





お願い

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

本書は、WebSphere Process Server for Multiplatforms バージョン 6、リリース 1、モディフィケーション 0 (製品番号 5724-L01)、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

IBM 発行のマニュアルに関する情報のページ

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

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

(URL は、変更になる場合があります)

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

原 典： WebSphere® Process Server for Multiplatforms
Version 6.1.0
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

目次

図	v
表	vii
第 1 章 WebSphere Process Server の管理の概要	1
WebSphere Process Server の管理コンソール	1
管理コンソールのガイド付きアクティビティ	4
管理コンソール・ページ	5
管理コンソールのボタン	6
コマンド行ツール、スクリプト、およびプログラミング・インターフェース	8
Business Process Choreographer Explorer について	9
Business Process Choreographer Observer について	10
ビジネス・ルール・マネージャー	12
構成情報	13
第 2 章 管理インターフェース入門	15
管理コンソール入門	15
管理コンソールの始動と停止	16
管理コンソール設定の指定	19
管理コンソール・フィルターの設定	19
「マイ・タスク」の使用法	20
Business Process Choreographer Explorer 入門	21
Business Process Choreographer Explorer の開始	27
Business Process Choreographer Explorer のカスタマイズ	28
Business Process Choreographer Observer 入門	38
第 3 章 サーバーおよびクラスターの管理	43
サーバーの作成	43
サーバーまたはクラスターの SCA サポートの構成	44
サーバーおよびクラスター上での Service Component Architecture サポートに関する考慮事項	46
サーバーおよびクラスターの開始または停止	47
スタンドアロン・サーバーの開始	47
スタンドアロン・サーバーの停止	49
デプロイメント・マネージャーの開始	51
デプロイメント・マネージャーの停止	52
ノード・エージェントの開始	54
ノード・エージェントの停止	55
ノード・エージェントの再開	56
デプロイメント環境の開始と停止	57
クラスターの開始	58
クラスターの停止	59
サーバーの開始	60
サーバーの停止	62
第 4 章 デプロイメント環境の管理	65

カスタム・デプロイメント環境のレイアウト構成の概要	66
デプロイメント環境の作成	73
デプロイメント環境の機能の構成	76
ホスト別名の構成	77
デプロイメント環境の認証別名の構成	78
カスタム・デプロイメント環境の構成	79
デプロイメント環境の据え置かれた構成の構成	82
コマンド行を使用したデプロイメント環境の構成	82
デプロイメント・トポロジーの表示	83
デプロイメント環境の編集	86
データ・ソース構成の編集	88
データベース・プロバイダーの編集	89
デプロイメント環境のデータ・ソースの構成	90
デプロイメント環境内のデータ・ソースの編集	91
デプロイメント・マネージャーの停止と再始動	92
クラスター・メンバーの停止および再始動	93
デプロイメント環境の開始と停止	94
デプロイメント環境定義のエクスポート	95
コマンド行を使用したデプロイメント環境定義のエクスポート	96
デプロイメント環境定義のインポート	98
コマンド行を使用したデプロイメント環境定義のインポート	100
デプロイメント環境の除去	102

第 5 章 Business Process Choreographer の管理 105

第 6 章 Common Event Infrastructure の構成および管理 107

第 7 章 サービス・コンポーネントの管理 109

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	109
ビジネス・ステート・マシンのインスタンスの検索	110
表示状態の表示	110
ビジネス・ルールおよびセレクターの管理	111
ビジネス・ルールとセレクターの監査ログの構成	111
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	113
ビジネス・ルールおよびセレクターを含むモジュールの考慮事項	116
ビジネス・ルールの概要	118
ビジネス・ルール・マネージャー	122
セレクター・コンポーネントの概要	165
バインディング	173
SCA EIS バインディング	174

JMS バインディング	181
汎用 JMS バインディング	191
WebSphere MQ JMS バインディング	201
WebSphere MQ バインディング	210
HTTP バインディング	221
データ・バインディング	232
エクスポート・バインディングとイベント順序付け	236
バインディングのトラブルシューティング	240

第 8 章 アダプター 277

WebSphere Adapter	280
WebSphere Business Integration Adapter	280
アダプターのインストールおよび構成	281
WebSphere Business Integration Adapter の管理のセットアップ	281
WebSphere Business Integration Adapter の管理	283

第 9 章 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 285

サービス・アプリケーションの管理	285
サービス・アプリケーションの使用開始	285
実動サーバーへのモジュールのインストール	294
メディアーション・モジュールの管理	298
WSRR へのアクセスの管理	309
SCA 要求のスループットの管理	312
サービス・アプリケーションのその他の用途	314
デプロイされた SCA モジュールの WebSphere MQ JMS 宛先の変更	328
Service Component Architecture モジュールと WebSphere MQ	330
ターゲットの使用	331
ターゲットの変更	332
J2C アクティベーション・スペックの削除	334
SIBus 宛先の削除	335
エンタープライズ・アプリケーションの管理	336
Application Scheduler	338
Application Scheduler へのアクセス	339
Application Scheduler MBean インターフェースを使用した Application Scheduler へのアクセス	339
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	341
リレーションシップの管理	345
リレーションシップの表示	346
リレーションシップ詳細の表示	347
ロール詳細の表示	347
リレーションシップの照会	348
リレーションシップ・インスタンスの表示	349
リレーションシップ・インスタンス詳細の表示	350
リレーションシップ・インスタンスの詳細の編集	351
リレーションシップ・インスタンスの新規作成	352
リレーションシップ・インスタンスの削除	353
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	354
ロール・インスタンス詳細の表示	355

ロール・インスタンス・プロパティの編集	355
ロール・インスタンスの新規作成	356
ロール・インスタンスの削除	357
リポジトリからのリレーションシップ・インスタンス・データの除去	358
チュートリアル: Relationship Manager の管理	360
リレーションシップ・サービスの管理	363
リレーションシップ・サービスの構成	364
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	365
リレーションシップ・プロパティの表示	365
拡張メッセージング・リソースの管理	366
拡張メッセージング・サービスの使用可能化	367
遅延応答を処理するリスナー・ポート拡張の構成	368
拡張メッセージング・プロバイダーの選択	369

第 10 章 イベントの使用 375

イベントの順次処理	375
例: イベント順序付け	376
イベント順序付けを実装する場合の考慮事項	378
ロックの管理	382
イベント順序付けのトラブルシューティング	384
失敗したイベントの管理	385
リカバリーのセキュリティ考慮事項	387
失敗したイベントの検索	387
失敗したイベントのデータの操作	394
失敗したイベントの再サブミット	398
失敗したイベントに関連する Common Base Event の検索	400
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	401
失敗したイベントの削除	401
Failed Event Manager のトラブルシューティング	402

第 11 章 WebSphere Process Server のトラブルシューティング管理 405

Failed Event Manager のトラブルシューティング	405
Service Component Architecture と WebSphere MQ の通信のトラブルシューティング	407
イベント順序付けのトラブルシューティング	408
デプロイメント環境のトラブルシューティング	409
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	410
ログイン・エラーの解決	410
ログイン競合エラーの解決	411
アクセス競合エラーの解決	412
バインディングのトラブルシューティング	412
JMS および WebSphere MQ JMS バインディングのトラブルシューティング	413
汎用 JMS バインディングのトラブルシューティング	416
WebSphere MQ バインディングのトラブルシューティング	417

特記事項 419



1. ガイド付きアクティビティ	5	15. WebSphere MQ エクスポート・バインディング・リソース	212
2. Business Process Choreographer Observer のアーキテクチャー	10	16. JMS インポート・バインディングのリソース	254
3. Business Process Choreographer Observer のスタンドアロン・セットアップ構成	11	17. JMS エクスポート・バインディングのリソース	255
4. 実稼働時のパフォーマンスに合わせた Business Process Choreographer Observer のセットアップ構成	12	18. 汎用 JMS インポート・バインディング・リソース	257
5. ビジネス・ルール・マネージャーのイベント・シーケンス	123	19. 汎用 JMS エクスポート・バインディング・リソース	258
6. デシジョン・テーブル	158	20. WebSphere MQ JMS インポート・バインディングのリソース	260
7. JMS インポート・バインディングのリソース	183	21. WebSphere MQ JMS エクスポート・バインディングのリソース	261
8. JMS エクスポート・バインディングのリソース	184	22. WebSphere MQ インポート・バインディング・リソース	262
9. 単純なユース・ケース・シナリオ: 外部クライアントとサーバー・アプリケーションの対話	190	23. WebSphere MQ エクスポート・バインディング・リソース	263
10. 汎用 JMS インポート・バインディング・リソース	193	24. WebSphere Adapter の詳細な図式	278
11. 汎用 JMS エクスポート・バインディング・リソース	194	25. WebSphere Business Integration Adapter の詳細な図式	279
12. WebSphere MQ JMS インポート・バインディングのリソース	202	26. メディエーション・モジュールの簡単な例	288
13. WebSphere MQ JMS エクスポート・バインディングのリソース	203	27. 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	289
14. WebSphere MQ インポート・バインディング・リソース	211	28. 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	294

表

1. WebSphere Process Server 構成ファイル	14	15. 提供される HTTP ヘッダー情報	226
2. コンポーネント、コンポーネント間の相互関係、および考慮事項	68	16. JMS インポート・バインディング: サーバー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	264
3. デプロイメント環境の状況	84	17. JMS エクスポート・バインディング: サーバー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	264
4. エンティティの状態の集合	85	18. 汎用 JMS インポート: サーバー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	265
5. トポロジー・インスタンスの使用可能状況 (使用可能性が低い順から高い順)	86	19. 汎用 JMS エクスポート: サーバー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	265
6. 機能ボタン	133	20. WebSphere MQ インポート: サーバー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	266
7. JMS インポート・バインディング: サーバー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	187	21. WebSphere MQ エクスポート: サーバー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	266
8. JMS エクスポート・バインディング: サーバー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	187	22. MQ JMS インポート: サーバー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	268
9. 汎用 JMS インポート: サーバー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	197	23. MQ JMS エクスポート: サーバー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	268
10. 汎用 JMS エクスポート: サーバー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	197	24. WebSphere Adapter および WebSphere Business Integration Adapter 間の差異	279
11. MQ JMS インポート: サーバー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	206	25. 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	337
12. MQ JMS エクスポート: サーバー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	207	26. Network Deployment 環境でのイベント順序付けのサポート	380
13. WebSphere MQ インポート: サーバー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	216	27. esAdmin listLocks コマンドの出力例	383
14. WebSphere MQ エクスポート: サーバー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	217		

第 1 章 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。以下のトピックを読み、管理タスクに使用されるインターフェースと構成ファイルについて理解してください。

WebSphere Process Server の管理コンソール

管理コンソールは、WebSphere Process Server のアプリケーション、サービス、およびその他のリソースをセル、ノード、またはサーバーの有効範囲で管理するため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、スタンドアロン・プロセス・サーバーから使用することもできますし、ネットワーク環境のセル内のすべてのサーバーを管理するデプロイメント・マネージャーから使用することもできます。

注: WebSphere Process Server 管理コンソールは、全体的に見れば Integrated Solutions Console フレームワークの一部であり、部分的に見れば 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。これらのタスクは、WebSphere Application Server 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
- Common Event Infrastructure の管理
- 製品セキュリティーの構成
- 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、およびその他の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、5 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・ブラウザー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。

- 「サーバー」>「デプロイメント環境」：デプロイメント環境管理機能と、新規デプロイメント環境の作成を支援するウィザードを使用できます。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
 - ビジネス・インテグレーション構成 (デプロイメント・ターゲット機能のタブ付きページ)
 - サービス・コンポーネント・アーキテクチャーの構成

- Common Event Infrastructure サーバーおよび宛先構成
- Business Process Choreographer 構成
- ビジネス・ルール構成
- セレクター
- WebSphere Business Integration Adapter サービス
- Application Scheduler
- 拡張メッセージング・サービス
- 「**サーバー**」 > 「**クラスター**」 > 「*cluster_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
 - ビジネス・インテグレーション構成 (デプロイメント・ターゲット機能のタブ付きページ)
 - サービス・コンポーネント・アーキテクチャの構成
 - Common Event Infrastructure サーバーおよび宛先構成
 - Business Process Choreographer 構成
 - ビジネス・ルール構成
- 「**リソース**」：ここから以下を利用できます。
 - WebSphere Business Integration Adapter
 - 担当者ディレクトリー・プロバイダー
 - 拡張メッセージング・プロバイダー
 - リモート成果物
- 「**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 (プロセス・サーバーの管理、Enterprise Service Bus の管理など) が提供されます。

また、メインのウェルカム・ページには、インストールした各製品の製品固有ウェルカム・ページへのリンクがあります。これらのページで、各製品の詳細情報を確認します。

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。

管理コンソールのガイド付きアクティビティー

ガイド付きアクティビティーは、複数の管理コンソール・ページを使用する必要がある一般的な管理タスクの実行を支援します。

ガイド付きアクティビティーは、タスクを実行するために必要な各管理コンソール・ページを表示し、そのタスクを適切に実行する上で役立つ次の情報を提供します。

- タスクとそれに関連する重要な概念の紹介
- タスクを実行する状況と理由の説明
- 現行タスクの実行前および実行後に行う必要がある他のタスクのリスト
- タスクを完了するための主なステップ
- 問題の回避または問題からの回復のためのヒント
- オンライン資料にあるフィールドの説明およびタスクに関する詳細情報へのリンク

5 ページの図 1 に、ガイド付きアクティビティーが表示された管理コンソールの例を示します。

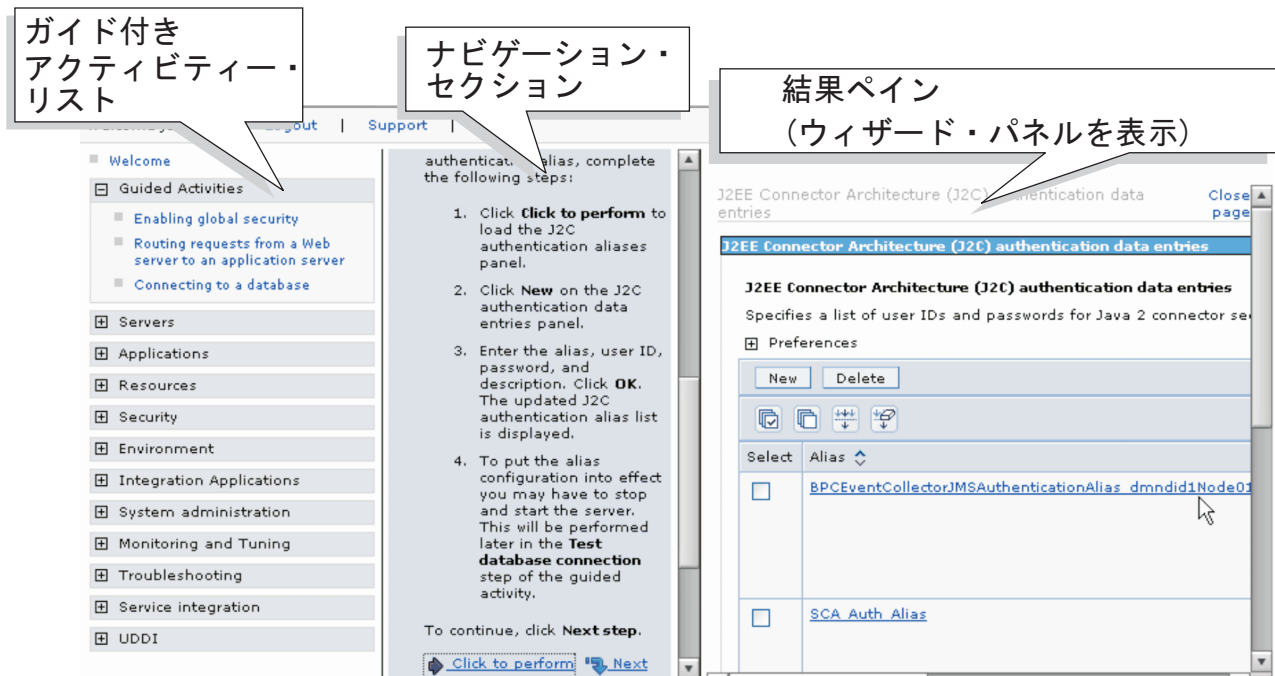


図1. ガイド付きアクティビティ

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 6 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『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。4 ページの『管理コンソールのガイド付きアクティビティ』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。19 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 (WebSphere Application Server および WebSphere Process Server リソースの両方を管理するための) ボタンの完全なリストについては、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。

コマンド行ツール、スクリプト、およびプログラミング・インターフェース

WebSphere Process Server には、ランタイム環境を管理するためのコマンド行ツール、スクリプト・インターフェース、およびプログラミング・インターフェース (管理プログラム) が組み込まれています。

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ついての詳細は、このインフォメーション・センターの「[参照](#)」>「[コマンドおよびスクリプト \(Commands and scripts\)](#)」を参照してください。

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。

プログラミング・インターフェースについての詳細は、このインフォメーション・センターの「[参照](#)」>「[コマンドおよびスクリプト \(Commands and scripts\)](#)」を参照してください。

管理プログラム

管理プログラムは、Java Management Extensions (JMX) 仕様に基づく一連の Java™ クラスおよびメソッドであり、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。

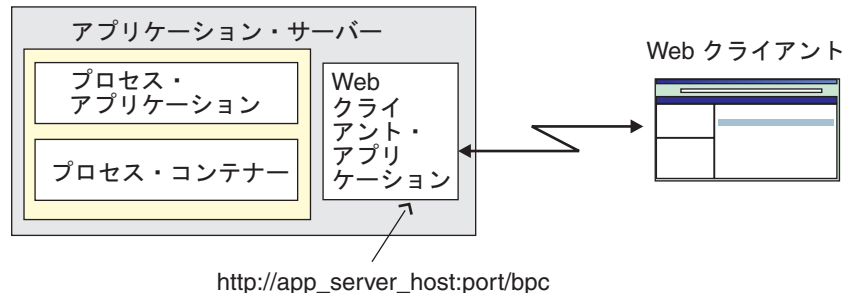
プログラミング・インターフェースについての詳細は、このインフォメーション・センターの「[参照](#)」>「[Javadoc](#)」を参照してください。

Business Process Choreographer Explorer について

Business Process Choreographer Explorer は、ビジネス・プロセスおよびヒューマン・タスクと対話するための汎用 Web ユーザー・インターフェースを実装している Web アプリケーションです。

サーバーまたはクラスター上で Business Process Choreographer Explorer インスタンスを 1 つ以上構成できます。WebSphere Process Server プロファイルを持つ WebSphere Process Server のインストール済み環境、または WebSphere Process Server クライアントのインストール済み環境があれば十分です。Business Process Choreographer をサーバーまたはクラスター上に構成する必要はありません。クライアントのインストール済み環境は、クライアントを WebSphere Process Server に接続するために必要な唯一のインフラストラクチャーですが、ここには Business Process Choreographer Explorer が組み込まれていません。ただし、デプロイメント・マネージャーがある場合は、WebSphere Process Server クライアントのインストール済み環境のサーバーにも Business Process Choreographer Explorer をインストールできます。

単一の Business Process Choreographer Explorer は、1 つの Business Process Choreographer 構成に接続できますが、ローカル構成に接続する必要はありません。ただし、Business Process Choreographer Explorer の複数のインスタンスを同じサーバーまたはクラスター上で構成して、各インスタンスを異なる Business Process Choreographer 構成に接続できます。



Business Process Choreographer Explorer を始動した場合、ユーザー・インターフェースに表示されるオブジェクトおよび実行できるアクションは、所属先のユーザー・グループとそのグループに与えられている権限に応じて異なります。例えば、ビジネス・プロセス管理者の場合は、デプロイされたビジネス・プロセスの処理を円滑に進める責任があります。したがって、プロセス・テンプレート、タスク・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・インスタンスの開始、タスクの作成および開始、失敗したアクティビティの修復および再開、作業項目の管理、完了したプロセス・インスタンスやタスク・インスタンスの削除などを実行できます。ただし、ユーザーの場合は、割り当てられているタスクについてのみ、表示や処理が可能になります。

Business Process Choreographer Observer について

Business Process Choreographer Observer について

Business Process Choreographer Observer を使用すると、完了しているプロセスに関するレポートを作成できます。実行中のプロセスの状況を表示する場合にも使用できます。ここでは、アーキテクチャーと考えられる構成パスについて説明します。

Business Process Choreographer Observer では、Common Event Infrastructure (CEI) を使用して、WebSphere Process Server から出力されるイベントを収集します。プロセス、アクティビティ、またはそのほかの集合データの数の概況をつかむには、いくつかの事前定義レポートを使用するか、独自のレポートを定義します。特定のプロセスまたはアクティビティについての情報を入手することもできます。

Business Process Choreographer Observer は、次の図に示す 2 つの J2EE エンタープライズ・アプリケーションが基本になっています。

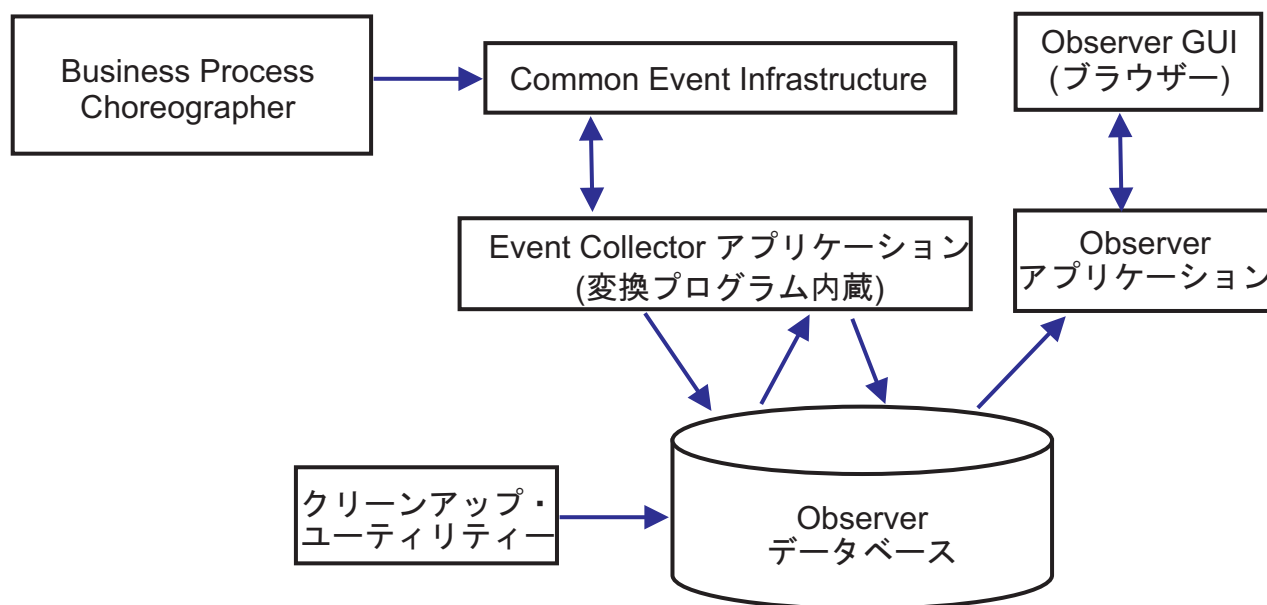


図 2. Business Process Choreographer Observer のアーキテクチャー

- Event Collector アプリケーションは、CEI パスからイベント情報を読み取り、その情報を Business Process Choreographer Observer データベースの Event Collector 表に格納します。
- Observer データベースは、イベント・データを格納する一連のデータベース表です。
- イベント変換プログラムである J2EE エンタープライズ・アプリケーションが定期的に起動し、未加工のイベント・データを、Business Process Choreographer Observer からの照会に適した形式に変換します。
- Observer アプリケーションは、レポートを生成し、ユーザーが Observer のグラフィカル・ユーザー・インターフェース (GUI) を使用して開始できるほかの操作を実行します。

- レポートを生成するときには、GUI を使用できます。定義したレポートを保管したり取り出したりすることもできます。
- クリーンアップ・ユーティリティは、Observer データベースからレコードを除去するときに使用でき、これはパフォーマンスの向上に役立ちます。

単純な構成

パフォーマンスを重視しない単純な構成を次の図に示します。

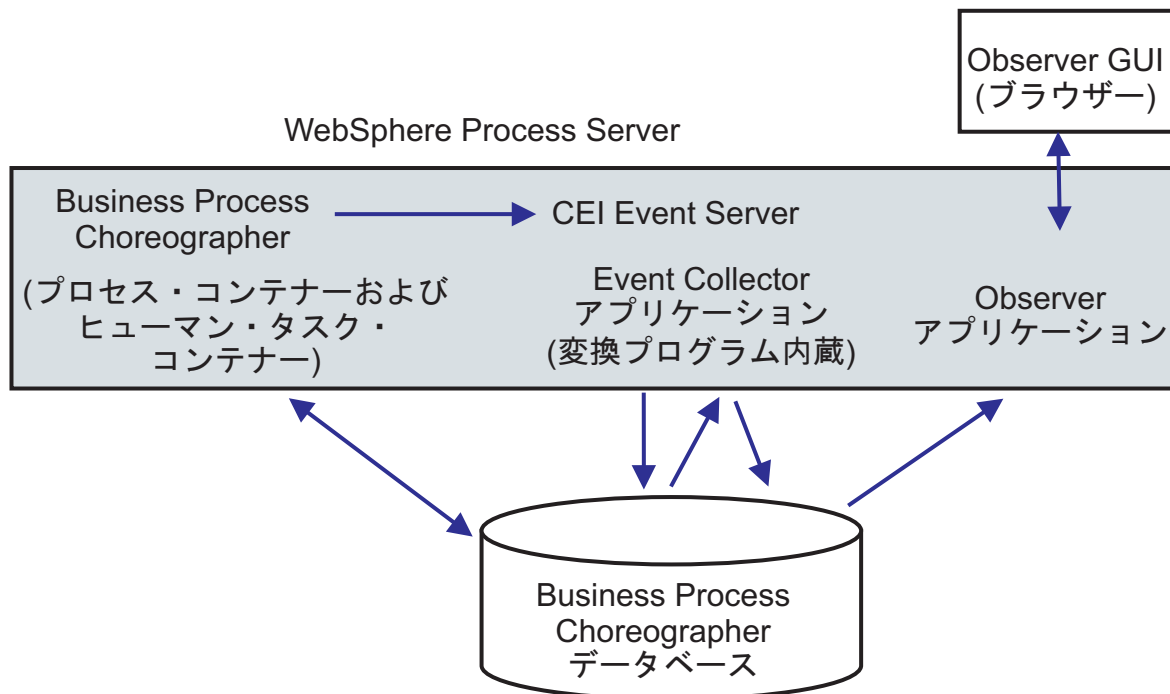


図3. Business Process Choreographer Observer のスタンドアロン・セットアップ構成

すべてが 1 台のマシンにインストールされており、Business Process Choreographer と Business Process Choreographer Observer が同じデータベースを使用します。

この種の単純な構成が作成されるのは、Business Process Choreographer のサンプル構成を作成する場合です。また、bpeconfig.jacl ツールのデフォルト設定では、Observer アプリケーションと Event Collector アプリケーションが Business Process Choreographer 構成と同じデプロイメント・ターゲット上に構成されます。Common Event Infrastructure (CEI) ログインが使用可能になり、必要なデータベース・スキーマが Business Process Choreographer の Derby データベース BPEDB に作成されます。この構成パスは、パフォーマンスを重視していない場合には理想的です。

ハイパフォーマンス構成

Business Process Choreographer Observer アーキテクチャーの潜在能力をフル活用するための自由な設定が可能になる対話式の構成ツールが用意されています。例えば、実動システムでパフォーマンスが理想的な構成の場合は、3 つの Business Process Choreographer エlementが別のマシン上で稼働し、Business Process Choreographer Observer には独自のデータベースがあります。

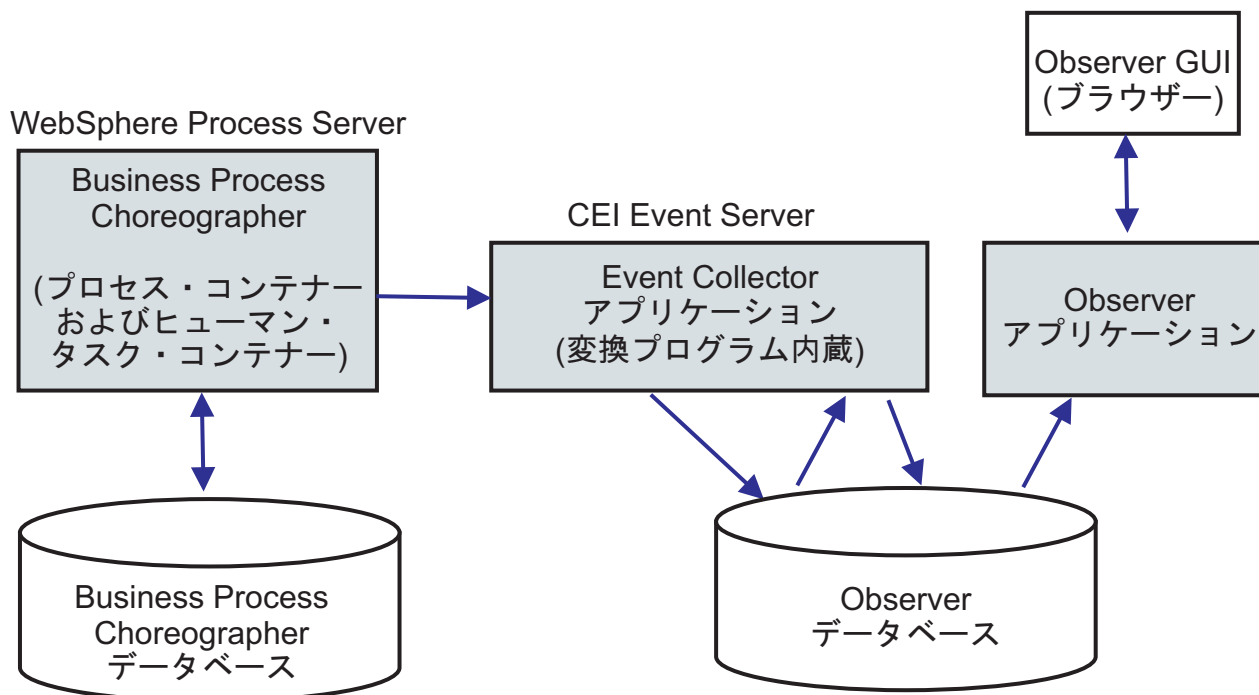


図4. 実稼働時のパフォーマンスに合わせた Business Process Choreographer Observer のセットアップ構成

クラスター化されたセットアップ構成で、Business Process Choreographer Observer に別個のデータベースを使用する場合や Business Process Choreographer Observer を既存の Business Process Choreographer 構成に追加する場合、またはより高度なデータベース・オプションを使用する場合は、Business Process Choreographer Observer の構成を実行してください。

Network Deployment 環境の場合

Business Process Choreographer Observer を Network Deployment 環境で構成する場合は、以下の制約事項が適用されます。

- ご使用のセルで CEI を構成する必要があります。
- 前の図に示したように、CEI Event Server が構成されているデプロイメント・ターゲットで Business Process Choreographer Event Collector を構成する必要があります。CEI Event Server を Business Process Choreographer とは異なるクラスターで構成した場合は、CEI Event Server が構成されているデプロイメント・ターゲットで Business Process Choreographer Event Collector を構成する必要があります。Business Process Choreographer Observer アプリケーションを Event Collector と同じマシン上にインストールする必要はありません。

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、プロファイル作成時またはサーバーの初期インストール後に選択してインストールできます。

ビジネス・ルールは、WebSphere Integration Developer 内で、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設計および開発します。ビジネス・ルールは、WebSphere Business Modeler 内で作成することもできます。ただし、Modeler では、ビジネス・ルール・タスクの作成のみがサポートされます (ビジネス・ルール・タスクは、Modeler からエクスポートしたときにルール・セットになります)。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。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・ルールの構築および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。

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) のディレクトリーに保持されている 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ーによって、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ー内のファイルは、そのサーバーにのみ適用されます。
- アプリケーション・ディレクトリー内のファイルは、そのアプリケーションにのみ適用されます。
- クラスター・レベルのディレクトリー内のファイルは、そのクラスターにのみ適用されます。
- ノード・レベルのディレクトリー内のファイルは、そのノード上のすべてのサーバーに適用されます。
- セル・ディレクトリー内のファイルは、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	サーバーと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、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 Adapters が含まれます。
cell-wbi.xml	セルを識別します。このファイルは、リレーションシップ・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。
server-bpc.xml	Business Process Choreographer コンテナおよびそのコンポーネントを識別します。
resources-bpc.xml	Business Process Choreographer コンテナ用の稼働環境リソース (担当者ディレクトリー・プロバイダー用の構成情報を含む) 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	コア WebSphere Process Server 構成 (Artifact Loader Service および Business Context Data Service を含む) 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に関する完全な情報は、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。

第 2 章 管理インターフェース入門

WebSphere Process Server をセットアップ、使用、および管理するには、以下のトピックの情報を使用します。

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、このトピックのタスクを使用します。

以下のタスクは、サーバーと管理コンソールの始動、コンソールのスコープと設定の変更、およびマスター・リポジトリへの作業の保存に役立ちます。

- **サーバーを始動します。**

管理コンソールを使用するには、その前にスタンドアロン・サーバーまたはデプロイメント・マネージャーを始動しておく必要があります。スタンドアロン・サーバーを始動する方法については、『スタンドアロン・サーバーの開始』を参照してください。Network Deployment の場合は、『デプロイメント・マネージャーの開始』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、16 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。19 ページの『管理コンソール設定の指定』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・コレクション・テーブルに表示されます。スコープ設定について詳しくは、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。

- **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。19 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 インフォメーション・センターで説明されているように、deployment.xml 構成ファイルを編集してこの値を変更できます。

- **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。

始める前に

管理コンソールが必要とするアプリケーション・サーバーを始動していることを確認します。

このタスクについて

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。

プロシージャ

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、「WSVR0221I: アプリケーションが始動されました: 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。

- `profile_root/config/cells/cell_name/nodes/node_name/serverindex.xml`
- `profile_root/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 9060) 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。

i5/OS コンソール・ポートがマシン上で既に実行中のアプリケーションと競合するために管理コンソールを始動できない場合は、`chgwassvr` スクリプト・コマンドを使用してポート番号を変更します。詳しくは、『*Changing the ports associated with an application server*』を参照してください。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、アプリケーション・サーバーを始動します。

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。ユーザー ID は、ログインするために使用された対象セッションの期間中にしか存続しません。

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。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もあります。
- 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ーを閉じた場合は、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。

関連タスク

47 ページの『サーバーおよびクラスターの開始または停止』

サーバーでアプリケーションを開始する前に、サーバーが稼働している必要があります。サーバーを開始する方法はいくつかありますが、スタンドアロン・サーバーと管理対象サーバーのいずれであるかによって、使用方法が異なります。管理対象サーバーの場合、サーバーを開始する前に、ノード・エージェントが稼働している必要があります。管理対象サーバーは、デプロイメント・マネージャーの管理コンソールから開始できます。デプロイメント環境またはクラスターを導入している場合、デプロイメント・マネージャーの管理コンソールですべてのサーバーを一括に開始または停止できます。

47 ページの『スタンドアロン・サーバーの開始』

管理コンソールを使用する前、およびサーバーでアプリケーションを実行する前に、サーバー・プロセスを開始しておく必要があります。

55 ページの『ノード・エージェントの停止』

管理対象ノードのノード・エージェントは、停止する必要があるサーバー・プロセスです。

56 ページの『ノード・エージェントの再開』

管理対象ノードのノード・エージェントは、再始動する必要があるサーバー・プロセスです。

57 ページの『デプロイメント環境の開始と停止』

IBM 提供のパターンに基づいて、管理コンソールから直接デプロイメント環境を開始または停止することができます。

58 ページの『クラスターの開始』

クラスター内のすべてのサーバー (クラスター・メンバー) を 1 回の操作で開始できます。ripple 始動によりまず各サーバーが停止され、その後順に開始されます。クラスターを開始すると、ワークロード管理が自動的に有効になります。

59 ページの『クラスターの停止』

クラスター内のすべてのサーバー (クラスター・メンバー) を 1 回の操作で停止できます。

60 ページの『サーバーの開始』

管理対象サーバーでアプリケーションを実行する前に、サーバー・プロセスを開始しておく必要があります。単一サーバー (クラスター・メンバーを含む) を開始できます。

62 ページの『サーバーの停止』

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

管理コンソール設定の指定

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。

このタスクについて

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**」: 検索の結果として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。

プロシージャ

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。

次のタスク

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。

プロシージャ

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

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。

「マイ・タスク」の使用法

コンソール・ナビゲーションをカスタマイズするには、タスク・ビューを作成および編集します。

このタスクについて

「**マイ・タスク**」では、コンソール・ナビゲーションに表示するタスクのリストを作成および編集できます。タスクには、そのタスクを完了するために使用される 1 つ以上の Web アプリケーション (コンソール・モジュール) を表示するページが含まれます。コンソールを初めて表示すると、アクセスできるすべてのタスクがナビゲーションに表示されます。「**マイ・タスク**」は、使用する頻度の高いタスクのみを表示するようにナビゲーションをカスタマイズするときに特に便利です。タスクのカスタマイズ後には、コンソールにログインするたびに、最初に「**マイ・タスク**」が表示されます。

プロシージャ

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. ナビゲーションの「表示」選択リストから「マイ・タスク」を選択します。「マイ・タスク」を初めて使用する場合に「マイ・タスク」を開くには、「タスクの追加」をクリックする必要があります。
3. 「マイ・タスク」リストに追加するタスクを選択します。
4. 変更を保存するには、「適用」をクリックします。
5. 変更を取り消すには、「リセット」をクリックします。

結果

「適用」をクリックすると、カスタマイズされたタスク・リストがナビゲーションに表示されます。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。

Business Process Choreographer Explorer 入門

持っているユーザー役割に応じて、Business Process Choreographer Explorer を使用してビジネス・プロセスおよびヒューマン・タスクを管理するか、割り当てられたタスクを処理することができます。

このタスクについて

Business Process Choreographer Explorer を使用して、以下のタスクを実行できます。

- ビジネス管理者である場合は、ビジネス・プロセスのライフ・サイクルを管理し、ビジネス・プロセスを修復することができます。例えば、単一のアクティビティの実行を再始動または強制実行するか、ビジネス・プロセス全体を補正することができます。補正に失敗した場合は、そのプロセス・インスタンスを再試行するか、スキップするか、停止することができます。さらに、ビジネス・プロセスおよびアクティビティのカスタム・プロパティを追加および更新することもできます。
- ヒューマン・タスク管理者であれば、ヒューマン・タスクのライフ・サイクルを管理でき、作業割り当てを管理できます。例えば、ユーザーに責任を割り当てたり、ユーザーの不在処理や代替ユーザーを管理したりできます。また、ヒューマン・タスクの優先順位とビジネス・カテゴリーを変更し、カスタム・プロパティの追加または更新を行うこともできます。
- ビジネス・ユーザーである場合は、Business Process Choreographer Explorer を使用して、割り当てられたタスクを処理できます。例えば、ビジネス・プロセス、サービス、およびヒューマン・タスクを開始することができ、ヒューマン・タスクの処理、編集、保存、実行、または解放を行うことができます。そのほか、不在のフラグを立て、代理人を定義できます。

さらに、Business Process Choreographer Explorer には検索機能があり、これを使用すると、注意が必要なビジネス・プロセスとそれに関連するアクティビティおよびヒューマン・タスクを検出できます。例えば、それらのインスタンスの状況を検査し、関連するインスタンスおよびテンプレートの間をナビゲートし、関連するアクティビティおよびヒューマン・タスクを含んでいるプロセス状態のグラフィカル・ビューを取得することができます。

関連タスク

タスク・テンプレートとタスク・インスタンスの管理

タスク・テンプレートを管理するには、管理コンソールまたは管理コマンドを使用します。タスク・インスタンスを処理するには、Business Process Choreographer Explorer を使用します。

作業割り当ての管理

タスクの開始後に、例えば、ワークグループのメンバー間でワークロードを効率よく分散するために、そのタスクに対する作業割り当ての管理が必要になる場合があります。

タスク・インスタンスの作成と開始

タスク・インスタンスは、使用を許可されている任意のタスク・テンプレートから作成し、開始することができます。

タスクの操作

タスクを処理するには、タスクを要求してから、タスクの完了に必要なアクションを実行する必要があります。

Business Process Choreographer Explorer のユーザー・インターフェース

Business Process Choreographer Explorer は、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一連の管理機能を備えたスタンドアロン Web アプリケーションです。このインターフェースは、タスクバー、ナビゲーション・ペイン、およびワークスペースで構成されます。

次の図に、Business Process Choreographer Explorer のユーザー・インターフェースのレイアウトを示します。



このユーザー・インターフェースの主領域について、以下に説明します。

タスクバー

すべてのユーザーがタスクバーによって提供されるオプションは、Business Process Choreographer Explorer からのログアウト、不在設定の指定、およびオンライン・ヘルプへのアクセスです。システム管理者権限を保持している場合には、タスクバーに以下のオプションも組み込まれます。

カスタマイズ

このオプションは、Business Process Choreographer Explorer のこのインスタンスのナビゲーション・ペインに対してビューの追加およびビューの除去を行う場合に選択します。ユーザーがログインしたときに、ユーザーに対して表示されるビューを定義することもできます。

ビューの定義

このオプションは、ユーザー・グループのカスタマイズ・ビューを定義する場合に選択します。

代理人の定義

このオプションは、ユーザーの不在設定を定義する場合に選択します。

ナビゲーション・ペイン

ナビゲーション・ペインには、オブジェクトを管理するときに使用するビューへのリンクがあります。例えば、開始したプロセス・インスタンスや管理する権限があるヒューマン・タスクなどのオブジェクトが対象です。デフォルトのユーザー・インターフェースには、ビジネス・プロセスやタスクの事前定義ビューへのリンクがあります。

システム管理者は、ナビゲーション・ペインに対して事前定義ビューの追加および除去を行い、カスタム・ビューを定義してナビゲーション・ペインに追加することにより、デフォルトのナビゲーション・ペインのコンテンツをカスタマイズできます。すべてのユーザーは、ナビゲーション・ペインから個別設定ビューを定義できます。

ワークスペース

ワークスペースには、ビジネス・プロセスおよびヒューマン・タスクに関連したオブジェクトを表示および管理するときに使用するページがあります。これらのページにアクセスするには、ナビゲーション・ペインのリンクをクリックする方法、アクション・バーのアクションをクリックする方法、またはワークスペース・ページ内のリンクをクリックする方法があります。

Business Process Choreographer Explorer のナビゲーション・ペイン

プロセス・インスタンスや作業割り当てなどのビジネス・プロセス・オブジェクトおよびヒューマン・タスク・オブジェクトを管理するときに使用するビューにアクセスするには、ナビゲーション・ペインを使用します。デフォルトのユーザー・インターフェースには、ビジネス・プロセスやタスクの事前定義ビューへのリンクがあります。ナビゲーション・ペインに追加する専用の個別設定ビューを定義することもできます。さらに、システム管理者は、すべてのユーザーが使用できるカスタマイズ・ビューを定義できます。

使用可能なアクション

ナビゲーション・ペインでは、以下のアクションを使用できます。

- ビューにナビゲートする。

あるビューにナビゲートするには、そのビュー名をクリックします。

- グループを縮小表示または展開する。

ナビゲーション・ペイン内のある項目を展開または縮小表示するには、その項目の横にある矢印をクリックします。

- 新規の検索を定義する。




オブジェクトの検索や個別設定ビューの定義を行うには、「新規検索」アイコン

() をクリックします。

ビューの種類に応じて、その他のアクションをポップアップ・メニューから選択できます。アイコンは、ポップアップ・メニューが使用できるかどうかを示しています。

- ビューを削除するには、「削除」アイコン () をクリックします。

- ビューを変更するには、「編集」アイコン () をクリックします。

- ビューのコピーを作成して、コピーを変更するには、「コピー」アイコン () をクリックします。
- リスト内でのビューの位置を上下に移動するには、「上へ」アイコン () または「下へ」アイコン () をクリックします。

ナビゲーション・ペインの事前定義ビュー

デフォルトのナビゲーション・ペインには、以下のビュー・グループがあります。ご使用の Business Process Choreographer Explorer のナビゲーション・ペインに表示されるビューは、システム管理者がナビゲーション・ペインに対してビューを追加または削除したかどうかによって異なります。ビューのグループにビューが定義されていない場合、そのグループは表示されません。

プロセス・テンプレート

プロセス・テンプレート・グループには、以下のビューがあります。

ユーザーのプロセス・テンプレート

このビューには、プロセス・テンプレートのリストが表示されます。このビューからは、プロセス・テンプレートとその構造に関する情報の表示、テンプレートに関連付けられているプロセス・インスタンスのリストの表示、およびプロセス・インスタンスの開始を実行できます。

プロセス・インスタンス

プロセス・インスタンス・グループには、以下のビューがあります。

ユーザーが開始

このビューには、開始したプロセス・インスタンスが表示されます。このビューからは、プロセス・インスタンスの進行状況のモニター、プロセス・インスタンスに関連するアクティビティー、プロセス、またはタスクのリスト表示を実行できます。

ユーザーが管理

このビューには、管理する権限があるプロセス・インスタンスが表示されます。このビューからは、プロセス・インスタンスを処理できます。例えば、プロセスの中断および再開、またはプロセス・インスタンスでのアクティビティーの進行状況のモニターを実行できます。

重大なプロセス

このビューには、実行状態のプロセス・インスタンスのうち、停止状態のアクティビティーを含むものが表示されます。このビューからは、プロセス・インスタンスを処理したり、アクティビティーのリストを表示後にアクティビティーを処理したりすることができます。

強制終了したプロセス

このビューには、強制終了状態のプロセス・インスタンスが表示されます。このビューからは、これらのプロセス・インスタンスを処理できます。

失敗した補正

このビューには、microflow について失敗した補正アクションが表示されます。

アクティビティー・インスタンス

デフォルトでは、アクティビティー・インスタンス・グループにはビューがありません。このため、このグループはデフォルトのナビゲーション・ペインには表示されません。

タスク・テンプレート

タスク・テンプレート・グループには、以下のビューがあります。

ユーザーのタスク・テンプレート

このビューには、タスク・テンプレートのリストが表示されます。このビューからは、タスク・インスタンスの作成および開始、テンプレートに関連付けられているタスク・インスタンスのリストの表示を実行できます。

タスク・インスタンス

タスク・インスタンス・グループには、以下のビューがあります。

ユーザーの予定

このビューには、操作する権限があるタスク・インスタンスのリストが表示されます。このビューからは、タスク・インスタンスの処理、要求したタスク・インスタンスのリリース、または他のユーザーへのタスク・インスタンスの転送を実行できます。

すべてのタスク

このビューには、お客様が所有者、潜在的な所有者、編集者のいずれかになっているすべてのタスクが表示されます。このビューからは、タスク・インスタンスの処理、要求したタスク・インスタンスのリリース、または他のユーザーへのタスク・インスタンスの転送を実行できます。

ユーザーが開始

このビューには、開始したタスク・インスタンスが表示されます。このビューからは、タスク・インスタンスの処理、要求したタスク・インスタンスのリリース、または他のユーザーへのタスク・インスタンスの転送を実行できます。

ユーザーが管理

このビューには、管理する権限があるタスク・インスタンスが表示されます。このビューからは、タスク・インスタンスを処理できます。例えば、プロセスの中断および再開、タスク・インスタンスの作業項目の作成、またはタスク・インスタンスの現在の作業項目リストの表示を実行できます。

ユーザーのエスカレーション

このビューには、ログオンしているユーザーのすべてのエスカレーションが表示されます。

ビューの種類



ナビゲーション・ペインには、以下の種類のビューを収容できます。ビューによっては、ポップアップ・メニューからその他のアクションを選択できます。

デフォルトのナビゲーション・ペイン内の事前定義ビュー。


これらのビュー・グループが表示されるのは、「ナビゲーション・ツリーおよびログイン・ビューのカスタマイズ」ページで、システム管理者によってナビゲーション・ペインが変更されなかった場合に限られます。これらのビューでは、ポップアップ・メニューは使用できません。

システム管理者によりナビゲーション・ペインに追加されたカスタマイズ・ビューおよび事前定義ビュー。

ビジネス・ユーザーは、ビュー名をクリックしてビューまでナビゲートできます。システム管理者の場合は、ポップアップ・メニューを使用できます。

- 事前定義ビューは、「事前定義ビュー」アイコン () で示されます。システム管理者は、ポップアップ・メニューを使用して、ナビゲーション・ペイン内でのこれらのビューの位置を変更できます。
- カスタマイズ・ビューは、「カスタム・ビュー」アイコン () で示されます。システム管理者は、これらのビューの削除、編集、コピー、および移動を実行できます。

個別設定ビュー。

これらのビューは、「カスタム・ビュー」アイコン () で示されます。これらのビューは、これらのビューを作成したユーザーに対してのみ表示されます。ユーザーは、これらのビューの削除、編集、コピー、および移動を実行できます。

Business Process Choreographer Explorer の開始

Business Process Choreographer Explorer は、ビジネス・プロセス・コンテナの構成の一部としてインストールできる Web アプリケーションです。Web ブラウザーから Business Process Choreographer Explorer の使用を開始するには、前もってビジネス・プロセス・コンテナ、ヒューマン・タスク・コンテナ、および Business Process Choreographer Explorer アプリケーションのインストールが完了済みで、さらにこのアプリケーションが稼働している必要があります。

このタスクについて

Business Process Choreographer Explorer を開始するには、以下の手順を実行します。

プロシージャ

1. Web ブラウザーで、Business Process Choreographer Explorer の URL にアクセスします。

この URL は、次の形式をとります。URL の値は、使用するシステムで仮想ホストとコンテキスト・ルートがどのように構成されているかによって異なります。

`http://app_server_host:port_no/context_root`

各部の意味は、次のとおりです。

app_server_host

作業に使用するビジネス・プロセス・アプリケーションを提供するアプリケーション・サーバーのホストのネットワーク名。

port_no

Business Process Choreographer Explorer が使用するポート番号。ポート番号はシステム構成によって異なります。デフォルト・ポート番号は 9080 です。

context_root

Business Process Choreographer Explorer アプリケーションの、アプリケーション・サーバー上のルート・ディレクトリー。デフォルトは bpc です。

2. セキュリティーが有効になっている場合は、ユーザー ID とパスワードを入力して、「ログイン」をクリックする必要があります。

結果

Business Process Choreographer Explorer の最初のページが表示されます。デフォルトでは、このページには「ユーザーの予定」ビューが表示されます。

Business Process Choreographer Explorer のカスタマイズ

Business Process Choreographer Explorer は、管理者がビジネス・プロセスとヒューマン・タスクを管理し、ビジネス・ユーザーが割り当てられたタスクを処理するためのユーザー・インターフェースを提供します。これは汎用インターフェースであるため、特定の Business Process Choreographer Explorer インスタンスに割り当てられたユーザー・グループのビジネス・ニーズに合わせて、インターフェースをカスタマイズする必要がある場合があります。

このタスクについて

このユーザー・インターフェースは、さまざまな方法でカスタマイズできます。

さまざまなユーザー・グループ用の Business Process Choreographer Explorer インターフェースのカスタマイズ

デフォルトの Business Process Choreographer Explorer ユーザー・インターフェースのナビゲーション・ペインには、事前定義ビューへの一連のリンクが入っています。「ユーザーの予定」ビューは、ログインした後に表示されるデフォルト・ビューです。Business Process Choreographer のいずれかのシステム管理者ロールを持っている場合は、ナビゲーション・ペインに表示されるリンク、ログインしたユーザーに対して表示されるビュー、および各ビューに表示される情報をカスタマイズできます。

始める前に

インターフェースをカスタマイズするには、BPCSystemAdministrator 許可を持っている必要があります。

このタスクについて

例えば、Business Process Choreographer Explorer のデフォルトのユーザー・インターフェースには、ビジネス・ステート・マシンで作業するためのビューが含まれていません。その場合、ビジネス・ステート・マシンのプロセス・テンプレートおよびプロセス・インスタンスを操作するための事前定義ビューを追加できます。

あるいは、カスタマー・オーダーを処理するユーザーに、カスタマー・サービスの照会を処理するユーザーとは異なるインターフェースを提供する必要がある場合があります。Business Process Choreographer Explorer のインスタンスを、そのインスタンスに割り当てられたユーザーのワークフロー・パターンに合うようにカスタマイズできます。

Business Process Choreographer Explorer のデフォルトのユーザー・インターフェースをカスタマイズするには、以下の手順を実行します。

プロシージャ

1. ナビゲーション・ペイン内の一連のビューとデフォルトのログイン・ビューをカスタマイズします。
 - a. タスクバーで「**カスタマイズ**」をクリックします。
 - b. 「ナビゲーション・ツリーおよびログイン・ビューのカスタマイズ」ページで、組み込むビューを選択し、ナビゲーション・ペインから除去するビューの選択を解除します。
 - c. ユーザーが Business Process Choreographer Explorer にログインしたときに表示されるビューを選択します。

リストには前のステップで選択したビューと、「カスタマイズ・ビューの検索および定義」ページで作成したカスタマイズ済みビュー (手順 2 を参照) が含まれています。

- d. 変更を保存するには、「**保管**」をクリックします。

このインスタンスのビューをデフォルトのビューに戻すには、「**デフォルトの復元**」をクリックします。このアクションは、ナビゲーション・ペインを事前定義ビューのリストにリセットします。ナビゲーション・ペイン内のカスタマイズされたビューは、このアクションの影響を受けません。

2. ビューをカスタマイズします。

この Business Process Choreographer Explorer インスタンスのビューに表示される情報を指定できます。

- a. タスクバーで「**ビューの定義**」をクリックします。
- b. 「カスタマイズ・ビューの検索および定義」ページで、例えばプロセス・テンプレートなど、カスタマイズしたいビューのタイプを選択します。
- c. そのビューの「カスタマイズ・ビューの検索および定義」ページで、検索基準を指定します。
- d. 「プロパティの表示」タブを使用して、ビューに組み込むプロパティを選択し、リストのプロパティを指定します。

これがタスク・インスタンス・ビューまたはプロセス・インスタンス・ビューである場合は、「**ビューの設定**」をクリックして、ビューのアクション・

バーに追加する一連のアクションを選択し、システム管理者およびシステム・モニター用のビューに組み込まれる項目を指定します。

- ビューのタイプを選択します。
 - ビューのアクション・バーに管理アクションを追加するには、「**インスタンスの管理**」を選択します。
 - ログオン・ユーザーがインスタンスを処理できるよう、一連のアクションをアクション・バーに追加するには、「**インスタンスの処理**」を選択します。
- システム管理者およびシステム・モニターの場合は、検索結果を各自のインスタンスに制限できます。
 - 検索基準に一致するすべての項目をビューに表示するには、「**すべてのインスタンス**」を選択します。システム管理者が当該の項目についての作業項目を持っているかどうかに関係なく、すべての項目が表示されます。
 - ログオン・ユーザーが作業項目を持っている項目だけを表示するには、「**個人用インスタンス**」を選択します。
- e. ビューの表示名を「**ビュー名**」フィールドに入力し、「**保管**」をクリックします。

新しいビューがナビゲーション・ペインに表示されます。各ユーザーが次回に Business Process Choreographer Explorer にログインすると、その新しいビューが表示されます。

ビジネス・ステート・マシンのプロセス・テンプレート用ビューの定義:

ビジネス・ステート・マシンのプロセス・テンプレートには事前定義ビューが用意されていますが、このタイプのテンプレート用に独自のビューを定義することもできます。

始める前に

カスタマイズ・ビューを作成するには、BPCSystemAdministrator 許可を持っている必要があります。

プロシージャ

1. タスクバーで「**ビューの定義**」をクリックします。
2. 「カスタマイズ・ビューの検索および定義」ページで、「**プロセス・テンプレートの検索およびカスタマイズ・ビューの定義**」を選択します。
3. 「**プロパティ・フィルター**」 → 「**カスタム・プロパティ・フィルター**」をクリックします。
 - a. 以下の設定でカスタム・プロパティを追加します。
 - 「**プロパティ名**」フィールドに generatedBy と入力します。
 - 「**プロパティ値**」フィールドに BusinessStateMachine と入力します。
 - b. 「**追加**」をクリックします。
 - c. 必要に応じて他のカスタム・プロパティを追加します。
4. 「**プロパティの表示**」 → 「**リスト列**」をクリックします。

- a. カスタム・プロパティのリスト列で、以下の設定により、カスタム・プロパティを追加します。
 - 「プロパティ名」フィールドに generatedBy と入力します。
 - 「表示名」フィールドに列の表示名を入力し、「追加」をクリックします。
 - b. 他の列を追加するか、選択した列のリストから列を除去します。
5. 照会の表示名を「ビュー名」フィールドに入力し、「保管」をクリックします。

結果

デフォルトでは、新しいビューへのリンクがナビゲーション・ペインの「プロセス・テンプレート」グループに追加されます。ユーザーが次回に Business Process Choreographer Explorer にログインすると、このビューが表示されます。

ビジネス・ステート・マシンのプロセス・インスタンス用ビューの定義:

ビジネス・ステート・マシンのプロセス・インスタンスには、事前定義ビューが用意されていますが、このタイプのプロセス・インスタンス用に独自のビューを定義することもできます。

