説

明しません。失敗したイベントの管理については、インフォメーション・センターの『WebSphere Process Server の失敗したイベントの管理』を参照してください。

以下の表から、発生した問題を選択します。

問題	参照先
拡張検索中にパフォーマンスが低下する	『拡張検索機能が最適化されていない』
「検索」ページの「日付ごと」タブに値を入力できない	『入力した値が正しくない場合に「日付ごと」タブの値が自動的にデフォルトに変更される』
期限切れのイベントを削除できない	330 ページの『「期限切れイベントを削除」機能を実行すると Failed Event Manager が中断するように見える』
失敗したイベントが作成されない	330 ページの『失敗したイベントが作成されない』

## 拡張検索機能が最適化されていない

Failed Event Manager の拡張検索機能が最適化されていません。したがって、失敗したイベントの大量のセットに対して「拡張検索」タブを使用する場合に、パフォーマンスが低下する可能性があります。

## 入力した値が正しくない場合に「日付ごと」タブの値が自動的にデフォルトに変更される

「検索」ページの「日付ごと」タブには、「開始日」と「終了日」という 2 つのフィールドがあります。どちらのフィールドも必須です。値はロケールに応じて異なり、フィールドの上に示される例に従って正確な形式で指定する必要があります。値の形式に不整合があると (年数に 2 桁ではなく 4 桁の数字を使用した、時間値を省略した、など)、Failed Event Manager は以下の警告メッセージを発行し、フィールドをデフォルト値に置き換えます。

CWMAN0017E: 入力された日付 *your\_incorrectly\_formatted\_date* が正しく解析されませんでした。日付 *default\_date* が使用されます。

「開始日」フィールドのデフォルト値は、January 1, 1970, 00:00:00 GMT に定義されています。

**重要:** Failed Event Manager のインプリメンテーションで表示される実際のデフォルト値は、ロケールおよびタイム・ゾーンによって異なります。例えば、米東部標準時 (EST) のタイム・ゾーンにあるマシンでロケールが en\_US に設定されている場合、「開始日」フィールドのデフォルト値は 12/31/69 7:00 PM になります。

「終了日」フィールドのデフォルト値は、常に現在の日付および時刻であり、ご使用のロケールおよびタイム・ゾーンに応じた形式で表示されます。

この問題を回避するには、各フィールドの上に示される例に従って、日付と時刻を常に注意深く入力してください。

## 「期限切れイベントを削除」機能を実行すると Failed Event Manager が中断するよう見える

現在の検索結果に失敗したイベントが多く含まれている場合、または失敗したイベントに大量のビジネス・データが含まれている場合に、「期限切れイベントを削除」ボタンを使用すると、Failed Event Manager が無期限に中断しているように見える場合があります。

この場合、Failed Event Manager は大量のデータ・セットを処理しているので、実際には中断していません。コマンドが完了すると、結果セットが最新表示されます。

### 失敗したイベントが作成されない

失敗したイベントが Recovery サブシステムによって作成されない場合には、以下のチェックリストを参照して、問題の原因を確認してください。

- wpsFEMgr アプリケーションが実行中であることを確認します。必要な場合は、再始動します。
- Failed Event Manager のデータベースが作成済みであり、接続がテスト済みであることを確認します。
- 必要とされる失敗したイベントの宛先が SCA システム・バス上で作成済みであることを確認します。各デプロイメント・ターゲットに対して 1 つの失敗したイベントの宛先が必要です。
- Recovery サービスに処理させるイベントに参加するすべての Service Component Architecture (SCA) インプリメンテーション、インターフェース、またはパートナー参照に対して、サービス品質 (QoS) の **Reliability** 修飾子が Assured に設定されていることを確認します。

---

## WebSphere Process Server 管理のトラブルシューティング

トラブルシューティングは、問題の原因を検出して除去するプロセスです。以下の各トピックは、標準的な管理タスク中に発生する可能性のある問題を識別して解決するために役立ちます。

### Failed Event Manager のトラブルシューティング

このトピックでは、Failed Event Manager の使用中に発生する可能性がある問題について説明します。

**注:** このトピックでは、システムで失敗したイベントを検出、変更、再サブミット、または削除するために Failed Event Manager を使用方法については説明しません。失敗したイベントの管理について詳しくは、インフォメーション・センターの『*WebSphere Process Server* の失敗したイベントの管理』を参照してください。