始める前に

カスタマイズ・ビューを作成するには、BPCSystemAdministrator 許可を持っている必要があります。

プロシージャ

1. タスクバーで「ビューの定義」をクリックします。
2. 「カスタマイズ・ビューの検索および定義」ページで、「プロセス・インスタンスの検索およびカスタマイズ・ビューの定義」を選択します。
3. 「プロパティ・フィルター」 → 「カスタム・プロパティ・フィルター」をクリックします。
 - a. 以下の設定でカスタム・プロパティを追加します。
 - 「プロパティ名」フィールドに generatedBy と入力します。
 - 「プロパティ値」フィールドに BusinessStateMachine と入力します。
 - b. 「追加」をクリックします。
 - c. 必要に応じて他のカスタム・プロパティを追加します。
4. 「プロパティの表示」 → 「リスト列」をクリックします。
 - a. 照会プロパティのリスト列で、以下の照会プロパティを追加します。
 - ビューにビジネス状態情報を追加するには、「プロパティ名」フィールドに name、「変数名」フィールドに DisplayState、「ネーム・スペース」フィールドに tns と入力します。ここで、tns は *-process* のサフィックスが付けられたビジネス・ステート・マシンのターゲット・ネーム・スペースです。また、「表示名」フィールドに列の表示名も指定し、「追加」をクリックします。
 - ビューに相関情報を追加するには、「プロパティ名」フィールド、「変数名」フィールド、および「ネーム・スペース」フィールドに該当する情

報を入力します。これらの値は、ビジネス・ステート・マシンの定義から派生します。また、「表示名」フィールドで列の表示名も指定します。

プロパティ名

ビジネス・ステート・マシンに定義した関連プロパティの名前。

変数名 関連セットが着信パラメーターによって開始される場合、変数名は以下の形式になります。

`operation_name_Input_operation_parameter_name`

ここで、`operation_name` は、初期状態からの遷移操作の名前です。

関連セットが発信パラメーターによって開始される場合、変数名は以下の形式になります。

`operation_name_Output_operation_parameter_name`

ネーム・スペース

照会プロパティのネーム・スペース。ここで、`tns` は `-process` のサフィックスが付けられたビジネス・ステート・マシンのターゲット・ネーム・スペースです。

- b. 他のカスタム・プロパティまたは照会プロパティを追加するか、選択した列のリストに列を追加するか、リストから列を除去します。

5. 照会の名前を「ビュー名」フィールドに入力し、「保管」をクリックします。

結果

デフォルトでは、新しいビューへのリンクがナビゲーション・ペインの「プロセス・インスタンス」グループに追加されます。ユーザーが次回に Business Process Choreographer Explorer にログインすると、このビューが表示されます。

Business Process Choreographer Explorer インターフェースの個別設定


デフォルトの Business Process Choreographer Explorer ユーザー・インターフェースのナビゲーション・ペインには、事前定義ビューおよびシステム管理者が定義したビューへの一連のリンクが入っています。ナビゲーション・ペインには、例えば特定のタスクまたはプロセスをモニターするために、独自のビューを追加することができます。

このタスクについて

ユーザー・インターフェースの個別設定を行うには、Business Process Choreographer Explorer で以下の手順を実行します。

プロシージャ

1. ナビゲーション・ツリーの新規ビューを定義したいセクションで、「新規検索」

アイコン () をクリックします。

2. そのビューの「個別設定ビューの検索および定義 (Search and Define Personalized Views)」ページで、検索基準を指定します。
3. 「プロパティの表示」 タブを使用して、ビューに組み込むプロパティを選択し、リストのプロパティを指定します。

これがタスク・インスタンス・ビューまたはプロセス・インスタンス・ビューである場合は、「**ビューの設定**」をクリックして、ビューのアクション・バーに追加する一連のアクションを選択します。システム管理者およびシステム・モニターである場合は、ビューに含める項目を指定してください。

- ビューのタイプを選択します。
 - ビューのアクション・バーに管理アクションを追加するには、「**インスタンスの管理**」を選択します。
 - ログオン・ユーザーがインスタンスを処理できるよう、一連のアクションをアクション・バーに追加するには、「**インスタンスの処理**」を選択します。
 - システム管理者およびシステム・モニターである場合は、検索結果を独自のインスタンスだけに制限できます。
 - 検索基準に一致するすべての項目をビューに表示するには、「**すべてのインスタンス**」を選択します。システム管理者が当該の項目についての作業項目を持っているかどうかに関係なく、すべての項目が表示されます。
 - ログオン・ユーザーが作業項目を持っている項目だけを表示するには、「**個人用インスタンス**」を選択します。
4. ビューの表示名を「**ビュー名**」フィールドに入力し、「**保管**」をクリックします。

結果

新しいビューがナビゲーション・ペインに表示されます。

デフォルトの Web アプリケーションの外観の変更

Business Process Choreographer Explorer は、JavaServer Pages (JSP) ファイルと JavaServer Faces (JSF) コンポーネントに基づいた、すぐに使用できる Web ユーザー・インターフェースを備えています。この Web インターフェースのレンダリング方法は、カスケーディング・スタイル・シート (CSS) によって制御されます。このスタイル・シートを変更して、ユーザー・インターフェースを特定のルック・アンド・フィールに合わせることができ、新たにコードを記述する必要はありません。

始める前に

スタイル・シートを変更するには、カスケーディング・スタイル・シートに関する深い知識が必要です。

このタスクについて

CSS の変更は、例えば、デフォルト・インターフェースをコーポレート・アイデンティティのガイドラインに合わせるために行うことができます。

プロシージャ

スタイル・シートを変更します。デフォルトのスタイル・シートである `style.css` には、ヘッダー、ナビゲーション・ペイン、およびコンテンツ・ペイン内に置かれるエレメントのスタイルが入っています。

関連概念

22 ページの『Business Process Choreographer Explorer のユーザー・インターフェース』

Business Process Choreographer Explorer は、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一連の管理機能を備えたスタンドアロン Web アプリケーションです。このインターフェースは、タスクバー、ナビゲーション・ペイン、およびワークスペースで構成されます。

Business Process Choreographer Explorer のインターフェースで使用されるスタイル:

style.css ファイルには、デフォルトのユーザー・インターフェースのルック・アンド・フィールを適合させるために変更できるスタイルが収録されています。

style.css ファイルには、以下に示すデフォルトのユーザー・インターフェースの要素のスタイルが収録されています。

- 『バナー』
- 『フッター』
- 35 ページの『メニュー・バー』
- 35 ページの『ログイン・ページ』
- 35 ページの『ナビゲーター』
- 36 ページの『コンテンツ・パネル』
- 36 ページの『コマンド・バー』
- 36 ページの『リスト』
- 36 ページの『詳細パネル』
- 36 ページの『メッセージ・データ』
- 37 ページの『タブ付きペイン』
- 37 ページの『検索ページ』
- 37 ページの『エラー詳細』

このファイルは、次のディレクトリにあります。

```
<profile_root>%installedApps%<node_name>%<explorer_instance>%bpexplorer.war%theme
```

バナー

スタイル名	説明
.banner	バナーの分割線。
.banner_left	バナー内の分割線。アプリケーションのタイトル画像を埋め込むときに使用します。
.banner_right	バナー内の分割線。例えば、ロゴを追加して表示するときに使用できます。

フッター

スタイル名	説明
.footer	フッターの分割線。

スタイル名	説明
.footer_left	フッター内の分割線。例えば、アプリケーションの会社ロゴを表示する場合に使用できます。
.footer_right	フッター内の分割線。例えば、ロゴを追加して表示するときに使用できます。

メニュー・バー

スタイル名	説明
.menubar	JSF サブビュー。
.menuContainer	ラベルやリンクなどのメニュー項目を含むコンテナー・パネル。
.menuItem	メニュー・バーに表示される項目。

ログイン・ページ

スタイル名	説明
.loginPanel	ログイン・フォームを含むパネル。
.loginTitle	フォームのタイトル。
.loginText	手順を示すテキスト。
.loginForm	入力コントロールが組み込まれているフォーム。
.loginValues	コントロールのレイアウトを決定するテーブル。
.loginField	ログオン・フィールドに使用されるラベル (「名前」や「パスワード」など)。
.loginValue	テキスト入力フィールド。

ナビゲーター

スタイル名	説明
.pageBodyNavigator	ナビゲーターが表示される領域。
.navigator	ナビゲーターの JSF サブビュー。リストへのリンクが収録されている。
.navigatorTitle	各ナビゲーター・ボックスのタイトル。
.taskNavigatorTitle	あるクラスに属する、ナビゲーション・ボックスのタイトル。これらは、ビジネス・プロセス・オブジェクトのリストへのリンクとヒューマン・タスク・オブジェクトのリストへのリンクを区別するときに使用します。
.navigatorFrame	各ナビゲーター・ボックスの分割線 (例えば、境界線を引くための線)。
.navigatorLink	ナビゲーター・ボックス内のリンク。
.expanded	ナビゲーター・ボックスが展開されている場合に使用される。
.collapsed	ナビゲーター・ボックスが縮小表示されている場合に使用される。

コンテンツ・パネル

スタイル名	説明
.pageBodyContent	コンテンツが表示される領域。
.panelContainer	リスト、詳細、メッセージのいずれかが入っている分割パネル。
.panelTitle	表示されるコンテンツのタイトル。例えば、「ユーザーの予定」。
.panelHelp	ヘルプ・テキストおよびアイコンが入っている分割コンテナ。
.panelGroup	コマンド・バー、コマンド・リスト、詳細、またはメッセージが入っている分割コンテナ。

コマンド・バー

スタイル名	説明
.commandbar	コマンド・バー領域を囲む分割コンテナ。
.button	コマンド・バーのボタンに使用されるスタイル。

リスト

スタイル名	説明
.list	行のある表。
.listHeader	リストの見出し行に使用されるスタイル。
.ascending	リストをこの列で昇順にソートする場合のリスト・ヘッダー・クラスのスタイル。
.descending	リストをこの列で降順にソートする場合のリスト・ヘッダー・クラスのスタイル。
.unsorted	リストをこの列でソートしない場合のリスト・ヘッダー・クラスのスタイル。

詳細パネル

スタイル名	説明
.details	詳細パネルを囲む分割コンテナ。
.detailsProperty	プロパティ名のラベル。
.detailsValue	プロパティ値のテキスト。

メッセージ・データ

スタイル名	説明
.messageData	メッセージを囲む分割コンテナ。
.messageDataButton	メッセージ・フォームでの「追加」ボタンおよび「除去」ボタンのボタン・スタイル。

スタイル名	説明
.messageDataOutput	レンダリングの読み取り専用テキストに使用。
.messageDataValidInput	有効なメッセージ値に使用。
.messageDataInvalidInput	無効なメッセージ値に使用。

タブ付きペイン

スタイル名	説明
.tabbedPane	すべてのタブ付きペインを囲む分割コンテナ。
.tabHeader	タブ付きペインのタブ・ヘッダー。
.selectedTab	アクティブなタブ・ヘッダー。
.tab	非アクティブのタブ・ヘッダー。
.tabPane	タブ付きペインを囲む分割コンテナ。
.tabbedPaneNested	検索ページで使用されるネストされたタブ付きペインを囲む分割コンテナ。
.tabHeaderSimple	ネストされたタブ付きペインのタブ・ヘッダー。
.tabHeaderProcess	プロセス・フィルター用のネストされたタブ付きペインのタブ・ヘッダー。
.tabHeaderTask	タスク・フィルター用のネストされたタブ付きペインのタブ・ヘッダー。
.tabPaneSimple	ネストされたタブ付きペインを囲む分割コンテナ。

検索ページ

スタイル名	説明
.searchPane	検索パネル用のタブ付きペイン。タブ付きペインも参照。
.searchPanelFilter	検索書式のテーブル・コンテナ。
.searchLabel	検索書式コントロールのラベル。
.summary	検索要約ペインを囲むコンテナ。
.summaryTitle	検索要約ペインのすべてのタイトルの共通スタイル。
.summaryTitleProcess	検索要約ペインのプロセス関連セクションのタイトルのスタイル。
.summaryTitleTask	検索要約ペインのタスク関連セクションのタイトルのスタイル。

エラー詳細

スタイル名	説明
.errorPage	エラー・ページのタブ付きペイン。
.errorLink	ページのボタン・リンクをレンダリングするときに使用するスタイル。
.errorDetails	エラー詳細のあるタブ付きペイン。
.errorDetailsStack	例外スタックのあるタブ付きペイン。
.errorDetailsMessage	エラー・メッセージのテキスト・スタイル。

Business Process Choreographer Observer 入門

ビジネス・プロセスおよびタスクの実行中に、WebSphere® Process Server はプロセス・インスタンスとそれに関連するアクティビティの状態変更に関する情報が入ったイベントを発行できます。Business Process Choreographer Observer を使用すると、それらのイベントに基づいた統計情報を取り出し、プロセスとアクティビティに関するレポートを作成できます。

このタスクについて

特定のプロセス・インスタンス、アクティビティ・インスタンス、またはインライン・ヒューマン・タスクに関する詳細な情報を取得するために、独自のレポートを定義するか、ドリルダウン・アプローチを使用できます。また、報告された結果をエクスポートし、外部でさらに処理することもできます。

Business Process Choreographer Observer は、IT レベルのモニターとビジネス・レベルのモニターの間をギャップを埋めます。また、Business Flow Manager コンポーネントでのイベントを報告する手段を提供することにより、Business Process Choreographer 内で起きていることを理解するのに役立ちます。

関連概念

『Business Process Choreographer Observer のユーザー・インターフェース』
Business Process Choreographer Observer は、プロセス・イベントおよびアクティビティ・イベントを通知するための一連の機能を備えたスタンドアロン Web アプリケーションです。このインターフェースは、タスクバー、ナビゲーション・ペイン、およびワークスペースで構成されます。

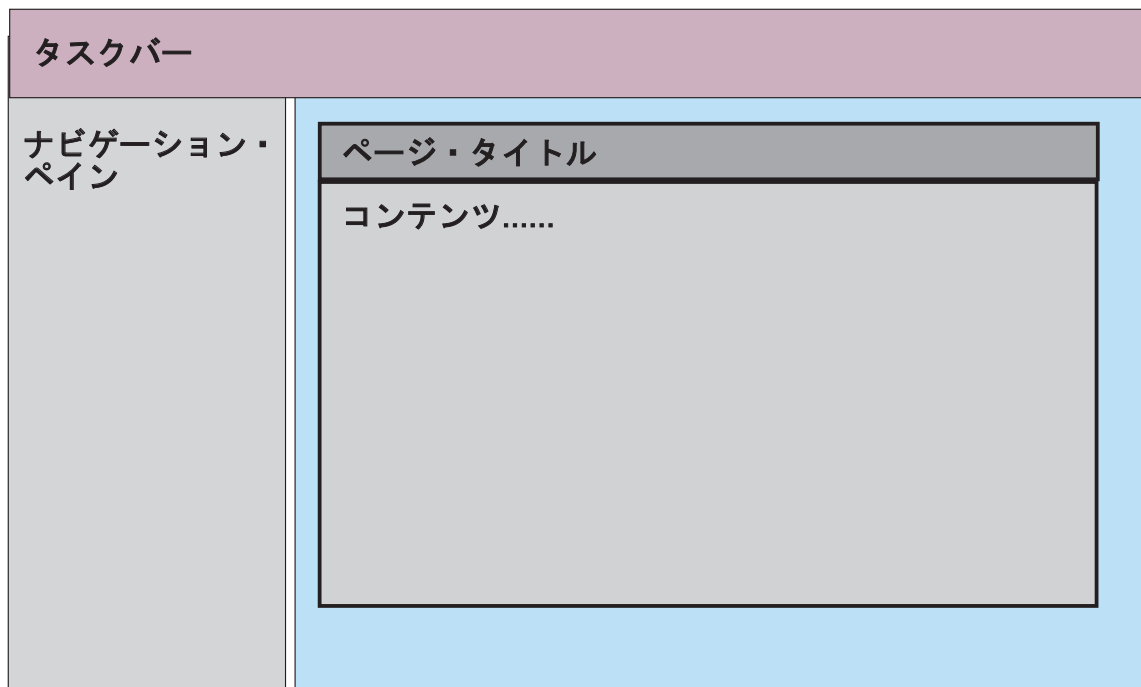
40 ページの『Business Process Choreographer Observer のナビゲーション・ペイン』

作成するレポートの種類 (プロセス・レポートやアクティビティ・レポートなど) を選択する場合は、ナビゲーション・ペインを使用します。独自のレポート定義を保管して、これらの定義をナビゲーション・ペインに追加することもできます。

Business Process Choreographer Observer のユーザー・インターフェース

Business Process Choreographer Observer は、プロセス・イベントおよびアクティビティ・イベントを通知するための一連の機能を備えたスタンドアロン Web アプリケーションです。このインターフェースは、タスクバー、ナビゲーション・ペイン、およびワークスペースで構成されます。

次の図に、Business Process Choreographer Observer のユーザー・インターフェースのレイアウトを示します。



このユーザー・インターフェースの主領域について、以下に説明します。


タスクバー

タスクバーによって提供されるオプションは、Business Process Choreographer Observer からのログアウト、および一般的な「ヘルプ」ページへのリンクです。

ナビゲーション・ペイン

ユーザー・インターフェースの左側にあるナビゲーション・ペインには、作成するレポートの種類を選択するときに使用するリンクが表示されます。例えば、アクティビティ・インスタンスのデータをグラフで表示できます。

ワークスペース

ユーザー・インターフェースの右側にあるワークスペースには、レポート定義を指定してレポートを表示するときに使用するページが表示されます。これらのページにアクセスするには、ナビゲーション・ペインのリンクをクリックします。ページについて詳しくは、それぞれのページの「ヘルプ」アイコン  をクリックしてください。

関連タスク

38 ページの『Business Process Choreographer Observer 入門』

ビジネス・プロセスおよびタスクの実行中に、WebSphere® Process Server はプロセス・インスタンスとそれに関連するアクティビティの状態変更に関する情報が入ったイベントを発行できます。Business Process Choreographer Observer を使用すると、それらのイベントに基づいた統計情報を取り出し、プロセスとアクティビティに関するレポートを作成できます。

Business Process Choreographer Observer のナビゲーション・ペイン

作成するレポートの種類 (プロセス・レポートやアクティビティー・レポートなど) を選択する場合は、ナビゲーション・ペインを使用します。独自のレポート定義を保管して、これらの定義をナビゲーション・ペインに追加することもできます。

使用可能なアクション

ナビゲーション・ペインでは、以下のアクションを使用できます。

- グループを縮小表示または展開する。

ナビゲーション・ペイン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。

- 事前定義のリストまたはグラフにナビゲートする。

レポートを作成するインスタンスの種類をクリックします。

- プロセス・レポート・ウィザードまたはアクティビティー・レポート・ウィザードへナビゲートする。


レポートのタイプ、レポートの内容、レポートのフィルター基準を指定するに

は、「新規レポート」アイコン () をクリックします。

- 保存済みのプロセス・レポートまたはアクティビティー・レポートを実行する。

レポートを実行するには、レポート名をクリックします。


- 保存済みのプロセス・レポート定義またはアクティビティー・レポート定義のポップアップ・メニューを開く。

保存済みのレポート定義を操作するには、「ポップアップ・メニューの表示」アイコン () をクリックします。


– レポート定義を編集するには、「編集」アイコン () をクリックします。

– レポート定義をコピーするには、「コピー」アイコン () をクリックします。

– レポート定義を削除するには、「削除」アイコン () をクリックします。

– レポート結果をエクスポートするには、「エクスポート」アイコン () をクリックします。

– 検索を非同期で実行するには、「非同期検索 (Asynchronous Search)」アイコン () をクリックします。

– 非同期検索が正常に完了すると、ナビゲーション・ペインに「非同期検索完了 (Asynchronous Search Completed)」アイコン () が表示されます。検索結果を表示する場合は、レポートの名前をクリックします。

- 非同期検索が正常に完了しなかった場合は、「非同期検索失敗 (Asynchronous Search Failed)」アイコン (🚫) が表示されます。

ナビゲーション・ペインの事前定義リストおよびグラフ

ナビゲーション・ペインには、以下に示す事前定義リストおよびグラフのグループがあります。

リスト このグループには、以下のリストがあります。

プロセス

このリストは、指定した期間内にプロセス・イベントを出力したプロセスを表示する場合に使用します。プロセスのリストは、プロセス状態に応じて表示されます。

アクティビティ

このリストは、選択したアクティビティが、指定した期間内に到達した状態を表示する場合に使用します。アクティビティのリストは、アクティビティ状態に応じて表示されます。

ユーザー

このリストは、指定した期間内に、選択したユーザーが実行したアクティビティと、そのアクティビティが到達した状態を表示する場合に使用します。アクティビティは、その状態に応じて表示されます。対応するユーザーは、アクティビティごとに表示されます。

重要: 従業員の業務遂行能力に関するデータを収集することがプライバシー保護およびデータ保護の法律に違反する国では、個々のユーザーのアクティビティ・イベントを出力しないようプロセス・モデルを定義する必要があります。

グラフ このグループには、以下のグラフがあります。

プロセス・スナップショット

このグラフは、指定した時刻に状態の異なるプロセス・インスタンスの数を調べる場合に使用します。データは棒グラフまたは円グラフで表示できます。

期間によるプロセス

このグラフは、指定した期間内に指定した状態に到達したプロセス・インスタンスの分布を調べる場合に使用します。各インスタンスは、それが指定の状態に到達したタイム・スライス内に表示されます。データは折れ線グラフ、棒グラフ、または円グラフで表示できます。


アクティビティ・スナップショット

このグラフは、指定した時刻に状態の異なるアクティビティ・インスタンスの数を調べる場合に使用します。データは棒グラフまたは円グラフで表示できます。

期間によるアクティビティ

このグラフは、指定した期間内に指定した状態に到達したアクティビティ・インスタンスの分布を調べる場合に使用します。各インスタンスは、それが指定の状態に到達したタイム・スライス内に表示されます。データは折れ線グラフ、棒グラフ、または円グラフで表示できます。

プロセス・レポートおよびアクティビティ・レポート

ナビゲーション・ペインには、以下のレポート・ウィザードへのリンクがあります。レポート・ウィザードは「**新規レポート**」アイコン () で示されます。

プロセス・レポート

プロセス・レポートは、プロセス・インスタンス・イベントを照会する場合に使用します。これらのイベントには、プロセス・インスタンスの状態変更が記述されています。作成するレポートのデータを定義する場合は、レポート・ウィザードを使用してください。レポート定義を保管して検索することができます。

アクティビティ・レポート

アクティビティ・インスタンス・イベントを照会する場合は、アクティビティ・レポートを使用します。これらのイベントには、アクティビティ・インスタンスの状態変更が記述されています。個々のレポートを指定する場合は、レポート・ウィザードを使用します。レポート定義を保管して検索することができます。

関連タスク

38 ページの『Business Process Choreographer Observer 入門』

ビジネス・プロセスおよびタスクの実行中に、WebSphere® Process Server はプロセス・インスタンスとそれに関連するアクティビティの状態変更に関する情報が入ったイベントを発行できます。Business Process Choreographer Observer を使用すると、それらのイベントに基づいた統計情報を取り出し、プロセスとアクティビティに関するレポートを作成できます。

第 3 章 サーバーおよびクラスターの管理

サーバーの管理には、サーバーの作成、Service Component Architecture (SCA) モジュールをサポートするためのサーバーの構成、およびサーバーの開始と停止が含まれます。

サーバーの作成

管理対象ノードでサーバーを作成するには、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 `wsadmin createApplicationServer` コマンドを使用します。

始める前に

管理対象ノードを作成するには、カスタム・プロファイルを作成し、このプロファイルをデプロイメント・マネージャーに統合しておく必要があります。

このタスクについて

以下の手順は、デプロイメント・マネージャーで管理されるサーバーを作成するためのものです。スタンドアロン・サーバーを作成するには、スタンドアロン・サーバー・プロファイルを作成します。

管理コンソールを使用する方法以外の方法でも、サーバーを作成できます (`wsadmin createApplicationServer` コマンドなど)。詳しくは、『Commands for the AdminTask object』を参照してください。

プロシージャ

1. 管理コンソールで「アプリケーション・サーバー」 → 「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」ウィザードを開始します。
2. 『Creating application servers』の手順に従います。「サーバー・テンプレートの選択」パネルで **defaultProcessServer** テンプレートまたは適切なユーザー定義テンプレートを選択します。
3. **オプション:** 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がサーバーで実行される場合は、**Business Process Choreographer** を構成します。詳しくは、『Business Process Choreographer の構成』を参照してください。
4. **オプション:** 『Business Process Choreographer の構成』の説明に従って、**SchedulerCalendars** 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。

結果

これで、サーバーを始動し、モジュールをサーバーにデプロイできます。

サーバーまたはクラスタの SCA サポートの構成

Service Component Architecture コンソール・ページでは、サービス・アプリケーションまたはそれに必要なメッセージング・エンジンと宛先 (あるいはこの両方) をホストできるように、Network Deployment 環境内のサーバーまたはクラスタを設定できます。

始める前に

SCA サポートを構成する前に、以下の点を確認します。

- スタンドアロン・サーバー・プロファイルを使用しているかどうか。使用している場合は SCA サポートが既に構成されているので、Service Component Architecture ページでこのサポートを除去することはできません。ただし、このページでデータベース・データ・ソースの一部のプロパティを変更することはできます。
- メッセージング・エンジンと宛先をホストする場所 (リモート・バス・メンバーまたはローカル・バス・メンバーを使用)。
- SCA システム・バスのみを構成する必要があるのか、それとも SCA アプリケーション・バスも構成する必要があるのか。アプリケーション・バスはデフォルトで構成されます。WebSphere Business Integration Adapters を使用する SCA アプリケーションをデプロイする場合は、アプリケーション・バスが必要です。

このタスクに必要なセキュリティ・ロール: 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。

このタスクについて

サーバーまたはクラスタ上で SCA サポートを構成するには、以下のステップを実行します。

プロシージャ

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」 → 「アプリケーション・サーバー」 → 「*serverName*」 → 「Service Component Architecture」
 - 「サーバー」 → 「クラスタ」 → 「*clusterName*」 → 「Service Component Architecture」
2. 「Service Component Architecture コンポーネントのサポート」をクリックします。
3. 「バス・メンバー・ロケーション」パネルで、SCA アプリケーションに必要な宛先とメッセージング・エンジンをホストする場所を指定します。2 つのオプションがあります。
 - **ローカル。** SCA アプリケーション、宛先、およびメッセージング・エンジンを現行サーバーまたはクラスタでホストすることを指定します。
 - **リモート。** SCA アプリケーションを現行サーバーまたはクラスタでホストし、宛先およびメッセージング・エンジンをリモート・サーバーまたはクラスタ (デプロイメント・ターゲット ともいう) でホストすることを指定します。

4. (リモート・バス・メンバーのみ) 前のステップで「リモート」を選択した場合は、ホスト・アプリケーションの宛先およびメッセージング・エンジンをホストするリモート・サーバーまたはクラスターを指定します。ドロップダウン・メニューから、既存のデプロイメント・ターゲット (既に SCA システム・バスのメンバーとして構成されているターゲット) を選択するか、または「新規」をクリックして、「デプロイメント・ターゲットの参照」ページから新規サーバーまたはクラスターを選択します。

「デプロイメント・ターゲットの参照」ページから新規サーバーまたはクラスターを選択する場合は、このトピックで説明する SCA の構成が完了すると、そのターゲットで必要なメッセージング・エンジンが自動的に構成されます。

5. 「システム・バス・メンバー」パネルの表で、システム・バス・データ・ソース構成を確認または変更します。
 - a. 「データベース・インスタンス」、「スキーマ」、「テーブルの作成」、「ユーザー名」、「パスワード」、「サーバー」、および「プロバイダー」フィールドのデフォルト値を確認します。各フィールドと受け入れ可能な値の詳細については、オンライン・ヘルプを参照してください。
 - b. これらのフィールドにデフォルト値がない場合、またはデフォルト値が適切でない場合は、システム・バス・データ・ソースに対する適切な値を入力します。フィールドに値を直接入力するか、または「編集」をクリックしてデータ・ソースの詳細ページで値を編集します。
 - c. オプション: 「接続のテスト」をクリックして、データ・ソースがデータベースに接続でき、正しく認証されることを確認します。
6. 「アプリケーション・バス・メンバー」パネルの表で、アプリケーション・バス・データ・ソース構成を確認または変更します。
 - a. 「WebSphere Business Integration Adapter コンポーネントの使用可能化」オプションが選択されていることを確認します。

注: アプリケーション・バスを使用しない場合は、「WebSphere Business Integration Adapter コンポーネントの使用可能化」オプションをクリアし、ステップ 7 に進みます。
 - b. 「データベース・インスタンス」、「スキーマ」、「テーブルの作成」、「ユーザー名」、「パスワード」、「サーバー」、および「プロバイダー」フィールドのデフォルト値を確認します。各フィールドと受け入れ可能な値の詳細については、オンライン・ヘルプを参照してください。
 - c. これらのフィールドにデフォルト値がない場合、またはデフォルト値が適切でない場合は、アプリケーション・バス・データ・ソースに対する適切な値を入力します。フィールドに値を直接入力するか、または「編集」をクリックしてデータ・ソースの詳細ページで値を編集します。
 - d. オプション: 「接続のテスト」をクリックして、データ・ソースがデータベースに接続でき、正しく認証されることを確認します。
7. 「OK」をクリックすると、SCA 構成が完了します。

関連情報

デプロイメント環境の計画

 メディエーション・モジュールのリソースの管理

サーバーおよびクラスター上での Service Component Architecture サポートに関する考慮事項

サーバーおよびクラスターは、Service Component Architecture (SCA) アプリケーションまたはアプリケーション宛先 (あるいはその両方) をサポートできます。

SCA アプリケーション (サービス・アプリケーションともいいます) では、自動的に作成された 1 つ以上のサービス統合バスを使用する必要があります。各アプリケーションは、宛先 と呼ばれる一連のメッセージング・リソースを使用します。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ーか、リモートのサーバーまたはクラスター上でホストできます。一般に、メッセージング・エンジンはデータベース・データ・ソースを使用します。スタンドアロン・サーバー・プロファイルの作成中にファイル・ストアを使用するオプションが選択された場合は、データベース・データ・ソースの代わりにファイル・ストアを使用できます。

デフォルトでは、Network Deployment 環境または管理対象ノード環境内の新規サーバーおよびクラスターは、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。

注: スタンドアロン・サーバーでは、SCA サポートが自動的に構成されます。この構成を使用不可にすることはできません。

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。サーバーの場合、アプリケーション・クラス・ローダー・ポリシーが「複数」に設定されていることを確認します。

Network Deployment 環境または管理対象ノード環境でサーバーまたはクラスターの SCA サポートを使用可能にする前に、以下に示す構成のうちのどちらを実装するかを決定します。

- **リモート・バス・メンバー構成:** 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・エンジンと共に、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。

リモート・メッセージングを使用するには、サービス統合バスとそのメンバーの計画と構成のために初期投資が必要です。しかし、この構成は、アプリケーション・クラスター内の複数のメンバーによって再利用できます。メッセージはすべてのメンバーに配信されます。さらに、フェイルオーバー・サポートを提供するように初期構成を構築できます。

- **ローカル・バス・メンバー構成:** 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の両方をホストします。必要なメッセージング・エンジンは、サーバーまたはクラスター上のローカル・バス・メンバーを使用して構成されます。

ご使用の環境に適した構成を判別するために役立つ情報については、計画に関するトピックを参照してください。

関連情報

 サーバーのクラス・ローダーの構成

 サービス統合バスについて

 メッセージング・エンジン


サーバーおよびクラスタの開始または停止

サーバーでアプリケーションを開始する前に、サーバーが稼働している必要があります。サーバーを開始する方法はいくつかありますが、スタンドアロン・サーバーと管理対象サーバーのいずれであるかによって、使用方法が異なります。管理対象サーバーの場合、サーバーを開始する前に、ノード・エージェントが稼働している必要があります。管理対象サーバーは、デプロイメント・マネージャーの管理コンソールから開始できます。デプロイメント環境またはクラスタを導入している場合、デプロイメント・マネージャーの管理コンソールですべてのサーバーを一括に開始または停止できます。

このタスクについて

クラスタを使用している場合、「アプリケーション・サーバー」サブコンポーネント（「サーバー」>「アプリケーション・サーバー」>「*server_name*」>「管理」>「サーバー・コンポーネント」>「アプリケーション・サーバー」）の「初期状態」プロパティは、クラスタの開始時にクラスタ内の個々のサーバーの状態を制御する目的では使用できません。これは、サーバーの「アプリケーション・サーバー」サブコンポーネントの状態を制御するためのものです。クラスタの個々のメンバーを開始および停止するには、管理コンソールの「サーバー」のオプションを使用するか、またはコマンド行のコマンド (**startServer** および **stopServer**) を使用する方法をお勧めします。

関連タスク

 ファースト・ステップ・コンソール (First steps console)

WebSphere Process Server をインストールしたら、ファースト・ステップ・コンソールを使用して、製品ツールの開始、製品資料へのアクセス、個別プロファイルに関連するサーバーおよび管理コンソールなどのエレメントへの指示を行います。汎用バージョンのコンソールと、インストール内のプロファイルごとのバージョンが使用可能です。

16 ページの『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。

スタンドアロン・サーバーの開始

管理コンソールを使用する前、およびサーバーでアプリケーションを実行する前に、サーバー・プロセスを開始しておく必要があります。


始める前に

この時点でサーバーが実行されてはなりません。インストール検査プログラムを実行すると、サーバーが開始される点に注意してください。

このタスクについて

スタンドアロン・サーバーを開始する方法はいくつかあります。複数のスタンドアロン・サーバー・プロファイルを作成したホストでは、正しいプロファイルを使用するように注意してください。

サーバーを開始するには、以下のいずれかの手順を実行します。

-  「スタート」メニューで「IBM WebSphere」 → 「Process Server 6.1」 → 「プロファイル」 → 「*profile_name*」 → 「サーバーの始動」をクリックします。

ここで、*profile_name* はプロファイルの名前です (例: ProcSrv01)。

- ファースト・ステップ・コンソールで、「サーバーの始動」をクリックします。
- startServer コマンドを使用します。

install_root/profiles/profile_name/logs/server_name/startServer.log に進行状況が記録されます。サーバーが正常に始動すると、「サーバー *server_name* が e-business 用にオープンされました。プロセス ID は *nnnn* です。」というメッセージがログに記録されます。

次のタスク

これで、管理コンソールを始動し、スタンドアロン・サーバーを管理し、アプリケーションをサーバーにデプロイできます。WebSphere Process Server を今回初めて使用する場合、既にサンプルをインストールしていれば、ファースト・ステップ・コンソールからサンプル・ギャラリーを開くことができます。

関連概念

プロファイル

プロファイルでは、個別のコマンド・ファイル、構成ファイル、ログ・ファイルを持つ固有のランタイム環境を定義します。プロファイルでは、スタンドアロン・サーバー、デプロイメント・マネージャー、および管理対象ノードの 3 つのタイプの環境を定義します。

サーバー

サーバーは、WebSphere Process Server の中核機能を提供します。プロセス・サーバーは、アプリケーション・サーバーが Service Component Architecture (SCA) モジュールを操作する能力を拡張します。その他のサーバー (デプロイメント・マネージャーおよびノード・エージェント) は、プロセス・サーバーの管理に使用されます。

関連タスク

49 ページの『スタンドアロン・サーバーの停止』

サーバーの構成を変更する場合、またはサーバーにデプロイされているモジュールを変更する場合は、サーバー・プロセスを停止する必要があります。

16 ページの『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。

61 ページの『サーバーを Java デバッグ可能にする』

サーバーの始動後に、サーバーをトレースおよびデバッグする必要がある場合は、Java デバッグ可能にできます。



ファースト・ステップ・コンソール (First steps console)

WebSphere Process Server をインストールしたら、ファースト・ステップ・コンソールを使用して、製品ツールの開始、製品資料へのアクセス、個別プロファイルに関連するサーバーおよび管理コンソールなどのエレメントへの指示を行います。汎用バージョンのコンソールと、インストール内のプロファイルごとのバージョンが使用可能です。



ファースト・ステップ・コンソールのオプション

WebSphere Process Server をインストールしたら、ファースト・ステップ・コンソールを使用して、製品ツールの開始、製品資料へのアクセス、個別プロファイルに関連するサーバーおよび管理コンソールなどのエレメントへの指示を行います。汎用バージョンのコンソールと、インストール内のプロファイルごとのバージョンが使用可能です。各コンソールのオプションは、インストールした機能と、当該オペレーティング・システムで使用可能なエレメントに応じて動的に表示されます。オプションには、インストールの検証、サーバーまたはデプロイメント・マネージャーの開始と停止、管理コンソールへのアクセス、プロファイル管理ツールの開始、サンプル・ギャラリーへのアクセス、製品資料へのアクセス、マイグレーション・ウィザードの開始などがあります。

関連情報



startServer コマンド

スタンドアロン・サーバーの停止

サーバーの構成を変更する場合、またはサーバーにデプロイされているモジュールを変更する場合は、サーバー・プロセスを停止する必要があります。

始める前に

1. サーバーは実行されていなければなりません。
2. 処理が完了するまで待ちます。十分な期間にわたって待機するか、またはパフォーマンス・モニター・インフラストラクチャー・カウンターを使用して、キューに格納されているすべての処理が完了したかどうかを判別します。
3. 新規処理が開始しないようにします。

このタスクについて

サーバーを停止する方法はいくつかあります。

サーバーを停止するには、以下のいずれかの手順を実行します。

- **Windows** 「スタート」メニューで「IBM WebSphere」 → 「Process Server 6.1」 → 「プロファイル」 → 「*profile_name*」 → 「サーバーの停止」の順に選択します。

ここで、*profile_name* はプロファイルの名前です (例: ProcSrv01)。

- ファースト・ステップ・コンソールで、「サーバーの停止」をクリックします。
- stopServer コマンドを使用します。

サーバー・ログ *install_root/profiles/profile_name/logs/server_name/stopServer.log* に進行状況が記録されます。サーバーが正常に停止すると、「サーバー *server_name* の停止が完了しました。」というメッセージがサーバー・ログに記録されます。

次のタスク

サーバーが停止するまで待ち、停止したら、サーバーを再始動します。

関連概念

プロファイル

プロファイルでは、個別のコマンド・ファイル、構成ファイル、ログ・ファイルを持つ固有のランタイム環境を定義します。プロファイルでは、スタンドアロン・サーバー、デプロイメント・マネージャー、および管理対象ノードの 3 つのタイプの環境を定義します。

サーバー

サーバーは、WebSphere Process Server の中核機能を提供します。プロセス・サーバーは、アプリケーション・サーバーが Service Component Architecture (SCA) モジュールを操作する能力を拡張します。その他のサーバー (デプロイメント・マネージャーおよびノード・エージェント) は、プロセス・サーバーの管理に使用されます。

プロファイル

プロファイルでは、個別のコマンド・ファイル、構成ファイル、ログ・ファイルを持つ固有のランタイム環境を定義します。プロファイルでは、スタンドアロン・サーバー、デプロイメント・マネージャー、および管理対象ノードの 3 つのタイプの環境を定義します。

関連タスク

47 ページの『スタンドアロン・サーバーの開始』

管理コンソールを使用する前、およびサーバーでアプリケーションを実行する前に、サーバー・プロセスを開始しておく必要があります。

ファースト・ステップ・コンソール (First steps console)

WebSphere Process Server をインストールしたら、ファースト・ステップ・コンソールを使用して、製品ツールの開始、製品資料へのアクセス、個別プロファイ

ルに関連するサーバーおよび管理コンソールなどのエレメントへの指示を行います。汎用バージョンのコンソールと、インストール内のプロファイルごとのバージョンが使用可能です。

関連情報

 stopServer コマンド

デプロイメント・マネージャーの開始

デプロイメント・マネージャーの管理コンソールを使用してセルを管理する前に、そのデプロイメント・マネージャーを開始する必要があります。


始める前に

デプロイメント・マネージャーが稼働してはなりません。インストール検査プログラムを実行すると、デプロイメント・マネージャーが開始される点に注意してください。

このタスクについて

デプロイメント・マネージャーを開始する方法はいくつかあります。

デプロイメント・マネージャーを開始するには、次のようにします。

-  「スタート」メニューで「IBM WebSphere」 → 「Process Server 6.1」 → 「プロファイル」 → 「*profile_name*」 → 「デプロイメント・マネージャーの開始」の順に選択します。

ここで、*profile_name* はプロファイルの名前です (例: Dmgr01)。

- ファースト・ステップ・コンソールで、「デプロイメント・マネージャーの始動」をクリックします。
- startManager コマンドを使用します。

install_root/profiles/profile_name/logs/server_name/startServer.log に進行状況が記録されます。サーバーが正常に始動すると、「サーバー *server_name* が e-business 用にオープンされました。プロセス ID は *nnnn* です。」というメッセージがログに記録されます。

次のタスク

これで、管理コンソールを開始し、セルを管理できます。ノードをデプロイメント・マネージャーに統合してから、サーバーとクラスターを作成できます。

関連概念

 プロファイル

プロファイルでは、個別のコマンド・ファイル、構成ファイル、ログ・ファイルを持つ固有のランタイム環境を定義します。プロファイルでは、スタンドアロン・サーバー、デプロイメント・マネージャー、および管理対象ノードの 3 つのタイプの環境を定義します。

 デプロイメント・マネージャー

デプロイメント・マネージャーとは、他のサーバーの論理グループ (セル) の操作を管理するサーバーです。デプロイメント・マネージャーは、サーバーおよびクラスターを管理するための中央の場所になっています。

関連タスク

『デプロイメント・マネージャーの停止』

デプロイメント・マネージャーはサーバー・プロセスです。このため、オペレーティング・システムの保守作業を行う場合などに停止する必要があります。デプロイメント・マネージャーは、ドメイン内のその他のサーバーの運用に影響せずにいつでも停止できます。



ファースト・ステップ・コンソール (First steps console)

WebSphere Process Server をインストールしたら、ファースト・ステップ・コンソールを使用して、製品ツールの開始、製品資料へのアクセス、個別プロファイルに関連するサーバーおよび管理コンソールなどのエレメントへの指示を行います。汎用バージョンのコンソールと、インストール内のプロファイルごとのバージョンが使用可能です。



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

addNode コマンドを使用して、カスタム・ノードをデプロイメント・マネージャー・セルに統合する方法について説明します。

16 ページの『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。

関連情報



startManager コマンド

デプロイメント・マネージャーの停止

デプロイメント・マネージャーはサーバー・プロセスです。このため、オペレーティング・システムの保守作業を行う場合などに停止する必要があります。デプロイメント・マネージャーは、ドメイン内のその他のサーバーの運用に影響せずにいつでも停止できます。

始める前に

デプロイメント・マネージャーは稼働している必要があります。

このタスクについて

デプロイメント・マネージャーを停止する方法はいくつかあります。

デプロイメント・マネージャーを停止するには、以下のいずれかの手順を実行します。

- **Windows** 「スタート」メニューで「IBM WebSphere」 → 「Process Server 6.1」 → 「プロファイル」 → 「*profile_name*」 → 「デプロイメント・マネージャーの停止」をクリックします。

ここで、*profile_name* はプロファイルの名前です (例: **Dmgr01**)。

- ファースト・ステップ・コンソールで、「デプロイメント・マネージャーの停止」をクリックします。
- 管理コンソールで「システム管理」 → 「デプロイメント・マネージャー」 → 「停止」 → 「OK」の順にクリックします。管理コンソールが閉じ、その後サーバーが実行停止します。
- stopManager コマンドを使用します。

サーバー・ログ *install_root/profiles/profile_name/logs/server_name/stopServer.log* に進行状況が記録されます。サーバーが正常に停止すると、「サーバー *server_name* の停止が完了しました。」というメッセージがサーバー・ログに記録されます。

次のタスク

デプロイメント・マネージャーが停止するまで待ち、停止したら、サーバーを再始動します。

関連概念

プロファイル

プロファイルでは、個別のコマンド・ファイル、構成ファイル、ログ・ファイルを持つ固有のランタイム環境を定義します。プロファイルでは、スタンドアロン・サーバー、デプロイメント・マネージャー、および管理対象ノードの 3 つのタイプの環境を定義します。

デプロイメント・マネージャー

デプロイメント・マネージャーとは、他のサーバーの論理グループ (セル) の操作を管理するサーバーです。デプロイメント・マネージャーは、サーバーおよびクラスターを管理するための中央の場所になっています。

関連タスク

51 ページの『デプロイメント・マネージャーの開始』

デプロイメント・マネージャーの管理コンソールを使用してセルを管理する前に、そのデプロイメント・マネージャーを開始する必要があります。

ファースト・ステップ・コンソール (First steps console)

WebSphere Process Server をインストールしたら、ファースト・ステップ・コンソールを使用して、製品ツールの開始、製品資料へのアクセス、個別プロファイルに関連するサーバーおよび管理コンソールなどのエレメントへの指示を行います。汎用バージョンのコンソールと、インストール内のプロファイルごとのバージョンが使用可能です。

関連情報

ノード・エージェントの開始

管理対象ノードのノード・エージェントは、ノードでサーバーを開始する前に開始する必要があるサーバー・プロセスです。デプロイメント・マネージャーがノード・エージェントと通信するには、ノード・エージェントが開始されている必要があります。

始める前に

ノード・エージェントが稼働してはなりません。管理コンソールからノード・エージェントが稼働しているかどうかを判断する手順をここで説明します。

このタスクについて

ノード・エージェントは、ノードが構成されているホストのコマンド行で、`install_root/bin` ディレクトリーから開始する必要があります。

プロシージャ

1. デプロイメント・マネージャーの管理コンソールからサーバーの正しいノード・エージェントを判別するには、以下のようにします。「サーバー」 → 「アプリケーション・サーバー」をクリックします。
 2. ノード・エージェントが稼働しているかどうかを判断するには、以下のようにします。「システム管理」 → 「ノード・エージェント」をクリックします。
 3. ノード・エージェントを開始するには、次のようにします。 `startNode` コマンドを実行します。このコマンドで使用できるオプションを確認するには、**-help** オプションを使用します。オプションを指定しないと、ノードが開始されます。「サーバー `nodeagent` が `e-business` 用にオープンされました。プロセス ID は `nnnn` です。」というメッセージが表示されるまで待機します。
- デフォルト・プロファイルのノード・エージェントを開始するには、`startNode` と入力します。
 - オプションのリストを表示するには、`startNode -help` と入力します。
 - `Custom03` プロファイルのノード・エージェントを開始するには、`startNode -profileName Custom03` と入力します。
 - `Custom03` プロファイルのノード・エージェントを開始し、トレース情報をログ・ファイル `install_root/profiles/Custom03/logs/startServer.log` に書き込むには、`startNode -logfile -profileName Custom03` と入力します。

次のタスク

これで、デプロイメント・マネージャーからこのノードを管理できます (ノード上のサーバーを開始することもできます)。

関連概念

管理対象ノードとは、デプロイメント・マネージャー・セルに統合されているノードのことです。管理対象ノードでは、管理対象サーバーを構成して実行できません。

関連タスク

56 ページの『ノード・エージェントの再開』

管理対象ノードのノード・エージェントは、再始動する必要があるサーバー・プロセスです。

『ノード・エージェントの停止』

管理対象ノードのノード・エージェントは、停止する必要があるサーバー・プロセスです。

関連情報



startNode コマンド

ノード・エージェントの停止

管理対象ノードのノード・エージェントは、停止する必要があるサーバー・プロセスです。

始める前に

1. ノード・エージェントは実行されていなければなりません。
2. すべてのサーバーが停止している必要があります。

プロシージャ

1. デプロイメント・マネージャーの管理コンソールからノード・エージェントを停止するには、以下のようにします。「システム管理」 → 「ノード・エージェント」を選択します。
2. ノード・エージェントを選択します。
3. 「停止」をクリックします。

関連概念



管理対象ノード

管理対象ノードとは、デプロイメント・マネージャー・セルに統合されているノードのことです。管理対象ノードでは、管理対象サーバーを構成して実行できません。

関連タスク

54 ページの『ノード・エージェントの開始』

管理対象ノードのノード・エージェントは、ノードでサーバーを開始する前に開始する必要があるサーバー・プロセスです。デプロイメント・マネージャーがノード・エージェントと通信するには、ノード・エージェントが開始されている必要があります。

16 ページの『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。

『ノード・エージェントの再開』

管理対象ノードのノード・エージェントは、再始動する必要があるサーバー・プロセスです。

関連情報

 stopNode コマンド

ノード・エージェントの再開

管理対象ノードのノード・エージェントは、再始動する必要があるサーバー・プロセスです。

始める前に

ノード・エージェントは実行されていない必要があります。

プロシージャ

1. デプロイメント・マネージャーの管理コンソールからノード・エージェントを再開するには、以下のようにします。「システム管理」 → 「ノード・エージェント」をクリックします。
2. ノード・エージェントを選択します。
3. 「再始動」をクリックします。

関連概念

 管理対象ノード

管理対象ノードとは、デプロイメント・マネージャー・セルに統合されているノードのことです。管理対象ノードでは、管理対象サーバーを構成して実行できません。

関連タスク

54 ページの『ノード・エージェントの開始』

管理対象ノードのノード・エージェントは、ノードでサーバーを開始する前に開始する必要があるサーバー・プロセスです。デプロイメント・マネージャーがノード・エージェントと通信するには、ノード・エージェントが開始されている必要があります。

55 ページの『ノード・エージェントの停止』

管理対象ノードのノード・エージェントは、停止する必要があるサーバー・プロセスです。

16 ページの『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。

関連情報

 startNode コマンド

 stopNode コマンド

デプロイメント環境の開始と停止

IBM 提供のパターンに基づいて、管理コンソールから直接デプロイメント環境を開始または停止することができます。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

デプロイメント環境が存在している必要があります。

このタスクについて

デプロイメント環境を開始または停止する場合は必ず、以下のステップを使用します。

制限: カスタム・デプロイメント環境には、この手順を使用することはできません。クラスターを個別に開始および停止する必要があります。

プロシージャ

1. 開始するデプロイメント環境を制御するデプロイメント・マネージャーの管理コンソールにログインします。
2. デプロイメント環境の管理コンソール・ページにナビゲートします。「サーバー」→「デプロイメント環境」をクリックします。
3. 開始および停止するデプロイメント環境の名前の隣にあるチェック・ボックスをクリックします。
4. 選択したデプロイメント環境を開始または停止します。


アクション	結果
「開始」をクリックします。	デプロイメント・マネージャーにより、デプロイメント環境を構成するクラスターが始動されます。
「停止」をクリックします。	デプロイメント・マネージャーにより、デプロイメント環境を構成するクラスターが停止されます。

注: このプロセスは、ご使用のデプロイメント環境のサイズによっては数分かかる場合があります。

結果

画面が最新表示され、デプロイメント環境の状況が示されます。正常な場合、『デプロイメント環境の状況』に説明されているように、デプロイメント環境にデプロイメント環境の状態が表示されます。

関連概念

 デプロイメント環境

デプロイメント環境は、構成済みクラスター、サーバー、およびミドルウェアの集合であり、これらが連携して、Service Component Architecture (SCA) との対話をホストする環境を実現します。例えば、デプロイメント環境には、メッセージ宛先、ビジネス・イベントのプロセッサ、および管理プログラムのホストが含まれていることがあります。

関連タスク

16 ページの『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。

関連情報

 Administration Thin Client の使用

クラスターの開始

クラスター内のすべてのサーバー (クラスター・メンバー) を 1 回の操作で開始できます。ripple 始動によりまず各サーバーが停止され、その後順に開始されます。クラスターを開始すると、ワークロード管理が自動的に有効になります。

始める前に

1. ノード・エージェントは実行されていなければなりません。
2. クラスターにデプロイされているアプリケーションに必要なすべてのリソースが使用可能であることを確認し、すべての前提条件サブシステムを開始します。

プロシージャ

1. デプロイメント・マネージャーの管理コンソールからクラスターを開始するには、以下のようにします。「サーバー」 → 「クラスター」をクリックします。
2. クラスターを選択します。
3. サーバーが稼働していない場合は、「開始」をクリックします。
4. サーバーが稼働している場合は、「ripple 始動」をクリックします。まず各サーバーが停止され、その後順に開始されます。

次のタスク

これで、アプリケーションをクラスターで開始できるようになりました。いずれかのサーバーで実行中のプロセスをトレースおよびデバッグする必要がある場合は、Java デバッグ可能にできます。

関連概念

 クラスター

クラスターを利用すると、単一サーバーよりもアプリケーションのキャパシティと可用性が高くなります。

関連タスク

16 ページの『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。

『クラスタの停止』

クラスタ内のすべてのサーバー (クラスタ・メンバー) を 1 回の操作で停止できます。

クラスタの停止

クラスタ内のすべてのサーバー (クラスタ・メンバー) を 1 回の操作で停止できます。

始める前に

1. 処理が完了するまで待ちます。十分な期間にわたって待機するか、またはパフォーマンス・モニター・インフラストラクチャー・カウンターを使用して、キューに格納されているすべての処理が完了したかどうかを判別します。
2. 新規処理が開始しないようにします。

プロシージャ

1. デプロイメント・マネージャーの管理コンソールからクラスタを停止するには、以下のようにします。「サーバー」→「クラスタ」をクリックします。
2. クラスタを選択します。
3. サーバーを再始動する場合は、「**ripple 始動**」をクリックします。まず各サーバーが停止され、その後順に開始されます。
4. サーバーが稼働していない場合は、「**停止**」をクリックします。

次のタスク

「**ripple 始動**」を使用していない場合は、サーバーが停止するまで待機してから、再始動します。

関連概念

クラスタ

クラスタを利用すると、単一サーバーよりもアプリケーションのキャパシティと可用性が高くなります。

関連タスク

58 ページの『クラスタの開始』

クラスタ内のすべてのサーバー (クラスタ・メンバー) を 1 回の操作で開始できます。ripple 始動によりまず各サーバーが停止され、その後順に開始されます。クラスタを開始すると、ワークロード管理が自動的に有効になります。

16 ページの『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。

サーバーの開始

管理対象サーバーでアプリケーションを実行する前に、サーバー・プロセスを開始しておく必要があります。単一サーバー (クラスター・メンバーを含む) を開始できます。

始める前に

1. ノード・エージェントは実行されていなければなりません。
2. サーバーにデプロイされているアプリケーションに必要なすべてのリソースが使用可能であることを確認し、すべての前提条件サブシステムを開始します。
3. この時点でサーバーが実行されていなければなりません。サーバーで障害が発生しており、新規処理の開始前にリカバリー機能を完了させたい場合は、`startServer` コマンドと `-recovery` オプションを使用してサーバーをリカバリー・モードで再始動する必要があります。

プロシージャ

1. デプロイメント・マネージャーの管理コンソールからサーバーを開始するには、以下のようにします。「サーバー」 → 「アプリケーション・サーバー」をクリックします。
2. サーバーを選択します。
3. 「開始」をクリックします。 `install_root/profiles/profile_name/logs/server_name/startServer.log` に進行状況が記録されます。サーバーが正常に始動すると、「サーバー `server_name` が e-business 用にオープンされました。プロセス ID は `nnnn` です。」というメッセージがログに記録されます。

結果

その他のプロセスでは、新規開始サーバーが直ちにディスカバーされないことがあります。

次のタスク

これで、アプリケーションをサーバーで開始できるようになりました。サーバーで実行中のプロセスをトレースおよびデバッグする必要がある場合は、Java デバッグを可能にできます。

関連概念

 管理対象サーバー

管理対象サーバーとは、管理対象ノード上で構成されているサーバーです。デプロイ環境内でアプリケーションを実行するリソースを提供します。

関連タスク

16 ページの『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。

62 ページの『サーバーの停止』

サーバーの構成を変更する場合、またはサーバーにデプロイされているモジュール

ルを変更する場合は、サーバー・プロセスを停止する必要があります。単一サーバー (クラスター・メンバーを含む) を停止できます。

関連情報

 startServer コマンド

i5/OS コマンド行からのサーバーの開始

他に 2 つのサーバー開始方法があります。

- startServer Qshell コマンドを使用できます。
- Submit Job (SBMJOB) CL コマンドを使用できます。以下の CL コマンドを i5/OS コマンド行から実行できます。

```
SBMJOB CMD(CALL PGM(QWBI61/QWASSTRSVR) PARM('-profilePath'  
      'profile_root' '-server' 'server_name')) JOB(server_name)  
      JOBD(QWAS61/QWASJOBDD) JOBQ(QWAS61/QWASJOBQ) USER(QEJBSVR)  
      LANGID(*USRPRF)  
      CENTRYID(*USRPRF) CCSID(*USRPRF) OUTQ(QWAS61/QWASOUTQ) ALWMLTTHD(*YES)
```

server_name はサーバーの名前です。

profile_root はサーバーのプロファイル・ディレクトリーです。

サーバーを Java デバッグ可能にする

サーバーの始動後に、サーバーをトレースおよびデバッグする必要がある場合は、Java デバッグ可能にできます。

始める前に

サーバーは実行されていない必要があります。

プロシージャ

1. デプロイメント・マネージャーの管理コンソールから Java デバッグを可能にするには、以下のようにします。「サーバー」 → 「アプリケーション・サーバー」をクリックします。
2. サーバーを選択します。
3. 「サーバー・インフラストラクチャー」 → 「Java およびプロセス管理」 → 「プロセス定義」 → 「Java 仮想マシン」をクリックします。
4. 標準 Java デバッガーを有効にするには、以下のようにします。「Java 仮想マシン」ページで「デバッグ・モード」をクリックします。
5. 必要に応じて「デバッグ・モード」の引数を設定します。
6. 「OK」をクリックします。
7. 変更を保管します。

次のタスク

この時点でサーバーを停止する必要があります。停止後、スタンドアロン・サーバーの場合は再始動してください。

関連タスク

47 ページの『スタンドアロン・サーバーの開始』

管理コンソールを使用する前、およびサーバーでアプリケーションを実行する前に、サーバー・プロセスを開始しておく必要があります。

サーバーの停止

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

始める前に

1. サーバーは実行されていなければなりません。
2. 処理が完了するまで待ちます。十分な期間にわたって待機するか、またはパフォーマンス・モニター・インフラストラクチャー・カウンターを使用して、キューに格納されているすべての処理が完了したかどうかを判別します。
3. 以下の手順で、新規処理が開始しないようにします。
 - IBM HTTP Server を使用している場合は、`plugin_cfg.xml` ファイルを変更し、HTTP トラフィックのためにサーバーを除去します。その他の HTTP サーバーを使用している場合は、ご使用のサーバーの手順に従いサーバーを除去します。
 - IOP トラフィックの場合は、サーバーのランタイム・ウェイトをゼロに設定します。
 - サービス統合バスを静止します。

プロシージャ

1. デプロイメント・マネージャーの管理コンソールからサーバーを停止するには、以下のようにします。「サーバー」 → 「アプリケーション・サーバー」をクリックします。
2. サーバーを選択します。
3. 「停止」をクリックします。 `install_root/profiles/profile_name/logs/server_name/stopServer.log` に進行状況が記録されます。サーバーが正常に始動すると、「サーバー `server_name` の停止が完了しました。」というメッセージがログに記録されます。

次のタスク

サーバーが停止するまで待ち、停止したら、サーバーを再始動します。

関連概念

管理対象サーバー

管理対象サーバーとは、管理対象ノード上で構成されているサーバーです。デプロイ環境内でアプリケーションを実行するリソースを提供します。

関連タスク

60 ページの『サーバーの開始』

管理対象サーバーでアプリケーションを実行する前に、サーバー・プロセスを開始しておく必要があります。単一サーバー (クラスター・メンバーを含む) を開始できます。

16 ページの『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。

関連情報

 stopServer コマンド

第 4 章 デプロイメント環境の管理

デプロイメント・マネージャー上の管理コンソールによって、そのデプロイメント・マネージャーに対して定義されているデプロイメント環境を管理するための機能が提供されます。また、管理コンソールでデプロイメント環境を作成、削除、インポート、およびエクスポートすることもできます。

始める前に

デプロイメント・マネージャーを開始している必要があり、また管理コンソールにログインしている必要があります。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

「**デプロイメント環境**」管理コンソール・ページは、特定のデプロイメント・マネージャーに対して定義されているデプロイメント環境の管理と定義に関連するすべてのタスクの開始点です。

プロシージャ

1. 管理コンソール・ページにナビゲートします。「サーバー」>「**デプロイメント環境**」をクリックします。
2. デプロイメント環境のコンポーネントを表示するには、「**デプロイメント環境名**」をクリックします。
3. 既存の環境の場合は、管理するデプロイメント環境の隣にあるチェック・ボックスをクリックし、次に実行する機能のボタンをクリックします。

機能	タスク
開始または停止	デプロイメント環境を開始および停止します。
除去	デプロイメント環境からリソースを除去します。このオプションでは、作成されたリソースは削除されません。
エクスポート...	デプロイメント環境をエクスポートします。

4. 新規デプロイメント環境をデプロイメント・マネージャーに追加するには、「**新規作成...**」または「**インポート...**」を使用します。

次のタスク

デプロイメント環境エンティティを管理します。

カスタム・デプロイメント環境のレイアウト構成の概要

この概要では、カスタム・デプロイメント環境の 2 つの主要な構成セクションについて説明します。

67 ページの『このデプロイメント環境で使用するクラスターと単一サーバーの選択』では、デプロイメント環境を構成するクラスターとサーバーを定義します。機能ごとにクラスターを作成するパターン・デプロイメント環境とは異なり、カスタム・デプロイメント環境の機能を実行するために必要なクラスターとサーバーを追加します。

67 ページの『デプロイメント環境構成の指定』では、クラスターとサーバーが提供する機能の構成について説明します。これらの機能は、メッセージング、Common Event Infrastructure、およびアプリケーション・サポートです。

システム内でデプロイメント環境構成が完了したら、環境を生成する前に、構成に戻って変更することができます。システム内でデプロイメント環境構成を生成した後でも、現行構成を確認できます。また、サーバーとクラスターを追加したり、追加の機能を構成したりすることもできます。このデプロイメント環境での管理対象からサーバーとクラスターを除去することもできます。生成済みの機能構成を取り消すことや、デプロイメント環境のサーバーおよびクラスターが使用しているデプロイメント環境定義からサーバーまたはクラスターを除去することはできません。

管理コンソールの動作

カスタム・デプロイメント環境レイアウトには、以下の制限があります。

- ユーザーが構成を完了し、デプロイメント環境を生成すると、関連するコントロールがチェックされ、使用不可になります。つまり、構成を取り消すことができなくなります。
- コンポーネントのコントロールがチェックされず、使用不可にならない場合は、関連付けられているメッセージング・エンジンまたは Common Event Infrastructure (CEI) サーバーを最初に構成する必要があります。機能は、グループ (メッセージング、CEI、および アプリケーション・サポート) 別に構成します。これについては、このトピックで後述します。
- システムに存在する構成は、トポロジー・レイアウト構成よりも優先されます。したがって、カスタム・トポロジーをエクスポートすると、トポロジーに含まれるサーバーの実際の構成が反映されます。

「トポロジー・レイアウト」ページには、カスタム・トポロジーにおいて構成する必要のある 4 つのセクションがあります。

- クラスターおよび単一サーバーの選択
- メッセージング
- Common Event Infrastructure
- コンポーネント

カスタム・トポロジー・レイアウト構成の完了を妨げる注意すべき制限があります。以下の各セクションでは、特定の要件または制限を示します。

このデプロイメント環境で使用するクラスターと単一サーバーの選択

このセクションでは、デプロイメント環境内のクラスターとサーバーを管理し、クラスターとサーバーが提供する機能を定義します。

ドロップダウン・リストには、このデプロイメント環境の一部として構成できるクラスターとサーバーが表示されます。機能構成のコラボレーション単位にクラスターとサーバーを割り当てます。各コラボレーション単位は、デプロイメント環境において 1 つの機能を提供するクラスターおよびサーバーのグループ全体を表します。このデプロイメント環境での管理対象からクラスターまたはサーバーを除去できます。ただし、除去できるクラスターおよびサーバーは、構成内の他のクラスターまたはサーバーで必要としなくなったもののみです。

デプロイメント環境構成の指定

メッセージング

制約事項: 分割されたメッセージング・エンジンはサポートされません。

選択したターゲットのメッセージング宛先ロケーションを構成するには、「メッセージング」タブを使用します。「メッセージング」セクションには、複数のテーブルが存在する場合があります。各テーブルは、1 つのコラボレーション単位を表します。それぞれのユニットのローカル構成のオプションには、ターゲット(クラスター/サーバー)を 1 つのみ選択する必要があります。このユニットのその他のすべてのターゲットでは、リモート宛先が想定されます。リモート宛先構成のターゲットに送信されるメッセージは、そのユニットのローカル・ターゲットにルーティングされます。

メッセージング構成は、SCA システム・バス、CEI システム・バス、および Business Process Choreographer システム・バスに適用されます。

トポロジー構成内のローカル宛先との競合を防止するために、以下のルールが適用されます。

- SCA システム・バス・メッセージング・エンジン構成により、ローカル宛先ロケーションとリモート宛先ロケーションが決定します。SCA アプリケーション、CEI バス構成、および Business Process Choreographer バス構成は、SCA システム・バス構成に従います。
- ユニット内の別のターゲット上でその他のバスのメッセージング・エンジンを探す場合、そのユニット内の別のターゲットでは、リモート宛先ロールが想定されます。CEI と Business Process Choreographer の構成が異なる場合は、テーブル上に視覚的なフィードバックは示されません。情報メッセージには、指定のバスのメッセージング・エンジンが、SCA メッセージング・エンジンが配置されているのと同じターゲットに配置されていないことが示されます。
- リモートまたはローカル宛先が既に構成されており、指定のユニットの現在のバス設定と競合するターゲットは、追加することはできません。この操作を試行すると、エラー・メッセージが生成されます。

Common Event Infrastructure

メッセージングと同様、Common Event Infrastructure (CEI) は、複数のテーブルを持つことができます(各テーブルが 1 つのユニットになります)。各テーブル内には、CEI クラスター/サーバーが 1 つあります。これは、「サーバー」ラジ

オ・ボタンを選択すると、サーバーとして機能します。サーバーとして構成されていないすべてのターゲットでは、宛先ロールが想定されます。対応するターゲットでは、このターゲットで発行される Common Base Event が該当するコラボレーション単位のサーバーに送信されるように、Event Infrastructure エミッター・ファクトリーの Java Naming and Directory Interface (JNDI) 名が構成されています。

アプリケーション・サポート

「アプリケーション・サポート」タブには、特定のデプロイメント・ターゲットに対して構成可能なコンポーネントがすべてリストされます。関連するコラボレーション単位内のコンポーネント機能を構成します。例えば、単位内で構成されている Business Process Choreographer Container から発行される Common Base Event を収集するには、同じ単位内で Business Process Choreographer Event Collector を構成します。各コンポーネント構成には、他のコンポーネント構成に対する制限と依存関係があります。依存関係は、チェックされていない使用不可のコントロールにより表現されます。このようなコントロールを使用可能にするには、依存コントロールを最初に構成する必要があります。

重要: 依存コントロールは、「メッセージング」タブまたは「CEI」タブに表示されます。

コンポーネント間の関係を表 2 に示します。

表 2. コンポーネント、コンポーネント間の相互関係、および考慮事項

コンポーネント	目的	関連コンポーネント	考慮事項
Service Component Architecture (SCA)	SCA アプリケーション・サポート用にデプロイメント・ターゲットを構成します。 SCA システムおよびアプリケーションのバス・メンバーは、対応するメッセージング構成がローカルの場合はローカルに構成されます。それ以外の場合は、対応するメッセージング単位の指定に基づき、リモート宛先ロケーションを使用してリモートに構成されます。	メッセージング	デプロイメント・ターゲットがメッセージング用に構成されていない場合、SCA 構成は使用できません。

表 2. コンポーネント、コンポーネント間の相互関係、および考慮事項 (続き)

コンポーネント	目的	関連コンポーネント	考慮事項
Business Process Choreographer Container	<p>ビジネス・フローと ヒューマン・タスク の両方をサポートす るようデプロイメン ト・ターゲットを 構成します。</p> <p>この構成では、SCA 構成に従って Business Process Choreographer システ ム・バスをセットア ップします。</p>	<p>メッセージング Service Component Architecture Business Process Choreographer Explorer</p>	<p>デプロイメント・タ ーゲットがメッセー ジング用に構成され ていない場合、また は Service Component Architecture サポート 用に構成されていな い場合、Business Process Choreographer 構成は使用できませ ん。</p> <p>1 つのコラボレーシ ョン単位が 1 つの Business Process Choreographer 構成を サポートします。 「アプリケーション・サポ ート」タブ で、Business Process Choreographer 構成に 必要な数の単位を追 加できます。追加で きる単位の数に制限 はありません。</p> <p>Container を管理する 場合は、Business Process Choreographer Explorer を構成する ことを検討してくだ さい。</p>

表2. コンポーネント、コンポーネント間の相互関係、および考慮事項 (続き)

コンポーネント	目的	関連コンポーネント	考慮事項
Business Process Choreographer Explorer	<p>選択されているデプロイメント・ターゲットで Business Process Choreographer Explorer を構成します。</p> <p>Business Process Choreographer Explorer は、同じコラボレーション単位内で構成されている Business Process Choreographer Container を管理する Web アプリケーションです。</p>	Business Process Choreographer Container	<p>同じコラボレーション単位内の Business Process Choreographer Container 構成を選択すると、Business Process Choreographer Explorer 構成が使用可能になります。</p> <p>Web アプリケーション・サポート用にデプロイメント・ターゲットを構成する必要があります。</p> <p>任意のデプロイメント・ターゲットで、必要な数の Business Process Choreographer Explorer を構成できます。構成できる Explorer の数に制限はありません。</p> <p>Container が構成されているコラボレーション単位に選択したデプロイメント・ターゲットを追加し、Explorer の構成コントロールをチェックします。</p>

表2. コンポーネント、コンポーネント間の相互関係、および考慮事項 (続き)

コンポーネント	目的	関連コンポーネント	考慮事項
Business Process Choreographer Event Collector	<p>選択されているデプロイメント・ターゲットで Business Process Choreographer Event Collector を構成します。</p> <p>Business Process Choreographer Event Collector は、同じコラボレーション単位内で構成されている Business Process Choreographer Container から発行される Common Base Event を収集します。監視対象 Container に関する統計情報がデータベースに記録されます。</p>	<p>Business Process Choreographer Container</p> <p>Common Event Infrastructure</p> <p>Business Process Choreographer Observer</p>	<p>最初に、Business Process Choreographer Event Collector に使用するのと同じデプロイメント・ターゲットで Common Event Infrastructure サーバーを構成します。Business Process Choreographer Event Collector は、同じコラボレーション単位内で Business Process Choreographer Container が構成されている場合のみ使用可能です。</p> <p>特定の Business Process Choreographer Container を監視する必要があるかどうかはわからない場合は、この機能を後で構成できます。</p> <p>監視データを調べる場合は、Business Process Choreographer Observer を構成することを検討してください。</p>

表2. コンポーネント、コンポーネント間の相互関係、および考慮事項 (続き)

コンポーネント	目的	関連コンポーネント	考慮事項
Business Process Choreographer Observer	<p>選択されているデプロイメント・ターゲットで Business Process Choreographer Observer を構成します。</p> <p>Business Process Choreographer Observer は、同じコラボレーション単位内で構成されている Business Process Choreographer Container の Business Process Choreographer Event Collector により収集された統計情報を調査するための Web アプリケーションです。</p>	<p>Business Process Choreographer Container</p> <p>Business Process Choreographer Event Collector</p>	<p>同じコラボレーション単位内の Business Process Choreographer Event Collector 構成を選択すると、Business Process Choreographer Observer 構成が使用可能になります。</p> <p>Web アプリケーション・サポート用にデプロイメント・ターゲットを構成する必要があります。</p> <p>任意のデプロイメント・ターゲットで、必要な数の Business Process Choreographer Observer を構成できます。構成できる Observer の数に制限はありません。Event Collector が構成されているコラボレーション単位に選択したデプロイメント・ターゲットを追加し、Observer の構成コントロールをチェックします。</p>

表2. コンポーネント、コンポーネント間の相互関係、および考慮事項 (続き)

コンポーネント	目的	関連コンポーネント	考慮事項
ビジネス・ルール・マネージャー	<p>選択されているデプロイメント・ターゲットでビジネス・ルール・マネージャーを構成します。</p> <p>ビジネス・ルール・マネージャーでは、ビジネス・プロセスの動作を決定するビジネス・ルールを構成できます。</p>	Service Component Architecture	<p>同じデプロイメント・ターゲットでSCA サポートを構成すると、ビジネス・ルール・マネージャー構成コントロールが使用可能になります。</p> <p>ビジネス・ルール・マネージャーは、各デプロイメント環境で1つのみ構成できます。</p> <p>1つのビジネス・ルール・マネージャーでセル全体のビジネス・ルール構成が管理されるため、システムで構成する必要があるビジネス・ルール・マネージャーは1つのみです。</p>

デプロイメント環境の作成

要件を満たすパターンを決定したら、「デプロイメント環境構成」ウィザードを使用して、そのパターンに基づくデプロイメント環境を作成します。

始める前に

デプロイメント・マネージャーの管理コンソールを起動しておく必要があります。このページにナビゲートするには、「サーバー」>「デプロイメント環境」をクリックします。

このタスクに必要なセキュリティー・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

このタスクでは、特定のパターンに従ってデプロイメント環境が作成されます。「デプロイメント環境構成」ウィザードは、適切な順序でステップを指示するので、作業を簡単に行うことができます。

注: ウィザードで作業中にエラーが発生した場合は、「戻る」をクリックして前に戻ることができます。

プロシージャ

1. 「デプロイメント環境」ページで「新規」をクリックして、「デプロイメント環境構成」ウィザードを起動します。

システムは、「デプロイメント環境構成」ウィザードの最初のページを表示します（「新規デプロイメント環境の作成」が既に選択されています）。

2. デプロイメント環境に固有の名前を指定し、リストからランタイムの機能を選択して「次へ」をクリックします。

「ランタイムの機能」は、次のいずれかです。

- **WESB**。これを指定すると、メディエーションをサポートするデプロイメント環境が作成されます。
- **WPS**。これを指定すると、メディエーション、ビジネス・プロセス、ヒューマン・タスク、およびビジネス・ルールをサポートするデプロイメント環境が作成されます。

「ランタイムの機能」のデフォルトは、デプロイメント・マネージャーの機能と同じになります。ウィザードは「デプロイメント環境パターン」ページを表示します。

3. このデプロイメント環境のデプロイメント環境パターンを選択して、「次へ」をクリックします。

このパターンは、計画フェーズでこの環境に選択したパターンと同じでなければなりません。パターンについてよく理解していない場合は、『デプロイメント環境パターン』を参照してください。

注: カスタム・パターンを選択した場合、ユーザーがカスタム・デプロイメント環境構成を定義します。詳しくは、『デプロイメント環境パターン』を参照してください。

ウィザードは「ノードの選択」ページを表示します。

4. このデプロイメント環境に含めるノードを選択して、「次へ」をクリックします。

ノードを含めるには、ノード名の隣にあるチェック・ボックスを選択します。選択されているノードを別のノード名にマップするには、「ノード・マッピング」を使用します。ウィザードは「クラスター」ページを表示します。

5. デプロイメント環境の各機能に使用する各ノードのクラスター・メンバーの数を割り当てます。

デフォルトでは、各機能について、各ノード上に 1 つのクラスター・メンバーが割り当てられます。この数を変更するには、各列の数値を上書きします。機能についてよく理解していない場合は、『デプロイメント環境の機能』を参照してください。ノードが 0 の場合、ノードはその機能に寄与しません。ウィザードは「データベース」ページを表示します。

6. デプロイメント環境のデータベースを構成して、「次へ」をクリックします。

このページでは、このデプロイメント環境に含めるコンポーネントのデータベース情報を定義します。可能な場合、ウィザードはパラメーターのデフォルト情報を提供しますが、ユーザーが環境の計画時に定義した値と一致するようにこれらの値を変更します。

7. オプション: Business Process Choreographer の構成を定義して、「次へ」をクリックします。

このページでは、以下の値を指定します。

- コンテキスト・ルート
- セキュリティー・ロール
- 認証別名
- Human Task Manager メール・セッション (必要な場合)

注: このページは、「ランタイムの機能」が WPS の場合にのみ表示されます。

ウィザードは「ビジネス・ルール・マネージャー」ページを表示します。

8. オプション: ビジネス・ルール・マネージャーのコンテキスト・ルートを指定して、「次へ」をクリックします。

注: このページは、「ランタイムの機能」が WPS の場合にのみ表示されます。

ウィザードは「要約」ページを表示します。

9. 「要約」ページの情報が正しいことを確認したあと、「終了して環境を生成」をクリックして、デプロイメント環境の構成を保存して完了します。構成を完了せずに終了するには、「終了」をクリックします。

結果

構成が完了したら、構成ファイルを検査して、行った変更を表示できます。

次のタスク

変更内容をマスター構成に保存するか、変更内容を破棄します。

関連概念

デプロイメント環境パターン

デプロイメント環境パターンは、デプロイメント環境に含まれるコンポーネントとリソースの制約と要件を指定します。デプロイメント環境パターンは、お客様がデプロイメント環境を最も簡単な方法で作成できるよう支援することを目的とし、大半のビジネス要件を満たせるように設計されています。

デプロイメント・パターン内のデプロイメント環境機能

堅固なデプロイメント環境を設計するためには、各クラスターが特定の IBM 提供のデプロイメント環境パターンまたはカスタム・デプロイメント環境に提供できる機能について理解しておく必要があります。この知識は、ニーズに最も合うデプロイメント環境パターンを正しく判別するのに役立ちます。

データベース仕様書

WebSphere Process Server では、情報の保持、格納、追跡のためにさまざまなデータベース・テーブルを使用します。WebSphere Process Server のコンポーネントの中には、独自のデータベース・テーブルを使用するものもあります。これらのデータベース・テーブルは、プロファイル作成中に作成することも、スクリプトを使用して別途作成することもできます。

12 ページの『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、プロファイル作成時またはサーバーの初期インストール後に選択してインストールできます。

関連情報



Business Process Choreographer の構成

デプロイメント環境の機能の構成

1 つのページから、IBM 提供のパターンに基づいて、ノードをトポロジーに追加したり、デプロイメント環境内のクラスターに機能を割り当てたりできます。デプロイメント環境からノードを削除することもできます。

始める前に

- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- デプロイメント・マネージャーの管理コンソールを表示する必要があります。

「サーバー」 → 「デプロイメント環境」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「追加プロパティ」 → 「デプロイメント・トポロジー」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

重要: デプロイメント環境の機能は、IBM 提供のパターンに基づいてのみ構成できます。カスタム・デプロイメント環境から「デプロイメント・トポロジー」ページにアクセスすることはできません。

デプロイメント環境の要件が変化したときは、既存のデプロイメント環境に対してノードを追加/除去したり、デプロイメント環境内の各種の機能へのリソース割り振りを変更したりします。

- 統合ノードを追加するには、リストからノードを選択して、「ノードの追加」をクリックします。

このアクションにより、画面のノード・リストの最後にそのノードが配置されます。

画面の空の行にノードを追加するには、以下のステップを実行します。

1. リストからノードを選択する

2. 空の行のチェック・ボックスをクリックする
 3. 「**ノードの追加**」をクリックする
- 空のノードまたは存在しないノードを構成に追加するには、「**ノード名:**」フィールドに名前を入力して、「**ノードの追加**」をクリックします。
 - ノードを削除するには、削除するノードの隣にあるチェック・ボックスをクリックして、「**削除**」をクリックします。
 - ノードに機能を割り当てるには、各列のフィールドに数値を入力します。この数値は、指定の機能をサポートするためにシステムが構成するクラスター・メンバーの数を表します。
 - 構成を保管します。「**OK**」または「**適用**」をクリックします。構成するリソースが存在する場合、システムは、構成プロセスを開始します。構成の進行を示すダイアログが表示され、システムは、マスター構成に変更を保存するためのプロンプトを表示します。

注: 「**環境の生成**」をクリックするまでは、構成は完了しません。

結果

ページは、このデプロイメント環境のノードとロールの状況、およびユーザーの変更を反映するように更新されます。

関連資料

 [デプロイメント環境の状況情報](#)

デプロイメント環境内の特定のエンティティの状況を説明します。

 [デプロイメント環境の機能情報 \(Deployment environment function status\)](#)

構成されたデプロイメント環境に関して、最小限必要なエンティティと冗長エンティティの状態を説明します。

デプロイメント環境の状況

デプロイメント環境の状態を示す標識を説明します。トポロジー状況の警告アイコンは、そのデプロイメント環境に警告があることを示します。

ホスト別名の構成

IBM® HTTP Server または選択したサーバーを構成して、管理対象ノードとデプロイメント・マネージャー間の通信を可能にします。

始める前に

デプロイメント・マネージャーおよび関連ノードを作成して構成する必要があります。

このタスクについて

管理対象ノードとデプロイメント・マネージャー間の相互通信が可能になるように、デプロイメント・ターゲット・クラスター内の各ノードに対するホスト別名をデプロイメント・マネージャーに対して可視にする必要があります。ホスト別名に

は、DNS ホスト名およびポート番号が含まれます。この別名を URL アドレスの一部として使用し、アプリケーションがデプロイメント・ターゲットで実行された時点で、アプリケーションにアクセスします。

注: 以下の説明では、2 つのサーバー名 `AppCluster_member1` および `AppCluster_member2` を使用します。説明内ではご使用のサーバー名と置換して理解してください。

プロシージャー

1. 管理コンソールで、「サーバー」を展開して、「アプリケーション・サーバー」をクリックします。
2. 名前 `AppCluster_member1` をクリックします。
3. 「通信」の見出しの下の「ポート」を展開します。
4. `WC_defaulthost` にリストされているポート値をメモします。この値は後に使用する必要があります。
5. ステップ 1 から 4 を繰り返し、ステップ 2 以外では、`AppCluster_member2` を選択します。追加のアプリケーション・クラスター・メンバーごとにこれを繰り返します。

完了すると、クラスター・メンバーとそれらのデフォルト・ホスト用のポート番号のリストが表示されます。

6. 管理コンソールで、「環境」 → 「仮想ホスト」をクリックします。
7. 名前 `default_host` をクリックします。
8. 「追加プロパティ」で、「ホスト別名」をクリックします。
9. クラスター・メンバーに対するホスト名とポート値の正しい組み合わせのエントリが、表示されたリストにない場合、欠落しているエントリをリストに追加します。
10. リストに新規エントリを追加した場合、「保管」および「同期化」をクリックします。

次のタスク

テスト・アプリケーションをインストールして、インストール済み環境を検査します。

デプロイメント環境の認証別名の構成

1 つのページから、すべての認証別名を検討または編集できます。

始める前に

- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- デプロイメント・マネージャーの管理コンソールを表示する必要があります。

「サーバー」 → 「デプロイメント環境」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「追加プロパティ」 → 「認証別名」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

この統合された認証別名リストから、以下を行うことができます。

- 指定のデプロイメント環境のすべての別名の検討
- 「別名」リンクを通じての「認証構成」ページへのアクセス

「リセット」を使用すると、選択されている行が、現在構成されている値にリセットされます。「別名」リンクをクリックすると「認証構成」ページが表示され、ここで変更を行うことができます。

プロシージャ

1. 変更する行を選択します。
2. 以下のいずれかを実行します。

オプション	説明
行の編集	「別名」リンクをクリックします。
行のリセット	「リセット」をクリックします。

行の編集では、「認証構成」ページが表示され、ここで変更を行うことができます。

3. 変更を保存するには、「OK」または「適用」をクリックします。

カスタム・デプロイメント環境の構成

カスタム・デプロイメント環境を構成するには、「カスタム・デプロイメント・トポロジーの詳細」ページを使用します。

始める前に

- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- デプロイメント・マネージャーの管理コンソールを表示する必要があります。

「サーバー」 → 「デプロイメント環境」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「追加プロパティ」 → 「カスタム・デプロイメント・トポロジーの詳細」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

制約事項:

- システムに存在する構成は、デプロイメント環境の構成より優先されます。したがって、カスタム・デプロイメント環境をエクスポートすると、デプロイメント環境に含まれるサーバーの実際の構成が反映されます。

- コンポーネント・ユニットを構成する前にメッセージング・ユニットを構成する必要があります。チェック・ボックスが使用不可の場合は、メッセージング・サポートがまだ構成されていません。

このタスクについて

カスタム・デプロイメント環境の場合は、各機能をどのように構成するのかを必要に応じて決定できます。各機能は、クラスターまたは単一サーバーに対して構成します。カスタム・デプロイメント環境トポロジーの構成には、大きく分けて 3 つの領域があります。

- コンポーネント内通信をサポートするメッセージング。
- イベントおよびモニター機能を統合する **Common Event Infrastructure**。
- ビジネス・インテグレーションのサービス・コンポーネント (ビジネス・プロセスやヒューマン・タスクなど) をサポートするアプリケーション・サポート。

詳しくは、『カスタム・デプロイメント環境のレイアウト構成の概要』を参照してください。

プロシージャ

1. 「このデプロイメント環境で使用するクラスターとサーバーの選択」のリストから、クラスターまたはサーバーを選択します。
2. 「追加」をクリックします。下のテーブルにクラスターまたは単一サーバーが追加されます。
3. このデプロイメント環境に必要なクラスターとサーバーがすべて揃うまで、ステップ 1 および 2 を繰り返します。
4. 「メッセージング」タブをクリックします。
 - a. デプロイメント環境に必要な個別のメッセージング単位の数を決定し、「**新規単位の追加**」をクリックしてその数を追加します。

各単位の「メッセージング単位 x 」という名前が付けられます。この x は、単位の番号です。
 - b. ステップ 2 で作成されたテーブルからクラスターとサーバーを各単位の割り当てます。

単位に追加するクラスターまたはサーバーを選択してから、「**選択したものを単位に追加**」という名前のリストで単位を選択します。
 - c. 各単位のデプロイメント・ターゲットのうち、ローカル・メッセージング・サポートをホストするデプロイメント・ターゲットを決定します。
 - d. 単位内の該当デプロイメント・ターゲットを定義する行の「**ローカル・バス・メンバー**」をクリックし、ローカル・メッセージング・ホストを構成します。

その他のクラスターまたはサーバーはすべて、リモート・メッセージング宛先として自動的に構成されます。
5. 「**Common Events Infrastructure**」タブをクリックします。
 - a. デプロイメント環境に必要な個別の **Common Event Infrastructure** 単位の数を決定し、「**新規単位の追加**」をクリックしてその数を追加します。

各単位に「Common Event Infrastructure 単位 x 」という名前が付けられます。この x は、単位の番号です。

- b. ステップ 2 (80 ページ) で作成されたテーブルからクラスターとサーバーを各単位に割り当てます。

単位に追加するクラスターまたはサーバーを選択してから、「**選択したものを単位に追加**」という名前のリストで単位を選択します。

- c. 各単位のデプロイメント・ターゲットのうち、Common Event Infrastructure サーバーをホストするデプロイメント・ターゲットを決定します。
- d. 単位内の該当デプロイメント・ターゲットを定義する行の「**サーバー**」をクリックし、Common Event Infrastructure サーバー・ホストを構成します。

その他のクラスターまたはサーバーはすべて、リモート Common Event Infrastructure 宛先として自動的に構成されます。

6. 「**アプリケーション・サポート**」タブをクリックします。このタブには、指定のデプロイメント・ターゲットに対して構成可能なコンポーネントがすべて表示されます。

制限: このセクションでコンポーネントを構成する前に、各コンポーネントのメッセージング・ユニットを完全な状態にしておく必要があります。例えば、Service Component Architecture についてチェック・ボックスが使用不可の場合は、関連するメッセージング・ユニットが構成されていません。制限について詳しくは、『デプロイメント環境構成の概要』を参照してください。

- a. デプロイメント環境に必要な個別のアプリケーション・サポート単位の数を決定し、「**新規単位の追加**」をクリックしてその数を追加します。

必要な単位の数は、必要な Business Process Choreographer Container の数によって決まります。Business Process Choreographer Container が不要な場合は、Service Component Architecture アプリケーションには 1 つの単位で十分です。

各単位に「アプリケーション・サポート単位 x 」という名前が付けられます。この x は、単位の番号です。

- b. ステップ 2 (80 ページ) で作成されたテーブルからクラスターとサーバーを各単位に割り当てます。

単位に追加するクラスターまたはサーバーを選択してから、「**選択したものを単位に追加**」という名前のリストで単位を選択します。

- c. ユニット内で、デプロイメント環境の各コンポーネントに属させるクラスターまたはサーバーを選択します。
- d. デプロイメント環境に必要な各ユニットのコンポーネントをすべて構成するまで、ステップ 6b および 6c を繰り返します。

次のタスク

既存のデプロイメント環境を完成させるか編集すると、「カスタム・デプロイメント環境の構成 (Custom Deployment Environment Configuration)」ウィザードが開きます。情報を検討して、必要があれば変更を加えることができます。

関連概念

66 ページの『カスタム・デプロイメント環境のレイアウト構成の概要』この概要では、カスタム・デプロイメント環境の 2 つの主要な構成セクションについて説明します。

デプロイメント環境の据え置かれた構成の構成

データベースとテーブルの作成を据え置く必要がある場合は、「据え置かれた構成」ページを使用してください。このページには、データベースとテーブルを作成するためのスクリプトを見つけて実行する方法の説明が示されています。

始める前に

- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- デプロイメント・マネージャーの管理コンソールを表示する必要があります。

「サーバー」 → 「デプロイメント環境」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「追加プロパティ」 → 「据え置かれた構成」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

「据え置かれた構成」ページには、トポロジーのデータベースを正しく構成するために必要な構成ステップが示されています。通常、このページには、以下の内容が示されます。

- スクリプトの場所
- スクリプトの実行方法の説明

プロシージャ

1. 「据え置かれた構成」ページの指示を実行します。
2. 作業が完了したら、「実行された構成」をクリックします。

次のタスク

据え置かれた構成をいつ誰が最後に実行したかを示すテキスト・ボックスが表示されます。これらの説明は、将来参照するために残されます。

コマンド行を使用したデプロイメント環境の構成

wsadmin インターフェースを使用して、デプロイメント環境を構成することができます。この機能により、スクリプトを使用して、デプロイメント・マネージャー上の複数のデプロイメント環境を無人で構成できます。

始める前に

デプロイメント環境を構成するデプロイメント・マネージャーを使用している必要があります。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

デプロイメント・マネージャー上にデプロイメント環境をインポートまたは作成したあと、`generateDeploymentEnv` コマンドを使用してデプロイメント環境を構成できます。

プロシージャ

1. `wsadmin` 環境に入ります。
2. 構成するトポロジーごとに `generateDeploymentEnv` コマンドを入力します。

以下のコマンドは、ホスト `myDmgr` 上でトポロジー `eastEnvironment` および `westEnvironment` を構成します。

```
wsadmin -connType SOAP -host myDmgr -port 8879
> $AdminTask generateDeploymentEnv -topologyName eastTopology
> $AdminTask generateDeploymentEnv -topologyName westTopology
> $AdminConfig save
```

注: グローバル・セキュリティを有効にして実行している場合、`wsadmin` コマンドの処理後にユーザー ID とパスワードの入力を求めるプロンプトが出されます。

次のタスク


構成されたデプロイメント環境を保存します。コマンド行から、`$AdminConfig save` と入力できます。

関連資料

 [generateDeploymentEnv コマンド](#)

`generateDeploymentEnv` を使用して、デプロイメント・マネージャーでデプロイメント環境を構成します。

関連情報

 [ノード・エージェントの管理](#)

デプロイメント・トポロジーの表示

IBM 提供パターンの構成情報を確認するには、「デプロイメント・トポロジー」ページを使用します。

始める前に

「サーバー」 → 「**デプロイメント環境**」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「**追加プロパティ**」 → 「**デプロイメント・トポロジー**」をクリックします。

このタスクについて

このページを使用して、必要に応じてデプロイメント環境にノードを追加します。各ノードの特定の機能に関するクラスター・メンバーの数を変更することもできます。

- 名前なしノードを追加するには、「名前なしノード」をクリックしてから「追加」をクリックします。このアクションによって、空のノードが追加されます。
- 新規ノードを追加するには、「新規ノード」をクリックし、ノード名を入力し、「追加」をクリックします。
- 既存のノードを追加するには、「既存のノード」をクリックし、リストからノード名を選択し、「追加」をクリックします。
- デプロイメント環境でノードを置換するには、置換するノードの横の「選択」をクリックし、選択したノードを置換する方法に基づいて「名前なしノード」、「新規ノード」、または「既存のノード」のいずれかをクリックし、「選択したものを置換」をクリックします。
- 各機能に関するクラスター・メンバーの数を変更するには、入力フィールドに数を入力します。

機能には以下のものがあります。

- アプリケーション・デプロイメントのターゲット
- メッセージング・インフラストラクチャー
- サポート・インフラストラクチャー

要確認: 各機能には 1 つ以上のクラスター・メンバーが割り当てられている必要があります。

- 更新内容を保持し、引き続き「デプロイメント・トポロジー」ページを表示するには、「適用」をクリックします。更新内容を保持し、前のページに戻るには、「OK」をクリックします。

次のタスク

変更を保存または破棄します。

関連概念

66 ページの『カスタム・デプロイメント環境のレイアウト構成の概要』この概要では、カスタム・デプロイメント環境の 2 つの主要な構成セクションについて説明します。

クラスター、単一サーバー、およびノードの状況

デプロイメント環境内の特定のエンティティの状況を説明します。

表3. デプロイメント環境の状況







アイコン	状況	説明
	不明	システムは、エンティティの状況を判別できません。これは、構成が不完全であることを示す場合があります。
	使用不可	エンティティが構成されていますが、使用不可です。
	停止	エンティティが停止しています。

表 3. デプロイメント環境の状況 (続き)

アイコン	状況	説明
	一部が停止済み	一部のエンティティが停止しています。停止しているエンティティの数は関連エンティティの総数には達していません。
	実行中	エンティティが完全に機能しており、実行中です。
	一部が開始済み	一部のエンティティが開始しています。開始しているエンティティの数は関連エンティティの総数には達していません。

デプロイメント環境の機能の状況

構成されたデプロイメント環境に関して、最小限必要なエンティティと冗長エンティティの状態を説明します。

エンティティの状態の説明

機能の状況は一般に、特定の機能を実行するクラスターに対して使用されます。クラスターは、クラスター・メンバーが冗長部分を構成する冗長エンティティ状態の一例です。デプロイメント環境の状況は、デプロイメント環境が使用可能になるためにはすべての機能が使用可能になる必要がある最小エンティティ状態の一例です。

表 4. エンティティの状態の集合







アイコン	状況	最小エンティティの状態	冗長エンティティの状態
	不明	1 つ以上の最小エンティティの状態が不明であり、状態全体が不明です。	デプロイメント環境で構成が完了しませんでした。
	使用不可	1 つ以上の最小エンティティが使用不可です。	デプロイメント環境のエンティティがすべて使用不可です。
	停止	エンティティがすべて停止しています。	最小エンティティがすべて停止しています。いくつかのエンティティが停止していない場合は、それらのエンティティに問題があることを示します。
	一部が停止済み	一部が停止済みのエンティティが 1 つ以上存在しており、停止済みのエンティティが任意の数だけ存在しています。	一部が停止済みのエンティティまたは停止済みのエンティティが 1 つ以上存在しており、使用不可のエンティティが任意の数だけ存在しています。
	実行中	エンティティがすべて実行中です。	最小エンティティがすべて実行中です。いくつかのエンティティが実行中でない場合は、それらのエンティティに問題があることを示します。


表 4. エンティティの状態の集合 (続き)

アイコン	状況	最小エンティティの状態	冗長エンティティの状態
	一部が開始済み	実行中のエンティティが 1 つ以上存在しており、停止済みのエンティティ、一部が停止済みのエンティティ、または一部が実行中のエンティティが任意の数だけ存在しています。	一部が実行中のエンティティまたは実行中のエンティティが 1 つ以上存在しており、停止済みのエンティティ、一部が停止済みのエンティティ、または使用不可のエンティティが任意の数だけ存在しています。

デプロイメント環境の状況

デプロイメント環境の状態を示す標識を説明します。トポロジー状況の警告アイコンは、そのデプロイメント環境に警告があることを示します。

表 5. トポロジー・インスタンスの使用可能状況 (使用可能性が低い順から高い順)

状況アイコン	警告アイコン	状態	説明
	なし	不明	システムは、デプロイメント環境の現在の状態を判別できません。
		不完全	デプロイメント環境で欠落している要素はありませんが、不完全な部分があります。追加の詳細情報は、警告メッセージに含まれています。
		未構成	構成が認識され、完了していますが、生成されていません。
		据え置かれた構成	デプロイメント環境は生成していますが、据え置かれた構成が完了していません。
		使用不可	デプロイメント環境が完了していますが、使用不可の機能が 1 つ以上あります。
		一部が停止済み	デプロイメント環境が使用可能ですが、停止済みまたは一部が停止済みの機能が 1 つ以上あります。
		停止	すべての機能が停止済みです。
		一部が実行中	デプロイメント環境が使用可能ですが、一部が実行中の機能が 1 つ以上あります。
		実行中	デプロイメント環境が使用可能であり、機能がすべて実行中です。

デプロイメント環境の編集

デプロイメント環境の要件の発展とともに、デプロイメント環境を編集してリソースを追加することができます。

始める前に

- デプロイメント環境が存在している必要があります。

- デプロイメント・マネージャーの管理コンソールを起動しておく必要があります。このページにナビゲートするには、「サーバー」>「デプロイメント環境」をクリックします。
- デプロイメント環境からノードを削除する前に、削除するノードを完全に停止する必要があります。

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

特定のデプロイメント環境を構成するリソースを変更するには、デプロイメント環境を編集します。以下のことが可能です。

- サーバーおよびクラスターの追加または除去。
- 特定の機能に参加するノードの変更。
- データ・ソースの構成の変更。
- 認証別名の変更。
- データベースまたはテーブルの構成方法に関する情報の取得 (操作を遅延している場合)。

プロシージャ

1. 編集するデプロイメント環境を選択します。デプロイメント環境名をクリックします。システムは、「デプロイメント環境構成」ページを表示して、以下の内容をリストします。
 - **デプロイメント環境**
 - **デプロイメント環境パターン**
 - **説明**
 - **デプロイメント環境の状況**
 - **デプロイメント環境の機能**
 - **構成ページへのリンク**
2. 変更するデプロイメント環境の一部を選択します。

変更が完了するまで、各リンクを選択します。

「追加プロパティ」の下で、以下のリンクのいずれかを選択します。

- デプロイメント・トポロジー - IBM 提供パターンをベースにしたデプロイメント環境の場合。
- カスタム・デプロイメント・トポロジーの詳細 - カスタム・デプロイメント環境の場合。
- 据え置かれた構成

「関連項目」の下で、以下のリンクのいずれかを選択します。

- データ・ソース
- 認証別名

3. 構成を完了します。必要なアクションを実行するオプションを選択します。

注: 「環境の生成」をクリックするまでは、構成は完了しません。

アクション	結果
「OK」または「適用」をクリックします。	両方のオプションとも、構成を保存します。 「適用」をクリックしたときは現在のページのままですが、「OK」をクリックしたときは「デプロイメント環境」ページに戻ります。
「環境の生成」をクリックします。	構成を保存して、構成プロセスを開始します。 注: デプロイメント環境が最低限の制約を満たしていない場合、またはデプロイメント環境が不完全の場合は、このオプションを選択できません。

次のタスク

デプロイメント環境を管理します。

関連タスク

76 ページの『デプロイメント環境の機能の構成』

1 つのページから、IBM 提供のパターンに基づいて、ノードをトポロジーに追加したり、デプロイメント環境内のクラスターに機能を割り当てたりできます。デプロイメント環境からノードを削除することもできます。

79 ページの『カスタム・デプロイメント環境の構成』

カスタム・デプロイメント環境を構成するには、「カスタム・デプロイメント・トポロジーの詳細」ページを使用します。

90 ページの『デプロイメント環境のデータ・ソースの構成』

「データベース構成プロバイダー (Database Configuration Provider)」ページを使用して、最初にビジネス・インテグレーションのデータ・ソースを構成します。

78 ページの『デプロイメント環境の認証別名の構成』

1 つのページから、すべての認証別名を検討または編集できます。

82 ページの『デプロイメント環境の据え置かれた構成の構成』

データベースとテーブルの作成を据え置く必要がある場合は、「据え置かれた構成」ページを使用してください。このページには、データベースとテーブルを作成するためのスクリプトを見つけて実行する方法の説明が示されています。

データ・ソース構成の編集

デプロイメント環境のすべてのデータ・ソースを扱う中央の場所として「ビジネス・インテグレーション・データ・ソース (Business Integration Data Source)」ページを使用することにより、複数の編集を行うことができます。

始める前に

- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- デプロイメント・マネージャーの管理コンソールを表示する必要があります。

「サーバー」 → 「デプロイメント環境」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「追加プロパティ」 → 「データ・ソース」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

「データ・ソース」ページは、編集可能なデプロイメント環境のデータ・ソースをすべて集めたページです。「データ・ソース」ページでは、新しいデータ・ソースを追加することはできません。数値テキスト・フィールドは、各データ・ソースのコンポーネントおよびデータ・ソース・プロバイダーによって異なる場合があります。

重要: 競合する編集を行った場合 (例えば、別のデータ・ソースで使用されているスキーマ名を使用するなど)、システムは、警告メッセージを表示します。その場合でも変更を保存することはできますが、競合を解決するまでメッセージは存続します。

プロシージャ

1. 「データ・ソース」ページで、編集するデータ・ソースを含むコンポーネントを選択します。
2. 必要に応じて変更を加えます。
3. 「適用」または「OK」をクリックして、行った変更をすべて保存します。

データベース・プロバイダーの編集

データベース・プロバイダーを変更するには、「データベース・プロバイダー構成」ページを使用します。

始める前に

- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- デプロイメント・マネージャーの管理コンソールを表示する必要があります。

「サーバー」 → 「デプロイメント環境」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「追加プロパティ」 → 「データ・ソース」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

「データベース・プロバイダー構成」ページには、編集可能なテキスト・ボックスが表示されます。「コンポーネント固有のプロパティ」などの一部のセクションでは、データベース・プロバイダーによってテキスト・ボックスの数が異なる場合があります。

プロシージャ

1. 「データ・ソースの構成」ページで、編集するデータ・ソース・プロバイダーを含むコンポーネントを選択します。
2. 「プロバイダーの編集 (Edit Provider)」をクリックして、「データベース・プロバイダー構成」ページを開きます。
3. 必要な変更を加えます。
4. 「適用」または「OK」をクリックして、変更を保存します。

関連概念

データベース仕様書

WebSphere Process Server では、情報の保持、格納、追跡のためにさまざまなデータベース・テーブルを使用します。WebSphere Process Server のコンポーネントの中には、独自のデータベース・テーブルを使用するものもあります。これらのデータベース・テーブルは、プロファイル作成中に作成することも、スクリプトを使用して別途作成することもできます。

共通データベースの仕様

共通データベースの仕様には、サポートされるデータベース・タイプ、スクリプトとその場所、プロファイル作成の構成アクション、インストール・パラメーター、作成されるテーブルとユーザー ID の特権のタイプに関する情報が含まれません。

デプロイメント環境のデータ・ソースの構成

「データベース構成プロバイダー (Database Configuration Provider)」ページを使用して、最初にビジネス・インテグレーションのデータ・ソースを構成します。

始める前に

- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- デプロイメント・マネージャーの管理コンソールを表示する必要があります。

「サーバー」 → 「デプロイメント環境」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「追加プロパティ」 → 「データ・ソース」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

「データ・ソース」ページは、デプロイメント環境に必要なすべてのデータ・ソースを集めたページです。このページでは、使用可能なデータベースに基づいてデータ・ソースを構成します。

データ・ソースを必要とするコンポーネントは、ユーザーが選択した「データベース・プロバイダー」に基づいて、すべての必須フィールドを決定します。これらの必須フィールドにはデータを入力する必要があります。コンポーネントは、残りの

フィールドにデフォルト値を設定します。ユーザーは、デフォルト値を保持することもできますし、必要に応じてそれを変更することもできます。通常、コンポーネントは、「有効範囲」の値を決定します。

ビジネス・インテグレーションのデータ・ソースは、1 度だけ構成できます。データ・ソースを構成して保存すると、一部のテキスト・ボックスが使用不可になり、値を変更できなくなります。ページ内のそれ以外のテキスト・ボックスは、すべて編集できます。

プロシージャ

1. 「**ビジネス・インテグレーション・データ・ソース (Business Integration Data Source)**」 ページで、構成するデータ・ソースを選択します。
2. デフォルトでは表示されない追加のデータ・ソース・フィールドを編集する必要がある場合は「**編集**」をクリックします。
3. 情報を入力します。サポートされているデータベース・タイプのリストについては、『**データベース仕様書**』を参照してください。
4. 「**適用**」または「**OK**」をクリックして、変更を保存します。

関連概念

データベース仕様書

WebSphere Process Server では、情報の保持、格納、追跡のためにさまざまなデータベース・テーブルを使用します。WebSphere Process Server のコンポーネントの中には、独自のデータベース・テーブルを使用するものもあります。これらのデータベース・テーブルは、プロファイル作成中に作成することも、スクリプトを使用して別途作成することもできます。

共通データベースの仕様

共通データベースの仕様には、サポートされるデータベース・タイプ、スクリプトとその場所、プロファイル作成の構成アクション、インストール・パラメーター、作成されるテーブルとユーザー ID の特権のタイプに関する情報が含まれません。

デプロイメント環境内のデータ・ソースの編集

データ・ソースのプロパティを編集するには、「データ・ソース」ページを使用します。

始める前に

- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- デプロイメント・マネージャーの管理コンソールを表示する必要があります。

「サーバー」 → 「**デプロイメント環境**」をクリックして、このページまで移動します。デプロイメント環境を選択してから、「**追加プロパティ**」 → 「**データ・ソース**」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

「データ・ソース」ページは、デプロイメント環境のすべてのデータ・ソースを集めたページです。「データ・ソース」ページでは、新しいデータ・ソースを追加することはできません。データ・ソースの情報を編集するには、「データ・ソース」名をクリックするか、コンポーネントを選択します。一部のテキスト・ボックスは使用不可になっており、それらの値を変更することはできません。

重要: 競合する編集を行った場合は (例えば、別のデータ・ソースで使用されているスキーマ名を使用するなど)、警告メッセージが表示されます。その場合でも変更を保存することはできますが、競合を解決するまでメッセージは存続します。

プロシージャ

1. 「ビジネス・インテグレーション・データ・ソース (Business Integration Data Source)」ページで、編集するデータ・ソースを含むコンポーネントを選択します。
2. 「編集」をクリックすると、「データ・ソース」ページが開きます。
3. 情報を編集します。
4. 「適用」または「OK」をクリックして、変更を保存します。

デプロイメント・マネージャーの停止と再始動

デプロイメント・マネージャーで構成を変更した後は、変更を有効にするために、デプロイメント・マネージャーを停止してから再始動する必要があります。

始める前に

デプロイメント・マネージャーを開始している必要があります、また管理コンソールにログインしている必要があります。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

プロシージャ

1. デプロイメント・マネージャーの停止方法を選択します。

方法	アクション
管理コンソールの使用	<ol style="list-style-type: none">1. 「システム管理」 → 「デプロイメント・マネージャー」をクリックします。2. 「停止」をクリックします。

方法	アクション
コマンド行の使用	<ol style="list-style-type: none"> 1. デプロイメント・マネージャーの <code>profile_root/bin</code> ディレクトリーにナビゲートします。 2. ご使用のオペレーティング・システムの正しい形式の <code>stopManager</code> を入力します。 注: グローバル・セキュリティーが有効になっている場合は、ユーザー ID とパスワードの入力を求めるプロンプトが出されます。

2. デプロイメント・マネージャーの停止確認を待機します。
3. デプロイメント・マネージャーの `profile_root/bin` ディレクトリーにナビゲートします。
4. ご使用のオペレーティング・システムの正しい形式の `startManager` コマンドを入力します。

注: グローバル・セキュリティーが有効になっている場合は、ユーザー ID とパスワードの入力を求めるプロンプトが出されます。

次のタスク

アプリケーション・デプロイメント・ターゲットのクラスターが開始可能なことを確認します。

クラスター・メンバーの停止および再始動

構成を変更した場合、クラスター・メンバーを停止して、再始動します。

始める前に

1. そのメンバーに新しい処理が割り振られないようにする必要があります。
 - IBM HTTP Server を使用している場合は、`plugin_cfg.xml` ファイルを変更して、HTTP トラフィック用のクラスター・メンバーを除去します。別の HTTP サーバーを使用している場合は、ご使用の HTTP サーバーの説明に従ってクラスター・メンバーを除去します。
 - IIOP トラフィックの場合、クラスター・メンバーのランタイム・ウェイトをゼロに設定します。
 - サービス統合バスを静止します。
2. 現在クラスター・メンバーに転送されている処理は、完了する必要があります。ある期間待機するか、またはパフォーマンス・モニター・インフラストラクチャーのカウンターを使用して、キューに入れられたすべての処理をクラスターが完了するタイミングを判断します。

このタスクについて

構成変更を有効にするために、いくつかのサーバー・プロセスの停止および再始動が必要な場合があります。デプロイメント・マネージャー、クラスター・メンバ

一、およびノード・エージェントの停止および再始動も含まれます。以下の手順によって、クラスター・メンバーを停止および再始動します。

注: すべてのコマンド・ファイル (管理コンソールの使用に対する代替) は `install_root/bin` サブディレクトリーにあります。

プロシージャ

1. 管理コンソールから「**サーバー**」を選択し、次に「**アプリケーション・サーバー**」を選択します。
2. 停止するサーバーまたはクラスター・メンバーを選択します。
3. 「**停止**」をクリックします。
4. サーバーまたはクラスター・メンバーが停止するまで待機します。
5. 再始動するサーバーまたはクラスター・メンバーを選択し、「**開始**」をクリックします。
6. サーバーまたはクラスター・メンバーが開始されるまで待機します。

次のタスク

注: 代替方法として、オペレーティング・システムでコマンド行から `stopServer` および `startServer` コマンドをそれぞれ使用するか、または管理コンソールのクラスター・パネルで「**サーバー**」>「**クラスター**」> `cluster_name` > `cluster_member_name` を選択することによって、クラスター・メンバーを停止および再始動することができます。

デプロイメント環境の開始と停止

IBM 提供のパターンに基づいて、管理コンソールから直接デプロイメント環境を開始または停止することができます。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

デプロイメント環境が存在している必要があります。

このタスクについて

デプロイメント環境を開始または停止する場合は必ず、以下のステップを使用します。

制限: カスタム・デプロイメント環境には、この手順を使用することはできません。クラスターを個別に開始および停止する必要があります。

プロシージャ

1. 開始するデプロイメント環境を制御するデプロイメント・マネージャーの管理コンソールにログインします。
2. デプロイメント環境の管理コンソール・ページにナビゲートします。「**サーバー**」 → 「**デプロイメント環境**」をクリックします。

3. 開始および停止するデプロイメント環境の名前の隣にあるチェック・ボックスをクリックします。
4. 選択したデプロイメント環境を開始または停止します。

アクション	結果
「開始」をクリックします。	デプロイメント・マネージャーにより、デプロイメント環境を構成するクラスターが始動されます。
「停止」をクリックします。	デプロイメント・マネージャーにより、デプロイメント環境を構成するクラスターが停止されます。

注: このプロセスは、ご使用のデプロイメント環境のサイズによっては数分かかる場合があります。

結果

画面が最新表示され、デプロイメント環境の状況が示されます。正常な場合、『デプロイメント環境の状況』に説明されているように、デプロイメント環境にデプロイメント環境の状態が表示されます。

関連概念

デプロイメント環境

デプロイメント環境は、構成済みクラスター、サーバー、およびミドルウェアの集合であり、これらが連携して、Service Component Architecture (SCA) との対話をホストする環境を実現します。例えば、デプロイメント環境には、メッセージ宛先、ビジネス・イベントのプロセッサ、および管理プログラムのホストが含まれていることがあります。

関連タスク

16 ページの『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。

関連情報

Administration Thin Client の使用

デプロイメント環境定義のエクスポート

デプロイメント環境定義をエクスポートすると、各デプロイメント・マネージャー上で実行する必要のある構成を最小限に抑えることができるので、デプロイメント環境をより迅速に実装できるようになります。エクスポートしたデプロイメント環境は、デプロイメント環境のテンプレートとして、他のデプロイメント・マネージャー上で使用されます。また、これにより同じデプロイメント環境構成を大規模に複製できます。

始める前に

- 1 つ以上のデプロイメント環境をデプロイメント・マネージャー上で完全に定義します。
- デプロイメント・マネージャーの管理コンソールを起動しておく必要があります。このページにナビゲートするには、「サーバー」>「デプロイメント環境」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

同じ設計に基づくデプロイメント環境を多数実装する場合は、それらのデプロイメント環境定義をエクスポートすることにより、他のデプロイメント・マネージャー上でデプロイメント環境を容易に構成できます。

プロシージャ

1. 「デプロイメント環境」ページで、エクスポートするデプロイメント環境定義の隣にあるチェック・ボックスをクリックします。
2. 「エクスポート」をクリックします。システムの応答は、以下のいずれを選択したかに応じて異なります。

単一のデプロイメント環境

エクスポートするファイルの名前を入力するように要求されます。デフォルトの名前は、`deployment_environment_name.xml` です。ここでは、ファイルの絶対パスを指定します。

複数のデプロイメント環境

デプロイメント環境定義を含む、エクスポートする圧縮ファイルを配置する出力ディレクトリーを入力するように要求されます。デフォルトでは、システムは、圧縮ファイルの名前を `first_env_name.zip` のように設定します。

3. ファイル・パスを調べて、システムがファイルを作成したことを確認します。

次のタスク

以上で、エクスポートしたファイルを他のデプロイメント・マネージャーに移動して、そのデプロイメント・マネージャーにインポートすることができます。

注: 圧縮ファイルを直接インポートすることはできません。含まれているデプロイメント環境定義を抽出してから、それをターゲット・ファイル・システムに取り込む必要があります。

コマンド行を使用したデプロイメント環境定義のエクスポート

`wsadmin` コマンド・インターフェースを使用して、デプロイメント環境定義をエクスポートすることができます。この機能により、スクリプトを使用して、複数のデプロイメント環境定義をデプロイメント・マネージャーからエクスポートできます。これにより、作業構成を別のデプロイメント・マネージャーに複製することが可能です。

始める前に

- デプロイメント環境定義のエクスポート元のデプロイメント・マネージャーを起動しておく必要があります。

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

コマンド行を使用してデプロイメント環境定義をエクスポートするのは、以下の場合です。

- 複数のデプロイメント環境定義をエクスポートするときに、管理コンソールを使用したくない場合。
- 単一のデプロイメント環境定義を、複数のデプロイメント環境のテンプレートとして使用する場合。
- エクスポートするデプロイメント環境定義の数が非常に多いため、タスクでスクリプトを使用しなければならないと判断した場合。この場合は、適切なコマンドをスクリプトに配置して、そのファイルを実行します。

プロシージャ

1. デプロイメント・マネージャーにログインします。このとき使用するユーザー ID には、管理コマンドを入力するための適切な権限が必要です。
2. コマンド行を開きます。 デプロイメント・マネージャーを稼働しているオペレーティング・システムで適切な方法を使用します。
3. `wsadmin` 環境に入ります。
4. `exportDeploymentEnvDef` コマンドを入力して、デプロイメント・マネージャーから出力ファイルにデプロイメント環境定義をエクスポートします。 ファイル名の形式は、`depEnvName.xml` になります。