以下の表から、発生した問題を選択します。

問題	参照先
拡張検索中にパフォーマンスが低下する	329 ページの『拡張検索機能が最適化されていない』

問題	参照先
「検索」ページの「日付ごと」タブに値を入力できない	329 ページの『入力した値が正しくない場合に「日付ごと」タブの値が自動的にデフォルトに変更される』
期限切れのイベントを削除できない	330 ページの『「期限切れイベントを削除」機能を実行すると Failed Event Manager が中断するように見える』
失敗したイベントが作成されない	330 ページの『失敗したイベントが作成されない』

## 拡張検索機能が最適化されていない

Failed Event Manager の拡張検索機能が最適化されていません。したがって、失敗したイベントの大量のセットに対して「拡張検索」タブを使用する場合に、パフォーマンスが低下する可能性があります。

### 入力した値が正しくない場合に「日付ごと」タブの値が自動的にデフォルトに変更される

「検索」ページの「日付ごと」タブには、「開始日」と「終了日」という 2 つのフィールドがあります。どちらのフィールドも必須です。値はロケールに応じて異なり、フィールドの上に示される例に従って正確な形式で指定する必要があります。値の形式に不整合があると（年数に 2 桁ではなく 4 桁の数字を使用した、時間値を省略した、など）、Failed Event Manager は以下の警告メッセージを発行し、フィールドをデフォルト値に置き換えます。

CWMAN0017E: 入力された日付 *your\_incorrectly\_formatted\_date* が正しく解析されませんでした。日付 *default\_date* が使用されます。

「開始日」フィールドのデフォルト値は、January 1, 1970, 00:00:00 GMT に定義されています。

**重要:** Failed Event Manager のインプリメンテーションで表示される実際のデフォルト値は、ロケールおよびタイム・ゾーンによって異なります。例えば、米東部標準時 (EST) のタイム・ゾーンにあるマシンでロケールが en\_US に設定されている場合、「開始日」フィールドのデフォルト値は 12/31/69 7:00 PM になります。

「終了日」フィールドのデフォルト値は、常に現在の日付および時刻であり、ご使用のロケールおよびタイム・ゾーンに応じた形式で表示されます。

この問題を回避するには、各フィールドの上に示される例に従って、日付と時刻を常に注意深く入力してください。

### 「期限切れイベントを削除」機能を実行すると Failed Event Manager が中断するように見える

現在の検索結果に失敗したイベントが多く含まれている場合、または失敗したイベントに大量のビジネス・データが含まれている場合に、「期限切れイベントを削除」ボタンを使用すると、Failed Event Manager が無期限に中断しているように見える場合があります。

この場合、Failed Event Manager は大量のデータ・セットを処理しているので、実際には中断していません。コマンドが完了すると、結果セットが最新表示されます。

## 失敗したイベントが作成されない

失敗したイベントが Recovery サブシステムによって作成されない場合には、以下のチェックリストを参照して、問題の原因を確認してください。

- wpsFEMgr アプリケーションが実行中であることを確認します。必要な場合は、再始動します。
- Failed Event Manager のデータベースが作成済みであり、接続がテスト済みであることを確認します。
- 必要とされる失敗したイベントの宛先が SCA システム・バス上で作成済みであることを確認します。各デプロイメント・ターゲットに対して 1 つの失敗したイベントの宛先が必要です。
- Recovery サービスに処理させるイベントに参加するすべての Service Component Architecture (SCA) インプリメンテーション、インターフェース、またはパートナー参照に対して、サービス品質 (QoS) の **Reliability** 修飾子が Assured に設定されていることを確認します。

## Service Component Architecture と WebSphere MQ の通信のトラブルシューティング

Service Component Architecture (SCA) モジュールと WebSphere MQ キュー・マネージャーの間の通信は、SCA モジュール内のインポートおよびエクスポートと WebSphere MQ サーバー内のキューとの間のバインディングに依存します。この情報を使用して、WebSphere MQ メッセージを処理していないサーバーを判別します。

このタスクは、管理コンソールにアクセスできるユーザーが、WebSphere MQ に依存する要求が処理されていないことを認識した場合を前提とします。また、このユーザーは、WebSphere MQ キュー・マネージャーの変更を自分で実行できるか、あるいは変更を WebSphere MQ 管理者に依頼できる必要があります。

Service Component Architecture (SCA) モジュールは、サーバーと WebSphere MQ キュー・マネージャーの間のバインディングに依存します。2 つのエンティティーの間の通信が原因で、メッセージが完全に処理されないことがあります。以下のステップは、分断の原因を特定し、メッセージが再び処理されるようにするために役立ちます。