この例では、グローバル・セキュリティが有効なホスト `myDmgr` のデプロイメント環境 `myDepEnv` がエクスポートされます。

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgradmin -password dmgrpass  
> $Admintask exportDeploymentEnvDef -exportDirectory c:/dmgr01/DeploymentEnvs  
-topologyName myDepEnv
```

注: グローバル・セキュリティを無効にする場合は、ユーザー ID とパスワードを指定する必要はありません。

関連資料

 `exportDeploymentEnvDef` コマンド

`exportDeploymentEnvDef` を使用して、デプロイメント・マネージャーからトポロジをエクスポートします。

デプロイメント環境定義のインポート

既存のデプロイメント環境定義を別のデプロイメント・マネージャーからインポートして、それを新しいデプロイメント環境の基礎として使用することができます。

始める前に

- デプロイメント・マネージャーの管理コンソールを起動しておく必要があります。このページにナビゲートするには、「サーバー」>「デプロイメント環境」をクリックします。
- 別のデプロイメント・マネージャーからエクスポートされたデプロイメント環境定義のコピーが必要です。デプロイメント環境定義をインポートするデプロイメント・マネージャーからファイルにアクセスできる必要があります。
- デプロイメント環境定義をインポートするデプロイメント・マネージャーは、デプロイメント環境定義で定義されているすべての機能をサポートしている必要があります。例えば、WebSphere ESB デプロイメント・マネージャーで作成されたデプロイメント環境定義を WebSphere Process Server デプロイメント・マネージャーにインポートできますが、この逆はできません。

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

重要: 1 つの圧縮ファイルから複数のデプロイメント環境定義を同時にインポートすることはできません。圧縮ファイルから定義を抽出したら、それらの定義を一度に 1 つずつインポートする必要があります。

このタスクについて

要件に厳密に一致する既存のデプロイメント環境定義をインポートすると、デプロイメント環境の構成に費やす時間を最小限に抑えることができます。要件に厳密に一致する環境が存在する場合は、そのデプロイメント環境定義をエクスポートして、構成するデプロイメント・マネージャーにインポートします。

重要: ウィザードのどのパネルでも、「構成」をクリックすると、現行値を使用してデプロイメント環境が構成されます。

プロシージャ

1. 「デプロイメント環境」ページで、「インポート」をクリックします。システムは、「デプロイメント環境構成」ウィザードを起動します（「外部のトポロジ一定義のロード」が選択されています）。
2. インポートするデプロイメント環境定義を含むファイルの場所を指定します。入力フィールドに絶対パスを入力するか、「参照」をクリックして、ファイル・ダイアログを開いてファイルを選択します。

注:

3. 「次へ」をクリックします。構成が読み込まれ、「デプロイメント環境のインポート」ウィザードが開始されます。

すべてのノード名が現在の統合ノードに一致しない場合、ウィザードは「ノードの選択」ページを表示します。すべてのノードが一致する場合、ウィザードは「データベース」ページを表示します。

4. オプション: 使用可能なノードのリストから、デプロイメント環境に含めるノードを選択して「次へ」をクリックします。

ノードを含めるには、ノード名の隣にあるチェック・ボックスを選択します。選択されているノードを別のノード名にマップするには、「ノード・マッピング」を使用します。

重要: 選択したノードが、インポートするデプロイメント環境定義によって課せられた制約を満たしていない場合は、「次へ」が使用可能になりません。例えば、「Mandatory_Node」という名前のノードと、任意の名前のその他の 3 つのノードを含めるというデプロイメント環境の要件が存在する場合は、「Mandatory_Node」ノードを構成して、その他の 3 つのノードを選択するまで、作業を続行することはできません。

5. デプロイメント環境のデータベースを構成して、「次へ」をクリックします。

このページでは、このデプロイメント環境に含めるコンポーネントのデータベース情報を定義します。可能な場合、ウィザードはパラメーターのデフォルト情報を提供しますが、ユーザーが環境の計画時に定義した値と一致するようにこれらの値を変更します。

6. オプション: Business Process Choreographer の構成を定義して、「次へ」をクリックします。

このページでは、以下の値を指定します。

- コンテキスト・ルート
- セキュリティー・ロール
- 認証別名
- Human Task Manager メール・セッション (必要な場合)

注: このページは、「ランタイムの機能」が **WPS** の場合にのみ表示されます。

ウィザードは「ビジネス・ルール・マネージャー」ページを表示します。

7. オプション: ビジネス・ルール・マネージャーのコンテキスト・ルートを指定して、「次へ」をクリックします。

注: このページは、「ランタイムの機能」が **WPS** の場合にのみ表示されます。

ウィザードは「要約」ページを表示します。

8. 「要約」ページの情報が正しいことを確認したあと、「終了して環境を生成」をクリックして、デプロイメント環境の構成を保存して完了します。構成を完了せずに終了するには、「終了」をクリックします。

結果

構成が完了したら、構成ファイルを検査して、行った変更を表示できます。

次のタスク

変更内容をマスター構成に保存するか、変更内容を破棄します。

関連概念

デプロイメント環境パターン

デプロイメント環境パターンは、デプロイメント環境に含まれるコンポーネントとリソースの制約と要件を指定します。デプロイメント環境パターンは、お客様がデプロイメント環境を最も簡単な方法で作成できるよう支援することを目的とし、大半のビジネス要件を満たせるように設計されています。

デプロイメント・パターン内のデプロイメント環境機能

堅固なデプロイメント環境を設計するためには、各クラスターが特定の IBM 提供のデプロイメント環境パターンまたはカスタム・デプロイメント環境に提供できる機能について理解しておく必要があります。この知識は、ニーズに最も合うデプロイメント環境パターンを正しく判別するのに役立ちます。

データベース仕様書

WebSphere Process Server では、情報の保持、格納、追跡のためにさまざまなデータベース・テーブルを使用します。WebSphere Process Server のコンポーネントの中には、独自のデータベース・テーブルを使用するものもあります。これらのデータベース・テーブルは、プロファイル作成中に作成することも、スクリプトを使用して別途作成することもできます。

12 ページの『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、プロファイル作成時またはサーバーの初期インストール後に選択してインストールできます。

関連タスク

95 ページの『デプロイメント環境定義のエクスポート』

デプロイメント環境定義をエクスポートすると、各デプロイメント・マネージャー上で実行する必要のある構成を最小限に抑えることができるので、デプロイメント環境をより迅速に実装できるようになります。エクスポートしたデプロイメント環境は、デプロイメント環境のテンプレートとして、他のデプロイメント・マネージャー上で使用されます。また、これにより同じデプロイメント環境構成を大規模に複製できます。

関連情報

Business Process Choreographer の構成

コマンド行を使用したデプロイメント環境定義のインポート

wsadmin コマンド・インターフェースを使用して、デプロイメント環境定義をインポートすることができます。この機能により、スクリプトを使用して複数のデプロイメント環境定義をデプロイメント・マネージャーにインポートすることができます。このようにして、作業中の構成を他のデプロイメント・マネージャーに複製できます。

始める前に

- デプロイメント環境定義をエクスポートしておく必要があります。
- デプロイメント環境定義のインポート先のデプロイメント・マネージャーを起動しておく必要があります。
- インポートするデプロイメント環境定義と同名のデプロイメント環境がこのデプロイメント・マネージャーに存在していることを確認します。
- デプロイメント環境定義をインポートするデプロイメント・マネージャーは、デプロイメント環境定義で定義されているすべての機能をサポートしている必要があります。例えば、WebSphere ESB デプロイメント・マネージャーで作成されたデプロイメント環境を WebSphere Process Server デプロイメント環境にインポートできますが、この逆はできません。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

以下の場合に、コマンド行を使用してデプロイメント環境定義をインポートします。

- 複数のデプロイメント環境定義をインポートするときに、管理コンソールを使用したくない場合。
- 単一のデプロイメント環境定義を、複数のデプロイメント環境のテンプレートとして使用する場合。
- インポートするデプロイメント環境定義の数が非常に多いため、タスクでスクリプトを使用しなければならないと判断した場合。この場合は、適切なコマンドをスクリプトに配置して、そのファイルを実行します。

プロシージャ

1. コマンド行を開きます。 デプロイメント・マネージャーを稼働しているオペレーティング・システムで適切な方法を使用します。

注: グローバル・セキュリティが有効になっている場合は、ユーザー ID とパスワードの入力を求めるプロンプトが出されます。

2. システムにインポートするデプロイメント環境定義ファイルをコピーします。
3. wsadmin 環境に入ります。
4. importDeploymentEnvDef コマンドを入力して、デプロイメント・マネージャーにコピーしたファイルからデプロイメント環境定義をインポートします。 デプロイメント環境をインポートするときは、その名前を変更できます。

この例では、グローバル・セキュリティが有効になっているデプロイメント・マネージャー myDmgr にデプロイメント環境 myDepEnv がインポートされ、その名前が eastDepEnv に変更されます。

```
wsadmin -connType SOAP -host myDmgr -port 8879 -user dmgrAdmin -password -dmgrPass
> $AdminTask importDeploymentEnvDef
  -filePath c:/dmgr01/importedEnvironments/myDepEnv.xml
  -topologyName eastDepEnv
```

注: グローバル・セキュリティーを無効にする場合は、ユーザー ID とパスワードを指定する必要はありません。

次のタスク

インポートされたトポロジーを検証します。

関連資料



importDeploymentEnvDef コマンド

importDeploymentEnvDef を使用して、デプロイメント・マネージャーにトポロジーをインポートします。

関連情報



ノード・エージェントの管理

デプロイメント環境の除去

特定のデプロイメント環境でグループとして表現されるリソースに関して、それを管理する機能が不要になったら、そのデプロイメント環境定義をデプロイメント・マネージャーから除去します。

始める前に

- デプロイメント・マネージャーの管理コンソールを起動しておく必要があります。このページにナビゲートするには、「サーバー」>「デプロイメント環境」をクリックします。
- デプロイメント環境は、このデプロイメント・マネージャー上に存在します。
- リカバリーの目的でデプロイメント環境定義をエクスポートすることを検討します。

このタスクに必要なセキュリティー・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

デプロイメント環境を除去すると、デプロイメント環境を構成する構成要素を表す管理エンティティーが除去されます。デプロイメント環境を削除しても、デプロイメント環境を構成する各種のサーバー、ノード、およびクラスターの構成が除去または変更されることはありません。デプロイメント環境の削除は、あるデプロイメント・マネージャーから別のデプロイメント・マネージャーにデプロイメント環境を移動する最後のフェーズになる場合があります。

プロシージャー

1. 「デプロイメント環境」ページで、除去するデプロイメント環境を選択します。デプロイメント環境名の隣にあるチェック・ボックスをクリックします。
2. 「削除」をクリックします。

結果

システムは、画面からデプロイメント環境を削除します。

この変更内容をマスター構成に保存するには「**保存**」、マスター構成が更新されないようにするには「**破棄**」をクリックします。

関連タスク

95 ページの『デプロイメント環境定義のエクスポート』

デプロイメント環境定義をエクスポートすると、各デプロイメント・マネージャー上で実行する必要がある構成を最小限に抑えることができるので、デプロイメント環境をより迅速に実装できるようになります。エクスポートしたデプロイメント環境は、デプロイメント環境のテンプレートとして、他のデプロイメント・マネージャー上で使用されます。また、これにより同じデプロイメント環境構成を大規模に複製できます。

第 5 章 Business Process Choreographer の管理

Business Process Choreographer を管理する方法については、WebSphere Process Server for Multiplatforms バージョン 6.1 のインフォメーション・センターにアクセスし、『**WebSphere Process Server の管理**』>『**Business Process Choreographer の管理**』のトピックを参照してください。この情報は、「*Business Process Choreographer*」の PDF にも記載されています。

第 6 章 Common Event Infrastructure の構成および管理

Common Event Infrastructure を構成し、管理する方法については、WebSphere Process Server for Multiplatforms バージョン 6.1 のインフォメーション・センターにアクセスし、『**WebSphere Process Server の管理**』>『**Common Event Infrastructure の構成**』、および『**WebSphere Process Server の管理**』>『**Common Event Infrastructure の管理**』のトピックを参照してください。この情報は、「*Common Event Infrastructure*」の PDF にも記載されています。

第 7 章 サービス・コンポーネントの管理

このセクションのトピックを使用して、サービス・コンポーネントを管理します。

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 Multiplatforms バージョン 6.1 のインフォメーション・センターで『WebSphere Process Server の管理』>『サービス・コンポーネントの管理』のトピックを参照するか、「Business Process Choreographer」の PDF を参照してください。

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

一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、関連セット・プロパティを表示します。

始める前に

WebSphere Integration Developer 内で関連セットを定義し、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。

このタスクについて

関連セット・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、関連セット・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、関連セット・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関連セットは 1 つのみです。複数の関連セットは使用できません。

プロシージャ

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」の下に関連セット・プロパティを表示します。

次のタスク

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。

始める前に

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。

このタスクについて

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

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関連セット・プロパティの値が必要です。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。

プロシージャー

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」の下に関連セット・プロパティと表示状態を表示します。

次のタスク

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ターの管理

ビジネス・ルールおよびセレクトターは、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ター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。スタンドアロン・サーバー用または Network Deployment 用のビジネス・ルール動的リポジトリをインストールできます。

ビジネス・ルールまたはセレクトター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ターを変更するたびに、システム・ログ、またはビジネス・ルールおよびセレクトター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・ルールとセレクトターの監査ログの構成

ビジネス・ルールとセレクトターの新規作成、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。

始める前に

このタスクを実行するには、管理コンソールを使用する必要があります。

このタスクに必要なセキュリティー・ロール: セキュリティーと役割ベースの許可が有効になっている場合、このタスクを実行するには、コンフィギュレーターとしてログインする必要があります。

このタスクについて

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・ルールとセレクター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。

ビジネス・ルールとセレクターの監査ログを構成するには、以下のステップを実行します。

プロシージャ

1. 「サーバー」 > 「アプリケーション・サーバー」 > *servername* > 「ビジネス・ルール」 > 「ビジネス・ルールとセレクター監査」をクリックして、「ビジネス・ルールとセレクター監査」ページにナビゲートします。
2. 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"> 1. 「ランタイム」タブを選択します。 2. 必要な変更を入力します。 3. オプション: 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。 4. 変更を適用してから前のページに戻るには、「OK」をクリックします。変更の適用後もこのページを表示したままにするには、「適用」をクリックします。
遅延	<ol style="list-style-type: none"> 1. 「構成」タブを選択します。 2. 必要な変更を入力します。 3. 変更を適用してから前のページに戻るには、「OK」をクリックします。変更の適用後もこのページを表示したままにするには、「適用」をクリックします。 4. 変更内容を有効にするには、サーバーを再始動します。

結果

指定した属性が監査ログに適用されます。

注: WebSphere Application Server 6.1 でセキュリティーが有効に設定されている場合、サーバー・ユーザー ID の指定方法のために、ビジネス・ルールとセレクター監査の構成の変更が必要なことがあります。サーバー・ユーザー ID にデフォルト値が使用されている場合は、ビジネス・ルールまたはセレクターのインストール後にビジネス・ルールまたはセレクターが含まれているアプリケーションが開始されると、ビジネス・ルールまたはセレクターを使用する監査可能アクションが実行されると、自動生成されたサーバー ID 値がユーザーの監査レコードに記録されます。インストール後のアプリケーションの開始、管理クライアント、または管理コンソールでのインポートまたはエクスポートによってビジネス・ルールまたはセレクターの成果物に変更されると、監査可能なアクションが発生します。生成される値

は、他の監査レコードで使用されているユーザー ID の形式と一致していないことがあり、より一貫性のある値が必要な場合があります。

サーバー ID を指定するには、「リポジトリに保管されたサーバー ID」を使用するオプションを選択します。これにより、ユーザー・リポジトリのユーザー ID が、サーバー・プロセスに関連付けられます。ビジネス・ルールまたはセレクターの成果物がリポジトリにインストールされた後で、ビジネス・ルールまたはセレクターを含むアプリケーションが開始され、ビジネス・ルールまたはセレクターを使用する監査可能アクションが実行されると、監査レコードでこの ID が使用されます。

管理クライアント (ビジネス・ルール・マネージャーなど) による変更が行われる監査アクション、およびビジネス・ルール・グループのエクスポートやインポートなどの管理アクションには、サーバー ID 値は影響しません。このようなアクションの場合、監査レコードでは認証済みユーザーが使用されます。

サーバー・ユーザー ID の変更についての詳細は、『アプリケーションと環境の保護』の各トピックと、WebSphere Application Server の WebSphere Application Server Network Deployment バージョン 6 セキュリティーに関する資料を参照してください。

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。

始める前に

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。

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、このタスクを実行するには、コンフィギュレーターとしてログインする必要があります。

このタスクについて

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。以下のタスクは、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・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* > 「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」を選択します。

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を構成するには、以下のステップを実行します。

注: 以下のタスクでは、サーバー `server1` を構成します。サーバーの名前が `server1` でない場合は、以下の `server1` を、そのサーバーの名前で置き換えます。ステップ 3 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。

プロシージャ

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	<code>set mbean [\$AdminControl queryNames *:*,name=CustomizationAuditMBean,process=server1]</code>
監査ロギング構成の変更	<code>set auditconfig [\$AdminConfig list AuditLog]</code>

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。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。

管理コンソールを開き、「サーバー」>「アプリケーション・サーバー」>
servername > 「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* > 「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」を選択して変更を保存します。あるいは、\$AdminConfig save と入力します。

注: WebSphere Application Server 6.1 でセキュリティーが有効に設定されている場合、サーバー・ユーザー ID の指定方法のために、ビジネス・ルールとセレクター監査の構成の変更が必要なことがあります。サーバー・ユーザー ID にデフォルト値が使用されている場合は、ビジネス・ルールまたはセレクターのインストール後にビジネス・ルールまたはセレクターが含まれているアプリケーションが開始され、ビジネス・ルールまたはセレクターを使用する監査アクションが実行されると、自動生成されたサーバー ID 値がユーザーの監査レコードに記録されます。インストール後のアプリケーションの開始、管理クライアント、または管理コンソールでのインポートまたはエクスポートによってビジネス・ルールまたはセレクターの成果物に変更されると、監査可能なアクションが発生します。生成される値は、他の監査レコードで使用されているユーザー ID の形式と一致していないことがあり、より一貫性のある値が必要な場合があります。

サーバー ID を指定するには、「リポジトリに保管されたサーバー ID」を使用するオプションを選択します。これにより、ユーザー・リポジトリのユーザー ID が、サーバー・プロセスに関連付けられます。ビジネス・ルールまたはセレクターの成果物がリポジトリにインストールされた後で、ビジネス・ルールまたはセレクターを含むアプリケーションが開始され、ビジネス・ルールまたはセレクターを使用する監査可能アクションが実行されると、監査レコードでこの ID が使用されます。

管理クライアント (ビジネス・ルール・マネージャーなど) による変更が行われる監査アクション、およびビジネス・ルール・グループのエクスポートやインポートな

どの管理アクションには、サーバー ID 値は影響しません。このようなアクションの場合、監査レコードでは認証済みユーザーが使用されます。

サーバー・ユーザー ID の変更についての詳細は、『アプリケーションと環境の保護』の各トピックと、WebSphere Application Server の WebSphere Application Server Network Deployment バージョン 6 セキュリティーに関する資料を参照してください。

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

ここでは、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ときに考慮すべき情報をいくつか説明します。

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除去方法について説明します。

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。未使用の成果物を使用するアプリケーションをアンインストールした後で、データベースからその成果物を手動で削除します。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。この理由として、ビジネス・ルールとセクターに含まれているビジネス・ロジックが、アプリケーションのインストール時に更新されている可能性があること、またアプリケーションが除去された時点でこの重要なビジネス・データを削除せずにおく必要があることがあります。

始める前に

除去されるビジネス・ルールまたはセクターを使用するアプリケーションのコピーをすべてアンインストールします。ビジネス・ルールまたはセクターの成果物を削除する前に、それらのバックアップを作成してください。これを行うには、管理コンソールまたは `wsadmin` コマンドを使用して、ビジネス・ルールまたはセクターの成果物をサーバーからエクスポートします。

このタスクについて

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。

不要なビジネス・ルールおよびセクターの成果物をリポジトリから除去するには、以下のステップを実行します。

プロシージャ

1. 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のタイム・スタンプを保持するテーブル

CUSTPROPERTIES

ビジネス・ルール・グループ、ルール・セット、またはデシジョン・テーブルに対するユーザー定義のカスタム・プロパティ、およびシステム・プロパティを保持するテーブル。

2. 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、**ARTIFACTNS**、**ARTIFACTNAME**、および **ARTIFACTTYPE** です。
 - c. BYTESTORE テーブルから、ステップ 2a で検索された行を削除します。
 - d. ステップ 2b 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、1 行、または複数行に対応する場合があります。
 - e. BYTESTOREOVERFLOW テーブルから、ステップ 2d で検索された行を削除します。
 - f. ステップ 2b 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 CUSTPROPERTIES テーブルの行を探します。
 - g. CUSTPROPERTIES テーブルから、ステップ 2f で検索された行を削除します。
 - h. **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。

結果

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・ルールは、WebSphere Integration Developer の Eclipse ベースのビジネス・ルール・エディターを使用して開発およびデプロイします。ビジネス・ルールの値は、WebSphere Process Server のオプションである Web ベースのビジネス・ルール・マネージャーを使用して管理および変更します。これらのツールについて詳しくは、それぞれ 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・センターの該当トピックを参照してください。

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。

始める前に

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。

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

サーバーに存在するビジネス・ルール・グループを確認するには、以下のステップを実行します。

プロシージャ

1. 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」をクリックします。
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。

結果

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・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ます。これにより、開発環境にインポートできるファイルが作成され、開発の成果物を実際の実動システムの成果物と同期させることができます。

始める前に

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』の説明に従って、ビジネス・ルール・コンポーネントを事前に表示しておく必要があります。「サーバー」>「アプリケーション・サーバー」> *servername* >「ビジネス・ルール」>「ビジネス・ルール」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。セキュリティが有効でないときは、ユーザー ID を指定して管理コンソールにログインする必要があります。

このタスクについて

管理コンソールを使用してビジネス・ルールをエクスポートするには、以下のステップを実行します。

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。

プロシージャ

- 1 つ以上のビジネス・ルール・グループの隣にあるチェック・ボックスを選択して、「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、「ビジネス・ルールのエクスポート」ページです。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. 各ファイル名をクリックして、ファイルをシステムにダウンロードします。ファイルを保存するためのプロンプトがシステムから出されたら、「OK」をクリックします。

注: ファイルの保存を選択した場合は、ファイルのダウンロード時にファイル名を変更できます。

3. 「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。

結果

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。

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。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。

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:

インストールしたビジネス・ルールを、アプリケーションを再インストールせずに更新するには、ビジネス・ルールをインポートします。

始める前に

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。

ビジネス・ルールをインポートする前に、以下の条件が満たされていることを確認します (条件が満たされていない場合は、インポートが失敗します)。

- ファイルの拡張子が zip である。
- サーバーからビジネス・ルールをエクスポートして圧縮ファイルが作成されている。
- ビジネス・ルール・グループを使用するアプリケーションがセル内のサーバーに既にインストールされている。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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。

管理コンソールを使用してビジネス・ルールをインポートするには、以下のステップを実行します。

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。

プロシージャ

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。「サーバー」>「アプリケーション・サーバー」> *servername* > 「ビジネス・ルール」> 「ビジネス・ルール」をクリックします。
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」ページ上でファイルへのパスを指定します。

次のタスク

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、プロファイル作成時またはサーバーの初期インストール後に選択してインストールできます。

ビジネス・ルールは、WebSphere Integration Developer 内で、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設計および開発します。ビジネス・ルールは、WebSphere Business Modeler 内で作成することもできます。ただし、Modeler では、ビジネス・ルール・タスクの作成のみがサポートされます (ビジネス・ルール・タスクは、Modeler からエクスポートしたときにルール・セットになります)。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。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・ルールの構築および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・ルールのコピーを検索する
- 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

以下の図は、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。

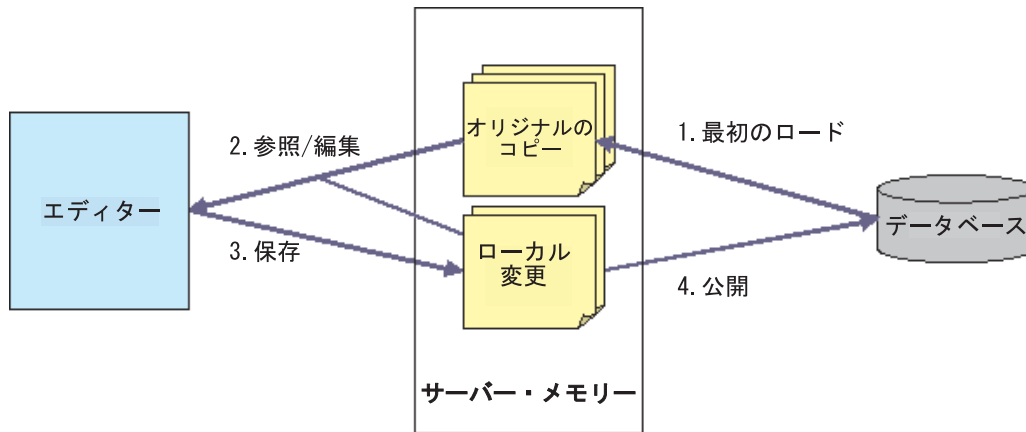


図5. ビジネス・ルール・マネージャーのイベント・シーケンス

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、リポジトリからビジネス・ルール・グループにアクセスして、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. ビジネス・ルール・グループとルール・ロジックが編集できるようになります。
3. ルール・セット、デシジョン・テーブル、およびビジネス・ルール・グループへの変更をコピーとしてサーバー・メモリーに保存できます。
4. ローカル・コピーを再びデータ・ソースに公開します。あるいは、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、すべてのサーバーが単一のビジネス・ルール・リポジトリを使用および共有します。ビジネス・ルール・リポジトリにアクセスすると、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。

管理コンソールを使用したビジネス・ルール・マネージャーのインストール:

ビジネス・ルール・マネージャーをエンタープライズ・アプリケーションとして WebSphere Process Server にインストールすると、実行時にビジネス・ルールを管理できます。WebSphere Process Server 6.1 以降では、プロファイル管理ツールの「ビジネス・ルール・マネージャーの構成」ページのチェック・ボックスを選択して WebSphere Process Server プロファイルを作成する時点で、同時にビジネス・ルール・マネージャーをインストールできます。ビジネス・ルール・マネージャーは、ほかの 3 つの方法を使用してインストールすることもできます。具体的には、管理コンソールの構成ページを使用する方法、オペレーティング・システムの JACL コマンドを使用する方法、Admin Tasks コマンドを使用する方法がありますが、Admin Tasks コマンドを使用する方法は、WebSphere Process Server 6.1 以降でのみ可能です。詳しくは、各インストール方法の個々のトピックを参照してください。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

管理コンソールを使用してビジネス・ルール・マネージャーをインストールするには、以下のステップを実行します。

プロシージャ

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、「サーバー」 > 「アプリケーション・サーバー」または「クラスター」をクリックします。
3. サーバーまたはクラスター・ターゲットの名前を選択します。

4. 「構成」タブを含むページで、「**Business Integration**」の下にある「ビジネス・ルール」を展開して、「**ビジネス・ルール・マネージャーの構成**」をクリックします。
5. 「**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。

注: ビジネス・ルール・マネージャーが既にインストールされている場合は、このチェック・ボックスがチェックされた状態になりますが、このページからビジネス・ルール・マネージャーをアンインストールすることはできないのでぼかし表示されます。ただし、アプリケーションのリストを表示し、このリストから手動でアンインストールできます。

6. 「**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` 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。
7. 「**OK**」をクリックします。
8. 構成を保管します。

ナビゲーション・ペインで、「**アプリケーション**」>「**エンタープライズ・アプリケーション**」をクリックして、「**ビジネス・ルール・マネージャーの始動 (Start Business Rules Manager)**」を選択します。

JACL コマンドを使用したビジネス・ルール・マネージャーのインストール:

Windows、Linux[®]、i5/OS[®] では、ビジネス・ルール・マネージャーをインストールするときに、管理コンソールの代わりとして JACL コマンドを使用できます。JACL コマンドを使用できるのは、WebSphere Process Server をインストールしてプロファイルを作成したときに、ビジネス・ルール・マネージャーをインストールしていなかった場合です。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、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のロールに割り当てられているユーザー ID を使用して認証される必要があります。

このタスクについて

JACL コマンドを使用してビジネス・ルール・マネージャーをインストールするには、以下のステップを実行します。

プロシージャ

1. WebSphere Process Server が始動していることを確認します。
2. オペレーティング・システムのシェル環境またはコマンド・プロンプトを開き、`install_root/bin` ディレクトリー (Windows の場合は `install_root\bin` ディレクトリー) に移動します。
3. オペレーティング・システム独自のインストール・コマンドを以下のように実行します。

- Windows で実行するコマンド: `wsadmin.bat -f installBRManager.jacl [-s servername -n nodename | -cl clustername] -ce cellname -r rootname`
- Linux で実行するコマンド: `wsadmin.sh -f ./installBRManager.jacl [-s servername -n nodename | -cl clustername] -ce cellname -r rootname`
- i5/OS で実行するコマンド: `wsadmin -f ./installBRManager.jacl [-s servername -n nodename | -cl clustername] -ce cellname -r rootname`

ビジネス・ルール・マネージャーをインストールして、それを複数のターゲットにマップするには、ご使用のオペレーティング・システムで以下の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 -m "{{target1} {target2} ... {targetn}}" -ce cellname -r rootname`
- Linux で実行するコマンド: `wsadmin.sh -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}" -ce cellname -r rootname`
- i5/OS で実行するコマンド: `wsadmin -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}" -ce cellname -r rootname`
- z/OS® で実行するコマンド: `wsadmin.sh -f installBRManager.jacl -m "{{target1} {target2} ... {targetn}}" -ce cellname -r rootname`

注: パラメーター「-m」（「複数 (multiple)」を意味する) を指定すると、ビジネス・ルール・マネージャーを複数のターゲットに一括にインストールしてマップできます。複数のターゲットを二重引用符で囲みます。

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。欠落している場合、*servername* のデフォルト値は「server1」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「-ce 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」です。

target_i ビジネス・ルール・マネージャーをインストールしてマップするターゲット (*i* は 1、2、...、*n*)。

ターゲットは (-s *servername* および -n *nodename*) または -cl *clustername* のいずれかです。

重要: 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* および -n *nodename*)
- または -cl *clustername*
- または -m "{{target1} {target2} ... {targetn}}"

他の引数ペアはオプションです。

例: ビジネス・ルール・マネージャー・アプリケーションを以下のターゲットにマップするとします。

- クラスタ 「BofACluster」
- Web サーバー 「RedirectorServer」 およびノード 「AIXNode01」
- アプリケーション・サーバー 「LinuxServer」 およびノード 「LinuxNode02」

コンテキスト・ルートは「bofa/brm」です。

以下のようにコマンドを実行します。

```
install_root/bin/wsadmin -f installBRManager.jacl -m "{{-cl BofACluster} {-n AIXNode01 -s RedirectorServer} {-s LinuxServer -n LinuxNode02}}" -r bofa/brm
```

AdminTask コマンドを使用したビジネス・ルール・マネージャーのインストール:

WebSphere Process Server 6.1 以降を使用して、Admin Task コマンドを使用してビジネス・ルール・マネージャーをインストールできます。 WebSphere Process Server をインストールしてプロファイルを作成するときにビジネス・ルール・マネージャーをインストールしなかった場合は、管理コンソールまたは JACL コマンドを使用するのと同様に Admin Task コマンドを使用します。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

管理タスク・コマンドを使用してビジネス・ルール・マネージャーをインストールするには、以下のステップを実行します。

プロシージャ

1. WebSphere Process Server が始動していることを確認します。
2. コマンド・ウィンドウで WebSphere Process Server のホーム・ディレクトリーに移動して、ディレクトリー `/bin` に切り替えます。
3. `wsadmin` コマンドを実行して、`wsadmin` モードに入ります。
4. 以下のいずれかのコマンドを入力して、ビジネス・ルール・マネージャーをインストールします。

```
wsadmin> $AdminTask configBusinessRulesManager {-serverName <serverName>
-nodeName <nodeName> -contextRoot <contextRoot>}
```

または

```
wsadmin> $AdminTask configBusinessRulesManager {-clusterName
<clusterName> -contextRoot <contextRoot>}
```

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。

nodeName

インストール・ノードの名前。

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。

contextRoot

アプリケーションを起動するために使用するコンテキスト・ルート。デフォルト値は `/br` です。

5. `wsadmin> $AdminConfig save` を実行して、構成を保存します。

ヒント: `$AdminTask help configBusinessRulesManager` を実行すると、そのパラメーターの詳細を知ることができます。

例: コンテキスト・ルート「`br`」を使用してサーバー「`cvuServer`」およびノード「`cvuNode01`」にビジネス・ルール・マネージャーをインストールするには、以下のコマンドを入力します。

```
wsadmin> $AdminTask configBusinessRulesManager {-serverName cvuServer
-nodeName cvuNode01 -contextRoot br}
```

次に構成を保存するには、以下のように入力します。

```
wsadmin> $AdminConfig save
```

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。

このタスクについて

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。サーバーのセキュリティを設定するには、以下のステップを実行します。

プロシージャ

1. ユーザー ID の作成時に各 ID にロールを割り当てることにより、ユーザー ID にグローバル・セキュリティを設定します。各ユーザー ID を作成して、それをロール `BusinessRuleUser` にマップします。

ロールを設定するには、ビジネス・ルール・マネージャー・アプリケーションにナビゲートし（「アプリケーション」>「エンタープライズ・アプリケーション」）、ビジネス・ルール・マネージャー・アプリケーションを選択します。「ユーザー/グループ・マッピングへのセキュリティ・ロール」を選択し、`BusinessRuleUser` ロールを更新します。

注: グローバル・セキュリティが有効に設定された ND 環境において、ポート $908n$ (n は正の整数) 上でビジネス・ルール・マネージャーを実行する場合は、ポート「 $944(n+3)$ とホスト値 "*"」を構成する必要があります。該当するポートがない場合は、ビジネス・ルール・マネージャーを起動する前に手動で構成します。

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。

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。

このタスクについて

ビジネス・ルール・マネージャー用の Web ブラウザーが正しく構成されていることを確認するには、以下のステップを実行します。

プロシージャ

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。

始める前に

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。

このタスクについて

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。

`http://hostname:port/br`

「hostname」は現行ホスト・システムの名前 (または IP アドレス)、「port」はアプリケーションがインストールされたアプリケーション・サーバーのポートです。

例えばサーバーが 1 つのみのスタンドアロン環境では、このリンクは次のようになります。

`http://hostname:9080/br`

注: グローバル・セキュリティーが有効になっている場合、前述のリンクが自動的にセキュア・リンクに切り替わります。例えばサーバーが 1 つのみのスタンドアロン環境では、`https://hostname:9443/br` です。

グローバル・セキュリティーが使用可能でない場合は、「ビジネス・ルール・グループ」ページが開きます。グローバル・セキュリティーがサーバー上で使用可能な場合は、「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。

プロシージャ

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。

結果

ビジネス・ルール・マネージャーの最初のページが開きます。ナビゲーション・ペインには、既存のビジネス・ルール・グループがリストされています。

次のタスク

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。

「ビジネス・ルール・グループ」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、「ビジネス・ルール・グループ」ページが表示されます。このページでは、ビジネス・ルール・グループとその定義済みの操作をすべて確認できます。

「ビジネス・ルール・グループ」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

ツールバーのコンポーネントは以下のとおりです。

ようこそ

現在ログオンしているユーザーの名前が表示されます。

ユーザー識別

「ウェルカム・ユーザー名 (Welcome User Name)」の後に現在のユーザーの名前が表示されます。

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、確認を求めるダイアログ・ボックスが表示されます。

検索

「ビジネス・ルール・グループの検索」ページを開きます。指定した一連の操作対象ビジネス・ルール・グループを迅速に検索したり、その絞り込みを行うことができます。

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、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ビジネス・ルール・グループ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。

重要: バージョン番号が現行モデルよりも大きいビジネス・ルール成果物を取得する場合は、シェルと呼ばれるビジネス・ルール成果物が、ナビゲーション・ペインでフラット・テキスト項目になります。その結果、シェルをさらに公開できなくなります。現在の WebSphere Process Server を最新版 (シェルのバージョンと同じかまたはそれより高いバージョン) に更新してください。

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、ビジネス・ルール・グループおよびルール・スケジュールの変更をデータベースに公開したり、ビジネス・ルール・グループまたはルール・スケジュールをデータベース上にあった元のコピーに戻したりすることができます。

ビジネス・ルール・グループ

「ビジネス・ルール・グループ」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。ビジネス・ルール・グループはナビゲーション・ツリーにリストされます。表示名の隣にある正符号 (+) または負符号 (-) をクリックすると、ビジネス・ルール・グループを展開または縮小できます。これにより、関連するルールをすべて表示することが可能です。左のペインのナビゲーション・ツリーでビジネス・ルール・グループを選択すると、右のペインにすべての子「ルール・スケジュール」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

コンテンツ領域は、右側にあるペインであり、これが主に表示および編集を行う領域です。コンテンツ領域には「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、「ビジネス・ルール・グループ」ページ、「ルール・スケジュール」ページ、「ルール・セット」ページ、「デシジョン・テーブル」ページ、「公開および復帰」ページ、および「ビジネス・ルール・グループの検索」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 (ビジネス・ルール・グループおよび「ルール・スケジュール」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

リソース表示名およびビジネス・ルールのタイプを以下の形式で指定します。

`Ruleset112 - Ruleset`

例: `calculateDiscount-Rule Schedule, CalculateDiscountRS - Rule Set`

機能ボタン

個々のページの目的に応じて各種のアクションを使用可能にします。すべての機能ボタンがページで使用できるわけではありません。ボタンによっては、コンテンツ領域の別のセクションに表示されるものもあります。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 6. 機能ボタン

ボタン名	機能
プロパティの追加	「ビジネス・ルール・グループ」ページでビジネス・ルール・グループにプロパティを追加するか、または「ビジネス・ルール・グループの検索」ページで検索照会を作成します。
戻る	前のページに戻ります。
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。デシジョン・テーブルまたはルール・セットを新規に作成するには、既存のデシジョン・テーブルまたはルール・セットをコピーしてから、値を変更する必要があります。
編集	ビジネス・ルール・グループ、ルール・スケジュール、ルール・セット、またはデシジョン・テーブルが編集できるようになります。
公開	ビジネス・ルール・グループまたはルール・スケジュールをリポジトリに公開します。
復帰	ローカル環境に保存されたルールに対する変更をすべて取り消し、サーバー・メモリー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。
保管	変更内容を検証してからローカル・コピーに保存し、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。
検索	「ビジネス・ルール・グループの検索」ページの検索照会を開始し、照会と一致するビジネス・ルール・グループを同じ「ビジネス・ルール・グループの検索」ページに検索結果として戻します。
ソート	ビジネス・ルール・グループのプロパティを、プロパティ名を基準にして、アルファベットの昇順でソートします。

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

「一般情報」セクションには以下の情報があります。

注: WebSphere Process Server 6.1 以降では、「ビジネス・ルール・グループ」ページに「一般情報」セクションが含まれます。「ビジネス・ルール・グループの検索」ページと「公開および復帰」ページには、このセクションは含まれません。

表示名 Websphere Process Server 6.1 以降のビジネス・ルール・グループ、ルール・セット、またはデシジョン・テーブルの表示名を示します。「ビジネス・ルール・グループ」、「ルール・セット」、および「デシジョン・テーブル」ページにおいて、表示名は、表示モードでは読み取り専用ですが、編集モードでは変更することが可能です。表示名には、どのストリング値でも使用でき、特殊文字を含めることもできます。同じタイプのビジネス・ルール成果物の表示名を固有にする必要はありません。ただし、ユース・ケースでは、ビジネス・ルール成果物の名前を固有にする必要があります。

表示名を設定した場合、その表示名は、名前の値が使用されるすべての場所(ナビゲーション・ペインを含む)で、名前の値の代わりに使用されます。また、成果物が詳しく表示されるときにも、その表示名が使用されます。ビジネス・ルール成果物の表示名を設定しない場合は、その名前の値が代わりに使用されます。「名前との同期」チェック・ボックスを選択すると、ターゲットのビジネス・ルール・グループ、ルール・セット、またはデシジョン・テーブルの対応する名前の値に表示名が同期されます。編集ページで行った変更を保存すると、ビジネス・ルール・マネージャーのすべてのページで新しい名前が有効になります。

最後に公開

ビジネス・ルール・グループ、ルール・スケジュール、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・スケジュール、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または公開されたかを表示します。

説明 ビジネス・ルール・グループ、ルール・スケジュール、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。これらのページの編集モードで、説明を編集できます。

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ります。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、「ビジネス・ルール・グループ」ページ、「ルール・スケジュール」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。

「ビジネス・ルール・グループ」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す。

ビジネス・ルール・リソース

ルール・スケジュール、ルール・セット、およびデシジョン・テーブルの表示名をリストします。

説明 リソースの要旨または名前を指定します。

アクション

対応するビジネス・ルール・リソース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、ビジネス・ルール・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、ローカル環境に保存されているビジネス・ルール・グループおよびルール・スケジュールに対する変更内容をリポジトリに公開するために使用します。また、ビジネス・ルール・リソースをローカル環境に保管する前にサーバー・メモリーにあった元のコピーにビジネス・ルール・グループおよびルール・スケジュールを戻すためにも使用します。

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。

「変更されたビジネス・ルール・リソース」セクション

このセクションでは、公開または復帰が可能なビジネス・ルール・グループおよびルール・スケジュールのリストを表示し、以下の情報を提供します。

ビジネス・ルール・リソース (Business Rule Resources)

変更したビジネス・ルール・グループおよびルール・スケジュールの名前を表示します。公開する準備が整っているリソースの隣には、公開を選択または選択解除するためのチェック・ボックスがあります。

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。リソースには、対応する「アクション」フィールドに「復帰」ボタンがあります。

「ビジネス・ルール・グループ」ページ:

「ビジネス・ルール・グループ」ページには、ビジネス・ルール・グループに関連するビジネス・ルール・リソースがすべて表示されます。

このページでは、ビジネス・ルール・グループの情報や関連するビジネス・ルール・リソースの情報を参照するだけでなく、編集ページを開いて情報を変更することもできます (ビジネス・ルール・グループのカスタム・プロパティの追加、削除、変更も含まれます)。

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。

「プロパティ」セクション

このセクションでは、ビジネス・ルール・グループのカスタム定義プロパティを表示します。

制約事項: ビジネス・ルール・グループにカスタム・プロパティがない場合、またはそのカスタム・プロパティのリストが空の場合、表示モードでは「プロパティ」セクションは表示されません。また、ビジネス・ルール・グループが WebSphere Process Server 6.1 より前のバージョンに属する場合、「ビジネス・ルール・グループ」ページには、ビジネス・ルール・グループの「プロパティ」セクションと「編集」ボタンは表示されません。

名前 プロパティの名前を指定します。この名前は固有である必要があります。また、空にすることはできません。各プロパティは、ビジネス・ルール・グループで 1 回のみ定義できます。

値 プロパティの値を指定します。プロパティごとに定義された値が必要です。空ストリングまたは長さゼロにすることはできますが、ヌルであってはなりません。プロパティをヌルに設定することは、プロパティを削除することと同じです。

「ビジネス・ルール・リソース」セクション

このセクションでは、ビジネス・ルール・グループに関連するルール・スケジュール、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

ビジネス・ルール・グループに関連するルール・スケジュール、ルール・セット、およびデシジョン・テーブルの表示名をリストします。

説明 ビジネス・ルール・グループ、ルール・スケジュール、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。

「ルール・スケジュール」ページ:

「ルール・スケジュール」ページには、スケジュール済みルール・ロジック項目内のビジネス・ルール・グループ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・スケジュール」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。

「スケジュール済みルール・ロジック」セクション

このセクションでは、ルールを構築するための基本単位である有効なビジネス・ルールのリストを表示します。ここでは、スケジュール済みルール・ロジック項目の追加やソートなどの操作が可能です。

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¹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。

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「終了日なし」のいずれかを選択できます。

注: 「終了日なし」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

スケジュール済みルール・ロジック項目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、他のスケジュール済みルール・ロジック項目のいずれとも日付が一致しない場合に選択されます。

「使用可能なルール・ロジック」セクション

このセクションでは、特定のビジネス・ルール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、関連する記述やアクションと共に表示します。

ルール・ロジック

ルール・セットまたはデシジョン・テーブルの名前を指定します。

説明 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。

アクション

ルールを容易に編集またはコピーするオプションを提供します。

「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た既存のルール・インスタンスの表示/編集、選択したテンプレートからの新規ルール・インスタンスの作

成、ルール実行順序の指定、ルールまたはルール・セットの名前変更、ルール・セットの表示名またはルール・セットのルールの表示/編集、ルール・セット、ルールの説明、テンプレート・パラメーターの説明の表示/編集、ルール・セットの作業用コピーとしての保存、ルールの削除が可能です。

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。

「ルール」セクション

このセクションでは、関連するルールのリストを表示し、以下の情報を提供します。

名前 ルールの名前を提供します。このフィールドは、編集モードでのみ表示されます。

表示名 ルールの表示名を指定します。表示名を指定しなかった場合は、「名前」の値が設定されます。表示モードでは読み取り専用ですが、編集モードでは編集が可能です。表示名には、どのストリング値でも使用でき、特殊文字を含めることもできます。固有である必要はありません。「アクション」フィールドの「名前の同期」チェック・ボックスを選択すると、表示名が、対応する名前と同期されます。

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。

説明 ルール・セット内の各ルールの情報を指定します。表示モードでは読み取り専用ですが、編集モードでは編集が可能です。

アクション

関連するボタンをクリックすることにより、ルールの再配列、ルールの削除、および表示名と名前の同期が可能です。このアクションは、編集モードでのみ使用可能です。

「テンプレート」セクション

このセクションでは、既存のテンプレートを使用して、編集モードで新規ルールを容易に作成できます。また、ルールに対して以下の情報を指定するため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および変更するためテキスト域を提供します。

表示名 ルールの表示名を入力するためテキスト域を提供します。表示名を指定しない場合は、「名前」の値が設定されます。表示名には、どのストリング値でも使用でき、特殊文字を含めることもできます。固有である必要はありません。「名前の同期」チェック・ボックスを選択すると、表示名が、ルール名の値と同期されます。編集ページで行った変更を保存すると、ビジネス・ルール・マネージャーのすべてのページで新しい名前が有効になります。

注: 「名前の同期」チェック・ボックスを選択すると、ルールの表示名が使用不可になり、変更できなくなります。

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。

説明 各テンプレート・パラメーターのルールを指定します。これが表示されるのは、ルール・セットが編集モードであり、マウスをターゲット・テンプレート・パラメーターの上に移動したときに限られます。読み取り専用です。

アクション

テンプレートへのルールの追加、テンプレートからのルールの削除、またはルール名の値と表示名との同期が可能です。

「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・スケジュール」 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た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設定アクションのルールの変更、デシジョン・テーブルと初期設定のルールの表示名およびその説明の表示と編集、デシジョン・テーブルの保存 (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。

「初期設定のルール」 セクション

このセクションでは、デシジョン・テーブルの初期設定のルールが示されます。ビジネス・ルール定義が WebSphere Integration Developer 内で初期設定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。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。編集モードでは、以下の情報を変更するためのフィールドが存在します。

名前 初期設定のルールの名前を指定します。

表示名 ルールの表示名を指定します。表示名を指定しなかった場合は、「名前」の値が設定されます。表示名には、任意のストリング値を使用でき、特殊文字を含めることもできます。また、固有名である必要はありません。「アクション」フィールドの「**名前の同期**」チェック・ボックスを選択すると、表示名が、対応する名前と同期されます。編集ページで行った変更を保存すると、新しい名前が有効になります。

注: 「**名前の同期**」チェック・ボックスを選択すると、ルールの表示名が使用不可になり、変更できなくなります。

ルール 初期設定の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。

説明 各初期設定のルールの情報を指定します。表示モードでは読み取り専用ですが、デシジョン・テーブルの編集モードでは編集が可能です。

アクション

「**名前の同期**」チェック・ボックスを選択することにより、表示名と名前を同期させることができます。

「デシジョン・テーブル」セクション

このセクションでは、テーブルの条件ケース（行見出しおよび列見出しとして表されます）とアクション（それらの条件ケースの交点として表されます）が示されます。「**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。

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、WebSphere Integration Developer で設計されたデシジョン・テーブル定義で指定された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

このセクションでは、既存のテンプレートを使用して新規ルールを容易に追加できます。

「ビジネス・ルール・グループの検索」ページ:

「**ビジネス・ルール・グループの検索**」ページでは、検索照会を作成することにより、指定した一連の操作対象ビジネス・ルール・グループを検索したり、その絞り込みを行ったりすることができます。「**ビジネス・ルール・グループの検索**」ページを開くには、ビジネス・ルール・マネージャーの上部にあるツールバーの「**検索**」をクリックします。

「**ビジネス・ルール・グループの検索**」ページでは、ターゲット・ネーム・スペース、ビジネス・ルール・グループ名、カスタム・プロパティ、またはこれらの任意の組み合わせを基準にして検索を実行できます。また、1 つ以上のカスタム・プロパティを追加したり、名前を基準にしてアルファベットの昇順でカスタム・プロパティをソートしたり、プロパティ・テーブル内でプロパティを上下に移動したり、カスタム・プロパティを削除したりできます。

「**ビジネス・ルール・グループの検索**」ページのコンテンツ領域には、「**メッセージ**」フィールドとページ固有の情報セクションがあり、以下のエレメントが含まれています。

「データの検索」セクション

このセクションには、以下のエレメントが含まれています。

名前 検索対象のビジネス・ルール・グループの名前を入力するためテキスト域を提供します。この値を空のままにすると、ビジネス・ルール・グループは検索コンテキストに含まれなくなります。入力した値は、名前と表示名の両方として使用されます。したがって、検索を実行すると、入力した名前の値と

一致する名前または表示名を持つビジネス・ルール・グループが検索されます。名前または表示名のいずれか一方で明示的に検索する場合は、プロパティ名を使用して、そのような検索を指定する必要があります。

例: プロパティの名前として `IBMSysName`、そのプロパティの値として `VIPGroup` を入力した場合、ビジネス・ルール・マネージャーは、`VIPGroup` に一致する名前 (表示名ではない) を持つビジネス・ルール・グループを検索します。

ターゲット・ネーム・スペース

ビジネス・ルール・グループの URL を入力するためテキスト域を提供します。この値を空のままにすると、ビジネス・ルール・グループは検索コンテキストに含まれなくなります。

「プロパティ」セクション

このセクションは、「プロパティの追加」をクリックすると開きます。以下のエレメントが表示されます。

論理演算子

「And」、「Or」、または「Not」を選択するためのドロップダウン・リストが表示されるので、複数のプロパティを含む検索照会を作成できます。

名前 プロパティの名前を入力するためテキスト域を提供します。検索コンテキストのプロパティ・テーブル内では、名前は固有でなければなりません。また、名前を空にすることはできません。

照会演算子

検索データ・フィールドごとに 4 つの照会演算子から選択するためのドロップダウン・リストが表示されます。照会演算子は以下のとおりです。

照会演算子	説明
次と等しい	ビジネス・ルール・グループ名、ターゲット・ネーム・スペース、またはプロパティの値が、指定したストリングと厳密に一致しなければならないことを示します。
次に近い	ビジネス・ルール・グループ名、ターゲット・ネーム・スペース、またはプロパティの値が、指定したストリングと類似するビジネス・ルール・グループを照会で検索することを示します。このストリングにはワイルドカード文字を含めることができます。任意の数の文字を表すワイルドカードを指定する場合はパーセント文字 ('%')、単一文字のワイルドカードを指定する場合は下線文字 ('_') を使用します。これらのワイルドカード文字は、SQL 構文に従う必要があります。
次と等しくない	ビジネス・ルール・グループ名、ターゲット・ネーム・スペース、またはプロパティの値が、指定したストリングと一致してはならないことを示します。

照会演算子	説明
次に近くない	ビジネス・ルール・グループ名、ターゲット・ネーム・スペース、またはプロパティの値が、指定したストリングと類似しないビジネス・ルール・グループを照会で検索することを示します。このストリングには、「like」演算子で定義されたワイルドカード文字を含めることができます。

値 プロパティ値を入力するためテキスト域を提供します。値は空にすることができます。また、値は検索コンテキストに取り込まれます。

例: プロパティ `PayMethod` の値を空のままにして、その照会演算子を「次と等しくない」に設定した場合は、`PayMethod` プロパティの値が空以外のストリングに設定されたビジネス・ルール・グループがすべて検索されます。

アクション

プロパティ・テーブル内でプロパティを上下に移動したり、カスタム・プロパティを削除したりできます。

「検索結果」セクション

このセクションには、以下のエレメントが含まれています。

ルール・グループ

検索照会で戻されたビジネス・ルール・グループの名前をリストします。

状況 ランタイムから検索結果として戻されたビジネス・ルール・グループの状況を表示します。以下の 4 種類のいずれかの状況になります。

ヒント: 結果のビジネス・ルール・グループをクリックすると、そのビジネス・ルール・グループのページが開きます。

状況	説明
ローカルと同じ	結果のビジネス・ルール・グループのコピーがビジネス・ルール・マネージャーに既に存在しており、その内容と結果のビジネス・ルール・グループの内容が厳密に同じであることを示します。したがって、検索後にさらにアクションが実行されることはありません。
ランタイムから変更	結果のビジネス・ルール・グループのコピーがビジネス・ルール・マネージャーに既に存在していることを示します。ただし、別のユーザー・セッションでマスター・コピーが変更されたため、ローカルの内容と結果のビジネス・ルール・グループの内容は異なります。ビジネス・ルール・マネージャーは、ランタイムからの新しい変更を取得するためにローカル・コピーを自動的に更新します。

状況	説明
ローカルで変更	結果のビジネス・ルール・グループのコピーがビジネス・ルール・マネージャーに既に存在していることを示します。ただし、それが現行ユーザーによって変更されています。ビジネス・ルール・マネージャーは、ユーザーによる以降のすべてのアクションに対してローカル・コピーを使用します。
ローカルで新規	結果のビジネス・ルール・グループのコピーがビジネス・ルール・マネージャーに存在していないことを示します。この場合、ビジネス・ルール・マネージャーは、結果のビジネス・ルール・グループのローカル・コピーを作成して、それをナビゲーション・ペインに表示します。

説明 ビジネス・ルール・グループの追加情報を提供します。

ビジネス・ルール・グループのプロパティの追加、削除、および変更

検索において、ビジネス・ルール・グループのカスタム・プロパティを使用すると、表示および変更したいビジネス・ルール・グループのサブセットを取得できます。新しいカスタム・プロパティを追加したり、既存のプロパティを削除または変更するには、ビジネス・ルール・グループの編集ページを使用します。ビジネス・ルール・グループのカスタム・プロパティの数に制限はありません。

始める前に

ビジネス・ルール・グループの編集モードにする必要があります。

制約事項: ビジネス・ルール・グループのプロパティ・サポートは、6.1 ビジネス・ルール・グループ以降で使用できます。

このタスクについて

ビジネス・ルール・グループのプロパティを追加、削除、または変更するには、以下のステップを実行します。

プロシージャ

1. 以下のオプションから選択します。

オプション	説明
オプション	ステップ

オプション	説明
プロパティのルールへの追加	<ol style="list-style-type: none"> 1. 「プロパティの追加」をクリックします。 2. 固有の「名前」を指定します。名前は空にすることはできません。 3. 固有の「値」を指定します。各プロパティは、ビジネス・ルール・グループで1回のみ定義できます。各プロパティには、定義済みの値を指定する必要があります。値は、空ストリングまたは長さゼロにすることはできますが、ヌルであってはなりません。プロパティをヌルに設定することは、プロパティを削除することと同じです。
プロパティの削除	選択したプロパティの「アクション」フィールドで、「削除」をクリックします。
プロパティの変更	対応するフィールドに新しい名前と値を入力します。
プロパティのソート	「ソート」をクリックすると、ビジネス・ルール・グループのプロパティが、プロパティ名を基準にしてアルファベットの昇順でソートされます。

2. 「保管」をクリックします。

結果

ビジネス・ルール・マネージャーは、ルールを検証してからプロパティをサーバーに送信します。

ビジネス・ルール・グループの検索

ビジネス・ルール・グループに対して検索照会を実行すると、指定した一連の操作対象ビジネス・ルール・グループを検索したり、その絞り込みを行ったりすることができます。検索照会は、名前、ターゲット・ネーム・スペース、カスタム・プロパティ、またはこれらの任意の組み合わせに基づいて作成します。

始める前に

「ビジネス・ルール・グループの検索」ページを起動する必要があります。このページを開くには、ビジネス・ルール・マネージャーのツールバーの「検索」をクリックします。

このタスクについて

検索照会を作成するには、以下のステップを実行します。

プロシージャ

1. 「名前」フィールドで、検索対象のビジネス・ルール・グループの名前を入力します。この値は、空のままにできます。ただし、検索コンテキストには含まれません。入力した値は、名前と表示名の両方として使用されます。したがっ

て、検索を実行すると、入力した名前の値と一致する名前または表示名を持つビジネス・ルール・グループが検索されます。名前または表示名のいずれか一方で明確に検索する場合は、プロパティ名を使用して、そのような検索を指定する必要があります。

例: プロパティの名前として `IBMSystemName`、そのプロパティの値として `VIPGroup` を入力した場合、ビジネス・ルール・マネージャーは、`VIPGroup` に一致する名前 (表示名ではない) を持つビジネス・ルール・グループを検索します。

2. 「ターゲット・ネーム・スペース」フィールドにビジネス・ルール・グループの URL を入力します。この値は、空のままにできます。ただし、検索コンテキストには含められません。
3. それぞれの「データの検索」フィールドで、以下の 4 つの照会演算子のいずれかを選択します。

オプション	説明
照会演算子	説明
次と等しい	ビジネス・ルール・グループ名、ターゲット・ネーム・スペース、またはプロパティの値が、指定したストリングと厳密に一致しなければならないことを示します。
次に近い	<p>ビジネス・ルール・グループ名、ターゲット・ネーム・スペース、またはプロパティの値が、指定したストリングと類似するビジネス・ルール・グループを照会で検索することを示します。このストリングにはワイルドカード文字を含めることができます。任意の数の文字を表すワイルドカードを指定する場合はパーセント文字 (<code>%</code>)、単一文字のワイルドカードを指定する場合は下線文字 (<code>_</code>) を使用します。これらのワイルドカード文字は、SQL 構文に従う必要があります。</p> <p>例:</p> <ol style="list-style-type: none"> 1. ビジネス・ルール・グループ名として <code>"is like" "Discount"</code>、ターゲット・ネーム・スペースとして <code>"http://calculateDiscounts"</code> を入力した場合、検索を実行すると、そのストリングを含み、またその URL を持つビジネス・ルール・グループがすべて戻されます。 2. ビジネス・ルール・グループ名として <code>"is like" "%Discount%"</code> を入力した場合、検索を実行すると、<code>AirlineTicketDiscount</code> や <code>MovieTicketDiscountRules</code> などの名前を持つすべてのビジネス・ルール・グループが戻されます。

オプション	説明
次と等しくない	ビジネス・ルール・グループ名、ターゲット・ネーム・スペース、またはプロパティの値が、指定したストリングと一致してはならないことを示します。
次に近くない	ビジネス・ルール・グループ名、ターゲット・ネーム・スペース、またはプロパティの値が、指定したストリングと類似しないビジネス・ルール・グループを照会で検索することを示します。このストリングには、「like」演算子で定義されたワイルドカード文字を含めることができます。

4. **オプション:** 検索コンテキストに必要な数だけプロパティを追加するには、「**プロパティの追加**」をクリックします。
- 「名前」を指定します。検索コンテキストのプロパティ・テーブル内では、名前は固有でなければなりません。また、名前を空にすることはできません。
 - 「照会演算子」を指定します。
 - 「値」を指定します。値は空にすることができます。また、値は検索コンテキストに取り込まれます。

例: プロパティ `PayMethod` の値を空のままにして、その照会演算子を「次と等しくない」に設定した場合は、`PayMethod` プロパティの値が空以外のストリングに設定されたビジネス・ルール・グループがすべて検索されます。

- 「アクション」フィールドで上矢印と下矢印をクリックして、プロパティを適切な順番に並べます。

ヒント: 「論理演算子」フィールドで「And」、「Or」、または「Not」を使用してプロパティを結合すると、複数のプロパティを含む検索照会を作成できます。

例: ターゲット・ネーム・スペース `"http://calculateDiscounts"` 内のビジネス・ルール・グループのうち、`DiscountedItem` プロパティにストリング `"men T-Shirts"` が含まれており、`Ship Handling` プロパティに値 `"Free"` が設定されているものをすべて検索するには、論理プロパティ `"And"` を使用します。

注: 「ビジネス・ルール・グループの検索」ページでプロパティを追加、削除、または変更しても、それは検索コンテキスト内でしか適用されません。ビジネス・ルール・マネージャー内のルール・オブジェクトのプロパティには、影響は及びません。

- 「検索」をクリックします。

結果

検索照会に一致したビジネス・ルール・グループは、「ビジネス・ルール・グループの検索」ページの「検索結果」セクションに表示されます。ランタイムから検索結果として戻されたビジネス・ルール・グループの状況は、以下の 4 種類の状況の

いずれかになります。

状況	説明
ローカルと同じ	結果のビジネス・ルール・グループのコピーがビジネス・ルール・マネージャーに既に存在しており、その内容と結果のビジネス・ルール・グループの内容が厳密に同じであることを示します。したがって、検索後にさらにアクションが実行されることはありません。
ランタイムから変更	結果のビジネス・ルール・グループのコピーがビジネス・ルール・マネージャーに既に存在していることを示します。ただし、別のユーザー・セッションでマスター・コピーが変更されたため、ローカルの内容と結果のビジネス・ルール・グループの内容は異なります。ビジネス・ルール・マネージャーは、ランタイムからの新しい変更を取得するためにローカル・コピーを自動的に更新します。
ローカルで変更	結果のビジネス・ルール・グループのコピーがビジネス・ルール・マネージャーに既に存在していることを示します。ただし、それが現行ユーザーによって変更されています。ビジネス・ルール・マネージャーは、ユーザーによる以降のすべてのアクションに対してローカル・コピーを使用します。
ローカルで新規	結果のビジネス・ルール・グループのコピーがビジネス・ルール・マネージャーに存在していないことを示します。この場合、ビジネス・ルール・マネージャーは、結果のビジネス・ルール・グループのローカル・コピーを作成して、それをナビゲーション・ペインにも表示します。

注: ビジネス・ルール・グループの変更の同期は、戻される検索結果と同時に実行され、ビジネス・ルール・マネージャーのコンテキストに適用されます。つまり、影響を受けるビジネス・ルール・グループに対する次の操作では、そのビジネス・ルール・グループの最新の更新が使用されます。

例: 次のプロパティが指定された 4 つのビジネス・ルール・グループがインストールされているとします。

ビジネス・ルール・グループ 1

- **名前:** BRDCR002BRG2.brg
- **ターゲット・ネーム・スペース:** <http://BRDCR002BRG2/com/ibm/br/rulegroup>
- **プロパティ:**
 - organization, 7GAA
 - department, accounting
 - ID, 0000047

- ID_cert45, ABC
- region, NorthRegion

ビジネス・ルール・グループ 2

- 名前: BRDCR002BRG3.brg
- ターゲット・ネーム・スペース: http://BRDCR002BRG3/com/ibm/br/rulegroup
- プロパティ:
 - organization, 7FAB
 - department, finance
 - ID, 0000053
 - ID_app45, DEF
 - region, NorthCentralRegion

ビジネス・ルール・グループ 3

- 名前: BRDCR002BRG4.brg
- ターゲット・ネーム・スペース: http://BRDCR002BRG4/com/ibm/br/rulegroup
- プロパティ:
 - organization, 7HAA
 - department, shipping
 - ID, 0000023
 - ID_app45, GHI
 - region, SouthRegion

ビジネス・ルール・グループ 4

- 名前: BRDCR002BRG5.brg
- ターゲット・ネーム・スペース: http://BRDCR002BRG5/com/ibm/br/rulegroup
- プロパティ:
 - organization, 8JAA
 - department, claims
 - ID, 00000567
 - region, SouthCentralRegion
 - manager, Joe Bean

1 つのプロパティを使用してビジネス・ルール・グループを取得します。

論理演算子	名前	照会演算子	値
	department	次と等しい	accounting

これは、ビジネス・ルール・グループ 1 を戻します。

2 つのプロパティと複数文字ワイルドカード「%」を使用してビジネス・ルール・グループを取得します。

論理演算子	名前	照会演算子	値
	region	次に近い	%Region
AND	ID	次に近い	00000%

これは、ビジネス・ルール・グループ 1 から 4 を戻します。

単一文字ワイルドカード「_」を使用してビジネス・ルール・グループを取得します。

論理演算子	名前	照会演算子	値
	ID	次に近い	00000_3

これは、ビジネス・ルール・グループ 2 と 3 を戻します。

複数の単一文字ワイルドカード「_」を使用してビジネス・ルール・グループを取得します。

論理演算子	名前	照会演算子	値
	region	次に近い	__uth%Region

これは、ビジネス・ルール・グループ 3 と 4 を戻します。

単一文字ワイルドカード「_」と NOT 演算子を使用してビジネス・ルール・グループを取得します。

論理演算子	名前	照会演算子	値
	organization	次に近くない	7_A

これは、ビジネス・ルール・グループ 4 を戻します。

複数文字ワイルドカード「%」と NOT 演算子を使用してビジネス・ルール・グループを取得します。

論理演算子	名前	照会演算子	値
	organization	次に近くない	7%

これは、ビジネス・ルール・グループ 4 を戻します。

次のタスク

結果のビジネス・ルール・グループをクリックすると、そのビジネス・ルール・グループのページが開きます。

スケジュール済みルール・ロジック項目の使用

スケジュール済みルール・ロジック項目は、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示します。

このタスクについて

ビジネス・ルール・マネージャーを使用して、スケジュール済みルール・ロジック項目の作成、変更、または削除を行います。

スケジュール済みルール・ロジック項目の作成:

スケジュール済みルール・ロジック項目は、既存の項目から作成します。

始める前に

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。

このタスクについて

新しいスケジュール済みルール・ロジック項目を作成するには、以下のステップを実行します。

プロシージャ

1. 「ルール・スケジュール」 ページで、「**選択レコードの追加**」をクリックします。

新しいスケジュール済みルール・ロジック項目がリストの下部に追加されます。「**開始日/時刻**」および「**終了日/時刻**」フィールドは、「**1 月 1 日**」に設定されています。日付/時刻フィールドの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。デフォルトのルール・ロジックが指定されている場合は、そのロジックがこの間隔で使用されます。ベスト・プラクティスとして、常にデフォルトのルール・ロジック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。

結果

スケジュール済みルール・ロジック項目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

162 ページの『スケジュール済みルール・ロジック項目の削除』

既存のスケジュール済みルール・ロジック項目をスケジュール済みルール・ロジック・テーブルから削除できます。スケジュール済みルール・ロジック項目を削除した場合、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。スケジュール済みルール・ロジック項目は、デフォルトのルール・ロジックとして、または特定の日時を指定して再度追加できます。

スケジュール済みルール・ロジック項目の変更:

既存のスケジュール済みルール・ロジック項目の日付および時刻の値を変更できます。

始める前に

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。

このタスクについて

スケジュール済みルール・ロジック項目を変更するには、以下のステップを実行します。

プロシージャ

1. 「ルール・スケジュール」ページで、スケジュール済みルール・ロジック項目の「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. スケジュール済みルール・ロジック項目の「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。デフォルトのルール・ロジックが指定されている場合は、そのロジックがこの間隔で使用されます。ベスト・プラクティスとして、常にデフォルトのルール・ロジック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。

結果

スケジュール済みルール・ロジック項目がローカル側で保管されます。スケジュール済みルール・ロジック項目をリポジトリに公開する準備ができました。詳しくは、162 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。

次のタスク

ビジネス・ルールの日付の設定について詳しくは、153 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

162 ページの『スケジュール済みルール・ロジック項目の削除』

既存のスケジュール済みルール・ロジック項目をスケジュール済みルール・ロジック・テーブルから削除できます。スケジュール済みルール・ロジック項目を削除した場合、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。スケジュール済みルール・ロジック項目は、デフォルトのルール・ロジックとして、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。どの時点であっても、操作に対して有効にできるルール・ロジックは 1 つのみです。また、どの時点であっても、有効なルール・ロジックが見つからない場合は、デフォルトのルール・ロジックが使用されます。

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」

終了日をスケジュール済みルール・ロジック項目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 つのスケジュール済みルール・ロジック項目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、最初のスケジュール済みルール・ロジック項目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 番目のスケジュール済みルール・ロジック項目の開始日は、最初のスケジュール済みルール・ロジック項目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。

始める前に

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。

このタスクについて

スケジュール済みルール・ロジック項目を分割するには、以下のステップを実行します。

プロシージャ

1. スケジュール済みルール・ロジック項目の横にある「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しいスケジュール済みルール・ロジック項目が作成されます。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. 新しいスケジュール済みルール・ロジック項目の開始日/時刻を選択します。

元のスケジュール済みルール・ロジック項目の終了日/時刻が継続 から新しいスケジュール済みルール・ロジック項目の開始日/時刻に変更され、新しいスケジュール済みルール・ロジック項目の終了日/時刻が前のスケジュール済みルール・ロジック項目の終了日/時刻に変更されます。

3. 新しいスケジュール済みルール・ロジック項目の「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・セット項目の作成:

新規のルール・セット項目を作成するには、既存のルール・セット項目をコピーし、その値を変更します。

このタスクについて

新規ルール・セット項目を作成するには、以下のステップを実行します。

プロシージャ

1. 選択したルール・セットのスケジュール済みルール・ロジック項目の横にある「**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項目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・セット項目の固有の名前を入力します。
3. 「表示名」フィールドで、新規ルール・セット項目の表示名を入力します。ルール・セットでは、表示名は固有である必要はありません。どのストリング値でも使用でき、特殊文字を含めることもできます。表示名を指定しない場合は、「名前」の値が表示名として使用されます。

注: 表示名と、ルール・セットの対応する名前とを同期させるには、「名前との同期」チェック・ボックスを選択します。

4. 新しいルール・セットの項目の簡単な説明を「説明」フィールドに入力します。
5. 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せん。

6. 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。
7. 「保管」をクリックします。

結果

ルール・セット項目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

162 ページの『スケジュール済みルール・ロジック項目の削除』

既存のスケジュール済みルール・ロジック項目をスケジュール済みルール・ロジック・テーブルから削除できます。スケジュール済みルール・ロジック項目を削除した場合、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。スケジュール済みルール・ロジック項目は、デフォルトのルール・ロジックとして、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。

始める前に

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。

このタスクについて

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。

プロシージャ

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. 「**表示名**」フィールドで、新規ルールの表示名を入力します。ルールでは、表示名は固有である必要はありません。どのストリング値でも使用でき、特殊文字を含めることもできます。表示名を指定しない場合は、「**名前**」の値が表示名として使用されます。

注: 表示名と名前の値とを同期させるには、「**アクション**」フィールドで対応する「**名前の同期**」チェック・ボックスを選択します。このチェック・ボックスが選択されている場合は、ルールの表示名が使用不可になり、変更できません。
 - c. 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。
 - d. ルールの説明を入力します。
 - e. 「**追加**」をクリックします。
3. 「**アクション**」フィールドで上矢印または下矢印をクリックして、ルールを適切な順序で配置します。

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。

次のタスク

ルール・セットを公開する準備ができました。詳しくは、162 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

162 ページの『スケジュール済みルール・ロジック項目の削除』

既存のスケジュール済みルール・ロジック項目をスケジュール済みルール・ロジック・テーブルから削除できます。スケジュール済みルール・ロジック項目を削除した場合、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。スケジュール済みルール・ロジック項目は、デフォルトのルール・ロジックとして、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。

始める前に

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。

このタスクについて

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。

プロシージャ

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。

次のタスク

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、162 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

162 ページの『スケジュール済みルール・ロジック項目の削除』

既存のスケジュール済みルール・ロジック項目をスケジュール済みルール・ロジック・テーブルから削除できます。スケジュール済みルール・ロジック項目を削除した場合、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

ク」セクション内にリストされます。スケジュール済みルール・ロジック項目は、デフォルトのルール・ロジックとして、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、表形式のスケジュール済みルール・ロジック項目です。このテーブルでは、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の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。

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、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」条件のテンプレートで定義したアクションを取り込むことはできません。「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。

下図は、デフォルト・メンバー・タイプをシルバーに設定する初期化アクション・ルール と、 500 ドル未満を使うゴールド顧客とシルバー顧客に適用される 「その他の場合」条件 を持つデシジョン・テーブルを示します。条件 PurchaseAmount および MemberType は、1 番目と 2 番目の行に沿って存在します。アクション Discount は、3 番目の行に沿って存在します。条件とアクションの方向を矢印によって示します。

Initialization Rule						
Display Name	Rule	Description				
Rule1	Default Member Type = Silver					

Decision Table						
PurchaseAmount	>= 500 && < 2000		>= 2000		Otherwise	
MemberType	Gold	Silver	Gold	Silver	Gold	Silver
Discount	8 %	3 %	10 %	5 %	2 %	0 %

図 6. デシジョン・テーブル

この例では、500 ドルから 1999 ドルを使うゴールド顧客は 8% の割引を受け、500 ドルから 2000 ドルを使うシルバー顧客は 3% の割引を受けることが示されています。2000 ドル以上を使うゴールド顧客は 10% の割引を受け、2000 ドル以上を使うシルバー顧客は 5% の割引を受けます。500 ドル未満を使うゴールド顧客は 2% の割引を受け、500 ドル未満を使うシルバー顧客は 0% の割引を受けます。

デシジョン・テーブル項目の作成:

新規のデシジョン・テーブル項目を作成するには、既存のデシジョン・テーブル項目をコピーし、その値を変更します。

このタスクについて

デシジョン・テーブル項目を作成するには、以下のステップを実行します。

プロシージャ

1. 選択したデシジョン・テーブルのスケジュール済みルール・ロジック項目の横にある「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項目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・テーブル項目の名前を入力します。
3. 「表示名」フィールドで、新規デシジョン・テーブル項目の表示名を入力します。デシジョン・テーブルでは、表示名は固有である必要はありません。どのストリング値でも使用でき、特殊文字を含めることもできます。表示名を指定しない場合は、「名前」の値が表示名として使用されます。

注: 表示名と名前の値とを同期させるには、対応する「名前との同期」チェック・ボックスを選択します。

4. 新しいデシジョン・テーブルの項目の簡単な説明を「説明」フィールドに入力します。
5. 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せん。

6. 「保管」をクリックします。

結果

デシジョン・テーブル項目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、162 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

162 ページの『スケジュール済みルール・ロジック項目の削除』

既存のスケジュール済みルール・ロジック項目をスケジュール済みルール・ロジック・テーブルから削除できます。スケジュール済みルール・ロジック項目を削除した場合、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。スケジュール済みルール・ロジック項目は、デフォルトのルール・ロジックとして、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、テンプレートの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	

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	
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・テーブル項目の変更:

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。

始める前に

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。

このタスクについて

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。

プロシージャ

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。ビジネス・ルール・マネージャーで「その他の場合」条件を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」条件のテンプレートで定義したアクションを取り込むことはできます。
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。

結果

ルールがローカル側で変更されます。ルールをリポジトリに公開する準備ができました。詳しくは、162 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

162 ページの『スケジュール済みルール・ロジック項目の削除』

既存のスケジュール済みルール・ロジック項目をスケジュール済みルール・ロジック・テーブルから削除できます。スケジュール済みルール・ロジック項目を削除した場合、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。スケジュール済みルール・ロジック項目は、デフォルトのルール・ロジックとして、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。

始める前に

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。

このタスクについて

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。

プロシージャ

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。

結果

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、『ビジネス・ルールの公開と復帰』を参照してください。

スケジュール済みルール・ロジック項目の削除

既存のスケジュール済みルール・ロジック項目をスケジュール済みルール・ロジック・テーブルから削除できます。スケジュール済みルール・ロジック項目を削除した場合、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。スケジュール済みルール・ロジック項目は、デフォルトのルール・ロジックとして、または特定の日時を指定して再度追加できます。

始める前に

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。

このタスクについて

スケジュール済みルール・ロジックを削除するには、以下のステップを実行します。

プロシージャ

1. 「ルール・スケジュール」ページで、スケジュール済みルール・ロジックを選択し、「削除」をクリックします。

スケジュール済みルール・ロジックが削除されます。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。

注: ビジネス・ルール・グループに対する各操作には、スケジュール済みルール・ロジック項目として、またはデフォルトのルール・ロジック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。すべてのスケジュール済みルール・ロジック項目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。

結果

スケジュール済みルール・ロジック項目が一時的に保管され、リポジトリ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・ソースに変更内容を格納するには、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。

始める前に

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。

このタスクについて

サーバーは、ビジネス・ルール・グループ・レベルとルール・スケジュール・レベルで変更を公開します。公開ステージでは、ビジネス・ルール・マネージャーは検証を行う必要はありません。なぜなら、ビジネス・ルール・マネージャーは、各編集ページで入力された変更内容を、情報の保存時にすべて検証するからです。

ビジネス・ルール・グループまたはルール・スケジュ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。

プロシージャ

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、リポジトリに送信するビジネス・ルール・グループおよびルール・スケジュールを選択します。すべてのビジネス・ルール・グループおよびルール・スケジュールをまとめて単一トランザクションとして公開するか、またはそれら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・グループまたはルール・スケジュールに対してローカル側で保存された変更をすべて取り消して、変更されたリソースをサーバー・メモリー内の元のコピーで置き換えるには、ビジネス・ルール・グループまたはルール・スケジュールのチェック・ボックスを選択して、「復帰」をクリックします。ビジネス・ルール・グループおよびルール・スケジュールを公開すると、サーバー・メモリー内にある元のコピーが変更されるため、公開の後でそれら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。

選択されているビジネス・ルール・グループとルール・スケジュールがサーバー・メモリーに書き込まれます。

次のタスク

ビジネス・ルールをデータ・ソース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・エラーの解決:

ログイン・エラーは、ログイン時に発生します。

このタスクについて

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。

プロシージャ

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。

次のタスク

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の解決:

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。

このタスクについて

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。

ログイン競合エラーを解決するには、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の解決:

1 人のユーザーがデータ・ソース内のビジネス・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と、アクセス競合エラーが発生します。

始める前に

ローカル側の変更をリポジトリに公開すると、このエラーが報告されます。

このタスクについて

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。

- 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、エラーが発生したビジネス・ルール・グループおよびルール・スケジュールをデータ・ソースから再ロードする必要があります。これは、エラーが発生したビジネス・ルール・グループおよびルール・スケジュールのローカル変更は使用できなくなるからです。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、ビジネス・ルール・グループまたはルール・スケジュールのページを再ロードします。エラー状態でない他のビジネス・ルール・グループおよびルール・スケジュールのローカル変更は引き続き使用できます。

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・テーブルに含まれています。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る) を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トター・コンポーネントの要件を識別し、セレクトター・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トター・テーブルを定義します。セレクトター・コンポーネントの開発に関連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。

セレクトター・コンポーネントの管理は、セレクトター・コンポーネントに関連した作業またはセレクトター・テーブルに関連した作業で構成されます。

制約事項: セレクトター・コンポーネント・ページ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。

セレクトター・コンポーネントの表示

セレクトター・コンポーネントの表示は、セレクトター・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、セレクトター・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、セレクトター・コンポーネントから構成されるセレクトター・テーブルを表示したりできます。

始める前に

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。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

サーバーに存在するセレクトター・コンポーネントを確認するには、以下のステップを実行します。

制約事項: セレクトター・コンポーネント・ページ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。

プロシージャ

1. ナビゲーション・ペインで「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. サーバー・リスト内のサーバーの名前をクリックします。

4. 「**Business Integration**」の下で「**セレクター**」>「**セレクター**」をクリックします。

コンソールには、すべてのセレクター・コンポーネントと各コンポーネントの説明が表示されます。

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) などをご画面から実行できます。

始める前に

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。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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。

このタスクについて

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確認して、その他のセレクター・テーブル関連のタスクを実行します。セレクター・テーブルを表示するには、以下のステップを実行します。

プロシージャ

1. 「サーバー」>「アプリケーション・サーバー」> *servername* > 「**Business Integration**」>「**セレクター**」>「**セレクター**」をクリックして、セレクター・コンポーネントを表示します。
2. セレクター・コンポーネントの表示からセレクター名をクリックして、選択したコンポーネントのセレクター・テーブルを表示します。
3. 画面上のセレクター・テーブルを 1 つクリックして、セレクター・テーブルを構成するターゲット・コンポーネントを表示します。

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。

始める前に

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。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、このタスクを実行するには、管理者としてログインする必要があります。

このタスクについて

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。ターゲット・コンポーネントを変更するには、以下のステップを実行します。

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、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。

プロシージャ

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。「サーバー」>「アプリケーション・サーバー」> *servername* > 「Business Integration」>「セレクター」>「セレクター」> *selectorname* をクリックします。
2. 画面上のセレクター・テーブルを 1 つクリックして、セレクター・テーブルを構成するターゲット・コンポーネントを表示します。
3. 変更するターゲット・コンポーネントのターゲット ID をクリックします。
4. 項目を変更します。

項目の変更する部分 ターゲット宛先

選択基準

変更の手順

1. ターゲット・コンポーネント・リストの隣にある矢印をクリックして、適格なターゲット・コンポーネントを表示します。
2. リストからターゲット・コンポーネントを選択します。
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・リストの隣にある矢印をクリックして、適格なターゲット・コンポーネントを表示します。
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。

5. 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。

6. この画面で作業を継続するには「適用」、ターゲット・コンポーネントの画面に戻るには「OK」をクリックします。
7. 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。

結果

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。

始める前に

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。

このタスクに必要なセキュリティー・ロール: セキュリティーと役割ベースの許可が有効になっている場合、このタスクを実行するには、管理者としてログインする必要があります。

このタスクについて

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。

ターゲット・コンポーネントを追加するには、以下のステップを実行します。

プロシージャ

1. 『セクター・テーブルの表示』で説明されているように、セクター・テーブルを表示します。「サーバー」>「アプリケーション・サーバー」> *servername* > 「Business Integration」> 「セクター」> 「セクター」> *selectername* をクリックします。
2. 画面上のセクター・テーブルを 1 つクリックして、セクター・テーブルを構成するターゲット・コンポーネントを表示します。
3. 「新規」をクリックして、データが事前に入力されたターゲット・コンポーネント・ページを表示します。
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。

結果

セクター・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。セクター・コンポーネントは、更新されたセクター・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・コンポーネントの削除

セクター・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、セクター・コンポーネントの処理が変更されます。

始める前に

このタスクを実行するには、セクター・テーブルが存在している必要があります。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、このタスクを実行するには、管理者としてログインする必要があります。

このタスクについて

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。

ターゲット・コンポーネントを削除するには、以下のステップを実行します。

プロシージャ

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。
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。

システムによってページが更新され、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。

結果

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・テーブルを変更したら、セレクトター・コンポーネントをエクスポートします。これにより、開発環境にインポートできるファイルが作成され、開発の成果物を実際の実動システムの成果物と同期させることができます。

始める前に

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』の説明に従って、セレクトター・コンポーネントを事前に表示しておく必要があります。「サーバー」>「アプリケーション・サーバー」> *servername* > 「Business Integration」> 「セレクトター」> 「セレクトター」をクリックします。

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、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する必要があります。セキュリティが有効でないときは、ユーザー ID を指定して管理コンソールにログインする必要があります。

このタスクについて

管理コンソールを使用してセレクトターをエクスポートするには、以下のステップを実行します。

プロシージャ

1. 1 つ以上のセレクトターの隣にあるチェック・ボックスを選択して、「エクスポート」をクリックします。

ブラウザーには、選択したセレクトター・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トター」ページです。各セレクトターのファイル拡張子は、.zip です。

2. 各ファイル名をクリックして、ファイルをファイル・システムにダウンロードします。ファイルを保存するためのプロンプトがシステムから出されたら、「OK」をクリックします。

注: ファイルの保存を選択した場合は、ファイルのダウンロード時にファイル名を変更できます。

3. 「戻る」をクリックして、セレクターのリストに戻ります。

結果

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

インストールしたセレクター・コンポーネントを、アプリケーションを再インストールせずに更新するには、セレクターをインポートします。

始める前に

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。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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。

管理コンソールを使用してセレクターをインポートするには、以下のステップを実行します。

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。

プロシージャ

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。「サーバー」>「アプリケーション・サーバー」> *servername* > 「**Business Integration**」>「セレクター」>「セレクター」をクリックします。
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」ページ上でファイルへのパスを指定します。

次のタスク

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。

バインディング

バインディングにより、さまざまなプロトコルとトランスポートを使用して対話できるように SCA インポートとエクスポートが構成されます。SCA アプリケーションは、選択されたバインディングのトランスポートとプロトコルを使用して、外部クライアントとサービスに接続します。バインディングは、特定のインターフェースのメッセージのフォーマットとプロトコルの詳細を指定します。

JMS バインディングは非同期であり、信頼性が重要な場合に使用します。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。

Web サービス、HTTP および EIS 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。

バインディングのタイプ

EIS バインディングは、外部 JMS アプリケーションとの通信に使用できます。EIS システムからアプリケーションまたはアプリケーションから EIS システムのどちらでも使用できます。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。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、サーバー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。

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。これらのバインディングでは、同じ JMS リソースを共有することにより、JMS を使用して直接通信できます。EIS バインディングでこれを行うことはできません。WebSphere 組み込み JMS プロバイダーである WebSphere MQ JMS プロバイダーと共に使用する専用 JMS バインディングと、サード・パーティー JMS プロバイダーと共に使用する汎用 JMS バインディングがあります。

WebSphere MQ (非 JMS) バインディングにより、ネイティブ MQ アプリケーション間のメディアーションおよびネイティブ MQ アプリケーションとの通信が実現するため、これらのアプリケーションがサービス指向アーキテクチャー・フレームワークに組み込まれ、MQ 固有のヘッダー情報にアクセスできるようになります。

サポートされているバインディング・タイプすべてのリストを、以下に示します。

- JMS (WebSphere 組み込み) バインディング
- 汎用 JMS バインディング
- WebSphere MQ JMS バインディング
- WebSphere MQ ネイティブ・バインディング (WebSphere MQ バインディング)

- HTTP バインディング
- Web サービス・バインディング
- ステートレス・セッション Bean バインディング (インポートのみ)

コマンド行からバインディングを構成できます。詳しくは、このインフォメーション・センターの「リファレンス」>「管理コマンド」のセクションを参照してください。

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。

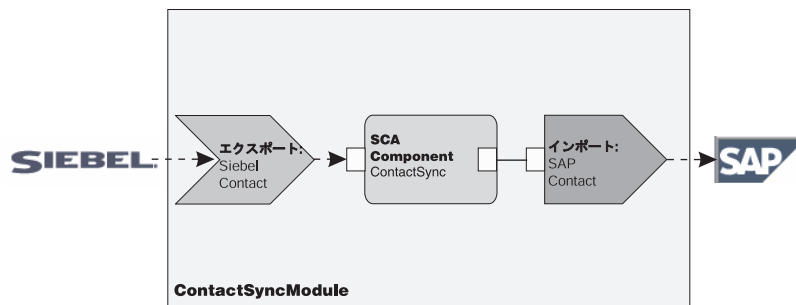
SCA コンポーネントでは、外部 EIS との間でのデータの転送が必要になる場合があります。このような接続を必要とする SCA モジュールを作成する場合は、SCA コンポーネントに加えて、特定の外部 EIS との通信のために EIS インポートまたは EIS エクスポート (あるいはこの両方) を組み込む必要があります。EIS インポートおよびエクスポートを作成する方法についての詳細情報は、WebSphere Integration Developer インフォメーション・センターにあります。

SCA ドメインは 1 つの Network Deployment セルに限定されています。このため、SCA バインディングを使用して 1 つのセルから別のセルのサービス呼び出す場合は、モジュール名が固有であることは保証されません。実動環境では、MQ や JMS などの別のタイプのバインディングを使用してください。テスト環境のモジュールで、SCA バインディングを使用して同一セル内のサービス呼び出す場合は、テスト・セル自体の中にサービスのスタブを作成することにより、テスト・セルから実動セルを呼び出す操作を回避してください。

EIS バインディングは WebSphere Integration Developer から管理されます。

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 アプリケーション・エクスポートを介して、Siebel への接続に対する変更を 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 アプリケーション・インポートを介して SAP アプリケーションを利用します。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

181 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。

191 ページの『汎用 JMS バインディング』

汎用 JMS インポート・バインディングとエクスポート・バインディングにより、サード・パーティーの JMS 1.1 準拠プロバイダーに対する接続が提供されます。このバインディングの操作は、JMS バインディングの操作と似ています。

201 ページの『WebSphere MQ JMS バインディング』

WebSphere MQ JMS バインディングは、WebSphere MQ JMS ベースのプロバイダーとの統合を提供します。

210 ページの『WebSphere MQ バインディング』

WebSphere MQ アプリケーションは、WebSphere MQ アプリケーションと Service Component Architecture との接続を提供します。

関連タスク

236 ページの『イベントの順序付けの有効化: EIS エクスポート』

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。

関連情報



アダプターを使用したサービスの開発

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。

EIS 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 アプリケーション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 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。

インポート

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。外部アプリケーションを 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、外部アプリケーション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 コンポーネントがモジュール外部のアプリケーションを統一された形式で認識できるようにします。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・アプリケーションによって公開されるインターフェースがあります。ここには、1 つ以上のメソッドがあり、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受け入れます。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。バインディングは 3 つのレベルで作成されます (インターフェース・バインディングがそれに含まれるメソッド・バインディングを使用し、さらにそのメソッド・バインディングがデータ・バインディングを使用します)。

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、アプリケーションを提供する 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アプリケーション・セットが EIS の特定のインスタンスによって提供されていて、このインスタンスには接続を介してアクセスできるという事実を反映しています。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・バインディングは、データ・バインディングを使用して、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。ハンドラーは最初にインターフェース・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、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。次に、データ・バインディングを使用して、リソース・アダプターが認識するフォーマットにメソッド引数をバインドします。ここで、CCI インターフェースを使用して対話が実行されます。呼び出しが戻されたら、データ・バインディングによって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。

エクスポート

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。外部アプリケーションを 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、外部アプリケーション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、EIS からの外部要求を listen するサブスクリプション・アプリケーションであると考えられます。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・アプリケーションとして認識します。

EIS エクスポートは、SCA コンポーネントがモジュール外部のアプリケーションを統一された形式で認識できるようにします。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。バインディングは 3 つのレベルで作成されます (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し、さらにそのネイティブ・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します)。

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・フォーマットをコンポーネントが認識するフォーマットにバインドします。

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トリガーします。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・ハンドラーを検出し、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、リスナー・バインディングに追加します。各ネイティブ・メソッド・バインディングと

共に、データ・バイ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・バイディングを呼び出します。リスナー・バイディングは、エクスポートされたコンポーネントを検出し、関数セクターを使用してネイティブ・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・バイディングを検出します。その後、ネイティブ・メソッド・バイ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。

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

J2EE リソース

EIS モジュール (EIS モジュール・パターンに準拠する 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、アプリケーションが 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るので、アプリケーションを実行する準備が整います。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。

JCA 対話仕様および接続仕様の動的プロパティー

SCA EIS バイディングは、ペイロード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 バイディングはペイロードを必要とします。このペイロードは、「**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。このプロパティー・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での対話仕様プロパティーの名前になります。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ー名が続かなければなりません。例えば、`InteractionId` という Java Bean プロパティーを持つ対話仕様は、プロパティー名を `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ー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` プロパティーを送受信するために、EIS インポート・バイディング・アプリケーションを呼び出します。ここで、サーバーのビジネス・グラフまたはビジネス・オブジェクトには下位の「**properties**」ビジネス・オブジェクトが含まれていて、このビジネス・オブジェクトによってペイロードを持つプロトコル固有データを送信でき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");
```