1. WebSphere MQ と通信している SCA モジュールを表示して、このモジュールが現在も処理中であることを確認します。このページにナビゲートするには、「アプリケーション」>「SCA モジュール」を使用します。
2. キュー・マネージャーを表示して、キュー・マネージャーが現在も作動可能であることを確認します。このタスクは、WebSphere MQ 管理コンソールで実行します。
3. SCA モジュールとキュー・マネージャーの間のバインディングを表示して、バインディングが正しいことを確認します。バインディングが正しくない場合は、バインディングを変更します。このページにナビゲートするには、「リソー



ス」>「JMS プロバイダー」>「WebSphere MQ」>「WebSphere MQ キュー宛先 (WebSphere MQ queue destinations)」を使用します。

4. 失敗したトランザクションを示している可能性のあるメッセージを見つけます。システム、SCA 固有のメッセージ領域、MQ 固有のメッセージ領域、失敗したイベント・キュー、およびその他の場所を調べて、障害の対象を特定する必要があります。
  - a. SystemOut.log を調べて、処理の障害を示すメッセージがないか確認します。

MQ エラーがあった場合、スタック・トレース内のどこかの場所にリンクした MQException と MQ 理由コード (例: 2059 は「キュー・マネージャーが使用不可です (queue manager unavailable)」) が存在します。
  - b. AMQERRxx.LOG および WebSphere MQ FFDC ファイルをチェックして、WebSphere MQ エラーの原因を確認します。
  - c. アプリケーション・キューを調べて、未処理のメッセージがあるかどうか確認します。WebSphere MQ とサービス統合バス (SIB) アプリケーションの両方を必ず調べます。
  - d. MQ の送達不能キューと SIB 例外の宛先を調べます。
  - e. 失敗イベント・キューを調べて、対象となるアプリケーションに関連したメッセージがあるかどうか確認します。失敗したイベントを探す方法については、『WebSphere Process Server の失敗したイベントの管理』を参照してください。

## イベント順序付けのトラブルシューティング

イベント順序付けで問題が発生した場合は、このトピックに記載されている情報を参照してください。

### イベント順序付け修飾子に関する問題

以下のようにして、コンポーネント定義が正しいことを確認してください。

- メソッドでイベント順序付け修飾子が設定されていますか? インターフェースで修飾子が誤って設定されていると、イベント順序付けの検証に失敗します。
- パラメーター名は有効ですか?
- xpath エレメントが有効であり、正しくプリミティブに解決されますか?
- メソッドに eventSequencing エレメントが 1 つだけありますか? 各メソッドがサポートする eventSequencing エレメントは 1 つのみです。
- メソッドに keySpecification エレメントが 1 つだけありますか? 各メソッドがサポートする keySpecification エレメントは 1 つのみです。

### デッドロック

ロックを行う操作が呼び出され、さらにその操作が、同じコンポーネントに対し、同じイベント順序付けキーおよびグループを使用して別の操作を呼び出すと、デッドロックが発生します。デッドロックを解決するには、esAdmin コマンドを使用して現行のロックをリストし、解放します。

デッドロックを避けるため、イベント順序付けを実装するときには依存関係を十分に考慮してください。循環依存関係を持つ操作は必ず別のイベント順序付けグループに入れてください。

### **BPEL プロセスでのデッドロック**

BPEL (Business Process Execution Language) プロセスでイベント順序付けを使用する場合にデッドロックが発生することがあります。以下の両方のアクティビティーに対応する操作でイベント順序付け修飾子を設定すると、デッドロックが発生します。

- 複数インスタンス化 `receive` または `pick` アクティビティーで `createInstance` 属性を `yes` に設定している
- 相関セット指定で `initiation` 属性を `join` に設定している

このタイプのデッドロックを解決するには、`esAdmin` コマンドを使用して現行のロックをリストし、解放します。以後のデッドロックを防ぐため、必ずこれらのタイプの従属操作を別のイベント順序付けグループに入れてください。

---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

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

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

〒106-8711  
東京都港区六本木 3-2-12  
IBM World Trade Asia Corporation  
Intellectual Property Law & Licensing

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

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

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

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

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

*IBM Corporation*  
*577 Airport Blvd., Suite 800*  
*Burlingame, CA 94010*  
*U.S.A.*

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

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

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

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

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

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

著作権使用許諾:

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

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。© (お客様の会社名) (西暦年)。このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. \_年を入れる\_。 All rights reserved.

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

---

## プログラミング・インターフェース情報

プログラミング・インターフェース情報は、このプログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツールのサービスを利用するアプリケーション・ソフトウェアを作成できます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

**警告:** 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

---

## 商標

IBM および関連する商標: <http://www.ibm.com/legal/copytrade.shtml>

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

この製品には、Eclipse Project (<http://www.eclipse.org/>) により開発されたソフトウェアが含まれています。



WebSphere Process Server for z/OS バージョン 6.0.2







Printed in Japan