ペイロードを作成します。

```
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 バインディングの構成

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サーバー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。

始める前に

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。

このタスクについて

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。

プロシージャ

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」という見出しの下でモジュール・コンポーネント「インポート」または「エクスポート」を展開し、次に管理するインポートまたは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: 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: 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、管理するインポート・バインディングまたはエクスポート・バインディングをクリックします。インポートまたは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

サーバーは、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、サーバーの SCA モジュールにメッセージを送信する必要があり、サーバー内からコンポーネントによって呼び出される必要もあります。

クライアントは、動的起動インターフェース (DII) または Java インターフェースを使用して、その他のアプリケーションの場合と同様に 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、サービス・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 アプリケーション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 などの外部アプリケーションが (EIS エク

サポート・アプリケーション・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 アプリケーションにバインドします。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アプリケーション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・アーキテクチャ・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・アーキテクチャ 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。

以下の JMS バインディングがあります。

- JMS JCA 1.5 に準拠したサービス統合バス (SIB) プロバイダー・バインディング (JMS バインディング)
- JMS 1.1 に準拠した非 JCA の汎用 JMS バインディング (汎用バインディング)
- WebSphere MQ JMS バインディング。WebSphere MQ に対し JMS プロバイダー・サポートを提供し、J2EE アプリケーションとの相互運用性を実現します。(WebSphere MQ JMS バインディング)

JMS バインディングを介して提供されるアプリケーションにより、Service Component Architecture (SCA) モジュールは、外部 JMS システムに対する呼び出しを実行したり、外部 JMS システムからメッセージを受信したりできます。

また、WebSphere MQ バインディング (WebSphere MQ バインディング) もサポートされています。このバインディングにより、ネイティブ MQ ユーザーは、着信メッセージと発信メッセージの任意のフォーマットを処理できます (WebSphere MQ が必要です)。

JMS アプリケーションは、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。

関連タスク

237 ページの『イベントの順序付けの有効化: JMS エクスポート』

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 バインディング: 全体像

JMS バインディングは、Service Component Architecture (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: SCA モジュールと外部 JMS システムの間の管理された両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・アプリケーション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・バインディングでは、SCA モジュール内で使用する外部 JMS アプリケーションをインポートできます。

JMS 宛先に関連する JMS プロバイダーへの接続を作成するには、JMS 接続ファクトリーを使用します。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。

外部 JMS システム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。

JMS インポート・バインディングでは、以下の 2 種類の使用シナリオがサポートされています。

- 片方向: インポート・ファイル内の `send` 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設定しません。
- 両方向: (要求応答) - インポート・ファイルで指定された `send` 宛先にメッセージを送信します。`receive` 宛先がアウトバウンド・メッセージの `replyTo` ヘッダー・プロパティに設定されます。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。要求応答シナリオでは、応答は要求の `messageId` を応答メッセージの `correlationId` フィールドにコピーするか (デフォルト)、または要求の `correlationId` を応答メッセージの `correlationId` フィールドにコピーすることができます。

片方向と両方向のいずれのシナリオを使用する場合も、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。

183 ページの図 7 は、インポートがどのように外部サービスにリンクされているのかを示しています。

JMS インポート

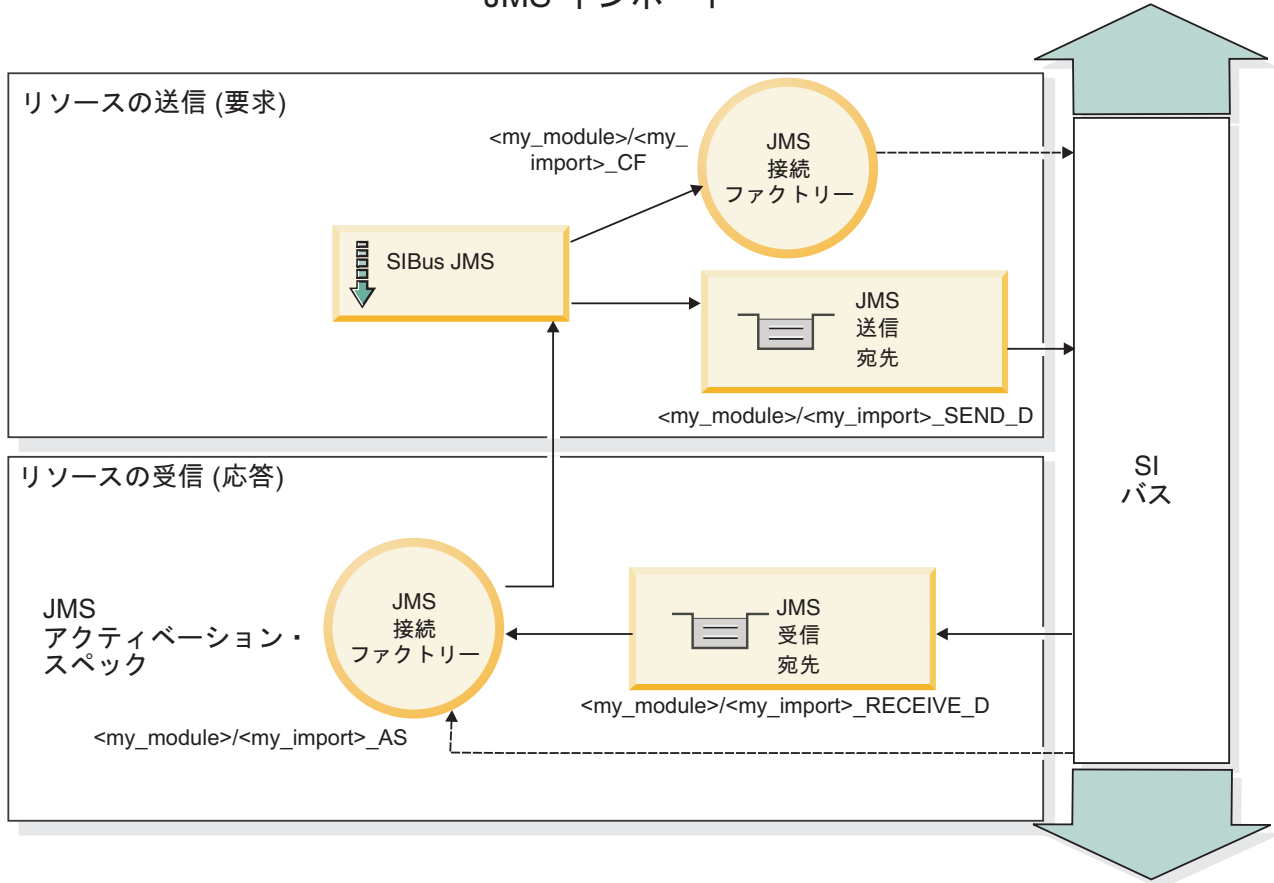


図7. JMS インポート・バインディングのリソース

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 エクスポートの接続部分は、構成可能なアクティベーション・スペックです。

JMS エクスポートには、send 宛先と receive 宛先があります。receive 宛先は、送信先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。send 宛先は、応答を送信する宛先です。ただし、着信メッセージで replyTo ヘッダー・プロパティを使用してこの宛先を置き換えている場合は、その宛先が優先されます。

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、呼び出されたアプリケーション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 replyTo 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。

184 ページの図 8 は、外部の要求側がどのようにエクスポートにリンクされているのかを示しています。

JMS エクスポート

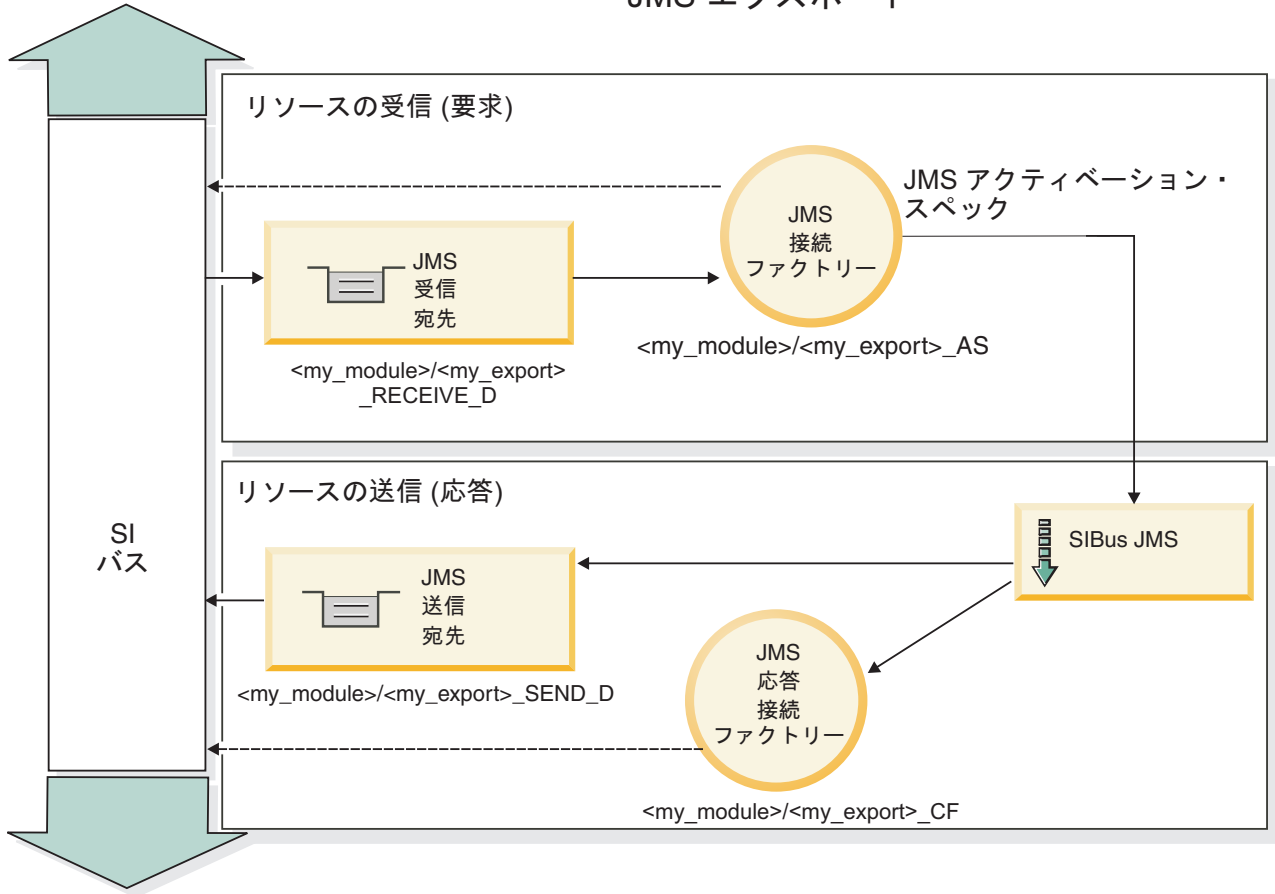


図8. JMS エクスポート・バインディングのリソース

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が、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、外部 JCA 1.5 準拠 JMS システムと統合する場合に使用します。JCA 1.5 に準拠した外部サービスは、SIB JMS リソース・アダプターを使用してメッセージを送受信でき、Service Component Architecture (SCA) コンポーネントと統合します。

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。

JMS バインディングの主な特徴

JMS のインポート・バインディングおよびエクスポート・バインディングの主な特徴は、ヘッダーと作成される J2EE リソースです。

EIS との相違点

インポート

JMS インポート・アプリケーションは、ランタイム呼び出しにおける応答の処理方法が EIS インポートとは異なります。JMS 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)、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 インポートのデプロイ時に、要求メッセージに対する応答を listen する目的で、ランタイム環境により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、要求と共に送信される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 replyTo ヘッダー・フィールドで指定されます。

エクスポート

JMS エクスポート・バインディングは、結果の戻りの処理方法が EIS エクスポート・バインディングとは異なります。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、次にその要求が SCA ランタイムで処理されるようにディスパッチします。

特殊ヘッダー

特殊ヘッダーのプロパティは、JMS インポートおよびエクスポートで使用され、ターゲットに対してメッセージの処理方法を指示します。

例えば、TargetFunctionName がネイティブ・メソッドから操作メソッドにマップされます。これは、デフォルトの関数セクター (JMSFunctionSelectorImpl) が JMS メッセージからネイティブ・メソッド名を抽出するときに使用されます。

J2EE リソース

JMS インポートおよびエクスポートを J2EE 環境にデプロイすると、さまざまな J2EE リソースが作成されます。

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

インポートでは、要求に対する応答を受信するときに使用されます。エクスポートでは、メッセージング・システムとの対話で MDB を表すメッセージ・エンドポイントを構成するときに使用されます。

宛先

- 送信宛先: インポートの場合は、要求または出力メッセージが送信される宛先になります。エクスポートの場合は、応答メッセージが送信される宛先にな

ります。ただし、着信メッセージの JMSReplyTo ヘッダー・フィールドにより置き換えられた場合は、その宛先が優先されます。

- 受信宛先: 着信メッセージが格納される宛先です。インポートの場合は応答になり、エクスポートの場合は要求になります。
- 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・タスクでは、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 が受信宛先で応答を listen できるようにします。これらのリソースのプロパティは、インポート・ファイルまたはエクスポート・ファイルに指定されます。

JMS バインディングの操作

JMS インポートおよびエクスポートは、サーバーで管理できます。

JMS インポートおよびエクスポートの構成および管理には、管理コンソールを使用します。

JMS インポートおよびエクスポートの詳細な作成手順は、WebSphere Integration Developer インフォメーション・センターの「**WebSphere Integration Developer**」 > 「**統合アプリケーションの開発 (Developing integration applications)**」 > 「**メッセージング・システムを使用した外部サービスへのアクセス (Accessing external services with messaging systems)**」 > 「**Java Message Service (JMS)**」 > 「**JMS バインディングの操作 (Working with JMS bindings)**」を参照してください。

JMS バインディングのパラメーター:

JMS のインポート・バインディングおよびエクスポート・バインディングは、すべての必要な接続ファクトリーが作成された状態でインストールできます。または、サーバー上で事前に構成された成果物のセットを指すように設計することもできます。

一般に、JMS インポート・バインディングおよび JMS エクスポート・バインディングは WebSphere Integration Developer で作成されます。開発時には、JMS バインディングに必要な接続と宛先をインストール時 (コンポーネントがサーバーにインストールされる時) に作成する方法を選択できます。もう 1 つの方法として、JMS インポートとエクスポートで使用する予定のサーバー上のリソースの 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 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。インポートおよびエクスポートのために生成される成果物の JNDI 名を以下の表に示します。

表 7. JMS インポート・バインディング: サーバー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

表 8. JMS エクスポート・バインディング: サーバー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・スベック	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。管理コンソールのスコープは、初期設定では「すべてのスコープ」に設定されています。新規リソースを作成するには、このスコープを「セル」または「ノード」に設定する必要があります。デフォルトのリストから既存のリソースを選択できます。

もう 1 つの方法を選択する場合は、バインディングが使用するリソースをサーバー上で検出することになるため、これらのリソースを事前にインストールし、インポート・ファイルとバインディング・ファイルで適切な JNDI 名を指定する必要があります。これにより、JMS バインディングとリソースの間の関連付けが行われます。

JMS バインディングの構成:

JMS インポート/エクスポート・バインディングを構成し、リソースの特殊機能を適用できます。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。

始める前に

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。

このタスクについて

JMS バインディングは、サーバー・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。

プロシージャ

1. SCA モジュールを選択します。管理コンソールから「アプリケーション」 → 「SCA モジュール」をクリックした後、*modulename* をクリックします。構成ページが開きます。
2. 「モジュール・コンポーネント」の下で、「インポート」または「エクスポート」を展開します。インストールされているインポートまたはエクスポートのリストが表示されます。
3. インポートまたはエクスポートを展開した後、「バインディング」を展開します。管理可能なバインディングのリストが表示されます。
4. 管理するバインディングをクリックします。バインディングの一般プロパティが表示されます。
 - 「送信リソース」カテゴリには、「接続ファクトリー」と「送信宛先」が含まれています。
 - 「受信リソース」カテゴリには、「応答接続ファクトリー」、「リスナー・ポート」、および「アクティベーション・スペック」が含まれています。
 - 「拡張リソース」カテゴリには、コールバック・リソースとその他の使用可能なリソースが含まれています。

注: リソースにアクセスするもう 1 つの方法として、テキスト・ボックスに JNDI 名を入力する方法があります。ただしこの方法では、まだ構成されていないリソースの名前の入力も可能になります。

5. 必要なリソースを管理します。
 - a. 「参照...」 ボタンをクリックすると、JNDI 名のリストを含むウィンドウが開きます。このウィンドウで、必要な JNDI 名を選択して「選択」をクリックします。
 - b. 「構成...」 ボタンをクリックすると、この JNDI 名によって参照される、対応するパネルが開きます。大部分のリソースはクラスター有効範囲で構成可能ですが、リスナー・ポートで「構成...」 オプションを選択すると、すべてのリスナー・ポートと、特定のクラスターのすべてのメンバーのクラスター・リスナー・ポート名を示すパネルが表示されます。このパネルでは、リスナー・ポートを 1 つ選択できます。

「構成...」を選択すると、対応するサーバー・パネルが開きます。

6. 設定を管理する準備ができている場合は、「適用」または「OK」をクリックします。

次のタスク

JNDI バインディングを反映するため、アプリケーションが自動的に再始動されません。

JMS および WebSphere MQ JMS ヘッダー

JMS ヘッダーおよび WebSphere MQ JMS ヘッダーは、JMS (または WebSphere MQ JMS) メッセージ・プロパティのすべてのプロパティを含んでいるサービス・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 ヘッダーおよび WebSphere MQ 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、ヘッダーを設定または取得するには、Service Component Architecture メッセージにアクセスします。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、ヘッダー値を取得したり、発信要求でヘッダー値を設定したりすることは可能です。

JMSMessage ヘッダーで設定できるプロパティを以下に示します。

- **JMSType** および **JMSCorrelationID** - 特定の事前定義メッセージ・ヘッダー・プロパティの値
- **JMSDeliveryMode** - 送達モードの値 (「永続的」または「非永続」。デフォルトは「永続的」)
- **JMSPriority** - 優先度の値 (0 から 9。デフォルトは JMS_Default_Priority)

JMS ヘッダーおよび WebSphere MQ JMS ヘッダーとメッセージ・プロパティが使用されるのは、基本 Service Component Architecture 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、JMS ヘッダー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。

関連概念

200 ページの『汎用 JMS ヘッダー』

汎用 JMS ヘッダーは、すべての汎用 JMS メッセージ・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。

219 ページの『WebSphere MQ ヘッダー』

WebSphere MQ ヘッダーには、Service Component Architecture (SCA) メッセージへの変換に関する特定の規則が組み込まれています。

外部クライアント

サーバーは、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 ポータルやエンタープライズ情報システムなど) は、サーバーの SCA モジュールにメッセージを送信できます。また、サーバー内からコンポーネントによって外部クライアントを呼び出すこともできます。

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受信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。送信フィールドで指定された宛先は、呼び出されたアプリケーション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。

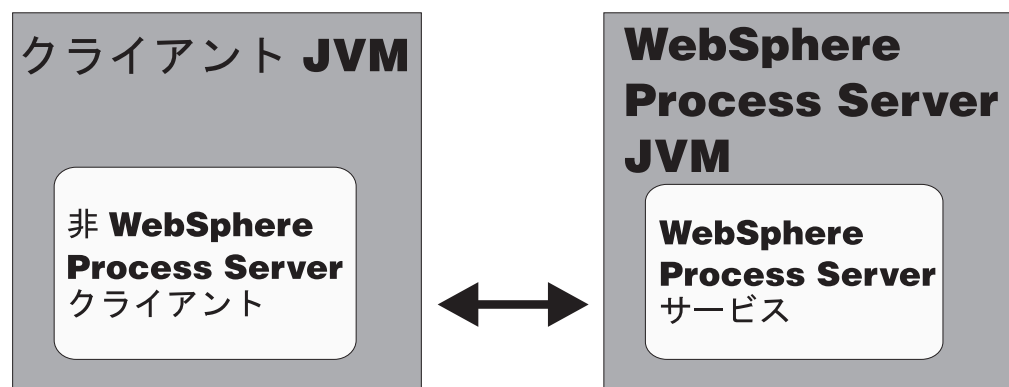
JMS インポートは外部クライアントにバインドし、外部クライアントにメッセージを配信できます。このメッセージは、外部クライアントからの応答を要求してもしなくても構いません。

外部クライアントの使用:

外部クライアント (サーバー外部のクライアント) が、サーバーにインストールされたアプリケーションと対話することが必要になる場合があります。

このタスクについて

外部クライアントがサーバー上の一般的なアプリケーションと対話する、非常に簡単なシナリオを考えましょう。以下の図は、標準的な単純なシナリオを表しています。



サービスの呼び出し

図9. 単純なユース・ケース・シナリオ: 外部クライアントとサーバー・アプリケーションの対話

アプリケーションは、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アプリケーションを使用できるようになります。

外部クライアントがサーバーとは別個の 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、このセクションの構成に関するトピックにリストされています。ただし、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。

プロシージャ

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します。
2. JNDI コンテキスト、SIB リソース・アダプターのポート、およびメッセージング・ブートストラッピング・ポートが正しいことを確認します。

サーバー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のポートが作成されます。管理コンソールを使用して、サーバー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」→ *your_server_name* → 「構成」を選択して、「通信 (Communication)」の下で「ポート」をクリックします。これで、使用する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て、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。

汎用 JMS バインディング

汎用 JMS インポート・バインディングとエクスポート・バインディングにより、サード・パーティーの JMS 1.1 準拠プロバイダーに対する接続が提供されます。このバインディングの操作は、JMS バインディングの操作と似ています。

JMS バインディングを介して提供されるサービスにより、Service Component Architecture (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、外部システムからメッセージを受信したりできます。このシステムとして、外部 JMS システムを使用できます。

汎用 JMS バインディング・アプリケーションは、JCA 1.5 非準拠 JMS プロバイダー (JMS プロバイダーが JMS 1.1 をサポートし、オプションの JMS Application Server Facility を実装している場合) との統合を実現します。汎用 JMS バインディングは、JCA 1.5 には対応していないが、JMS 1.1 仕様の Application Server Facility を実装している JMS プロバイダーをサポートしています。これには、Oracle AQ、TIBCO、SonicMQ、WebMethods、BEA WebLogic、WebSphere MQ などが含まれます。SIB は JCA 1.5 JMS プロバイダーであるため、サポートされていません。

ユーザーは、SCA 環境内で JCA 1.5 非準拠の JMS ベース・システムと統合するときにこの汎用アプリケーションを使用できます。これにより、ターゲットの外部アプリケーションがメッセージを送受信でき、SCA コンポーネントと統合できます。

関連タスク

240 ページの『イベントの順序付けの有効化: 汎用 JMS エクスポート』

汎用 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。

汎用 JMS バインディング: 全体像

汎用 JMS バインディングは、Service Component Architecture (SCA) 環境と外部 JMS システム (JMS 1.1 に準拠し、オプションの JMS Application Server Facility を実装している場合) の間の接続を提供する非 JCA JMS バインディングです。

汎用 JMS バインディング

汎用 JMS インポートおよび汎用 JMS エクスポートの両方のバインディングの主要なコンポーネントを以下に示します。

- 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・アプリケーション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。

汎用 JMS インポート・バインディング

汎用 JMS インポート・バインディングは、SCA アプリケーションから JCA 1.5 非準拠 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 インポートの接続部分は、接続ファクトリーです。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あり、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 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。

JMS インポート・バインディングでは、以下の 2 種類の使用シナリオがサポートされています。

- 片方向: インポート・ファイルで指定された send 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・ファイル内の send 宛先にメッセージを送信します。receive 宛先が ReplyTo ヘッダー・プロパティに設定されます。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。要求応答シナリオでは、応答は要求の `messageId` を応答メッセージの `correlationId` フィールドにコピーするか (デフォルト)、または要求の `correlationId` を応答メッセージの `correlationId` フィールドにコピーすることができます。

片方向と両方向のいずれのシナリオを使用する場合も、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。

図 10 は、インポートがどのように外部サービスにリンクされているのかを示しています。

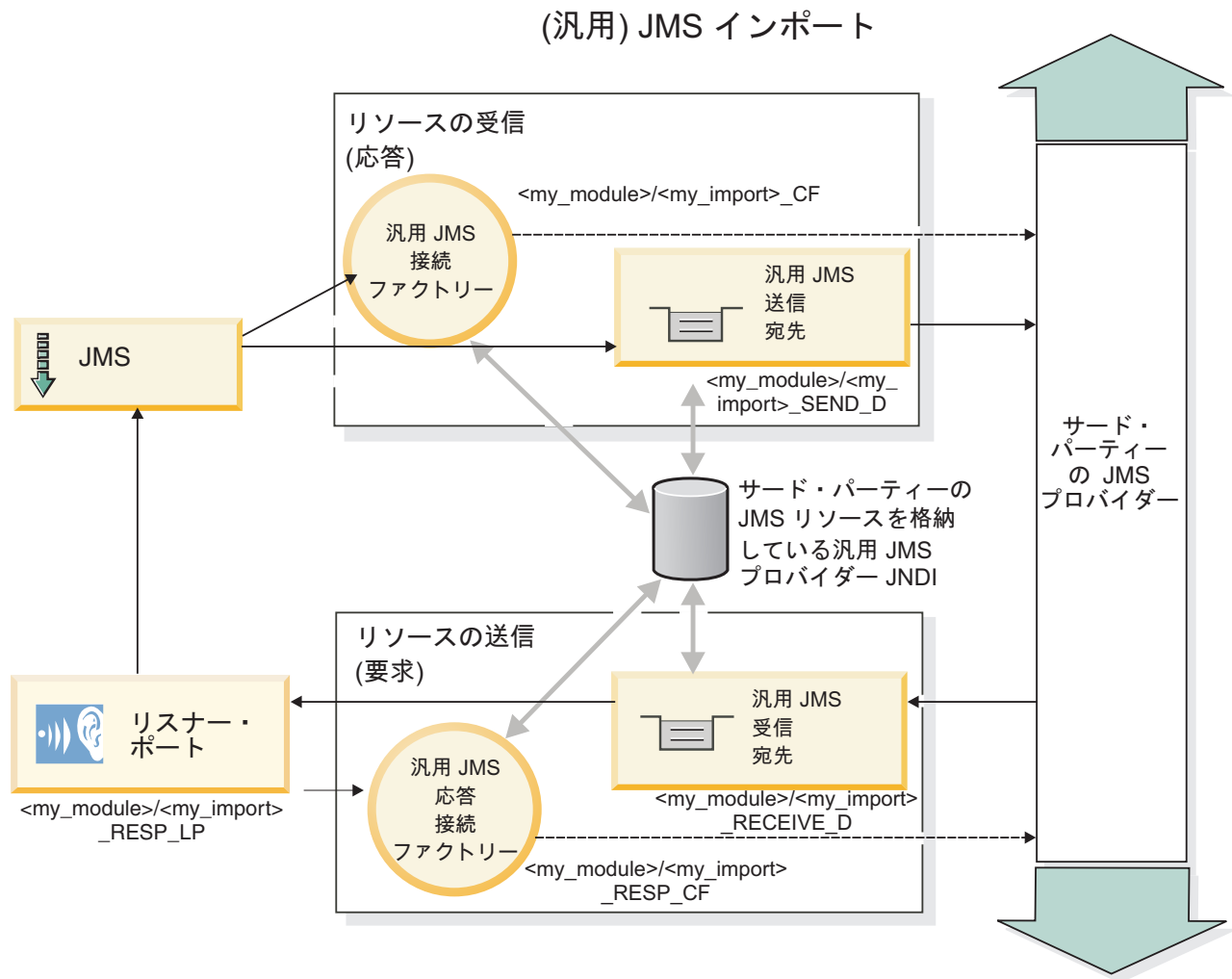


図 10. 汎用 JMS インポート・バインディング・リソース

汎用 JMS エクスポート・バインディング

汎用 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 エクスポートの接続部分は、ConnectionFactory および ListenerPort から構成されます。

汎用 JMS エクスポートには、send 宛先と receive 宛先があります。receive 宛先は、送信先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。send 宛先は、応答を送信する宛先です。ただし、着信メッセージで replyTo ヘッダー・プロパティを使用してこの宛先を置き換えている場合は、その宛先が優先されます。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、呼び出されたアプリケーション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 replyTo 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。
- 要求応答シナリオでは、応答は要求の messageId を応答メッセージの correlationId フィールドにコピーするか (デフォルト)、または要求の correlationId を応答メッセージの correlationId フィールドにコピーすることができます。

図 11 は、外部の要求側がどのようにエクスポートにリンクされているのかを示しています。

(汎用) JMS エクスポート

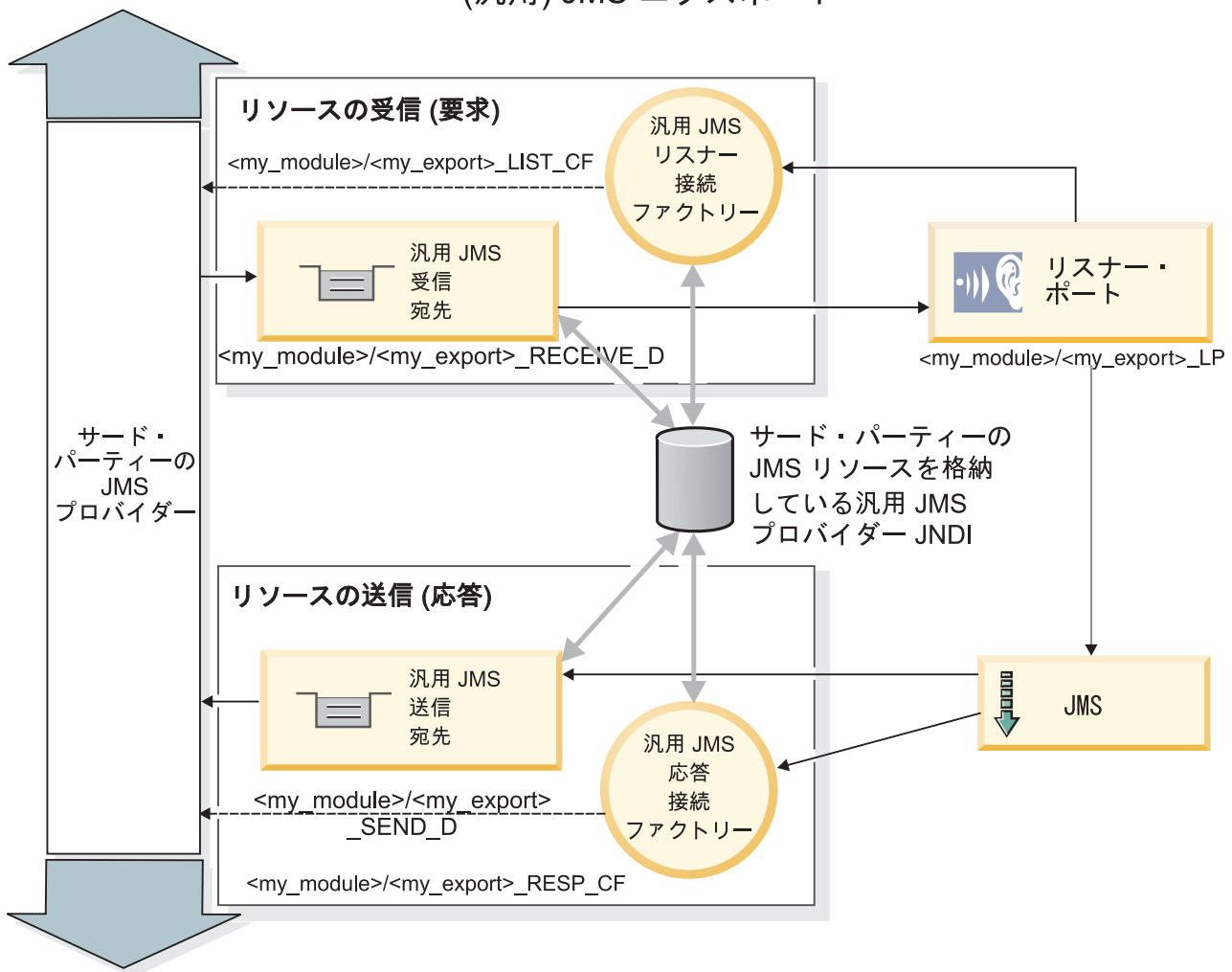


図 11. 汎用 JMS エクスポート・バインディング・リソース

汎用 JMS バインディングの主な特徴

汎用 JMS インポート・バインディングおよびエクスポート・バインディングの機能は、WebSphere 埋め込み JMS および MQ JMS インポート・バインディングの機能と整合性があります。主な機能は、ヘッダー定義および既存の J2EE リソースへのアクセスです。ただし、汎用という性質上、JMS プロバイダー固有の接続オプションはありません。また、このバインディングでは、デプロイメントおよびインストール時にリソースを生成できる機能が限定されています。

汎用インポート

MQ JMS インポート・アプリケーションと同様に、汎用 JMS 実装は非同期であり、3 つの呼び出し (片方向、両方向 (要求/応答ともいう)、およびコールバック) をサポートしています。

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る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 replyTo ヘッダー・フィールドで指定されます。

汎用エクスポート

汎用 JMS エクスポート・バインディングは、結果の戻りの処理方法が EIS エクスポート・バインディングとは異なります。汎用 JMS エクスポートでは、着信メッセージに指定された replyTo 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。

汎用 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 (汎用 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、次にその要求が SCA ランタイムで処理されるようにディスパッチします。

特殊ヘッダー

特殊ヘッダーのプロパティは、汎用インポートおよびエクスポートで使用され、ターゲット・バインディングに対してメッセージの処理方法を指示します。

例えば、TargetFunctionName プロパティは、デフォルトの関数セクターが、呼び出されるインポートまたはエクスポートのインターフェース内で操作の名前を識別するときに使用します。

J2EE リソース

JMS バインディングを 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。

- インポートの場合は受信宛先 (応答、両方向のみ)、エクスポートの場合は受信宛先 (要求) で listen するためのリスナー・ポート
- outboundConnection (インポート) および inboundConnection (エクスポート) の汎用 JMS 接続ファクトリー

- 送信宛先 (インポート) および受信宛先 (エクスポート、両方向のみ) の汎用 JMS 宛先
- responseConnection の汎用 JMS 接続ファクトリー (両方向のみ、オプション。これ以外の場合は、インポートに outboundConnection が使用され、エクスポートに inboundConnection が使用される)
- 受信宛先 (インポート) および送信宛先 (エクスポート) の汎用 JMS 宛先 (両方向のみ)
- SIB コールバック・キュー宛先にアクセスするときに使用されるデフォルトのメッセージング・プロバイダー・コールバック JMS 宛先 (両方向のみ)
- コールバック JMS 宛先にアクセスするときに使用されるデフォルトのメッセージング・プロバイダー・コールバック JMS 接続ファクトリー (両方向のみ)
- 応答処理中に使用される要求メッセージに関する情報を格納するための SIB コールバック・キュー宛先 (両方向のみ)

インストール・タスクにより、インポート・ファイルおよびエクスポート・ファイルの情報から ConnectionFactory、3 つの宛先、および ActivationSpec が作成されます。

汎用バインディングの操作

汎用 JMS バインディングはサーバーから管理できます。

汎用 JMS インポートおよびエクスポートの構成および管理には、管理コンソールを使用します。

汎用 JMS インポートおよびエクスポートの詳細な作成手順は、WebSphere Integration Developer インフォメーション・センターの「**WebSphere Integration Developer**」 > 「**統合アプリケーションの開発 (Developing integration applications)**」 > 「**メッセージング・システムを使用した外部サービスへのアクセス (Accessing external services with messaging systems)**」 > 「**汎用 JMS (Generic JMS)**」を参照してください。

汎用 JMS バインディングのパラメーター:

汎用 JMS のインポート・バインディングおよびエクスポート・バインディングは、すべての必要な接続ファクトリーが作成された状態でインストールできます。または、サーバー上で事前に構成された成果物のセットを指すように設計することもできます。

一般に、汎用 JMS バインディングは WebSphere Integration Developer で作成されます。開発時には、JMS バインディングに必要な接続と宛先を、コンポーネントがサーバーにインストールされるときに作成する方法を選択できます。または、JMS インポートまたはエクスポートで使用する予定のサーバー上のリソースの JNDI 名を指定できます。

このどちらの方法を選択するかによって、汎用 JMS 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 9. 汎用 JMS インポート: サーバー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	生成されたリソース JNDI 名
outboundConnection	[moduleName]/[importName]_CF
responseConnection	[moduleName]/[importName]_RESP_CF
送信宛先	[moduleName]/[importName]_SEND_D
受信宛先	[moduleName]/[importName]_RECEIVE_D
コールバック宛先	[moduleName]/[importName]_CALLBACK_D

表 10. 汎用 JMS エクスポート: サーバー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	生成されたリソース JNDI 名
inboundConnection	[moduleName]/[exportName]_LIS_CF
responseConnection	[moduleName]/[exportName]_RESP_CF
受信宛先	[moduleName]/[exportName]_RECEIVE_D
送信宛先	[moduleName]/[exportName]_SEND_D
コールバック宛先	[moduleName]/[exportName]_CALLBACK_D

注: リソースは、サーバー・スコープで作成されます。管理コンソールのスコープは、初期設定では「すべてのスコープ」に設定されています。新規リソースを作成するには、このスコープを「セル」または「ノード」に設定する必要があります。デフォルトのリストから既存のリソースを選択できます。

もう 1 つの方法を選択する場合は、JMS インポートが必要なリソースをサーバー上で検出することになるため、これらのリソースを事前にインストールし、インポート・ファイルおよびエクスポート・ファイルで JNDI 名を指定する必要があります。これにより、JMS バインディングとリソースの間の関連付けが行われます。

汎用 JMS バインディングの接続のセットアップ:

汎用 JMS バインディングを使用するには、サード・パーティー JMS プロバイダーとの接続をセットアップする必要があります。

始める前に

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。WebSphere Integration Developer および WebSphere Application Server で変更を行い、保存するための適切な権限が必要です。

このタスクについて

このタスクは手順の概要のみを示します。このトピックでは、個々のサード・パーティーの JMS プロバイダー固有の手順については説明しません。このシナリオのアプリケーションでは、汎用 JMS バインディングによりメディエーション・コン

ポーネントが両端で他のアプリケーションに接続されています。アプリケーションには、1つの両方向操作とのインターフェースが組み込まれています。

プロシージャー

1. サード・パーティー JMS プロバイダー固有のツールを使用して、キュー・マネージャー、キュー、および JMS 接続ファクトリーと宛先を作成するようにサード・パーティー JMS プロバイダーを構成します。
2. WebSphere Application Server で汎用メッセージング・プロバイダーを定義する必要があります。
3. WebSphere Integration Developer で、以下のタスクを実行する必要があります。
 - a. アプリケーションにエクスポートとインポートを追加し、以前に実装されたメディエーション・コンポーネントに接続します。
 - b. エクスポートとインポートの両方に汎用 JMS バインディングを追加します。
 - c. バインディング -> メッセージング・バインディング -> 汎用 JMS バインディングの順に生成します。
 - d. インポートとエクスポートの両方で、WebSphere Application Server で以前に定義されているプロパティと一致するように genericMessagingProviderName プロパティを設定します。
 - e. サード・パーティー JMS プロバイダーのツールでの定義と一致するように、接続と送信宛先/受信宛先の ExternalJNDIName を設定します。
4. アプリケーションを1つのサーバーにデプロイします。サード・パーティー JMS プロバイダーのキュー・マネージャーが稼働しており、接続に対して使用可能であること、および WebSphere Application Server で汎用メッセージング・プロバイダー定義が指し示すコンテキストが使用可能であることを確認します。WebSphere Integration Developer またはサービス・デプロイを使用して、アプリケーションを構築およびデプロイします。
5. アプリケーションを開始します。
6. アプリケーションを実行します。アプリケーションを実行するには、汎用 JMS エクスポート受信宛先で定義されているサード・パーティー JMS プロバイダー・キューにメッセージを格納します。応答が汎用 JMS エクスポート送信宛先に戻されます。同様に、アプリケーションから汎用 JMS インポート送信宛先に対して要求が発行されます。アプリケーションは、汎用 JMS インポート受信宛先に応答が戻されることを予期します。

汎用 JMS バインディングの構成:

汎用 JMS インポート・バインディングを管理し、リソースの特殊機能を構成できます。管理用タスクは、管理コンソールを使用して実行します。

始める前に

管理コンソールでプロファイルを変更し、変更内容を保存するための権限が必要です。また、接続セットアップ手順を完了している必要があります。

このタスクについて

汎用 JMS インポートは、サーバー・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。

プロシージャ

1. SCA モジュールを選択します。管理コンソールから「アプリケーション」 → 「SCA モジュール」を選択した後、*modulename* を選択します。構成ページが開きます。
2. 「モジュール・コンポーネント」の下で、「インポート」または「エクスポート」を展開します。インストールされているインポートまたはエクスポートのリストが表示されます。
3. インポートまたはエクスポートを展開した後、「バインディング」を展開します。管理可能なバインディングのリストが表示されます。
4. 管理するバインディングをクリックします。バインディングの一般プロパティが表示されます。
 - 「送信リソース」カテゴリには、「接続ファクトリー」と「送信宛先」が含まれています。
 - 「受信リソース」カテゴリには、「応答接続ファクトリー」、「リスナー・ポート」、および「アクティベーション・スペック」が含まれています。
 - 「拡張リソース」カテゴリには、コールバック・リソースとその他の使用可能なリソースが含まれています。

注: リソースにアクセスするもう 1 つの方法として、テキスト・ボックスに JNDI 名を入力する方法があります。ただしこの方法では、まだ構成されていないリソースの名前の入力も可能になります。

5. 必要なリソースを管理します。
 - a. 「参照...」 ボタンをクリックすると、JNDI 名のリストを含むウィンドウが開きます。このウィンドウで、必要な JNDI 名を選択します。選択された名前が、該当するテキスト・フィールドに取り込まれます。
 - b. 「構成...」 ボタンをクリックすると、この JNDI 名によって参照される、対応するパネルが開きます。大部分のリソースはクラスター有効範囲で構成可能ですが、リスナー・ポートで「構成...」 オプションを選択すると、すべてのリスナー・ポートと、特定のクラスターのすべてのメンバーのクラスター・リスナー・ポート名を示すパネルが表示されます。このパネルでは、リスナー・ポートを 1 つ選択できます。

「構成...」 を選択すると、対応する WebSphere Application Server パネルが開きます。

6. 設定を管理する準備ができている場合は、「適用」または「OK」をクリックします。

次のタスク

JNDI バインディングを反映するため、アプリケーションが自動的に再始動されます。

汎用 JMS ヘッダー

汎用 JMS ヘッダーは、すべての汎用 JMS メッセージ・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。

このヘッダーは、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、ヘッダーを設定または取得するには、Service Component Architecture メッセージにアクセスします。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、ヘッダー値を取得したり、発信要求でヘッダー値を設定したりすることは可能です。

methodBinding で静的に設定されるプロパティを以下に示します。

- JMSType
- JMSCorrelationID
- JMSDeliveryMode
- JMSPriority

汎用 JMS バインディングでは、JMS バインディングおよび MQJMS バインディングと同様の方法で、JMS ヘッダーとプロパティの動的変更もサポートされています。

一部の汎用 JMS プロバイダーでは、アプリケーションにより設定できるプロパティとその組み合わせが制限されます。詳しくは、サード・パーティー製品の資料を参照してください。ただし、methodBinding には、新しいプロパティ ignoreInvalidOutboundJMSProperties (例外の伝搬を許可するプロパティ) が追加されています。

汎用 JMS ヘッダーとメッセージ・プロパティが使用されるのは、基本 Service Component Architecture 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。

関連概念

189 ページの『JMS および WebSphere MQ JMS ヘッダー』

JMS ヘッダーおよび WebSphere MQ JMS ヘッダーは、JMS (または WebSphere MQ JMS) メッセージ・プロパティのすべてのプロパティを含んでいるサービス・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。

219 ページの『WebSphere MQ ヘッダー』

WebSphere MQ ヘッダーには、Service Component Architecture (SCA) メッセージへの変換に関する特定の規則が組み込まれています。

WebSphere MQ JMS バインディング

WebSphere MQ JMS バインディングは、WebSphere MQ JMS ベースの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・アプリケーションは、サーバー環境から外部 JMS または MQ JMS 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 からの インポートでは、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 への エクスポートでは、要求の受信と応答の送信が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します。

関連概念

210 ページの『WebSphere MQ バインディング』

WebSphere MQ アプリケーションは、WebSphere MQ アプリケーションと Service Component Architecture との接続を提供します。

関連タスク

239 ページの『イベントの順序付けの有効化: WebSphere MQ JMS エクスポート』

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 バインディング: 全体像

WebSphere MQ JMS アプリケーションは、JMS API を使用した WebSphere MQ との統合を提供します。

WebSphere MQ 管理用タスク

WebSphere MQ システム管理者は、基盤となる WebSphere MQ キュー・マネージャーを作成する必要があります。このキュー・マネージャーは、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、Service Component Architecture (SCA) アプリケーションから WebSphere MQ JMS ベースのプロバイダーへのアウトバウンド接続を提供します。サポートされているバージョンの

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。

WebSphere MQ JMS インポート・バインディングでは、以下の 2 種類の使用シナリオがサポートされています。

- 片方向: インポート・ファイル内の send 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・ファイル内の send 宛先にメッセージを送信します。receive 宛先が replyTo ヘッダー・フィールドに設定されます。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のシナリオを使用する場合も、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。

図 12 は、インポートがどのように外部サービスにリンクされているのかを示しています。

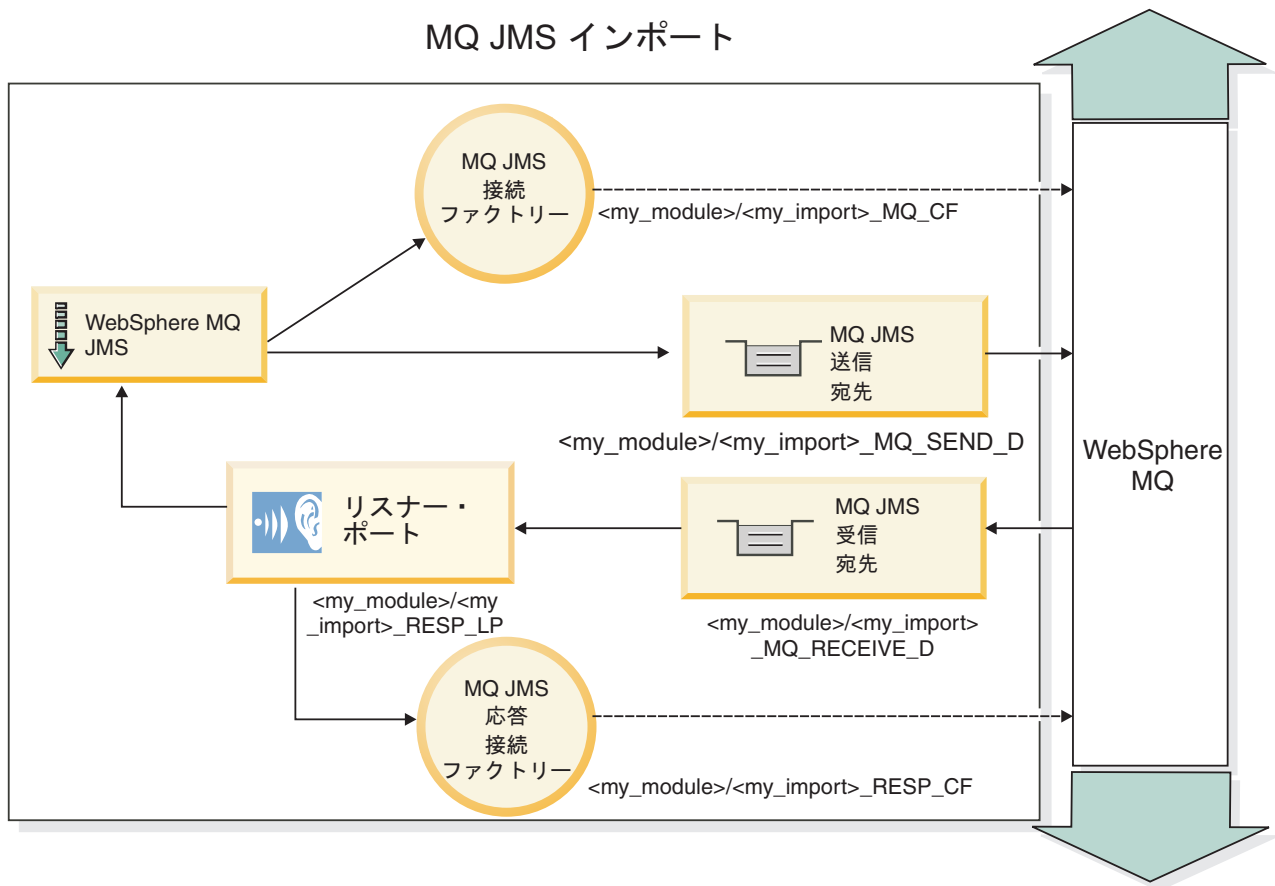


図 12. WebSphere MQ JMS インポート・バインディングのリソース

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、呼び出されたアプリケーション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 replyTo 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。

図 13 は、外部の要求側がどのようにエクスポートにリンクされているのかを示しています。

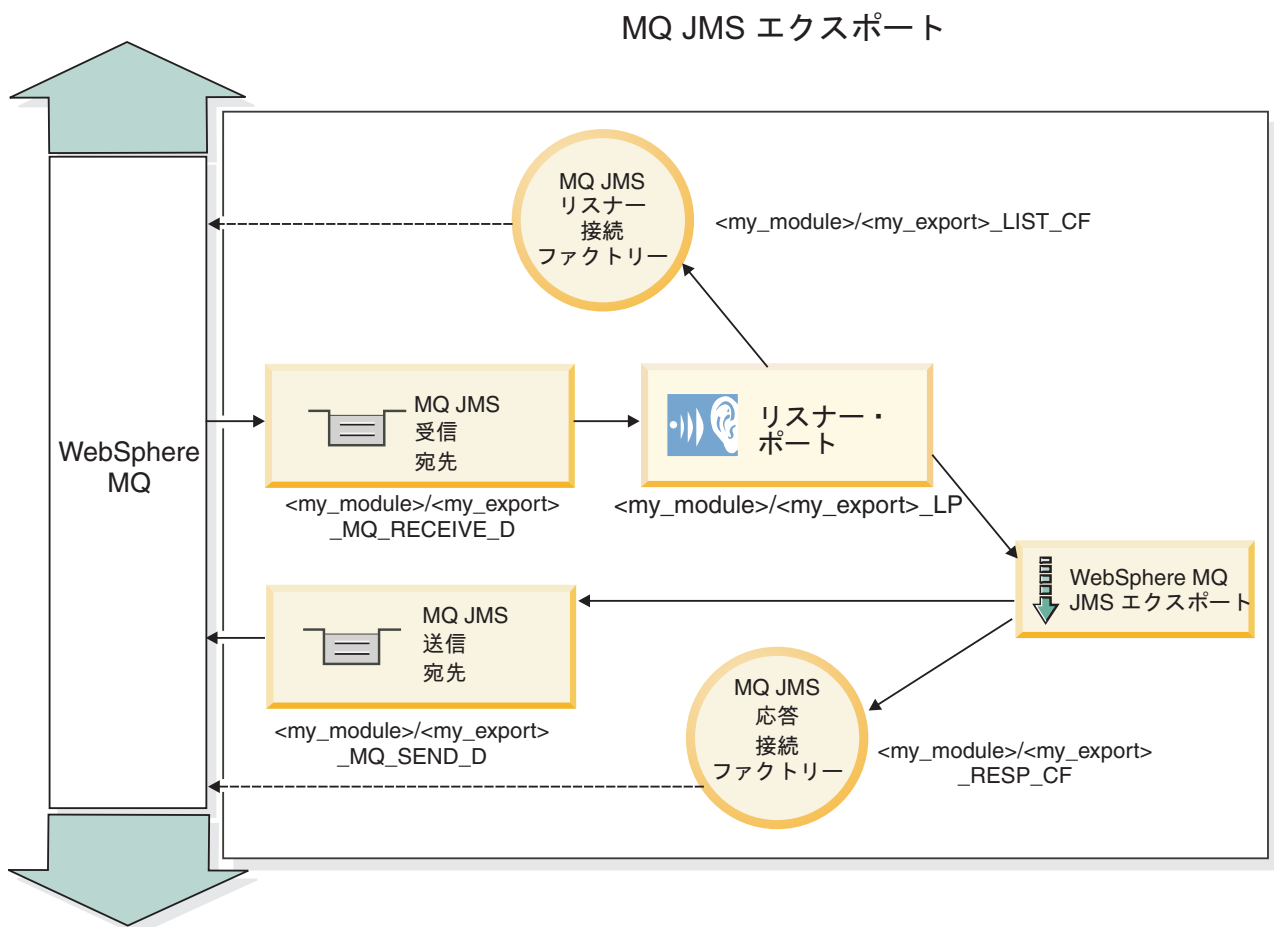


図 13. WebSphere MQ JMS エクスポート・バインディングのリソース

WebSphere MQ JMS バインディング・サポート

WebSphere MQ JMS アプリケーションは、WebSphere MQ JMS ベースの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・アプリケーションは、J2EE プログラミング環境内にある WebSphere MQ JMS ベースのシステムと直接統合したい場合に使用します。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 JMS からの インポートでは、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 JMS への エクスポートでは、要求の受信と応答の送信が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。

関連概念

212 ページの『WebSphere MQ バインディング・サポート』

WebSphere MQ アプリケーションにより、ネイティブ MQ ベースのアプリケーションとの統合が実現します。

WebSphere MQ JMS バインディングの主な特徴

WebSphere MQ JMS バインディングの主な特徴は、ヘッダー、J2EE 成果物、および作成される J2EE リソースです。

ヘッダー

JMS メッセージ・ヘッダーには、いくつかの定義済みフィールドがあります。これらのフィールドには、クライアントとプロバイダーの両方でメッセージの識別と送付に使用される値が格納されます。

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。ストリング 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 を使用する場合は、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 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。

RequestMsgIDToCorrelID

JMSMessageID は JMSCorrelationID フィールドにコピーされます。これはデフォルト設定です。

RequestCorrelIDToCorrelID

JMSCorrelationID は JMSCorrelationID フィールドにコピーされます。

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:
 - インポート: 要求または出力メッセージが送信される宛先です。
 - エクスポート: 応答メッセージが送信される宛先です。ただし、着信メッセージの JMSReplyTo ヘッダー・フィールドにより置き換えられた場合は、その宛先が優先されます。
- 受信宛先:
 - インポート: 応答メッセージまたは着信メッセージが格納される宛先です。
 - エクスポート: 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。

WebSphere MQ JMS バインディングの操作

WebSphere MQ JMS バインディングは、WebSphere Process Server で管理できません。

WebSphere MQ JMS バインディングにアクセスするには、管理コンソールを使用します。

WebSphere MQ JMS インポートおよびエクスポートを生成する方法の詳細については、WebSphere Integration Developer インフォメーション・センター [WID]で『MQ JMS バインディングの操作 (Working with MQ JMS bindings)』を参照してください。

関連概念

215 ページの『WebSphere MQ バインディングの操作』
 WebSphere MQ バインディングはサーバー内から管理できます。

関連タスク

239 ページの『イベントの順序付けの有効化: WebSphere MQ JMS エクスポート』

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。

MQ JMS バインディングのパラメーター:

MQ JMS バインディングは、すべての必要な接続ファクトリーが作成された状態でインストールできます。または、サーバー上で事前に構成された成果物のセットを指すように設計することもできます。

一般に、MQ JMS バインディングは WebSphere Integration Developer で作成されます。開発時には、MQ JMS バインディングに必要な接続と宛先をインストール時 (コンポーネントがサーバーにインストールされる時) に作成する方法を選択できます。あるいは、MQ JMS バインディングで使用する予定のサーバー上のリソースの JNDI 名を指定できます。

このどちらの方法を選択するかによって、MQ JMS 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. MQ JMS インポート: サーバー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

表 12. MQ JMS エクスポート: サーバー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	mqjms.module	my/export	mqjms.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	mqjms.module	my/export	mqjms.module/my/export_LIS_CF
応答接続ファクトリー	mqjms.module	my/export	mqjms.module/my/export_RESP_CF
受信	mqjms.module	my/export	mqjms.module/my/export_MQ_RECEIVE_D
送信	mqjms.module	my/export	mqjms.module/my/export_MQ_SEND_D
SIB コールバック宛先	mqjms.module	my/export	mqjms.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。
- 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 つの方法を選択する場合は、MQ JMS インポートまたはエクスポートが使用するリソースをサーバー上で検出することになるため、これらのリソースを事前にインストールし、インポート・ファイルで JNDI 名を指定する必要があります。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 バインディングの構成:

MQ JMS バインディングを管理し、リソースの特殊機能を構成できます。管理用タスクは、管理コンソールを使用して実行します。

始める前に

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ません。WebSphere MQ 管理者が WebSphere MQ 内で作成する必要があります。

このタスクについて

MQ JMS インポートまたはエクスポートは、サーバー・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。

プロシージャ

1. SCA モジュールを選択します。管理コンソールから「アプリケーション」→「SCA モジュール」をクリックした後、*modulename* をクリックします。構成ページが開きます。
2. 「モジュール・コンポーネント」の下で、「インポート」または「エクスポート」を展開します。インストールされているインポートまたはエクスポートのリストが表示されます。
3. インポートまたはエクスポートを展開した後、「バインディング」を展開します。管理可能なバインディングのリストが表示されます。
4. 管理するバインディングをクリックします。バインディングの一般プロパティが表示されます。
 - 「送信リソース」カテゴリには、「接続ファクトリー」と「送信宛先」が含まれています。
 - 「受信リソース」カテゴリには、「応答接続ファクトリー」、「リスナー・ポート」、および「アクティベーション・スペック」が含まれています。
 - 「拡張リソース」カテゴリには、コールバック・リソースとその他の使用可能なリソースが含まれています。

注: リソースにアクセスするもう 1 つの方法として、テキスト・ボックスに JNDI 名を入力する方法があります。ただしこの方法では、まだ構成されていないリソースの名前の入力も可能になります。

5. 必要なリソースを管理します。
 - a. 「参照...」 ボタンをクリックすると、JNDI 名のリストを含むウィンドウが開きます。このウィンドウで、必要な JNDI 名を選択します。選択された名前が、該当するテキスト・フィールドに取り込まれます。
 - b. 「構成...」 ボタンをクリックすると、この JNDI 名によって参照される、対応するパネルが開きます。大部分のリソースはクラスター有効範囲で構成可能ですが、リスナー・ポートで「構成...」 オプションを選択すると、すべてのリスナー・ポートと、特定のクラスターのすべてのメンバーのクラスター・リスナー・ポート名を示すパネルが表示されます。このパネルでは、リスナー・ポートを 1 つ選択できます。

「構成...」 を選択すると、対応する WebSphere Application Server パネルが開きます。

6. 設定を管理する準備ができている場合は、「適用」または「OK」をクリックします。

次のタスク

JNDI バインディングを反映するため、アプリケーションが自動的に再始動されます。

JMS および WebSphere MQ JMS ヘッダー

JMS ヘッダーおよび WebSphere MQ JMS ヘッダーは、JMS (または WebSphere MQ JMS) メッセージ・プロパティのすべてのプロパティを含んでいるサービス・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 ヘッダーおよび WebSphere MQ 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、ヘッダーを設定または取得するには、Service Component Architecture メッセージにアクセスします。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、ヘッダー値を取得したり、発信要求でヘッダー値を設定したりすることは可能です。

JMSMessage ヘッダーで設定できるプロパティを以下に示します。

- **JMSType** および **JMSCorrelationID** - 特定の事前定義メッセージ・ヘッダー・プロパティの値
- **JMSDeliveryMode** - 送達モードの値 (「永続的」または「非永続」。デフォルトは「永続的」)
- **JMSPriority** - 優先度の値 (0 から 9。デフォルトは JMS_Default_Priority)

JMS ヘッダーおよび WebSphere MQ JMS ヘッダーとメッセージ・プロパティが使用されるのは、基本 Service Component Architecture SCDDL 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、JMS ヘッダー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" contextPropagationEnabled="true">
```

デフォルトは true です。

関連概念

200 ページの『汎用 JMS ヘッダー』

汎用 JMS ヘッダーは、すべての汎用 JMS メッセージ・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。

219 ページの『WebSphere MQ ヘッダー』

WebSphere MQ ヘッダーには、Service Component Architecture (SCA) メッセージへの変換に関する特定の規則が組み込まれています。

外部クライアント

サーバーは、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、サーバーの SCA モジュールにメッセージを送信する必要があり、サーバー内からコンポーネントによって呼び出される必要もあります。

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、呼び出されたアプリケーション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。

WebSphere MQ バイディング

WebSphere MQ アプリケーションは、WebSphere MQ アプリケーションと Service Component Architecture との接続を提供します。

WebSphere MQ プロバイダー・アプリケーションは、サーバー環境から 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 からの インポートでは、データの送信先の宛先と応答を受信できる宛先が使用されます。データと MQ メッセージの間の相互変換は、MQ ヘッダー・バイディングおよび本体データ・バイディングを使用して行われます。

WebSphere MQ への エクスポートでは、要求の受信と応答の送信ができる宛先を使用します。データと MQ メッセージの間の相互変換は、MQ ヘッダー・バイディングおよび本体データ・バイディングを使用して行われます。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します。

関連概念

201 ページの『WebSphere MQ JMS バイディング』

WebSphere MQ JMS バイディングは、WebSphere MQ JMS ベースのプロバイダーとの統合を提供します。

WebSphere MQ バイディング: 全体像

WebSphere MQ アプリケーションにより、ネイティブ MQ ベースのアプリケーションとの統合が実現します。

WebSphere MQ 管理用タスク

WebSphere MQ システム管理者は、基盤となる WebSphere MQ キュー・マネージャーを作成する必要があります。このキュー・マネージャーは、WebSphere MQ バ

インディンク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 インポート・バインディング

WebSphere MQ インポート・バインディングは、Service Component Architecture (SCA) アプリケーションから WebSphere MQ ベースのアプリケーションへのアウトバウンド接続を提供します。サポートされているバージョンの 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 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。MQMD 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が replyTo MQMD ヘッダー・フィールドに設定されます。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。

図 14 は、インポートがどのように外部サービスにリンクされているのかを示しています。

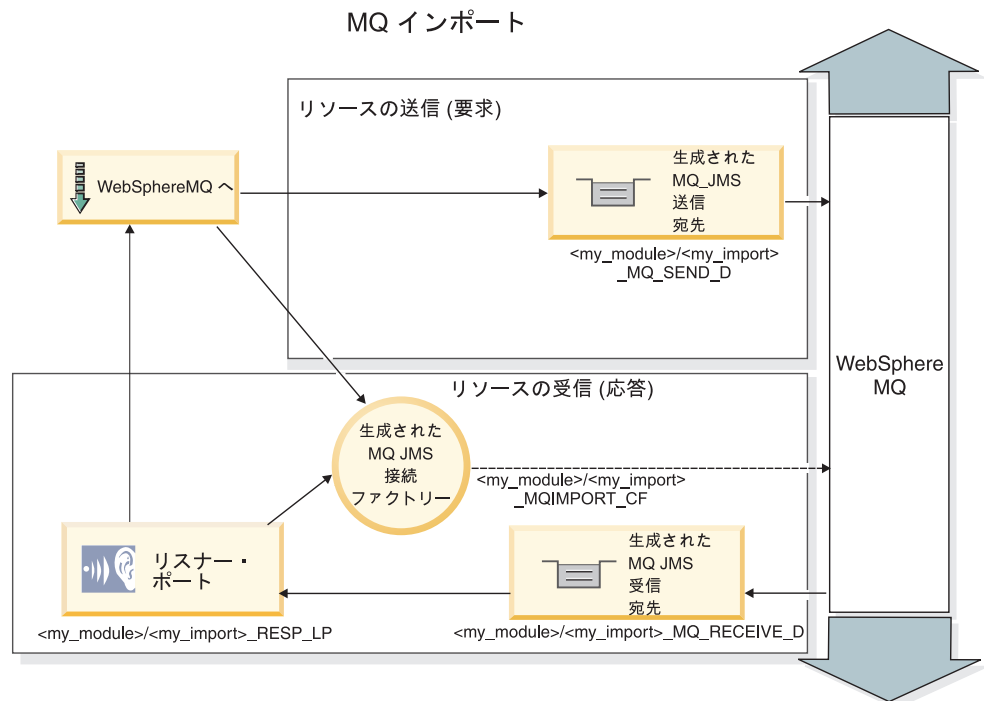


図 14. WebSphere MQ インポート・バインディング・リソース

WebSphere MQ エクスポート・バインディング

WebSphere MQ エクスポート・バインディングは、WebSphere MQ ベースのアプリケーションから 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、呼び出されたアプリケーション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 replyTo 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。

図 15 は、外部の要求側がどのようにエクスポートにリンクされているのかを示しています。

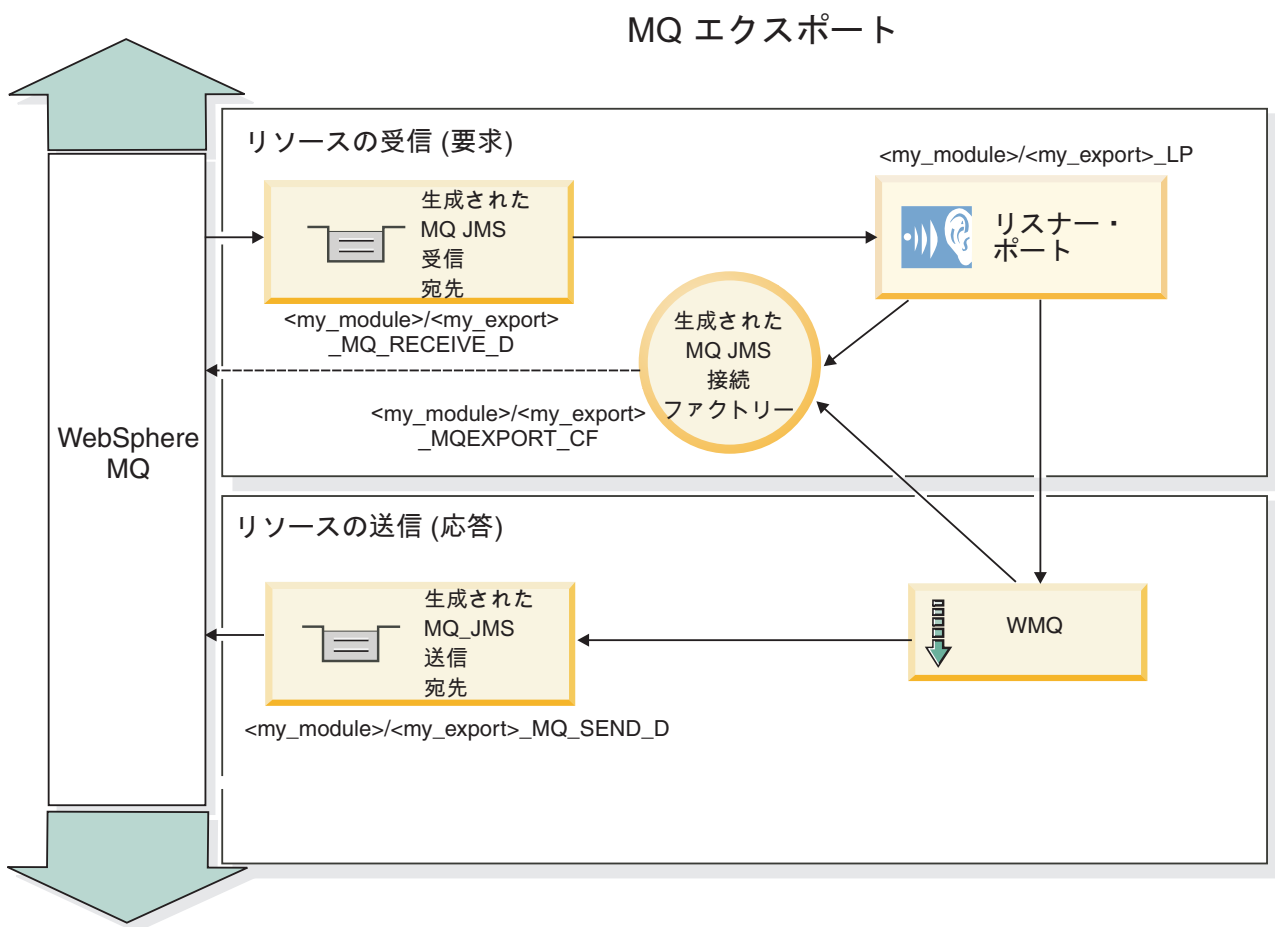


図 15. WebSphere MQ エクスポート・バインディング・リソース

WebSphere MQ バインディング・サポート

WebSphere MQ アプリケーションにより、ネイティブ MQ ベースのアプリケーションとの統合が実現します。

WebSphere MQ アプリケーションは、J2EE プログラミング環境内にあるネイティブ 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 からの インポートでは、データの送信先の宛先と応答を受信できる宛先が使用されます。データと MQ メッセージの間の相互変換は、MQ ヘッダー・バインディングおよび本体データ・バインディングを使用して行います。

WebSphere MQ への エクスポートでは、要求の受信と応答の送信ができる宛先を使用します。データと MQ メッセージの間の相互変換は、MQ ヘッダー・バインディングおよび本体データ・バインディングを使用して行います。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 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、サーバー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

203 ページの『WebSphere MQ JMS バインディング・サポート』

WebSphere MQ JMS アプリケーションは、WebSphere MQ JMS ベースのプロバイダーとの統合を提供します。

WebSphere MQ バインディングの主な特徴

WebSphere MQ バインディングの主な特徴は、ヘッダー、J2EE 成果物、および作成される J2EE リソースです。

関連スキーム

WebSphere MQ 要求/応答アプリケーションでは、応答メッセージと要求を関連付けるさまざまな手法の 1 つを使用できます。これらの手法には、MQMD の MessageID フィールドと CorrelID フィールドが関連します。ほとんどの場合、要求側はキュー・マネージャーに MessageID を選択させ、応答アプリケーションがこれを応答の CorrelID にコピーすることを想定します。多くの場合、要求側と応答アプリケーションは、使用される関連手法を暗黙的に認識しています。応答アプリケーションは、要求の Report フィールドに設定されているさまざまなフラグの指示に従って、これらのフィールドを処理することもあります。

WebSphere MQ メッセージのエクスポート・バインディングを構成するときには、以下のオプションを使用できます。

応答の MsgId のオプション:

新規 MsgID (New MsgID)

キュー・マネージャーが応答の固有の MsgId を選択できるようにします (デフォルト)。

要求の MsgID からコピーする (Copy from Request MsgID)

要求の MsgId フィールドから MsgId フィールドをコピーします。

SCA メッセージからコピーする (Copy from SCA message)

SCA 応答メッセージの WebSphere MQ ヘッダーに含まれる値を MsgId に設定します。この値が存在しない場合は、キュー・マネージャーに新規 Id を定義させます。

Report オプションを使用 (As Report Option)

要求の MQMD の Report フィールドを検査して、MsgId の処理方法を認識します。MQRO_NEW_MSG_ID オプションと MQRO_PASS_MSG_ID オプションがサポートされています。これらのオプションは、それぞれ「新規 MsgID (New MsgID)」および「要求の MsgID からコピーする (Copy from Request MsgID)」と同様に動作します。

応答の CorrelId のオプション:

要求の MsgID からコピーする (Copy from Request MsgID)

要求の MsgId フィールドから CorrelId フィールドをコピーします (デフォルト)。

要求の CorrelID からコピーする (Copy from Request CorrelID)

要求の CorrelId フィールドから CorrelId フィールドをコピーします。

SCA メッセージからコピーする (Copy from SCA message)

SCA 応答メッセージの WebSphere MQ ヘッダーに含まれる値を CorrelId に設定します。この値が存在しない場合は、空白のままにします。

Report オプションを使用 (As Report Option)

要求の MQMD の Report フィールドを検査して、CorrelId の処理方法を認識します。MQRO_COPY_MSG_ID_TO_CORREL_ID オプションと MQRO_PASS_CORREL_ID オプションがサポートされています。これらのオプションは、それぞれ「要求の MsgID からコピーする (Copy from Request MsgID)」および「要求の CorrelID からコピーする (Copy from Request CorrelID)」と同様に動作します。

WebSphere MQ メッセージのインポート・バインディングを構成するときには、以下のオプションを使用できます。

要求 の MsgId のオプション:

新規 MsgID (New MsgID)

キュー・マネージャーが要求の固有の MsgId を選択できるようにします (デフォルト)。

SCA メッセージからコピーする (Copy from SCA message)

SCA 要求メッセージの WebSphere MQ ヘッダーに含まれる値を MsgId に設定します。この値が存在しない場合は、キュー・マネージャーに新規 Id を定義させます。

応答関連のオプション:

応答の CorrelID を MsgId からコピーする (Response has CorrelID copied from MsgId)

応答メッセージの CorrelId フィールドが要求の MsgId に基づいて設定されていることを想定します (デフォルト)。

応答の MsgID を MsgId からコピーする (Response has MsgID copied from MsgId)

応答メッセージの MsgId フィールドが要求の MsgId に基づいて設定されていることを想定します。

応答の CorrelID を CorrelId からコピーする (Response has CorrelID copied from CorrelId)

応答メッセージの CorrelId フィールドが要求の CorrelId に基づいて設定されていることを想定します。

J2EE リソース

WebSphere MQ バインディングを 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 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: インポートの場合は、要求または出力メッセージが送信される宛先です。エクスポートの場合は、応答メッセージが送信される宛先です。ただし、着信メッセージの MQMD ReplyTo ヘッダー・フィールドにより置き換えられた場合は、その宛先が優先されます。
- 受信宛先: 応答/要求または着信メッセージが格納される宛先です。

WebSphere MQ バインディングの操作

WebSphere MQ バインディングはサーバー内から管理できます。

WebSphere MQ バインディングにアクセスするには、管理コンソールを使用します。

WebSphere MQ インポートおよびエクスポートの手順については、WebSphere Integration Developer インフォメーション・センターの「**WebSphere Integration Developer**」 > 「**統合アプリケーションの開発 (Developing integration applications)**」 > 「**メッセージング・システムを使用した外部サービスへのアクセス (Accessing external services with messaging systems)**」 > 「**WebSphere MQ**」 > 「**MQ バインディングの操作 (Working with MQ bindings)**」を参照してください。

関連概念

205 ページの『WebSphere MQ JMS バインディングの操作』

WebSphere MQ JMS バインディングは、WebSphere Process Server で管理できます。

関連タスク

239 ページの『イベントの順序付けの有効化: WebSphere MQ エクスポート』

WebSphere MQ 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 バインディングのパラメーター:

WebSphere MQ バインディングは、すべての必要な接続ファクトリーが作成された状態でインストールできます。または、サーバー上で事前に構成された成果物のセットを指すように設計することもできます。

一般に、WebSphere MQ インポート・バインディングおよび WebSphere MQ エクスポート・バインディングは WebSphere Integration Developer で作成されます。開発時には、WebSphere MQ インポートに必要な接続と宛先をインストール時 (コンポーネントがサーバーにインストールされる時) に作成する方法を選択できます。あるいは、WebSphere MQ バインディングで使用する予定のサーバー上のリソースの JNDI 名を指定できます。

このどちらの方法を選択するかによって、WebSphere MQ 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. WebSphere MQ インポート: サーバー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	mq.module	my/import	mq.module/my/import_MQ_CF
応答リスナー・ポート	mq.module	my/import	mq.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	mq.module	my/import	mq.module/my/import_RESP_CF
送信	mq.module	my/import	mq.module/my/import_MQ_SEND_D
受信	mq.module	my/import	mq.module/my/export_MQ_RECEIVE_D
SIB コールバック宛先	mq.module	my/import	mq.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.MQ/Callback_CF

表 14. WebSphere MQ エクスポート: サーバー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	mq.module	my/export	mq.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	mq.module	my/export	mq.module/my/export_LIS_CF
応答接続ファクトリー	mq.module	my/export	mq.module/my/export_RESP_CF
受信	mq.module	my/export	mq.module/my/export_MQ_RECEIVE_D
送信	mq.module	my/export	mq.module/my/export_MQ_SEND_D
SIB コールバック宛先	mq.module	my/export	mq.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.MQ/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。
- 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、WebSphere 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 つの方法を選択する場合は、WebSphere MQ バインディングが使用するリソースをサーバー上で検出することになるため、これらのリソースを事前にインストールし、インポート・ファイルまたはエクスポート・ファイルで JNDI 名を指定する必要があります。これにより、WebSphere MQ バインディングとリソースの間の関連付けが行われます。

WebSphere MQ バインディングの構成:

WebSphere MQ インポート・バインディングとエクスポート・バインディングを管理して、リソースの特殊機能を調整したり設定したりすることができます。管理用タスクは、管理コンソールを使用して実行します。

始める前に

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。

このタスクについて

WebSphere MQ インポートは、サーバー・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。

プロシージャー

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 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。

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、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 用の関連するキュー接続ファクトリーによって作成されます。

5. 変更内容をマスター・プロファイルに保管し、必要に応じてサーバーを再始動します。

WebSphere MQ ヘッダー

WebSphere MQ ヘッダーには、Service Component Architecture (SCA) メッセージへの変換に関する特定の規則が組み込まれています。

SCA モデルにおける WebSphere MQ ヘッダー表現には、以下の規則が組み込まれています。

- WebSphere MQ メッセージは、システム・ヘッダー (MQMD) とメッセージ本体で構成されます。
- メッセージ本体には、0 または 1 つ以上のメッセージ・ヘッダーを組み込むことができます。メッセージ内に複数のメッセージ・ヘッダーがある場合、ヘッダーの順序が意味を持ちます。
- 各ヘッダーには、次のヘッダー (最終ヘッダーの場合は本体) の構造を記述する情報が含まれています。MQMD は、1 番目のヘッダー (メッセージ・ヘッダーがない場合は本体) を記述します。

WebSphere MQ ヘッダーで使用される属性を以下に示します。

- MQMD は、WebSphere MQ メッセージ記述の内容を表します。ただし、本体の構造とエンコードを定義する情報は含まれません。
- MQControl には、メッセージ・ヘッダーまたは本体の構造とエンコードを定義する情報が含まれています。
- MQHeader は MQControl を拡張するヘッダー・データのコンテナですが、anyType の値エレメントが含まれています。
- MQHeaders には、MQHeader オブジェクトのリストが含まれています。

次の 3 つの SCA メッセージ・ヘッダーが定義されています。

- MQMD DataObject が含まれている SCA ヘッダー。
- MQControl DataObject が含まれている SCA ヘッダー。メッセージ本体の内容の構造とエンコードを記述します。
- MQHeaders DataObject (MQHeader DataObject のリスト) が含まれている SCA ヘッダー (オプション)。

関連概念

189 ページの『JMS および WebSphere MQ JMS ヘッダー』

JMS ヘッダーおよび WebSphere MQ JMS ヘッダーは、JMS (または WebSphere MQ JMS) メッセージ・プロパティのすべてのプロパティを含んでいるサー

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。

200 ページの『汎用 JMS ヘッダー』

汎用 JMS ヘッダーは、すべての汎用 JMS メッセージ・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。

Service Component Architecture モジュールと WebSphere MQ

SCA モジュールと WebSphere MQ キューを接続して、相互にサービスを提供することができます。

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。モジュールを作成するときは、SCA モジュールと WebSphere MQ キュー間の接続を定義します。

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。
- チャンネル圧縮は使用できません。
- データを正しく処理するには、必要なチャンネル出口をすべて設定する必要があります。これを行うには、「WebSphere MQ メッセージング・プロバイダー」 → 「WebSphere MQ 接続ファクトリー・オブジェクト (WebSphere MQ connection factory objects)」 → 「カスタム・プロパティ」をクリックして、以下のカスタム・プロパティを設定します。
 - **SENDEXIT**。このプロパティには、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl` を設定します。
 - **RECEXIT**。このプロパティには、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl` を設定します。

- **SENDEXITINIT**。このプロパティーには、選択したテキスト・ストリングを設定します。このプロパティーを設定すると、トレースで出口を見分けやすくなります。
- **RECEXITINIT**。このプロパティーには、選択したテキスト・ストリングを設定します。このプロパティーを設定すると、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。

外部クライアント

WebSphere Process Server は、WebSphere MQ 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、サーバーの SCA モジュールにメッセージを送信する必要があり、サーバー内からコンポーネントによって呼び出される必要もあります。

WebSphere MQ 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、呼び出されたアプリケーション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 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。

HTTP バインディング

HTTP バインディングは、Service Component Architecture (SCA) と HTTP の接続を提供する目的で設計されています。これにより、既存のまたは新規作成された HTTP アプリケーションをサービス指向アーキテクチャー (SOA) 環境内に組み込むことができます。

さらに、SCA ランタイム環境のネットワークは、既存の HTTP インフラストラクチャーを通して通信できます。

HTTP バインディングは、以下のような HTTP 機能を公開します。

- メッセージがメディエーション・コンポーネントに渡されるときに、HTTP フォーマットとメッセージ・ヘッダー情報が保持されます。これにより、HTTP アプリケーションのプログラマー、ユーザー、および管理者に、より分かりやすい表示が提供されます。
- 既存のデータ・バインディング・フレームワークが HTTP 規則に対応して拡張され、SCA メッセージと HTTP メッセージ・ヘッダーおよび本体との間のマッピングが提供されます。

- さまざまな共通 HTTP 機能をサポートするようにインポートおよびエクスポートを構成できます。
- HTTP インポートまたはエクスポートを含む SCA モジュールをインストールすると、HTTP 接続が可能になるように、適切なランタイム環境が自動的に構成されます。

HTTP インポートおよびエクスポートの詳細な作成手順は、インフォメーション・センターの「**WebSphere Integration Developer**」>「**統合アプリケーションの開発 (Developing integration applications)**」>「**HTTP データ・バインディング (HTTP data binding)**」を参照してください。

関連タスク

303 ページの『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、インポート・バインディングの詳細を変更したりすることもできます。

307 ページの『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。

HTTP バインディングの表示

アプリケーションのデプロイ後に、HTTP バインディングを検査して、それが正しいことを確認する必要がある場合があります。

始める前に

管理コンソールを表示する必要があります。

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、このタスクを実行するには、オペレーター、管理者、またはコンフィギュレーターとしてログインする必要があります。

このタスクについて

既存の HTTP バインディングを変更する最初のステップとして、HTTP バインディングを表示します。

プロシージャ

1. Service Component Architecture (SCA) アプリケーションを表示します。「**アプリケーション**」を展開して、「**SCA モジュール**」をクリックします。

インストール済みのアプリケーションが表示されます。

2. バインディングを表示する *module_name* をクリックします。

モジュールの構成が表示されます。

3. 表示するバインディングに応じて、「**インポート**」または「**エクスポート**」を展開します。
4. *path* を展開した後、「**バインディング**」を展開すると、モジュールに含まれるインポートまたはエクスポートのバインディングが表示されます。

含まれているインポート・バインディングまたはエクスポート・バインディングがすべて表示されます。

5. バインディング構成を表示するバインディング名をクリックします。

結果

選択したインポートまたはエクスポートのバインディングの構成が表示されます。

次のタスク

必要に応じてインポートまたはエクスポートを変更します。

HTTP エクスポート・バインディングの変更

管理コンソールでは、HTTP エクスポート・バインディングの構成を変更するときに、元のソースを変更してからアプリケーションを再デプロイする必要はありません。

始める前に

『HTTP バインディングの表示』の説明に従って、モジュールの HTTP バインディングを表示します。

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

バインディングのメソッドが ping 可能であるかどうかを変更する必要があるとき、またはメソッドもしくはバインディングがサポートするエンコードを変更する必要があるときは、HTTP エクスポート・バインディングを変更します。

プロシージャ

1. 『HTTP バインディングの表示』の説明に従って、モジュールのバインディングを表示します。
2. 変更するバインディングを選択します。バインディングの名前をクリックします。
3. バインディングの構成を変更します。

オプション	説明
バインディング・スコープの構成を変更するには	「バインディング・スコープ」タブをクリックします。
メソッド・スコープの構成を変更するには	「メソッド・スコープ」タブをクリックします。

両方の構成が存在するときは、メソッド・スコープの構成がバインディング・スコープの構成より優先されます。

4. 構成を変更した後、同じページに留まる場合は「適用」を、前のページに戻る場合は「OK」をクリックします。

結果

バインディングの構成が変更されます。

制約事項: モジュールを再デプロイすると、構成が新しい構成で置き換えられます。各デプロイメントでこの変更をモジュールで維持するには、WebSphere Integration Developer を使用してモジュールのソース・コードを変更する必要があります。

HTTP インポート・バインディングの変更

管理コンソールでは、HTTP インポート・バインディングの構成を変更するときに、元のソースを変更してからアプリケーションを再デプロイする必要はありません。

始める前に

『HTTP バインディングの表示』の説明に従って、モジュールの HTTP バインディングを表示します。

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

Service Component Architecture (SCA) モジュールで使用される HTTP アプリケーションのバインディング・プロパティが変更されたときは、HTTP インポート・バインディングを変更します。

プロシージャ

- 『HTTP バインディングの表示』の説明に従って、モジュールのバインディングを表示します。
- 変更するバインディングを選択します。バインディングの名前をクリックします。
- バインディングの構成を変更します。

オプション	説明
バインディング・スコープの構成を変更するには	「バインディング・スコープ」タブをクリックします。
メソッド・スコープの構成を変更するには	「メソッド・スコープ」タブをクリックします。

両方の構成が存在するときは、メソッド・スコープの構成がバインディング・スコープの構成より優先されます。

- 構成を変更した後、同じページに留まる場合は「適用」を、前のページに戻る場合は「OK」をクリックします。

結果

バインディングの構成が変更されます。

制約事項: モジュールを再デプロイすると、構成が新しい構成で置き換えられます。各デプロイメントでこの変更をモジュールで維持するには、WebSphere Integration Developer を使用してモジュールのソース・コードを変更する必要があります。

HTTP ヘッダー

HTTP インポートおよびエクスポート・バインディングを使用すると、HTTP ヘッダーを構成し、その値をアウトバウンド・メッセージに使用できます。HTTP インポートはそれらのヘッダーを要求に使用し、HTTP エクスポートは応答に使用します。

静的に構成されたヘッダーと制御情報は、実行時に動的に設定される値よりも優先されます。しかし、動的オーバーライド URL、バージョン、およびメソッド制御の値は、静的な値をオーバーライドします。それ以外の場合、静的な値がデフォルト値と見なされます。

バインディングは、実行時に HTTP ターゲットの URL、バージョン、およびメソッド値を判別することにより、HTTP インポート URL の動的性をサポートします。これらの値は、動的オーバーライド URL、バージョン、およびメソッドの値を Service Component Architecture (SCA) メッセージの HTTP 制御パラメーター・セクションから抽出して判別されます。

HTTP エクスポートおよびインポート・バインディングのアウトバウンド・メッセージに関する制御およびヘッダー情報は、次の順序で処理されます。

1. HTTP 動的オーバーライド URL、バージョン、およびメソッドを SCA メッセージから除外するヘッダーおよび制御情報。(最低の優先順位)
2. Simple Configuration Definition Language (SCDL) インスタンスのエクスポート/インポート・レベル
3. エクスポート/インポート・レベルでの管理コンソールからの変更
4. SCDL インスタンスのエクスポート/インポート・メソッド・レベル
5. エクスポート/インポートのメソッド・レベルでの管理コンソールからの変更
6. Service Component Architecture (SCA) メッセージからの HTTP 動的オーバーライド URL、バージョン、およびメソッド
7. データ・バインディングからのヘッダーおよび制御情報 (最高の優先順位)

HTTP エクスポートおよびインポートは、`contextPropagationEnabled` が `True` に設定されている場合、インバウンド方向のヘッダーと制御パラメーターに、着信メッセージからのデータ (`HTTPExportRequest` および `HTTPImportResponse`) の取り込みのみを行います。逆に、アウトバウンドのヘッダーと制御パラメーター (`HTTPExportResponse` および `HTTPImportRequest`) については、`contextPropagationEnabled` が `True` に設定されている場合、HTTP エクスポートおよびインポートは読み取りと処理のみを行います。SCDL からの構成済みのヘッダーおよび制御パラメーターは、`contextPropagationEnabled` プロパティの値に関係なく適用されます。

提供される HTTP ヘッダー構造とサポート

表 15 に、HTTPImport と HTTPExport の要求および応答の要求/応答パラメーターを項目別に示します。

表 15. 提供される HTTP ヘッダー情報

制御名	HTTPImport 要求	HTTPImport 応答	HTTPExport 要求	HTTPExport 応答
URL	無視される	設定されない	要求メッセージから読み取られる。 注: 照会ストリングも URL 制御パラメーターの一部です。	無視される
バージョン (可能な値: 1.0、1.1。デフォルトは 1.1)	無視される	設定されない	要求メッセージから読み取られる	無視される
メソッド	無視される	設定されない	要求メッセージから読み取られる	無視される
動的オーバーライド URL	設定された場合、HTTP Import URL をオーバーライドする。要求行のメッセージに書き込まれる。 注: 照会ストリングも URL 制御パラメーターの一部です。	設定されない	設定されない	無視される
動的オーバーライド・バージョン	設定された場合、HTTP Import バージョンをオーバーライドする。要求行のメッセージに書き込まれる。	設定されない	設定されない	無視される
動的オーバーライド・メソッド	設定された場合、HTTP Import メソッドをオーバーライドする。要求行のメッセージに書き込まれる。	設定されない	設定されない	無視される

表 15. 提供される HTTP ヘッダー情報 (続き)

制御名	HTTPImport 要求	HTTPImport 応答	HTTPExport 要求	HTTPExport 応答
メディア・タイプ (この制御パラメーターは、Content-Type HTTP ヘッダーの値の一部を搬送します。)	存在する場合、メッセージに Content-Type ヘッダーの一部として書き込まれる。 注: この制御エレメント値は、データ・バインディングによって提供してください。	応答メッセージ、Content-Type ヘッダーから読み取られる。	要求メッセージ、Content-Type ヘッダーから読み取られる。	存在する場合、メッセージに Content-Type ヘッダーの一部として書き込まれる。 注: この制御エレメント値は、データ・バインディングによって提供してください。
文字セット (デフォルト: UTF-8)	存在する場合、メッセージに Content-Type ヘッダーの一部として書き込まれる。 注: この制御エレメント値は、データ・バインディングによって提供してください。	応答メッセージ、Content-Type ヘッダーから読み取られる。	要求メッセージ、Content-Type ヘッダーから読み取られる。	サポート対象。メッセージに Content-Type ヘッダーの一部として書き込まれる。 注: この制御エレメント値は、データ・バインディングによって提供してください。
転送エンコード (可能な値: chunked、identity。デフォルトは identity)	存在する場合、メッセージにヘッダーとして書き込まれ、メッセージ変換のエンコードの方法を制御する。	応答メッセージから読み取られる	要求メッセージから読み取られる	存在する場合、メッセージにヘッダーとして書き込まれ、メッセージ変換のエンコードの方法を制御する。
コンテンツ・エンコード (可能な値: gzip、x-gzip、deflate、identity。デフォルトは identity)	存在する場合、メッセージにヘッダーとして書き込まれ、ペイロードのエンコードの方法を制御する。	応答メッセージから読み取られる	要求メッセージから読み取られる	存在する場合、メッセージにヘッダーとして書き込まれ、ペイロードのエンコードの方法を制御する。

表 15. 提供される HTTP ヘッダー情報 (続き)

制御名	HTTPImport 要求	HTTPImport 応答	HTTPExport 要求	HTTPExport 応答
Content-Length	存在する場合、メッセージにヘッダーとして書き込まれる (HTTP ヘッダー値は無視される)。 注: これは、制御パラメーター内でデータ・バイndィングによって設定してください。	応答メッセージから読み取られる	要求メッセージから読み取られる	存在する場合、メッセージにヘッダーとして書き込まれる (HTTP ヘッダー値は無視される)。 注: これは、制御パラメーター内でデータ・バイndィングによって設定してください。
StatusCode (デフォルト: 200)	サポート対象外	応答メッセージから読み取られる	サポート対象外	存在する場合、応答行のメッセージに書き込まれる。
ReasonPhrase (デフォルト: OK)	サポート対象外	応答メッセージから読み取られる	サポート対象外	制御値は無視される。メッセージ応答行の値は StatusCode から生成される。
認証 (複数のプロパティーを含む)	存在する場合、基本認証ヘッダーの作成に使用される。 注: このヘッダーの値は、HTTP プロトコル上でのみエンコードされます。SCA では、これはデコードされ、平文として渡されます。	該当なし	要求メッセージの基本認証ヘッダーから読み取られる。このヘッダーの存在は、ユーザーが認証済みであることを示すものではありません。認証は、サーブレット構成で制御する必要があります。 注: このヘッダーの値は、HTTP プロトコル上でのみエンコードされます。SCA では、これはデコードされ、平文として渡されます。	該当なし

表 15. 提供される HTTP ヘッダー情報 (続き)

制御名	HTTPImport 要求	HTTPImport 応答	HTTPExport 要求	HTTPExport 応答
プロキシ (次のような複数のプロパティを含む: Host, Port, Authentication)	存在する場合、プロキシによる接続を確立するために使用される。	該当なし	該当なし	該当なし
SSL (次のような複数のプロパティを含む: Keystore, Keystore Password, Trustore, Trustore Password, ClientAuth)	入力され、宛先 URL が HTTPS の場合、これは SSL による接続を確立するために使用される。	該当なし	該当なし	該当なし

HTTP 関数セレクター

関数セレクターは「ネイティブ」のオペレーション名の識別に使用され、そのオペレーション名はメソッド・バインディングを識別します。提供される 2 つの関数セレクターのいずれかを使用できます。あるいは、HTTPFunctionSelector インターフェースの Java 実装を提供することにより、カスタム関数セレクターを使用できます。

FunctionSelector は、HTTP ヘッダーからの情報、起動された HTTP エクスポート URL からの情報、および場合によっては HTTP メッセージ・ペイロードからの情報を使用します。

HTTP エクスポートは、要求メッセージを受け取ると、関数セレクターを使用して「ネイティブ」のメソッド名、つまり起動されるメソッドの HTTP ID (単数または複数) を判別します。次に、メソッド固有のバインディング構成 (「メソッド・バインディング」と呼ばれます) を使用して起動する操作の名前、要求に使用する形式 (つまり、データ・バインディング・クラス)、および応答に使用する形式 (つまり、データ・バインディング・クラス) が判別されます。

HTTP インポートも、メソッド・バインディングを使用して要求を判別し、オプションとして、応答データ・バインディングを判別します。インポートのインターフェース上で実行される操作ごとに、メソッド・バインディングがあります。

インポートまたはエクスポートのために HTTP メッセージを作成する場合、データ・バインディングは、ヘッダーのデータ・オブジェクトにアクセスでき、それらのオブジェクトを完全に制御できます。インバウンド HTTP メッセージを読み取る場合、FunctionSelector とデータ・バインディングは、どちらも HTTP ヘッダーのリストへの全アクセス権限を持ちます。更新情報は、SCA へ渡されるメッセージに反映されます。

提供される 2 つの関数セクターは、次のとおりです。

- HTTP エクスポート・コンテキスト・パス + HTTP メソッド

HTTP エクスポート・バインディング・コンテキスト・パス (HTTP エクスポート上の受信された要求の部分) と HTTP メソッドの組み合わせは、サブレット・インスタンスに固有のものなので、関数セクターはそれをネイティブ・オペレーション名として使用でき、そのオペレーション名がメソッド・バインディングを識別します。

例えば、HTTP エクスポートが受け取った要求の URL が「http://www.ibm.com/store/shoppingCart」で、要求の HTTP メソッドが「POST」の場合、上記の関数セクターは、エクスポート・コンテキスト・パスとして「/store/shoppingCart」を抽出し、区切り文字として「@」を使用して「POST」を付加します。その結果、エクスポートが探すネイティブ・メソッドは、「/store/shoppingCart@POST」であることが必要になります。エクスポート・コンテキスト・パスが設計時に構成され、必ずしも要求 URL の相対パスと同じでないことに注意してください。

- HTTP ヘッダー

この関数セクターは、TargetFunctionName HTTP カスタム・ヘッダーからネイティブ操作を読み取ります。

バインディングは、エクスポート内のすべてのネイティブ・メソッド名が固有であり、したがって実行時にメソッド・バインディングを判別するには、最初の一致を見つけるだけで十分であるという事実に依存しています。バリデータは、ネイティブ・メソッドが固有であることを確認します。

カスタム HTTP 関数セクターの実装

カスタム関数セクターを実装するには、HTTPFunctionSelector 抽象クラスを展開し、generateEISFunctionName メソッドを実装します。

HTTP データ・バインディング

Service Component Architecture (SCA) メッセージと HTTP プロトコル・メッセージ間のデータの異なるマッピングごとに、HTTP データ・バインディングを構成する必要があります。3 つのデータ・バインディング・クラスが提供されるほか、カスタム・データ・バインディングを作成することもできます。

HTTP インポートおよび HTTP エクスポートに使用する 3 つのデータ・バインディングが提供されます。それらは、バイナリー・データ・バインディング、XML データ・バインディング、および SOAP データ・バインディングです。応答データ・バインディングは、片方向の操作には必要ありません。

データ・バインディングは、インスタンスが HTTP と ServiceDataObject の間で両方向に変換可能な Java クラスの名前によって表されます。エクスポートでは関数セクターを使用する必要があります。関数セクターは、使用されるデータ・バインディングと呼び出される操作を、メソッド・バインディングと連動して判別できます。

提供されるデータ・バインディングは、次のとおりです。

- バイナリー・データ・バインディングは、本文を非構造化バイナリー・データとして取り扱います。バイナリー・データ・バインディングの XSD スキーマの実装は、次のとおりです。

```
<xsd:schema elementFormDefault="qualified"
  targetNamespace="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:tns="http://com.ibm.websphere.http.data.bindings/schema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:complexType name="HTTPBaseBody">
    <xsd:sequence/>
  </xsd:complexType>

  <xsd:complexType name="HTTPBytesBody">
    <xsd:complexContent>
      <xsd:extension base="tns:HTTPBaseBody">
        <xsd:sequence>
          <xsd:element name="value" type="xsd:hexBinary"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

- XML データ・バインディングは、本文を XML データとしてサポートします。XML データ・バインディングの実装は JMS XML データ・バインディングによく似ており、インターフェース・スキーマについての制限はありません。
- SOAP データ・バインディングは、本体を SOAP データとしてサポートします。SOAP データ・バインディングの実装には、インターフェース・スキーマについての制限はありません。

カスタム HTTP データ・バインディングの実装

HTTPStreamDataBinding は、カスタム HTTP メッセージを処理するための根元的なインターフェースです。このインターフェースは大きなペイロードを処理できるように設計されています。しかし、そのような実装を機能させるために、このデータ・バインディングは、ストリームにメッセージを書き込む前に制御情報とヘッダーを返す必要があります。

メソッドとその実行順序 (以下を参照) は、カスタム・データ・バインディングによって実装する必要があります。

HTTP バインディングは、以下の順序でデータ・バインディングを呼び出します。

アウトバウンド処理 (SDO から ネイティブ・フォーマットへ):

1. setDataObject(...)
2. setHeaders(...)
3. setControlParameters(...)
4. setBusinessException(...)
5. convertToNativeData()
6. getControlParameters()
7. getHeaders()
8. write(...)

インバウンド処理 (ネイティブ・フォーマットから SDO へ):

1. setControlParameters(...)
2. setHeaders(...)
3. convertFromNativeData(...)
4. isBusinessException()
5. getDataObject()
6. getControlParameters()
7. getHeaders()

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。

データ・バインディングの詳しい操作手順は、WebSphere Integration Developer インフォメーション・センターの「**WebSphere Integration Developer**」>「**統合アプリケーションの開発 (Developing integration applications)**」>「**外部サービスにアクセスする構成の操作 (Working with configurations that access external services)**」>「**データ・バインディング構成 (Data binding configuration)**」を参照してください。

EIS アプリケーションのデータ・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によって、引数データを J2C または JMS 実装が必要とするネイティブ・フォーマットに変換できます。

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、データ・オブジェクトの設定が可能です。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。レコード・ベースのデータ・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に発生します。このインスタンスでは、レコードをデータ・バインディング上で設定する必要があります。これにより、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、このインターフェースを実装する `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・バインディング上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・インターフェース `InboundNativeDataRecord` が提供されます。

`RecordDataBinding` と `RecordHolderDataBinding` の両方で、3 つの引数を取る呼び出しをサポートできます。リソース・アダプターに必要な引数が 2 つであるかまたは 3 つであるかは、`ResourceAdapterMetadata` で定義されています。

`ResourceAdapterMetadata` は、実行時に、`RecordDataBinding` と `RecordHolderDataBinding` に必要な引数の数を判別するために照会されます。

`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` インターフェース上で設定できるため、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、汎用 JMS、および WebSphere MQ JMS アプリケーションのデータ・バインディング

JMS メッセージでは、データ・バインディング・クラスを使用して、JMS メッセージと内部データ表現 (サービス・データ・オブジェクト) の間のマッピングが実行されます。

JMS、汎用 JMS、および WebSphere MQ JMS アプリケーションを対象として、あらゆるメッセージ本体に対応する 6 つの汎用データ・バインディングがあります。これらのデータ・バインディングは、定義済み `message` クラスと、`message` の 5 つのサブクラスです。`message` クラスには、プロパティを指定できますが、ペイロードは指定できません。各サブクラスには、それぞれ異なる本体タイプがあります。メッセージ処理に使用されるランタイム・インターフェースに対する操作は、`FunctionSelector` です。これは、メッセージの `JMSType` プロパティを使用して、操作名を取得します。

`message` の 5 つのサブクラスには、メッセージ・ペイロードを保持するためのプロパティがあります。

- `TextMessage` - メッセージ本体は Java ストリングです。`JMSTextDataBinding` に 1 つのストリング・プロパティが含まれています。
- `BytesMessage` - メッセージ本体はバイト配列です。`JMSBytesDataBinding` に 1 つの `hexBinary` プロパティが含まれています。

- `ObjectMessage` - メッセージ本体は直列化 Java オブジェクトです。
`JMSObjectDataBinding` に 1 つの `hexBinary` プロパティが含まれています。
- `StreamMessage` - メッセージ本体は単純 Java タイプのストリーム (シーケンス) です。`JMSStreamDataBinding` にエンタリー・エレメントのリストが含まれています。このリストの各エンタリーには、`type` プロパティと `value` プロパティが含まれています。
- `MapMessage` - メッセージ本体は、単純 Java タイプの値を持つ `name` により解決可能な名前/値のペアの集合です。`JMSMapDataBinding` にエンタリー・エレメントのリストが含まれています。このリストの各エンタリーには、`name`、`type`、および `value` プロパティが含まれています。

`FunctionSelector` はインポート・ノードとエクスポート・ノードで公開されます。エクスポート・ノードが作成され、インターフェースがこのノードに割り当てられている場合、ノードのバインディングが生成可能です。「JMS バインディング」が選択されると、ウィザードには、「直列化タイプ (Serialization type)」の 3 つのオプション (「オブジェクト」、「テキスト」、「ユーザー提供 (User Supplied)」) を示すドロップダウン選択機能が表示されます。

新規ユーザー提供関数セクターのクラス名をテキスト・ボックスに入力するには、JMS 関数セクターのチェック・ボックスをクリアします。

エクスポートまたはインポートのバインディング・プロパティのページでも、データ・バインディングを選択または変更できます。

直列化:

JMS および MQ JMS アプリケーションは、JMS または MQ JMS 形式との間での `DataObject` の直列化を実現するために、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 アプリケーション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>

```

WebSphere MQ アプリケーションのデータ・バインディング

WebSphere MQ メッセージのデータ・バインディングは、ヘッダー・データ・バインディングと 4 種類の本体データ・バインディングのセットとして提供されています。

本体データ・バインディングとヘッダー・データ・バインディングの両方で、WebSphere MQ メッセージの読み取りおよび書き込みに同じ API が使用されます。read() メソッドは MQDataInputStream オブジェクトを渡し、write() メソッドは MQDataOutputStream オブジェクトを渡します。

読み取り時でも書き込み時でも、ヘッダー・データ・バインディングは、ヘッダーに関連付けられている制御情報のフォーマット ID に基づいて選択されます。WebSphere MQ メッセージの読み取り時に、メッセージを処理できるヘッダー・バインディングがない場合は、本体バインディングが呼び出されます。WebSphere MQ

メッセージの書き込み時には、WMQ メッセージに書き込まれたフォーマット ID が本体データ・バインディングによって上書きされることがあります。

WebSphere MQ インポートまたはエクスポートで使用する本体データ・バインディングは、4 種類の中から選択できます。string および binary データ・バインディングでは、本体は構造化されていないテキスト・データまたはバイナリー・データとして処理されます。その他の 2 つの本体データ・バインディング (XML 形式を使用するデータ・バインディングと、Java オブジェクト直列化を使用するデータ・バインディング) では、SCA メッセージ本体が直列化形式でネイティブ MQ メッセージに格納されます。インポートでいずれかの本体データ・バインディングが使用される場合、インポートは、アウトバウンド・ネイティブ MQ メッセージに MQRFH2 ヘッダーを組み込みます。<usr> フォルダには、プロパティ TargetFunctionName があります。このプロパティには、呼び出された操作の名前が設定されます。

エクスポートで FunctionSelector を使用して、本体データ・バインディングのタイプを選択します。4 つの関数セクターがあります。そのうちの 2 つの関数セクターは、ネイティブ操作としてフォーマット情報を戻します。これらの関数セクターは、本体データ・バインディングがフォーマット・データに依存する場合に使用できます。3 番目の関数セクターは、MQRFH2 ヘッダーの <usr> フォルダの TargetFunctionName プロパティからネイティブ操作を読み取ります。これにより、JMS インポートとの相互運用が実現します。4 番目の関数セクターでは、WebSphere MQ メッセージの内容に関係なく常に handleMessage が戻されます。

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・バインディングが適切に構成されている場合は、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ように規定できます。

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。エクスポート・バインディングは、ターゲット SCA コンポーネントへのエントリー・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。

イベントの順序付けの有効化: EIS エクスポート

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。

このタスクについて

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確認してください。

一般に、Network Deployment 環境でイベントの順序付けが必要な場合は、エクスポートが含まれているモジュールを、スタンドアロン・サーバー、または高可用性のために有効にされているアクティブ・サーバーが 1 つのみのクラスターに移動する必要があります。

関連概念

174 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。

イベントの順序付けの有効化: JMS エクスポート

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。

始める前に

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、宛先が区画に分割されていない場合に限りクラスター環境でサポートされています。クラスターを含む Network Deployment 環境でイベントの順序付けが機能するためには、クラスターあたりのアクティブなメッセージ・エンジンは 1 つのみです。クラスターにないサーバーが含まれている Network Deployment 環境でイベントの順序付けが機能するためには、各サーバーにアクティブなメッセージ・エンジンを含めることができます。

このタスクについて

イベントの順序付けでは、イベントをターゲット・コンポーネントに実行のためにディスパッチする前に、イベントがロックを獲得する必要があります。実行が完了すると、イベントのロックが解放されます。イベントがロックを獲得できない場合は、呼び出しの実行が中断されます。その後イベントがロックを獲得すると、イベントがディスパッチされます。

特定のコンポーネントの特定のメソッドでイベントの順序付けが必要であることを宣言するには、コンポーネント定義でそのメソッドにイベントの順序付けの修飾子を追加します。

- `keySpecification` 属性は、順序付けする必要があるイベントの識別に使用するキーを定義します。
- `parameter` 属性は、キー属性の抽出元パラメーターを指定します。
- `name` 属性はパラメーターの名前です。
- `xpath` 属性は、キーの一部となる値の抽出元パラメーターに適用されます。

パラメーターごとに、キーの一部となるパラメーター・エレメントを指定する必要があります。

esadmin コマンド行ユーティリティーでは、アクティブなロックとキューに入れられているロックをリストおよび削除できます。このユーティリティーの使用法は **esAdmin** *[-h hostname] [-p soapPortNumber] method [parameters]* の形式です。ホスト名が指定されていない場合のデフォルトは *localhost* です。soapPortNumber が指定されていない場合のデフォルトは 8880 です。セキュリティーが有効になっている場合は、有効な *userId* とパスワードが必要です。

esadmin -help の使用法は **esAdmin** *[-h hostname] [-p soapPortNumber] [-u userId] [-p password] method [parameters]* の形式です。指定可能なメソッドは、以下のとおりです。

- listAll
- listlocks moduleName
- listLocks moduleName componentName
- listLocks moduleName componentName methodName
- deleteLocks moduleName
- unlock lockId

lockId と sequenceId は整数であり、moduleName、componentName、および methodName は文字列です。sequenceId は出力に示され、ロック要求がキューに入れられた順序を反映しています。

重要: *delete* コマンドを使用するときには十分に注意してください。可能な場合は *delete* コマンドの代わりに *unlock* コマンドを使用してください。*delete* コマンドを実行すると、データベースが削除されます。このコマンドを実行する前に、モジュールが停止している必要があります。モジュールを再始動する必要があります。

JMS エクスポートのイベントの順序付けは、WebSphere Integration Developer から有効にします。

プロシージャー

1. WebSphere MQ JMS エクスポート・バインディングの eventSequencing プロパティを True に設定します。
2. 変更をマスター構成に保管します。

結果

ご使用のバインディングでイベントの順序付けが可能になります。

注: 例外宛先を除去すると、何らかの障害が発生するたびにすべての着信メッセージが停止します。

関連概念

181 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。

イベントの順序付けの有効化: WebSphere MQ JMS エクスポート

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。

始める前に

Network Deployment 環境で WebSphere MQ JMS エクスポート・バインディングのイベントの順序付けが必要な場合、エクスポートが含まれているモジュールを、スタンドアロン・サーバーに移動するか、または高可用性のために有効にされているアクティブ・サーバーが 1 つのみのクラスターに移動する必要があります。

このタスクについて

WebSphere MQ JMS エクスポートのイベントの順序付けは、WebSphere Integration Developer から有効にします。

プロシージャ

1. WebSphere MQ JMS エクスポート・バインディングの `eventSequencing` プロパティを `True` 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

201 ページの『WebSphere MQ JMS バインディング』

WebSphere MQ JMS バインディングは、WebSphere MQ JMS ベースのプロバイダーとの統合を提供します。

205 ページの『WebSphere MQ JMS バインディングの操作』

WebSphere MQ JMS バインディングは、WebSphere Process Server で管理できます。

イベントの順序付けの有効化: WebSphere MQ エクスポート

WebSphere MQ 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。

始める前に

Network Deployment 環境で WebSphere MQ エクスポート・バインディングのイベントの順序付けが必要な場合、エクスポートが含まれているモジュールを、スタンドアロン・サーバーに移動するか、または高可用性のために有効にされているアクティブ・サーバーが 1 つのみのクラスターに移動する必要があります。

このタスクについて

WebSphere MQ エクスポートのイベントの順序付けは、WebSphere Integration Developer から有効にします。

プロシージャ

1. ネイティブ MQ エクスポート・バインディングの `eventSequencing` プロパティを `True` 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

215 ページの『WebSphere MQ バインディングの操作』
WebSphere MQ バインディングはサーバー内から管理できます。

イベントの順序付けの有効化: 汎用 JMS エクスポート

汎用 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。

始める前に

Network Deployment 環境で汎用 JMS エクスポート・バインディングのイベントの順序付けが必要な場合、エクスポートが含まれているモジュールを、スタンドアロン・サーバーに移動するか、または高可用性のために有効にされているアクティブ・サーバーが 1 つのみのクラスターに移動する必要があります。

このタスクについて

汎用 JMS エクスポートのイベントの順序付けは WebSphere Integration Developer から有効にされます。

プロシージャ

1. 汎用 JMS エクスポート・バインディングの `eventSequencing` プロパティを `True` 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

191 ページの『汎用 JMS バインディング』
汎用 JMS インポート・バインディングとエクスポート・バインディングにより、サード・パーティーの JMS 1.1 準拠プロバイダーに対する接続が提供されます。このバインディングの操作は、JMS バインディングの操作と似ています。

バインディングのトラブルシューティング

バインディングでは、特定のタイプのバインディングに固有の各種エラー状態が発生する可能性があります。

このタスクについて

エラー状態の処理方法は、該当するバインディングのタイプによって異なります。

JMS および WebSphere MQ JMS バインディングのトラブルシューティング

JMS バインディングおよび WebSphere 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。

実装例外: このタスクについて

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。

- `ServiceBusinessException`: 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。
- `ServiceRuntimeException`: 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>
```

以下のいずれかの例外を受け取る場合:

```
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: このタスクについて

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となります。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 との対話が可能となります。

Failed Event Manager に表示されない JMS ベースの WebSphere MQ SCA メッセージ: このタスクについて

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となります。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 との対話が可能となります。

障害: このタスクについて

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。プール値のヘッダー・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作業している場合は、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。

プロシージャ

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。

誤用例: WebSphere MQ バインディングとの比較: このタスクについて

WebSphere MQ JMS バインディングは、WebSphere MQ に対してデプロイされている JMS アプリケーションと相互協調処理するよう設計されています。これにより、メッセージは JMS メッセージ・モデルに基づいて公開されます。これに対し、WebSphere MQ インポートおよびエクスポートは、ネイティブ WebSphere MQ アプリケーションと相互協調処理することができ、WebSphere MQ メッセージ本体の内容全体をメディエーションに公開するように設計されています。

以下のシナリオでは、WebSphere MQ バインディングではなく WebSphere MQ JMS バインディングを使用して作成する必要があります。

- JMS メッセージ駆動型 Bean (MDB) を SCA モジュールから呼び出す。この MDB は、WebSphere MQ JMS プロバイダーに対してデプロイされています。これは、WebSphere MQ JMS インポートを使用して実装されます。
- SCA モジュールを J2EE コンポーネント・サーブレットまたは EJB から JMS を介して呼び出すことができるようにする。これは、WebSphere MQ JMS エクスポートを使用して実装されます。
- WebSphere MQ 上で転送中の JMS MapMessage の内容のメディエーションを実行する。この場合、WebSphere MQ JMS エクスポートとインポート、および適切なデータ・バインディングを組み合わせ使用します。

WebSphere MQ バインディングと WebSphere MQ JMS バインディングの相互協調処理が予期される状況があります。特に、J2EE WebSphere MQ アプリケーションと非 J2EE WebSphere MQ アプリケーション間をブリッジングする場合は、WebSphere MQ エクスポートと WebSphere MQ JMS インポート (あるいはこの逆) を、適切なデータ・バインディングまたはメディエーション・モジュール (あるいはこの両方) と組み合わせ使用します。

汎用 JMS バインディングのトラブルシューティング

汎用 JMS バインディングでは特定の失敗状態が生じることがあります。

このタスクについて

汎用 JMS バインディングに関するさまざまなエラー状態が発生する可能性があります。

汎用 JMS 例外のトラブルシューティング:

汎用 JMS のインポートおよびエクスポート実装が、さまざまなエラー状態に対応して例外を戻すことがあります。

このタスクについて

汎用 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。

- `ServiceBusinessException` - この例外は、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戻されます。
- `ServiceRuntimeException` - 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。JMS の場合これは `JMSException` です。

汎用 JMS メッセージの有効期限のトラブルシューティング:

JMS プロバイダーの要求メッセージには有効期限が設定されます。

このタスクについて

要求有効期限 は、JMS プロバイダーの要求メッセージの `JMSExpiration` の時刻に達した場合の、その要求メッセージの有効期限を指します。その他の JMS バインディングについては、汎用 JMS バインディングは要求有効期限を処理するため、インポートにより配置されたコールバック・メッセージの有効期限を、発信要求と同じ有効期限に設定します。コールバック・メッセージの有効期限の通知により、要求メッセージの有効期限が切れていることが示されます。クライアントには、ビジネス例外により期限切れを通知する必要があります。

コールバック宛先がサード・パーティー・プロバイダーに移動すると、このタイプの要求有効期限はサポートされなくなります。

応答有効期限 は、JMS プロバイダーの応答メッセージの `JMSExpiration` の時刻に達した場合の、その応答メッセージの有効期限を指します。

汎用 JMS バインディングでは応答有効期限はサポートされていません。これは、サード・パーティー JMS プロバイダーの正確な期限切れ動作が定義されていないためです。ただし、応答を受信する場合、応答を受信する時点で応答有効期限が切れていないかどうかを確認できます。

アウトバウンド要求メッセージでは、`JMSExpiration` 値は待機時間と、`asyncHeader` の `requestExpiration` 値 (設定されている場合) から算出されます。

汎用 JMS 接続ファクトリーのエラーのトラブルシューティング:

汎用 JMS プロバイダーで特定のタイプの接続ファクトリーを定義すると、アプリケーションの開始時にエラー・メッセージを受け取ることがあります。この問題を回避するには、外部接続ファクトリーを変更します。

このタスクについて

アプリケーションの起動時に、「MDB リスナー・ポート JMSConnectionFactory タイプが JMSDestination タイプと一致しません (MDB Listener Port JMSConnectionFactory type does not match JMSDestination type)」というエラー・メッセージが出力されることがあります。

この問題は、外部接続ファクトリーの定義時に発生します。特に、JMS 1.1 (統合) 接続ファクトリー (Point-to-Point 通信およびパブリッシュ/サブスクライブ通信の両方をサポート可能な接続ファクトリー) ではなく JMS 1.0.2 トピック接続ファクトリーを作成すると、例外がスローされることがあります。

この問題を解決するには、以下のステップを実行します。

プロシージャ

1. 使用している汎用 JMS プロバイダーにアクセスします。
2. 定義されている JMS 1.0.2 トピック接続ファクトリーを JMS 1.1 (統合) 接続ファクトリーに置き換えます。

結果

新規に定義した JMS 1.1 接続ファクトリーを使用してアプリケーションを起動すると、エラー・メッセージは表示されないはずで。

WebSphere MQ バインディングのトラブルシューティング

WebSphere MQ バインディングで発生する障害または失敗の状態を診断し、このような状態を修正できます。

このタスクについて

WebSphere MQ バインディングの主な障害状態は、トランザクションの意味構造、WebSphere MQ 構成、またはその他のコンポーネントの既存の動作への参照に基づいて判別されます。主な障害状態は以下のとおりです。

- WebSphere MQ キュー・マネージャーまたはキューに接続できない: WebSphere MQ に接続できず、メッセージを受信できない場合は、MDB ListenerPort が開始できません。この状態は、WebSphere Application Server ログに記録されます。永続メッセージは、正常に取得されるまで (または WebSphere MQ により期限切れとなるまで) WebSphere MQ キューに残ります。WebSphere MQ に接続できず、アウトバウンド・メッセージを送信できないと、送信操作を制御するトランザクションがロールバックされます。
- インバウンド・メッセージを構文解析できない、またはアウトバウンド・メッセージを構成できない: データ・バインディングが失敗すると、作業を制御するトランザクションがロールバックされます。
- アウトバウンド・メッセージを送信できない: メッセージを送信できないと、関連するトランザクションがロールバックされます。
- 複数の応答メッセージまたは予期しない応答メッセージが戻される: インポートでは、要求メッセージごとに応答メッセージが 1 つのみ戻されることが想定されています。応答を受信すると、レコードが削除されます。予期しない状態で受信した応答メッセージは、JMS インポートで破棄されます。

誤用例: WebSphere MQ JMS バインディングとの比較: このタスクについて

通常、WebSphere MQ インポートおよびエクスポートは、ネイティブ WebSphere MQ アプリケーションと相互協調処理し、WebSphere MQ メッセージ本体の内容全体をメディエーションに公開するように設計されています。一方、WebSphere MQ JMS バインディングは、WebSphere MQ に対してデプロイされている JMS アプリケーションと相互協調処理するように設計されています。これにより、メッセージは JMS メッセージ・モデルに基づいて公開されます。

以下のシナリオでは、WebSphere MQ バインディングではなく WebSphere MQ JMS バインディングを使用して作成する必要があります。

- JMS メッセージ駆動型 Bean (MDB) を SCA モジュールから呼び出す。この MDB は、WebSphere MQ JMS プロバイダーに対してデプロイされています。これは、WebSphere MQ JMS インポートを使用して実装されます。
- SCA モジュールを J2EE コンポーネント・サーブレットまたは EJB から JMS を介して呼び出すことができるようにする。これは、WebSphere MQ JMS エクスポートを使用して実装されます。
- WebSphere MQ 上で転送中の JMS MapMessage の内容のメディエーションを実行する。この場合、WebSphere MQ JMS エクスポートとインポート、および適切なデータ・バインディングを組み合わせて使用します。

WebSphere MQ バインディングと WebSphere MQ JMS バインディングの相互協調処理が予期される状況があります。特に、J2EE WebSphere MQ アプリケーションと非 J2EE WebSphere MQ アプリケーションを連携させる場合は、WebSphere MQ エクスポートと WebSphere MQ JMS インポート (あるいはこの逆) を、適切なデータ・バインディングまたはメディエーション・モジュール (あるいはこの両方) と組み合わせて使用します。

一般に構成エラーが原因で WebSphere MQ がメッセージを意図する宛先に配信できない場合、メッセージは指定されている送達不能キューに送信されます。このとき、メッセージ本体の先頭には、送達不能ヘッダーが追加されます。このヘッダーには、失敗の原因、元の宛先、およびその他の情報が含まれています。

WebSphere MQ JMS バインディングの主な特徴

WebSphere MQ JMS バインディングの主な特徴は、ヘッダー、J2EE 成果物、および作成される J2EE リソースです。

ヘッダー

JMS メッセージ・ヘッダーには、いくつかの定義済みフィールドがあります。これらのフィールドには、クライアントとプロバイダーの両方でメッセージの識別と送付に使用される値が格納されます。

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。ストリング 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 を使用する場合は、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 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。

RequestMsgIDToCorrelID

JMSMessageID は JMSCorrelationID フィールドにコピーされます。これはデフォルト設定です。

RequestCorrelIDToCorrelID

JMSCorrelationID は JMSCorrelationID フィールドにコピーされます。

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:
 - インポート: 要求または出力メッセージが送信される宛先です。
 - エクスポート: 応答メッセージが送信される宛先です。ただし、着信メッセージの JMSReplyTo ヘッダー・フィールドにより置き換えられた場合は、その宛先が優先されます。
- 受信宛先:
 - インポート: 応答メッセージまたは着信メッセージが格納される宛先です。
 - エクスポート: 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。

WebSphere MQ バインディングの主な特徴

WebSphere MQ バインディングの主な特徴は、ヘッダー、J2EE 成果物、および作成される J2EE リソースです。

関連スキーム

WebSphere MQ 要求/応答アプリケーションでは、応答メッセージと要求を関連付けるさまざまな手法の 1 つを使用できます。これらの手法には、MQMD の MessageID フィールドと CorrelID フィールドが関連します。ほとんどの場合、要求側はキュー・マネージャーに MessageID を選択させ、応答アプリケーションがこれを応答の CorrelID にコピーすることを想定します。多くの場合、要求側と応答アプリケーションは、使用される関連手法を暗黙的に認識しています。応答アプリケーションは、要求の Report フィールドに設定されているさまざまなフラグの指示に従って、これらのフィールドを処理することもあります。

WebSphere MQ メッセージのエクスポート・バインディングを構成するときには、以下のオプションを使用できます。

応答の MsgId のオプション:

新規 MsgID (New MsgID)

キュー・マネージャーが応答の固有の MsgId を選択できるようにします (デフォルト)。

要求の MsgID からコピーする (Copy from Request MsgID)

要求の MsgId フィールドから MsgId フィールドをコピーします。

SCA メッセージからコピーする (Copy from SCA message)

SCA 応答メッセージの WebSphere MQ ヘッダーに含まれる値を MsgId に設定します。この値が存在しない場合は、キュー・マネージャーに新規 Id を定義させます。

Report オプションを使用 (As Report Option)

要求の MQMD の Report フィールドを検査して、MsgId の処理方法を認識します。MQRO_NEW_MSG_ID オプションと MQRO_PASS_MSG_ID オプションがサポートされています。これらのオプションは、それぞれ「新規 MsgID (New MsgID)」および「要求の MsgID からコピーする (Copy from Request MsgID)」と同様に動作します。

応答の CorrelId のオプション:

要求の MsgID からコピーする (Copy from Request MsgID)

要求の MsgId フィールドから CorrelId フィールドをコピーします (デフォルト)。

要求の CorrelID からコピーする (Copy from Request CorrelID)

要求の CorrelId フィールドから CorrelId フィールドをコピーします。

SCA メッセージからコピーする (Copy from SCA message)

SCA 応答メッセージの WebSphere MQ ヘッダーに含まれる値を CorrelId に設定します。この値が存在しない場合は、空白のままにします。

Report オプションを使用 (As Report Option)

要求の MQMD の Report フィールドを検査して、CorrelId の処理方法を認識し

ます。MQRO_COPY_MSG_ID_TO_CORREL_ID オプションと MQRO_PASS_CORREL_ID オプションがサポートされています。これらのオプションは、それぞれ「要求の MsgID からコピーする (Copy from Request MsgID)」および「要求の CorrelID からコピーする (Copy from Request CorrelID)」と同様に動作します。

WebSphere MQ メッセージのインポート・バインディングを構成するときには、以下のオプションを使用できます。

要求 の MsgId のオプション:

新規 MsgID (New MsgID)

キュー・マネージャーが要求の固有の MsgId を選択できるようにします (デフォルト)。

SCA メッセージからコピーする (Copy from SCA message)

SCA 要求メッセージの WebSphere MQ ヘッダーに含まれる値を MsgId に設定します。この値が存在しない場合は、キュー・マネージャーに新規 Id を定義させます。

応答関連のオプション:

応答の CorrelID を MsgId からコピーする (Response has CorrelID copied from MsgId)

応答メッセージの CorrelId フィールドが要求の MsgId に基づいて設定されていることを想定します (デフォルト)。

応答の MsgID を MsgId からコピーする (Response has MsgID copied from MsgId)

応答メッセージの MsgId フィールドが要求の MsgId に基づいて設定されていることを想定します。

応答の CorrelID を CorrelId からコピーする (Response has CorrelID copied from CorrelId)

応答メッセージの CorrelId フィールドが要求の CorrelId に基づいて設定されていることを想定します。

J2EE リソース

WebSphere MQ バインディングを 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 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: インポートの場合は、要求または出力メッセージが送信される宛先です。エクスポートの場合は、応答メッセージが送信される宛先です。ただし、着信メッセージの MQMD ReplyTo ヘッダー・フィールドにより置き換えられた場合は、その宛先が優先されます。
- 受信宛先: 応答/要求または着信メッセージが格納される宛先です。

汎用 JMS バインディングの主な特徴

汎用 JMS インポート・バインディングおよびエクスポート・バインディングの機能は、WebSphere 埋め込み JMS および MQ JMS インポート・バインディングの機能と整合性があります。主な機能は、ヘッダー定義および既存の J2EE リソースへのアクセスです。ただし、汎用という性質上、JMS プロバイダー固有の接続オプションはありません。また、このバインディングでは、デプロイメントおよびインストール時にリソースを生成できる機能が限定されています。

汎用インポート

MQ JMS インポート・アプリケーションと同様に、汎用 JMS 実装は非同期であり、3 つの呼び出し (片方向、両方向 (要求/応答ともいう)、およびコールバック) をサポートしています。

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る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 replyTo ヘッダー・フィールドで指定されます。

汎用エクスポート

汎用 JMS エクスポート・バインディングは、結果の戻りの処理方法が EIS エクスポート・バインディングとは異なります。汎用 JMS エクスポートでは、着信メッセージに指定された replyTo 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。

汎用 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 (汎用 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、次にその要求が SCA ランタイムで処理されるようにディスパッチします。

特殊ヘッダー

特殊ヘッダーのプロパティは、汎用インポートおよびエクスポートで使用され、ターゲット・バインディングに対してメッセージの処理方法を指示します。

例えば、TargetFunctionName プロパティは、デフォルトの関数セクターが、呼び出されるインポートまたはエクスポートのインターフェース内で操作の名前を識別するときに使用します。

J2EE リソース

JMS バインディングを 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。

- インポートの場合は受信宛先 (応答、両方向のみ)、エクスポートの場合は受信宛先 (要求) で `listen` するためのリスナー・ポート
- `outboundConnection` (インポート) および `inboundConnection` (エクスポート) の汎用 JMS 接続ファクトリー
- 送信宛先 (インポート) および受信宛先 (エクスポート、両方向のみ) の汎用 JMS 宛先
- `responseConnection` の汎用 JMS 接続ファクトリー (両方向のみ、オプション。これ以外の場合は、インポートに `outboundConnection` が使用され、エクスポートに `inboundConnection` が使用される)
- 受信宛先 (インポート) および送信宛先 (エクスポート) の汎用 JMS 宛先 (両方向のみ)
- SIB コールバック・キュー宛先にアクセスするときに使用されるデフォルトのメッセージング・プロバイダー・コールバック JMS 宛先 (両方向のみ)
- コールバック JMS 宛先にアクセスするときに使用されるデフォルトのメッセージング・プロバイダー・コールバック JMS 接続ファクトリー (両方向のみ)
- 応答処理中に使用される要求メッセージに関する情報を格納するための SIB コールバック・キュー宛先 (両方向のみ)

インストール・タスクにより、インポート・ファイルおよびエクスポート・ファイルの情報から `ConnectionFactory`、3 つの宛先、および `ActivationSpec` が作成されます。

JMS バインディングの主な特徴

JMS のインポート・バインディングおよびエクスポート・バインディングの主な特徴は、ヘッダーと作成される J2EE リソースです。

EIS との相違点

インポート

JMS インポート・アプリケーションは、ランタイム呼び出しにおける応答の処理方法が EIS インポートとは異なります。JMS 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)、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 インポートのデプロイ時に、要求メッセージに対する応答を `listen` する目的で、ランタイム環境により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、要求と共に送信される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 `replyTo` ヘッダー・フィールドで指定されます。

エクスポート

JMS エクスポート・バインディングは、結果の戻りの処理方法が EIS エクスポート・バインディングとは異なります。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 インポートで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、次にその要求が SCA ランタイムで処理されるようにディスパッチします。

特殊ヘッダー

特殊ヘッダーのプロパティは、JMS インポートおよびエクスポートで使用され、ターゲットに対してメッセージの処理方法を指示します。

例えば、`TargetFunctionName` がネイティブ・メソッドから操作メソッドにマップされます。これは、デフォルトの関数セクター (`JMSFunctionSelectorImpl`) が JMS メッセージからネイティブ・メソッド名を抽出するときに使用されます。

J2EE リソース

JMS インポートおよびエクスポートを J2EE 環境にデプロイすると、さまざまな J2EE リソースが作成されます。

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

インポートでは、要求に対する応答を受信するときに使用されます。エクスポートでは、メッセージング・システムとの対話で MDB を表すメッセージ・エンドポイントを構成するときに使用されます。

宛先

- 送信宛先: インポートの場合は、要求または出力メッセージが送信される宛先になります。エクスポートの場合は、応答メッセージが送信される宛先になります。ただし、着信メッセージの `JMSReplyTo` ヘッダー・フィールドにより置き換えられた場合は、その宛先が優先されます。
- 受信宛先: 着信メッセージが格納される宛先です。インポートの場合は応答になり、エクスポートの場合は要求になります。
- 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・タスクでは、`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 が受信宛先で応答を `listen` できるようにします。これらのリソースのプロパティは、インポート・ファイルまたはエクスポート・ファイルに指定されます。

JMS バインディング: 全体像

JMS バインディングは、Service Component Architecture (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: SCA モジュールと外部 JMS システムの間の管理された両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・アプリケーション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・バインディングでは、SCA モジュール内で使用する外部 JMS アプリケーションをインポートできます。

JMS 宛先に関連する JMS プロバイダーへの接続を作成するには、JMS 接続ファクトリーを使用します。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。

外部 JMS システム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。

JMS インポート・バインディングでは、以下の 2 種類の使用シナリオがサポートされています。

- 片方向: インポート・ファイル内の `send` 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設定しません。
- 両方向: (要求応答) - インポート・ファイルで指定された `send` 宛先にメッセージを送信します。`receive` 宛先がアウトバウンド・メッセージの `replyTo` ヘッダー・プロパティに設定されます。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。要求応答シナリオでは、応答は要求の `messageId` を応答メッセージの `correlationId` フィールドにコピーするか (デフォルト)、または要求の `correlationId` を応答メッセージの `correlationId` フィールドにコピーすることができます。

片方向と両方向のいずれのシナリオを使用する場合も、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。

183 ページの図 7 は、インポートがどのように外部サービスにリンクされているのかを示しています。

JMS インポート

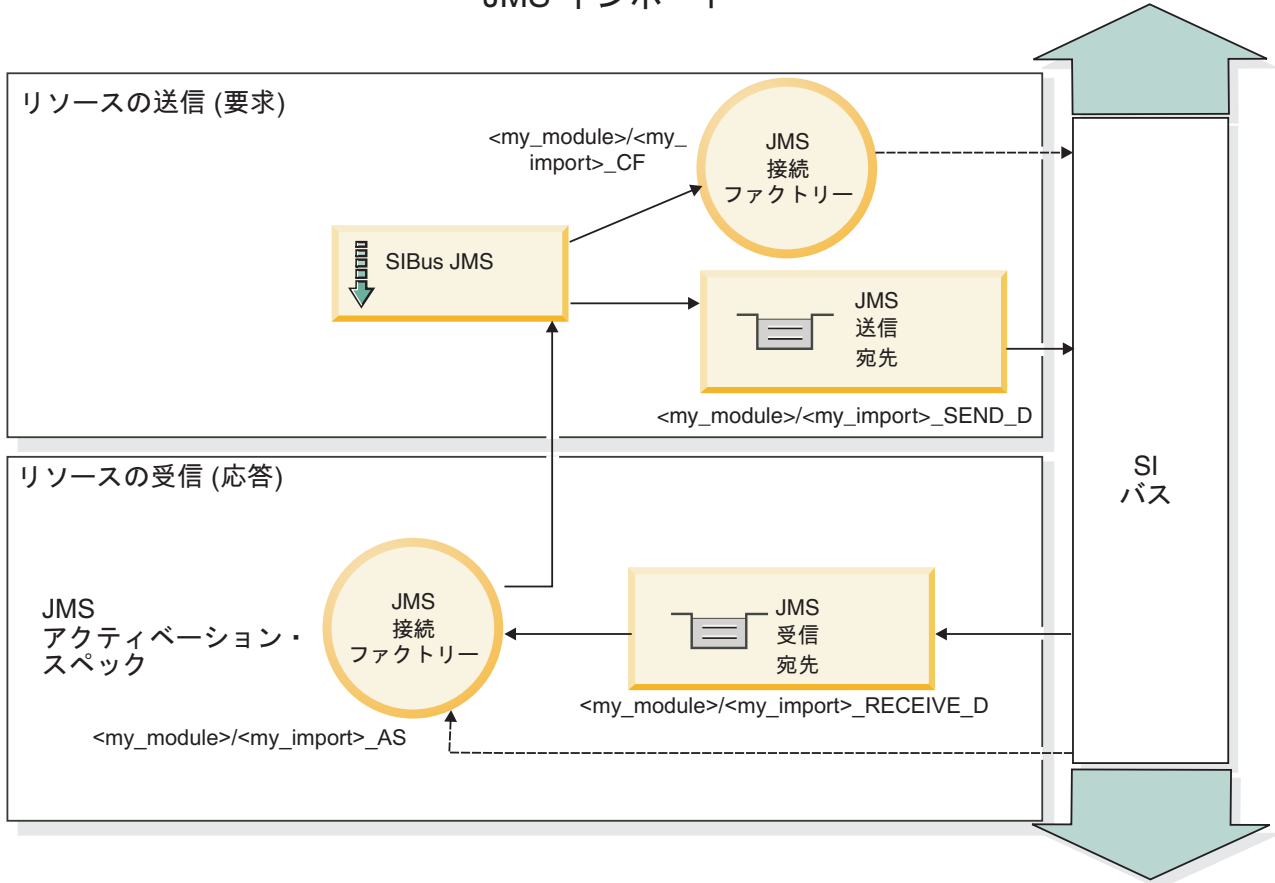


図 16. JMS インポート・バインディングのリソース

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 エクスポートの接続部分は、構成可能なアクティベーション・スペックです。

JMS エクスポートには、send 宛先と receive 宛先があります。receive 宛先は、送信先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。send 宛先は、応答を送信する宛先です。ただし、着信メッセージで replyTo ヘッダー・プロパティを使用してこの宛先を置き換えている場合は、その宛先が優先されます。

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、呼び出されたアプリケーション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 replyTo 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。

184 ページの図 8 は、外部の要求側がどのようにエクスポートにリンクされているのかを示しています。

JMS エクスポート

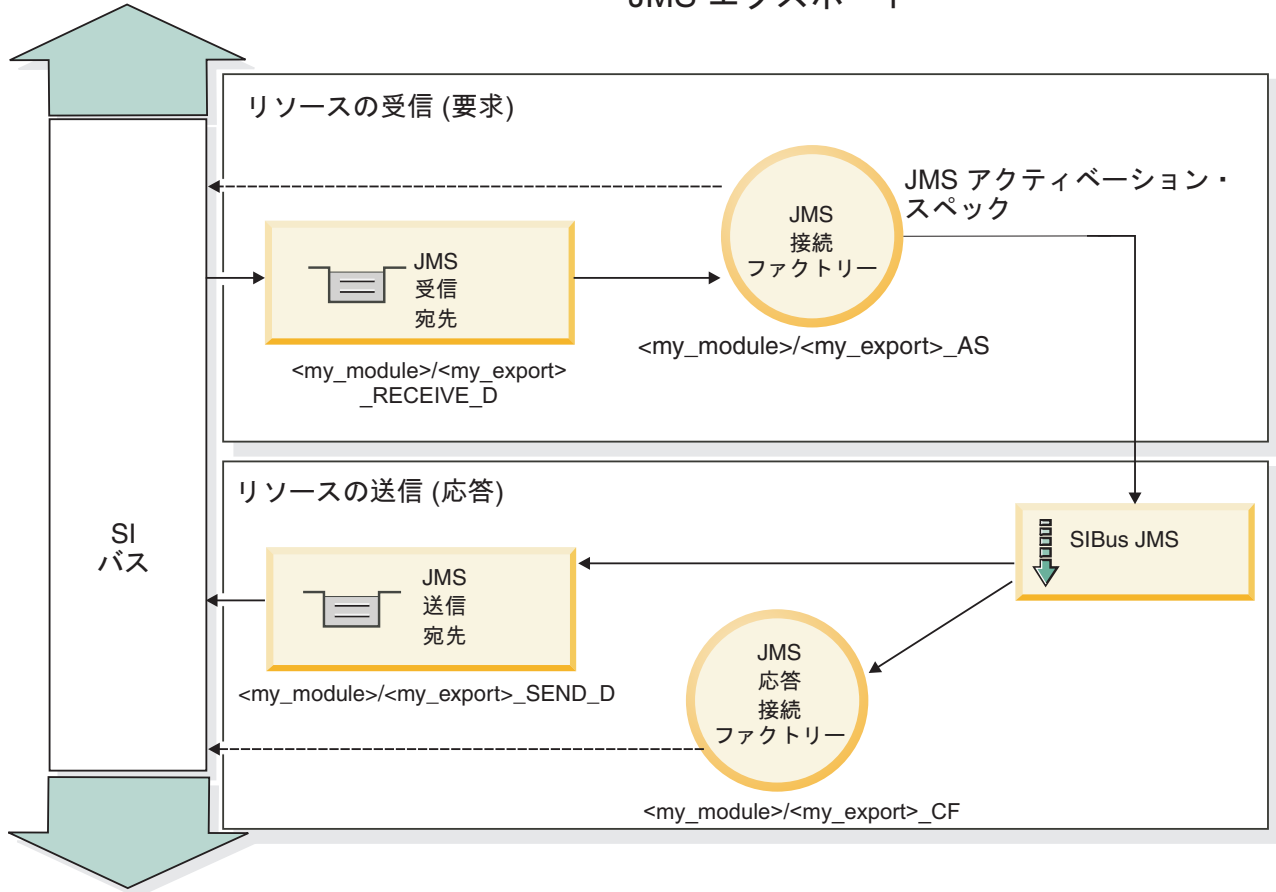


図 17. JMS エクスポート・バインディングのリソース

汎用 JMS バインディング: 全体像

汎用 JMS バインディングは、Service Component Architecture (SCA) 環境と外部 JMS システム (JMS 1.1 に準拠し、オプションの JMS Application Server Facility を実装している場合) の間の接続を提供する非 JCA JMS バインディングです。

汎用 JMS バインディング

汎用 JMS インポートおよび汎用 JMS エクスポートの両方のバインディングの主要なコンポーネントを以下に示します。

- 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・アプリケーション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。

汎用 JMS インポート・バインディング

汎用 JMS インポート・バインディングは、SCA アプリケーションから JCA 1.5 非準拠 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 インポートの接続部分は、接続ファクトリーです。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あり、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 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。

JMS インポート・バインディングでは、以下の 2 種類の使用シナリオがサポートされています。

- 片方向: インポート・ファイルで指定された send 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・ファイル内の send 宛先にメッセージを送信します。receive 宛先が ReplyTo ヘッダー・プロパティーに設定されます。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。要求応答シナリオでは、応答は要求の messageId を応答メッセージの correlationId フィールドにコピーするか (デフォルト)、または要求の correlationId を応答メッセージの correlationId フィールドにコピーすることができます。

片方向と両方向のいずれのシナリオを使用する場合も、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。

193 ページの図 10 は、インポートがどのように外部サービスにリンクされているのかを示しています。

(汎用) JMS インポート

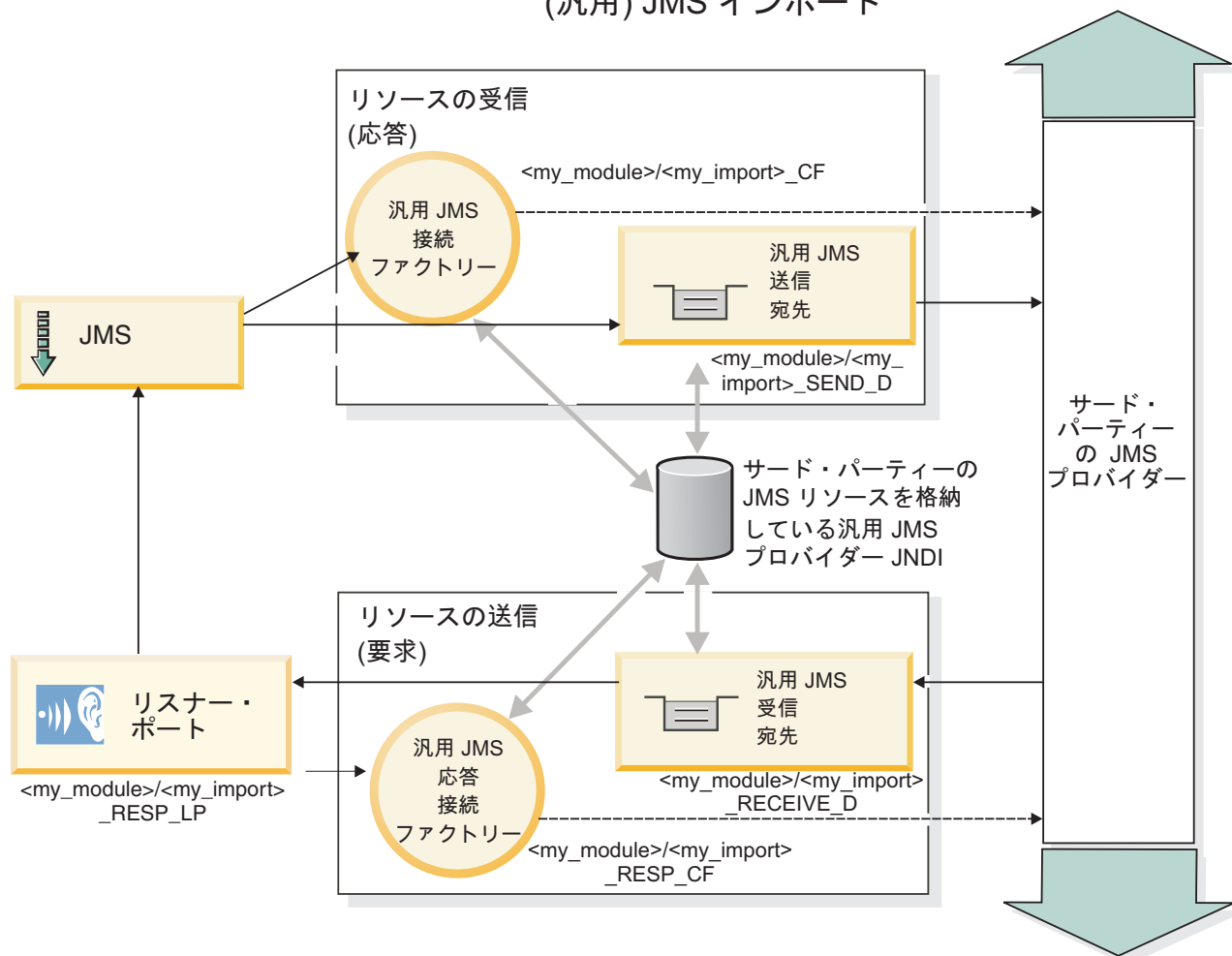


図 18. 汎用 JMS インポート・バインディング・リソース

汎用 JMS エクスポート・バインディング

汎用 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 エクスポートの接続部分は、ConnectionFactory および ListenerPort から構成されます。

汎用 JMS エクスポートには、send 宛先と receive 宛先があります。receive 宛先は、送信先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。send 宛先は、応答を送信する宛先です。ただし、着信メッセージで replyTo ヘッダー・プロパティを使用してこの宛先を置き換えている場合は、その宛先が優先されます。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、呼び出されたアプリケーション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 replyTo 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。

- 要求応答シナリオでは、応答は要求の `messageId` を応答メッセージの `correlationId` フィールドにコピーするか (デフォルト)、または要求の `correlationId` を応答メッセージの `correlationId` フィールドにコピーすることができます。

194 ページの図 11 は、外部の要求側がどのようにエクスポートにリンクされているのかを示しています。

(汎用) JMS エクスポート

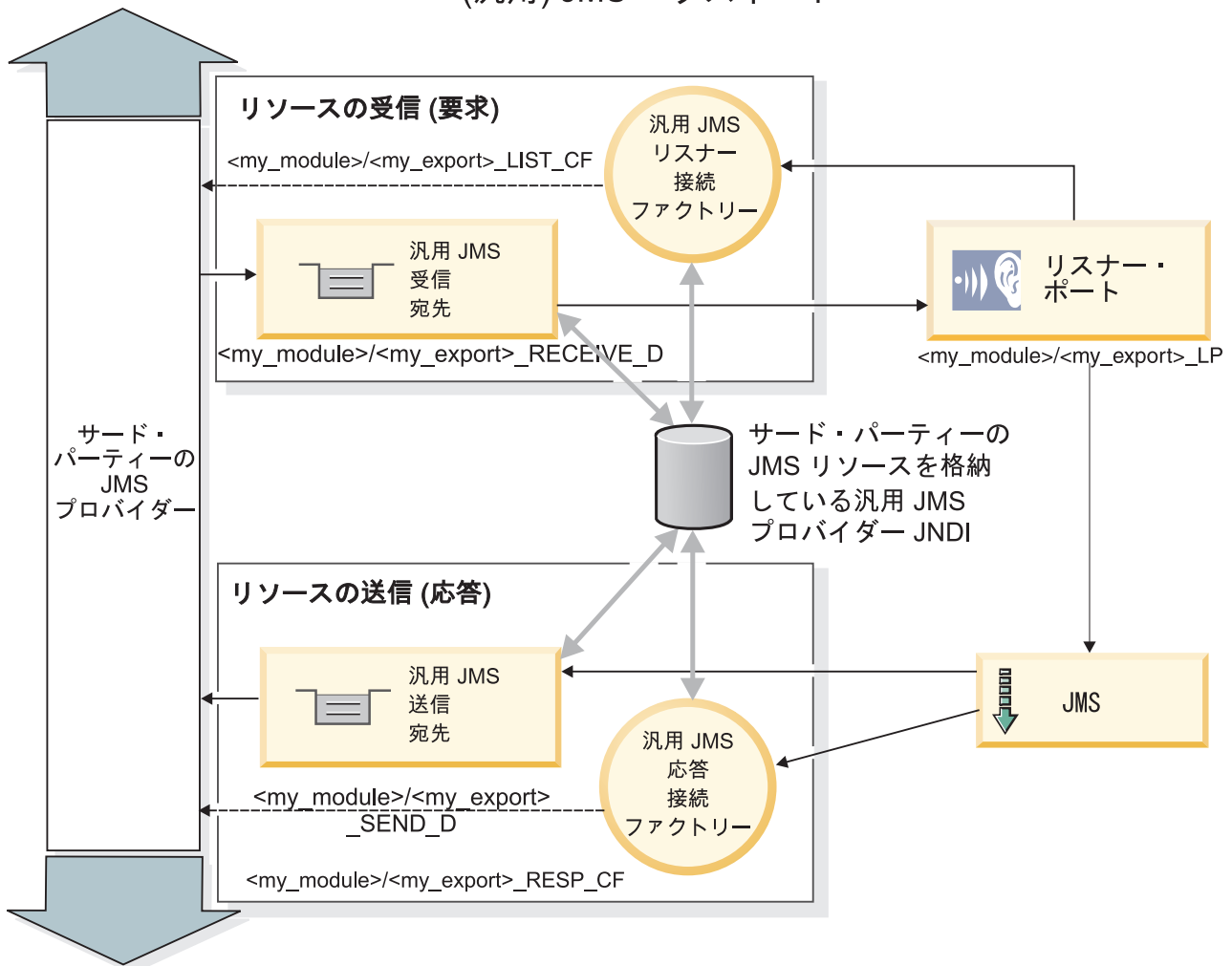


図 19. 汎用 JMS エクスポート・バインディング・リソース

WebSphere MQ JMS バインディング: 全体像

WebSphere MQ JMS アプリケーションは、JMS API を使用した WebSphere MQ との統合を提供します。

WebSphere MQ 管理用タスク

WebSphere MQ システム管理者は、基盤となる WebSphere MQ キュー・マネージャーを作成する必要があります。このキュー・マネージャーは、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、Service Component Architecture (SCA) アプリケーションから WebSphere MQ JMS ベースのプロバイダーへのアウトバウンド接続を提供します。サポートされているバージョンの 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。

WebSphere MQ JMS インポート・バインディングでは、以下の 2 種類の使用シナリオがサポートされています。

- 片方向: インポート・ファイル内の `send` 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・ファイル内の `send` 宛先にメッセージを送信します。receive 宛先が `replyTo` ヘッダー・フィールドに設定されます。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、デフォルト関連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のシナリオを使用する場合も、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。

202 ページの図 12 は、インポートがどのように外部サービスにリンクされているのかを示しています。

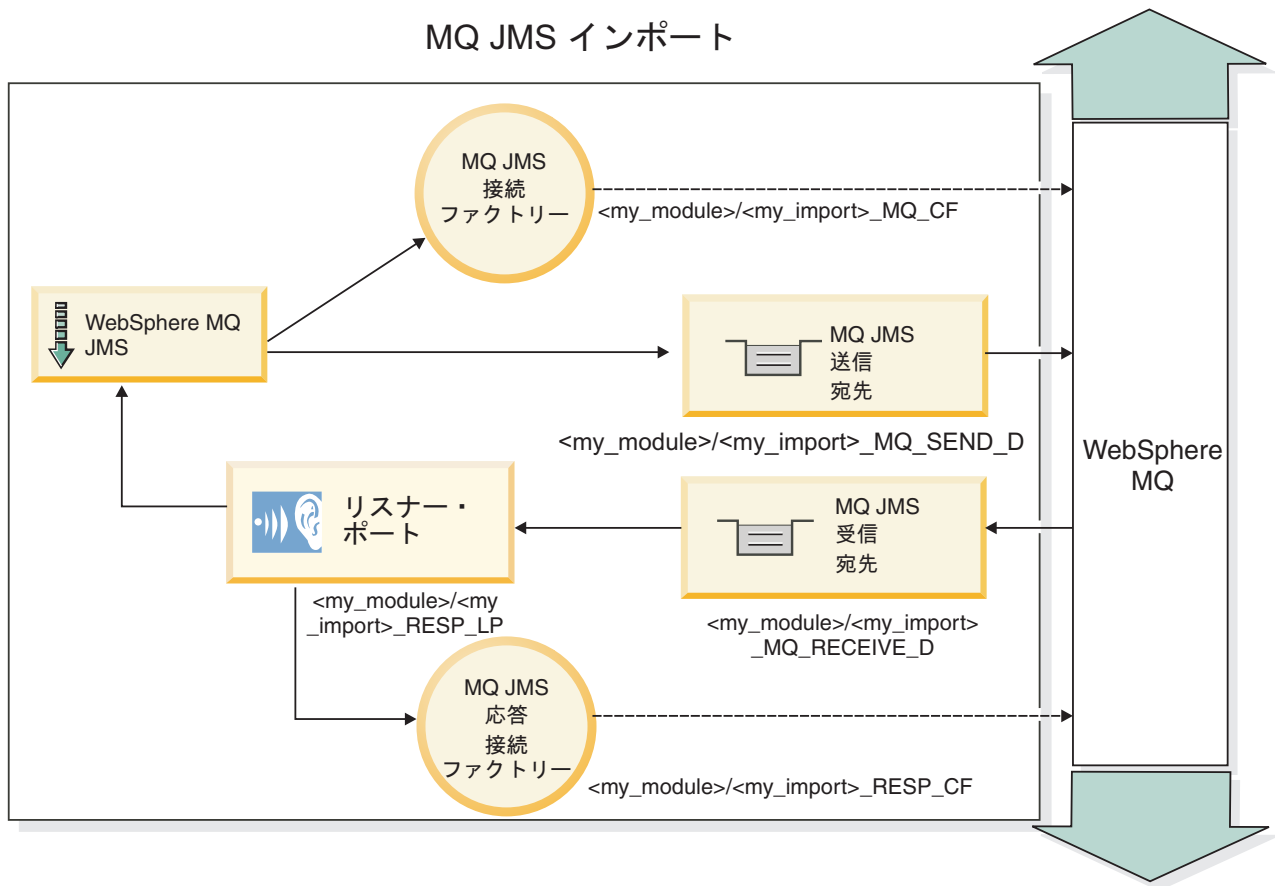


図 20. WebSphere MQ JMS インポート・バインディングのリソース

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、呼び出されたアプリケーション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 replyTo 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。

203 ページの図 13 は、外部の要求側がどのようにエクスポートにリンクされているのかを示しています。

MQ JMS エクスポート

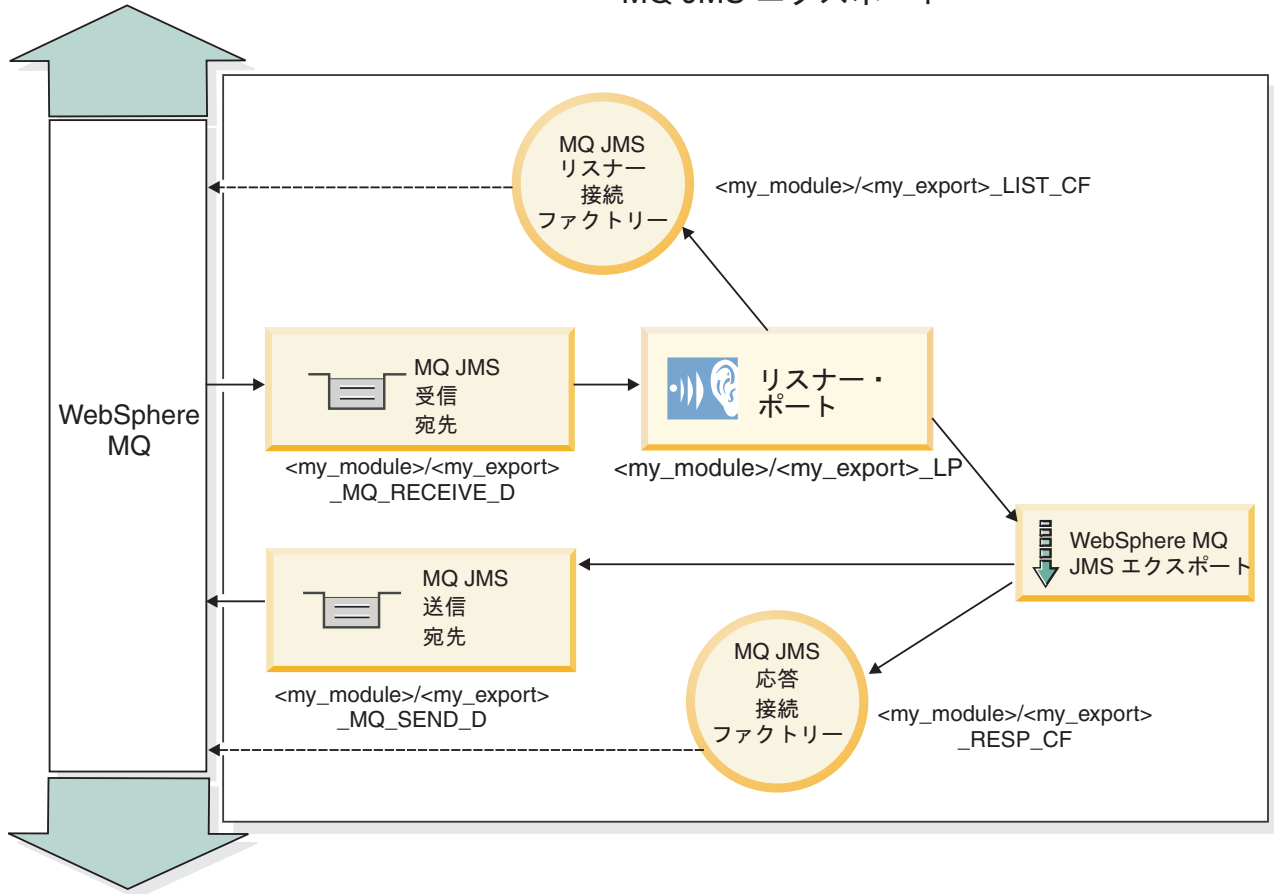


図 21. WebSphere MQ JMS エクスポート・バインディングのリソース

WebSphere MQ バインディング: 全体像

WebSphere MQ アプリケーションにより、ネイティブ MQ ベースのアプリケーションとの統合が実現します。

WebSphere MQ 管理用タスク

WebSphere MQ システム管理者は、基盤となる WebSphere MQ キュー・マネージャーを作成する必要があります。このキュー・マネージャーは、WebSphere MQ 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 インポート・バインディング

WebSphere MQ インポート・バインディングは、Service Component Architecture (SCA) アプリケーションから WebSphere MQ ベースのアプリケーションへのアウトバウンド接続を提供します。サポートされているバージョンの 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 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。MQMD 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が replyTo MQMD ヘッダー・フィールドに設定されます。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。

211 ページの図 14 は、インポートがどのように外部サービスにリンクされているのかを示しています。

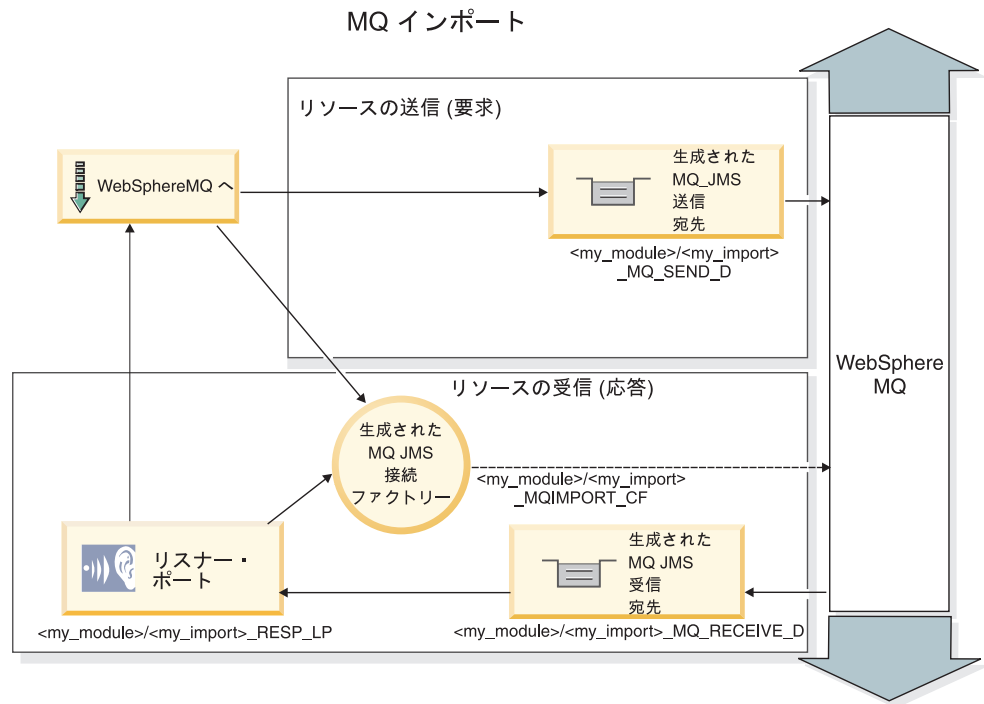


図 22. WebSphere MQ インポート・バインディング・リソース

WebSphere MQ エクスポート・バインディング

WebSphere MQ エクスポート・バインディングは、WebSphere MQ ベースのアプリケーションから 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、呼び出されたアプリケーション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 replyTo 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。

212 ページの図 15 は、外部の要求側がどのようにエクスポートにリンクされているのかを示しています。

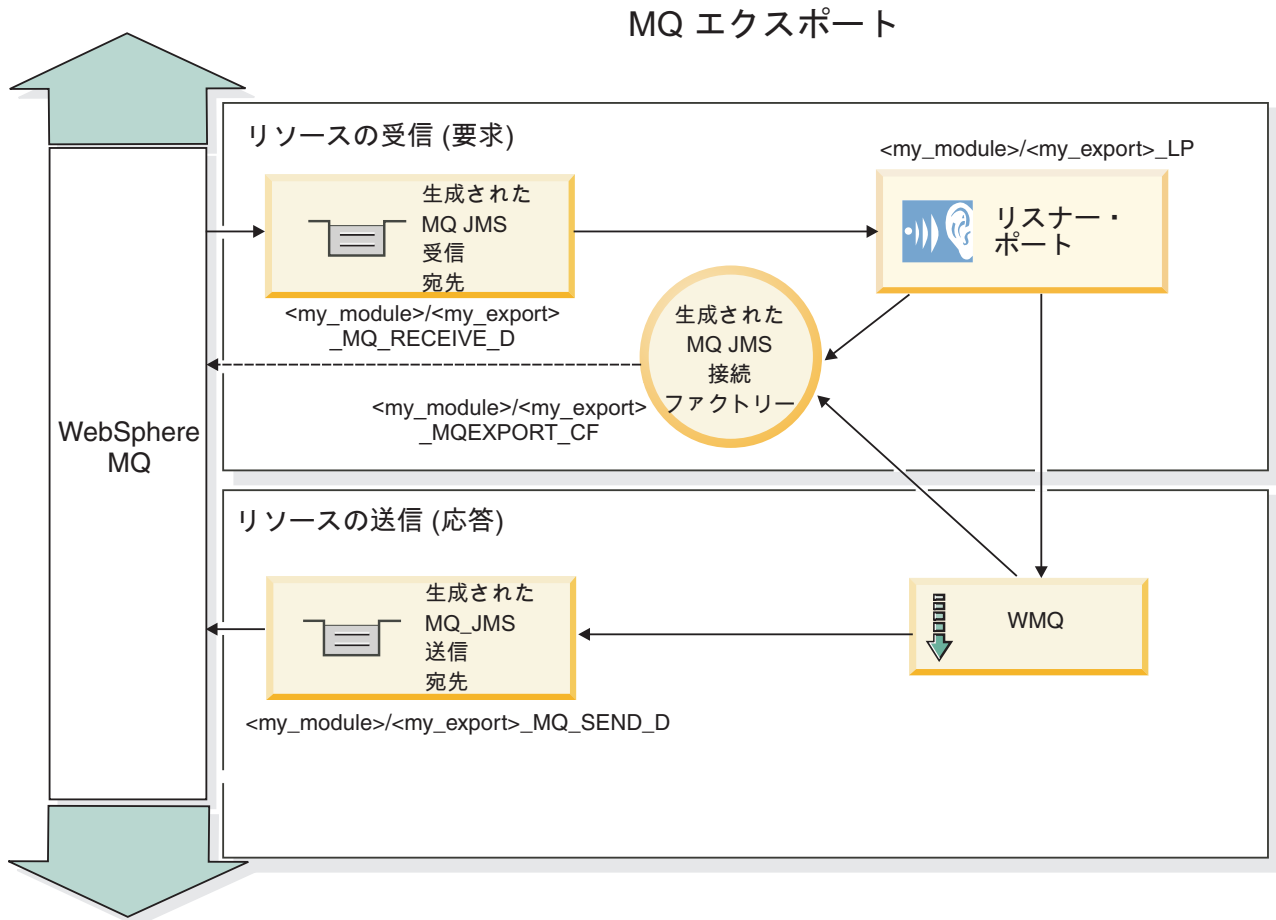


図 23. WebSphere MQ エクスポート・バインディング・リソース

JMS バインディングのパラメーター

JMS のインポート・バインディングおよびエクスポート・バインディングは、すべての必要な接続ファクトリーが作成された状態でインストールできます。または、サーバー上で事前に構成された成果物のセットを指すように設計することもできます。

一般に、JMS インポート・バインディングおよび JMS エクスポート・バインディングは WebSphere Integration Developer で作成されます。開発時には、JMS バインディングに必要な接続と宛先をインストール時 (コンポーネントがサーバーにインストールされる時) に作成する方法を選択できます。もう 1 つの方法として、JMS インポートとエクスポートで使用する予定のサーバー上のリソースの 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 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。インポートおよびエクスポートのために生成される成果物の JNDI 名を以下の表に示します。

表 16. JMS インポート・バインディング: サーバー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・スペック	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

表 17. JMS エクスポート・バインディング: サーバー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・スペック	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。管理コンソールのスコープは、初期設定では「すべてのスコープ」に設定されています。新規リソースを作成するには、このスコープを「セル」または「ノード」に設定する必要があります。デフォルトのリストから既存のリソースを選択できます。

もう 1 つの方法を選択する場合は、バインディングが使用するリソースをサーバー上で検出することになるため、これらのリソースを事前にインストールし、インポート・ファイルとバインディング・ファイルで適切な JNDI 名を指定する必要があります。これにより、JMS バインディングとリソースの間の関連付けが行われます。

汎用 JMS バインディングのパラメーター

汎用 JMS のインポート・バインディングおよびエクスポート・バインディングは、すべての必要な接続ファクトリーが作成された状態でインストールできます。または、サーバー上で事前に構成された成果物のセットを指すように設計することもできます。

一般に、汎用 JMS バインディングは WebSphere Integration Developer で作成されます。開発時には、JMS バインディングに必要な接続と宛先を、コンポーネントがサーバーにインストールされるときに作成する方法を選択できます。または、JMS インポートまたはエクスポートで使用する予定のサーバー上のリソースの JNDI 名を指定できます。

このどちらの方法を選択するかによって、汎用 JMS 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 18. 汎用 JMS インポート: サーバー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	生成されたリソース JNDI 名
outboundConnection	[moduleName]/[importName]_CF
responseConnection	[moduleName]/[importName]_RESP_CF
送信宛先	[moduleName]/[importName]_SEND_D
受信宛先	[moduleName]/[importName]_RECEIVE_D
コールバック宛先	[moduleName]/[importName]_CALLBACK_D

表 19. 汎用 JMS エクスポート: サーバー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	生成されたリソース JNDI 名
inboundConnection	[moduleName]/[exportName]_LIS_CF
responseConnection	[moduleName]/[exportName]_RESP_CF
受信宛先	[moduleName]/[exportName]_RECEIVE_D
送信宛先	[moduleName]/[exportName]_SEND_D
コールバック宛先	[moduleName]/[exportName]_CALLBACK_D

注: リソースは、サーバー・スコープで作成されます。管理コンソールのスコープは、初期設定では「すべてのスコープ」に設定されています。新規リソースを作成するには、このスコープを「セル」または「ノード」に設定する必要があります。デフォルトのリストから既存のリソースを選択できます。

もう 1 つの方法を選択する場合は、JMS インポートが必要なリソースをサーバー上で検出することになるため、これらのリソースを事前にインストールし、インポート・ファイルおよびエクスポート・ファイルで JNDI 名を指定する必要があります。これにより、JMS バインディングとリソースの間の関連付けが行われます。

WebSphere MQ バインディングのパラメーター

WebSphere MQ バインディングは、すべての必要な接続ファクトリーが作成された状態でインストールできます。または、サーバー上で事前に構成された成果物のセットを指すように設計することもできます。

一般に、WebSphere MQ インポート・バインディングおよび WebSphere MQ エクスポート・バインディングは WebSphere Integration Developer で作成されます。開発時には、WebSphere MQ インポートに必要な接続と宛先をインストール時 (コンポーネントがサーバーにインストールされる時) に作成する方法を選択できます。あるいは、WebSphere MQ バインディングで使用する予定のサーバー上のリソースの JNDI 名を指定できます。

このどちらの方法を選択するかによって、WebSphere MQ 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 20. WebSphere MQ インポート: サーバー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	mq.module	my/import	mq.module/my/import_MQ_CF
応答リスナー・ポート	mq.module	my/import	mq.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	mq.module	my/import	mq.module/my/import_RESP_CF
送信	mq.module	my/import	mq.module/my/import_MQ_SEND_D
受信	mq.module	my/import	mq.module/my/export_MQ_RECEIVE_D
SIB コールバック宛先	mq.module	my/import	mq.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.MQ/Callback_CF

表 21. WebSphere MQ エクスポート: サーバー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	mq.module	my/export	mq.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	mq.module	my/export	mq.module/my/export_LIS_CF

表 21. WebSphere MQ エクスポート: サーバー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 (続き)

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
応答接続ファクトリー	mq.module	my/export	mq.module/my/export_RESP_CF
受信	mq.module	my/export	mq.module/my/export_MQ_RECEIVE_D
送信	mq.module	my/export	mq.module/my/export_MQ_SEND_D
SIB コールバック宛先	mq.module	my/export	mq.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.MQ/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。
- 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、WebSphere 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 つの方法を選択する場合は、WebSphere MQ バインディングが使用するリソースをサーバー上で検出することになるため、これらのリソースを事前にインストールし、インポート・ファイルまたはエクスポート・ファイルで JNDI 名を指定する必要があります。これにより、WebSphere MQ バインディングとリソースの間の関連付けが行われます。

MQ JMS バインディングのパラメーター

MQ JMS バインディングは、すべての必要な接続ファクトリーが作成された状態でインストールできます。または、サーバー上で事前に構成された成果物のセットを指すように設計することもできます。

一般に、MQ JMS バインディングは WebSphere Integration Developer で作成されます。開発時には、MQ JMS バインディングに必要な接続と宛先をインストール時 (コンポーネントがサーバーにインストールされる時) に作成する方法を選択できます。あるいは、MQ JMS バインディングで使用する予定のサーバー上のリソースの JNDI 名を指定できます。

このどちらの方法を選択するかによって、MQ JMS 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 22. MQ JMS インポート: サーバー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

表 23. MQ JMS エクスポート: サーバー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	mqjms.module	my/export	mqjms.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	mqjms.module	my/export	mqjms.module/my/export_LIS_CF
応答接続ファクトリー	mqjms.module	my/export	mqjms.module/my/export_RESP_CF
受信	mqjms.module	my/export	mqjms.module/my/export_MQ_RECEIVE_D
送信	mqjms.module	my/export	mqjms.module/my/export_MQ_SEND_D
SIB コールバック宛先	mqjms.module	my/export	mqjms.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。
- 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 つの方法を選択する場合は、MQ JMS インポートまたはエクスポートが使用するリソースをサーバー上で検出することになるため、これらのリソースを事前にインストールし、インポート・ファイルで JNDI 名を指定する必要があります。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。

JMS バインディングの構成

JMS インポート/エクスポート・バインディングを構成し、リソースの特殊機能を適用できます。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。

始める前に

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。

このタスクについて

JMS バインディングは、サーバー・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。

プロシージャ

1. SCA モジュールを選択します。管理コンソールから「アプリケーション」 → 「SCA モジュール」をクリックした後、*modulename* をクリックします。構成ページが開きます。
2. 「モジュール・コンポーネント」の下で、「インポート」または「エクスポート」を展開します。インストールされているインポートまたはエクスポートのリストが表示されます。
3. インポートまたはエクスポートを展開した後、「バインディング」を展開します。管理可能なバインディングのリストが表示されます。
4. 管理するバインディングをクリックします。バインディングの一般プロパティが表示されます。
 - 「送信リソース」カテゴリーには、「接続ファクトリー」と「送信宛先」が含まれています。
 - 「受信リソース」カテゴリーには、「応答接続ファクトリー」、「リスナー・ポート」、および「アクティベーション・スペック」が含まれています。
 - 「拡張リソース」カテゴリーには、コールバック・リソースとその他の使用可能なリソースが含まれています。

注: リソースにアクセスするもう 1 つの方法として、テキスト・ボックスに JNDI 名を入力する方法があります。ただしこの方法では、まだ構成されていないリソースの名前の入力も可能になります。

5. 必要なリソースを管理します。
 - a. 「参照...」 ボタンをクリックすると、JNDI 名のリストを含むウィンドウが開きます。このウィンドウで、必要な JNDI 名を選択して「選択」をクリックします。
 - b. 「構成...」 ボタンをクリックすると、この JNDI 名によって参照される、対応するパネルが開きます。大部分のリソースはクラスター有効範囲で構成可

能ですが、リスナー・ポートで「構成...」オプションを選択すると、すべてのリスナー・ポートと、特定のクラスターのすべてのメンバーのクラスター・リスナー・ポート名を示すパネルが表示されます。このパネルでは、リスナー・ポートを 1 つ選択できます。

「構成...」を選択すると、対応するサーバー・パネルが開きます。

6. 設定を管理する準備ができている場合は、「適用」または「OK」をクリックします。

次のタスク

JNDI バインディングを反映するため、アプリケーションが自動的に再始動されます。

汎用 JMS バインディングの構成

汎用 JMS インポート・バインディングを管理し、リソースの特殊機能を構成できます。管理用タスクは、管理コンソールを使用して実行します。

始める前に

管理コンソールでプロファイルを変更し、変更内容を保存するための権限が必要です。また、接続セットアップ手順を完了している必要があります。

このタスクについて

汎用 JMS インポートは、サーバー・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。

プロシージャ

1. SCA モジュールを選択します。管理コンソールから「アプリケーション」 → 「SCA モジュール」を選択した後、*modulename* を選択します。構成ページが開きます。
2. 「モジュール・コンポーネント」の下で、「インポート」または「エクスポート」を展開します。インストールされているインポートまたはエクスポートのリストが表示されます。
3. インポートまたはエクスポートを展開した後、「バインディング」を展開します。管理可能なバインディングのリストが表示されます。
4. 管理するバインディングをクリックします。バインディングの一般プロパティが表示されます。
 - 「送信リソース」カテゴリには、「接続ファクトリー」と「送信宛先」が含まれています。
 - 「受信リソース」カテゴリには、「応答接続ファクトリー」、「リスナー・ポート」、および「アクティベーション・スペック」が含まれています。
 - 「拡張リソース」カテゴリには、コールバック・リソースとその他の使用可能なリソースが含まれています。

注: リソースにアクセスするもう 1 つの方法として、テキスト・ボックスに JNDI 名を入力する方法があります。ただしこの方法では、まだ構成されていないリソースの名前の入力も可能になります。

5. 必要なリソースを管理します。
 - a. 「参照...」 ボタンをクリックすると、JNDI 名のリストを含むウィンドウが開きます。このウィンドウで、必要な JNDI 名を選択します。 選択された名前が、該当するテキスト・フィールドに取り込まれます。
 - b. 「構成...」 ボタンをクリックすると、この JNDI 名によって参照される、対応するパネルが開きます。 大部分のリソースはクラスター有効範囲で構成可能ですが、リスナー・ポートで「構成...」 オプションを選択すると、すべてのリスナー・ポートと、特定のクラスターのすべてのメンバーのクラスター・リスナー・ポート名を示すパネルが表示されます。このパネルでは、リスナー・ポートを 1 つ選択できます。

「構成...」 を選択すると、対応する WebSphere Application Server パネルが開きます。

6. 設定を管理する準備ができている場合は、「適用」または「OK」 をクリックします。

次のタスク

JNDI バインディングを反映するため、アプリケーションが自動的に再始動されます。

MQ JMS バインディングの構成

MQ JMS バインディングを管理し、リソースの特殊機能を構成できます。管理用タスクは、管理コンソールを使用して実行します。

始める前に

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ません。WebSphere MQ 管理者が WebSphere MQ 内で作成する必要があります。

このタスクについて

MQ JMS インポートまたはエクスポートは、サーバー・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。

プロシージャ

1. SCA モジュールを選択します。 管理コンソールから「アプリケーション」 → 「SCA モジュール」 をクリックした後、*modulename* をクリックします。 構成ページが開きます。

2. 「モジュール・コンポーネント」の下で、「インポート」または「エクスポート」を展開します。インストールされているインポートまたはエクスポートのリストが表示されます。
3. インポートまたはエクスポートを展開した後、「バインディング」を展開します。管理可能なバインディングのリストが表示されます。
4. 管理するバインディングをクリックします。バインディングの一般プロパティが表示されます。
 - 「送信リソース」カテゴリには、「接続ファクトリー」と「送信宛先」が含まれています。
 - 「受信リソース」カテゴリには、「応答接続ファクトリー」、「リスナー・ポート」、および「アクティベーション・スペック」が含まれています。
 - 「拡張リソース」カテゴリには、コールバック・リソースとその他の使用可能なリソースが含まれています。

注: リソースにアクセスするもう 1 つの方法として、テキスト・ボックスに JNDI 名を入力する方法があります。ただしこの方法では、まだ構成されていないリソースの名前の入力も可能になります。

5. 必要なリソースを管理します。
 - a. 「参照...」 ボタンをクリックすると、JNDI 名のリストを含むウィンドウが開きます。このウィンドウで、必要な JNDI 名を選択します。選択された名前が、該当するテキスト・フィールドに取り込まれます。
 - b. 「構成...」 ボタンをクリックすると、この JNDI 名によって参照される、対応するパネルが開きます。大部分のリソースはクラスター有効範囲で構成可能ですが、リスナー・ポートで「構成...」 オプションを選択すると、すべてのリスナー・ポートと、特定のクラスターのすべてのメンバーのクラスター・リスナー・ポート名を示すパネルが表示されます。このパネルでは、リスナー・ポートを 1 つ選択できます。

「構成...」を選択すると、対応する WebSphere Application Server パネルが開きます。

6. 設定を管理する準備ができている場合は、「適用」または「OK」をクリックします。

次のタスク

JNDI バインディングを反映するため、アプリケーションが自動的に再始動されます。

WebSphere MQ バインディングの構成

WebSphere MQ インポート・バインディングとエクスポート・バインディングを管理して、リソースの特殊機能を調整したり設定したりすることができます。管理用タスクは、管理コンソールを使用して実行します。

始める前に

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。

このタスクについて

WebSphere MQ インポートは、サーバー・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。

プロシージャ

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 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。

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、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 用の関連するキュー接続ファクトリーによって作成されます。

5. 変更内容をマスター・プロファイルに保管し、必要に応じてサーバーを再始動します。

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。

以下の JMS バインディングがあります。

- JMS JCA 1.5 に準拠したサービス統合バス (SIB) プロバイダー・バインディング (JMS バインディング)
- JMS 1.1 に準拠した非 JCA の汎用 JMS バインディング (汎用バインディング)
- WebSphere MQ JMS バインディング。WebSphere MQ に対し JMS プロバイダー・サポートを提供し、J2EE アプリケーションとの相互運用性を実現します。(WebSphere MQ JMS バインディング)

JMS バインディングを介して提供されるアプリケーションにより、Service Component Architecture (SCA) モジュールは、外部 JMS システムに対する呼び出しを実行したり、外部 JMS システムからメッセージを受信したりできます。

また、WebSphere MQ バインディング (WebSphere MQ バインディング) もサポートされています。このバインディングにより、ネイティブ MQ ユーザーは、着信メッセージと発信メッセージの任意のフォーマットを処理できます (WebSphere MQ が必要です)。

JMS アプリケーションは、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。

関連タスク

237 ページの『イベントの順序付けの有効化: JMS エクスポート』
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。

汎用 JMS バインディング

汎用 JMS インポート・バインディングとエクスポート・バインディングにより、サード・パーティーの JMS 1.1 準拠プロバイダーに対する接続が提供されます。このバインディングの操作は、JMS バインディングの操作と似ています。

JMS バインディングを介して提供されるサービスにより、Service Component Architecture (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、外部システムからメッセージを受信したりできます。このシステムとして、外部 JMS システムを使用できます。

汎用 JMS バインディング・アプリケーションは、JCA 1.5 非準拠 JMS プロバイダー (JMS プロバイダーが JMS 1.1 をサポートし、オプションの JMS Application Server Facility を実装している場合) との統合を実現します。汎用 JMS バインディングは、JCA 1.5 には対応していないが、JMS 1.1 仕様の Application Server Facility を実装している JMS プロバイダーをサポートしています。これには、Oracle AQ、TIBCO、SonicMQ、WebMethods、BEA WebLogic、WebSphere MQ などが含まれます。SIB は JCA 1.5 JMS プロバイダーであるため、サポートされていません。

ユーザーは、SCA 環境内で JCA 1.5 非準拠の JMS ベース・システムと統合するときにこの汎用アプリケーションを使用できます。これにより、ターゲットの外部アプリケーションがメッセージを送受信でき、SCA コンポーネントと統合できます。

関連タスク

240 ページの『イベントの順序付けの有効化: 汎用 JMS エクスポート』
汎用 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 バインディング

WebSphere MQ JMS バインディングは、WebSphere MQ JMS ベースの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・アプリケーションは、サーバー環境から外部 JMS または MQ JMS 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 からの インポートでは、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 への エクスポートでは、要求の受信と応答の送信が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します。

関連概念

210 ページの『WebSphere MQ バインディング』

WebSphere MQ アプリケーションは、WebSphere MQ アプリケーションと Service Component Architecture との接続を提供します。

関連タスク

239 ページの『イベントの順序付けの有効化: WebSphere MQ JMS エクスポート』

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 バインディング

WebSphere MQ アプリケーションは、WebSphere MQ アプリケーションと Service Component Architecture との接続を提供します。

WebSphere MQ プロバイダー・アプリケーションは、サーバー環境から 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 からの インポートでは、データの送信先の宛先と応答を受信できる宛先が使用されます。データと MQ メッセージの間の相互変換は、MQ ヘッダー・バインディングおよび本体データ・バインディングを使用して行われます。

WebSphere MQ への エクスポートでは、要求の受信と応答の送信ができる宛先を使用します。データと MQ メッセージの間の相互変換は、MQ ヘッダー・バインディングおよび本体データ・バインディングを使用して行われます。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します。

関連概念

201 ページの『WebSphere MQ JMS バインディング』

WebSphere MQ JMS バインディングは、WebSphere MQ JMS ベースのプロバイダーとの統合を提供します。

第 8 章 アダプター

WebSphere Process Server では、WebSphere Adapters と WebSphere Business Integration Adapters の 2 種類のアダプターがサポートされています。ビジネス・アプリケーションは、アダプターを利用することにより、さまざまなエンタープライズ情報システム (EIS) (データベース、エンタープライズ・リソース・プランニング・システム、ファイル・システム、電子メール・システムなど) へ接続して、サービスとして機能します。

アプリケーションと EIS は、アダプターを利用することにより、「相互に対話」できます。つまり、一貫した方法で情報を送受信できます。アプリケーションをサービスとして機能させるために、アダプターはアプリケーションを WebSphere Process Server に接続します。これにより、Service Oriented Architecture (SOA) が実現します。アダプターを使用すれば、各 EIS または アプリケーション・サーバー用に専用の接続ユーティリティを提供する (またはカスタム接続ユーティリティを作成する) 必要はなくなります。

WebSphere Adapter 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 Adapter 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ります。

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 24 と 25 では、2 つのタイプのアダプターについてサーバーと EIS 間の通信の詳細を示しています。

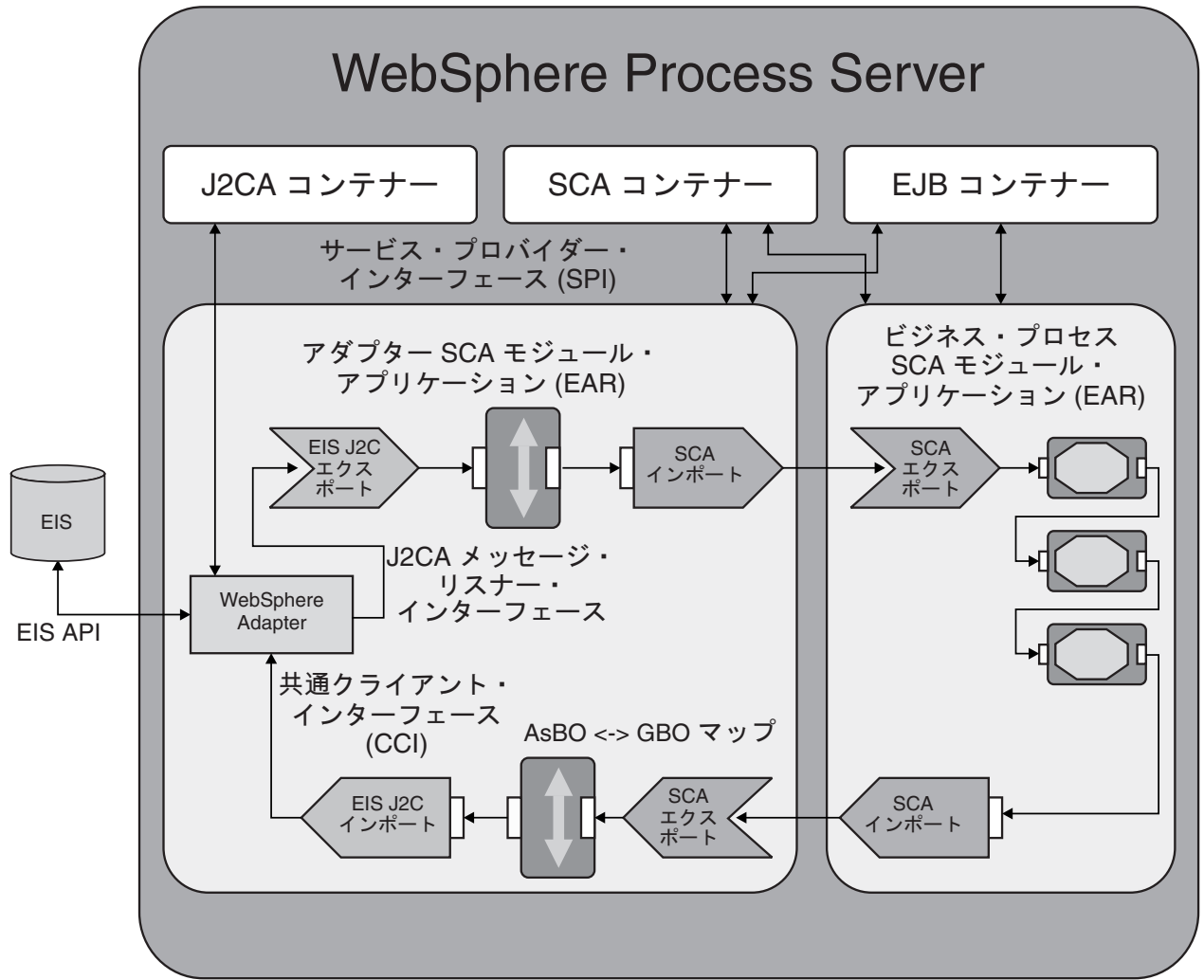


図 24. WebSphere Adapter の詳細な図式

図 24 は、サーバーでサポートされる J2EE コンポーネントと EIS 間の接続を管理する WebSphere Adapter を示しています。WebSphere Adapter はサーバー内にあります。

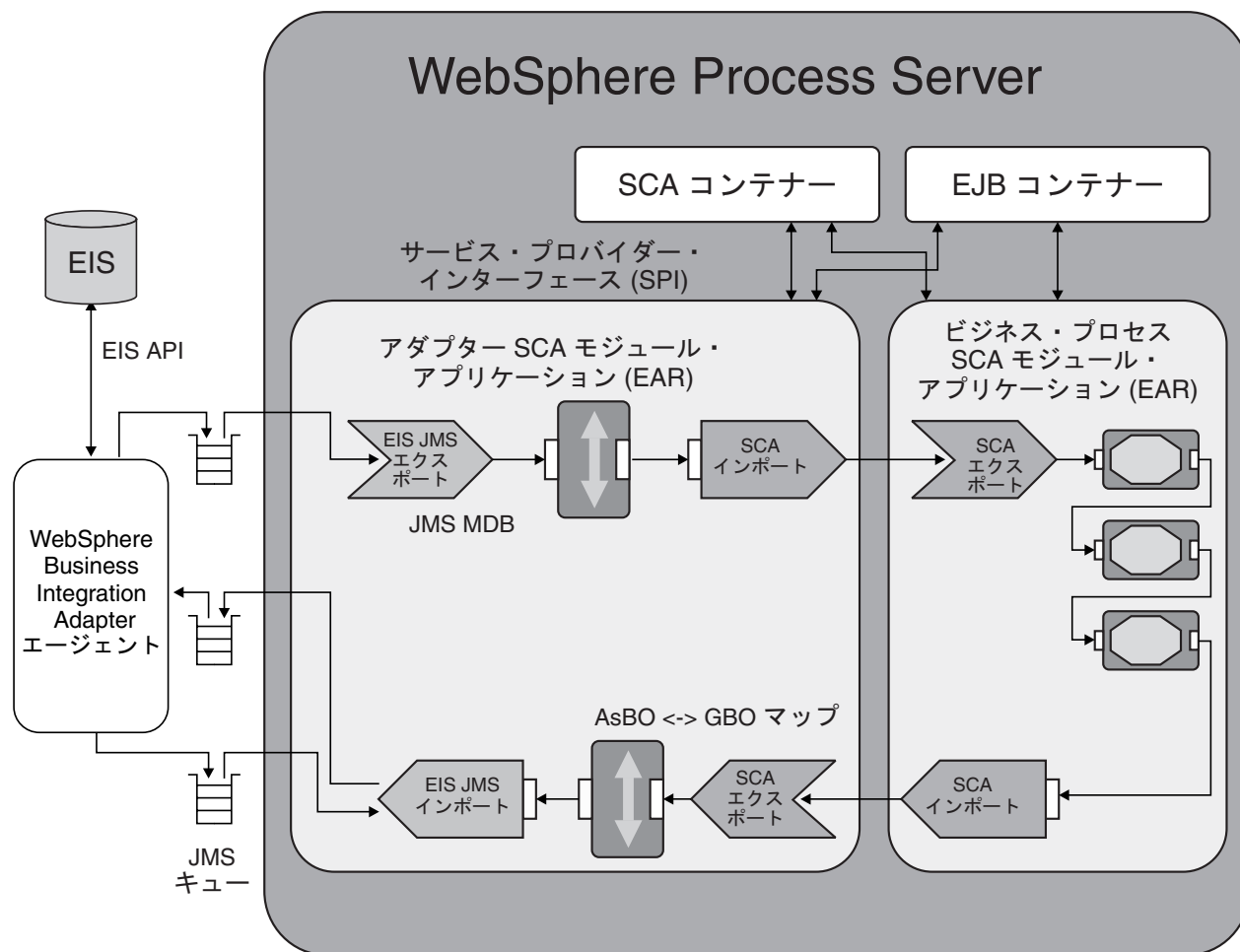


図 25. WebSphere Business Integration Adapter の詳細な図式

図 25 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 Message Service (JMS) トランスポート・レイヤーを使用することにより、WebSphere Business Integration Adapter と通信します。

表 24 に、2 つのアダプター・タイプの相違点を示します。

表 24. WebSphere Adapter 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。

表 24. WebSphere Adapter 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
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。	統合サーバーおよび WebSphere Adapter Framework を利用して、処理要求を開始および支援します。
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 Adapter 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	サーバーを実行します。	サーバー外部にあります。サーバーまたは統合ブローカーは、Java Message Service (JMS) トランスポート・レイヤーを介してアダプターと通信します。

WebSphere Adapters は推奨製品です。

WebSphere Adapter

WebSphere Adapter (リソース・アダプターともいう) は、エンタープライズ情報システム (EIS) と、サーバーでサポートされている J2EE コンポーネントの間で、管理された両方向接続を実現します。

WebSphere Adapter は、WebSphere Business Integration Adapter よりも優先されるため、この情報ライブラリーでは WebSphere Adapter について説明します。

詳細情報の参照先

WebSphere Adapters の構成方法と使用方法については、WebSphere Integration Developer インフォメーション・センターで、『アダプターの構成と使用 (Configuring and using adapters)』を参照してください。ご使用のアダプターのアダプター・ガイドで、ナビゲーションを展開し「**アダプター・モジュールの管理 (Administering the adapter module)**」をクリックしてください。

アダプターの一般情報については、WebSphere Integration Developer インフォメーション・センターの『アダプターを使用した外部サービスへのアクセス (Accessing external services with adapters)』を参照してください。

WebSphere Business Integration Adapter

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ています。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。

このセクションで説明する作業は、WebSphere Process Server と特に関連のある WebSphere Business Integration Adapter に関するものです。これらのタスクは、アダプターのインストール手順と構成手順の実行、アダプターの管理機能のセットアップ、および管理コンソールを使用したアダプターの管理です。

詳細情報の参照先

これらのアダプターを使用した作業の詳細については、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。

アダプターのインストールおよび構成

WebSphere Business Integration Adapter と WebSphere Process Server を連携させるには、インストールおよび構成の手順を実行する必要があります。

プロシージャ

1. アダプターをインストールします。
 - a. WebSphere Business Integration Adapters のインストール方法が説明されている『WebSphere Business Integration Adapters 製品のインストール』の手順に従います。
 - b. 『WebSphere Business Integration Adapters の資料』にアクセスした後、「**アダプター**」の下のナビゲーションを展開して、ご使用のアダプターを表示します。追加のインストール・タスクがある場合はそれが表示されるので、ご使用のアダプターに必要な特定の手順が他にあれば、その手順に従います。
2. アダプターを構成します。『WebSphere Business Integration Adapters の資料』にアクセスした後、「**アダプター**」の下のナビゲーションを展開して、ご使用のアダプターを表示したら、そのアダプターの構成の指示に従います。構成手順を実行すると、必要な成果物が生成されます。
3. 「モジュールの開発とデプロイ」(PDF) の『実動サーバーへのモジュールのインストール』の指示に従って、アプリケーション EAR ファイルをインストールします。

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。

始める前に

- 『WebSphere Business Integration Adapters 製品のインストール』に示されている手順を理解している必要があります。
- 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。

このタスクについて

WebSphere Business Integration Adapter を管理するには、以下の管理機能を実行します。

プロシージャ

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」を展開します。
- c. 「キュー接続ファクトリー (Queue connection factories)」を選択します。
- d. 「新規」をクリックして、新規 JMS キュー接続ファクトリーを作成します。管理入出力キューの有効範囲レベルに対応する有効範囲レベルを選択します。
- e. JMS リソース・プロバイダーを選択します。「デフォルトのメッセージング・プロバイダー」を選択します。

「OK」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. 新規の 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 サービスを使用可能とする予定の各サーバーごとに、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

実行中の WebSphere Business Integration Adapter を管理コンソールから管理できません。

始める前に

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。

このタスクについて

リソースを管理し、リソースに対してさまざまな管理アクションを実行するには、以下の手順に従います。

プロシージャ

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。
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。

第 9 章 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

WebSphere Process Server のアプリケーションでは、WebSphere Application Server の J2EE アプリケーションの場合と同様の管理タスクとインターフェースが必要ですが、その他に、サービス・アプリケーションや WebSphere MQ 宛先などのリソースに個別に関連する追加タスクがあります。アプリケーション・サービスの管理には、ビジネス・プロセスとタスク、ビジネス・ルール、およびスケジュールが含まれます。

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。

このタスクについて

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。

このタスクについて

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。

アプリケーションの統合環境

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 のエンタープライズ・アーカイブ (EAR) 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、任意の 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では、サービス・アプリケーションおよびそれに関連した SCA モジュールの詳細を表示することができます。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 モジュールの詳細は、SCA モジュールによって異なります。これには以下の属性が含まれます。

- SCA モジュール名
- SCA モジュールの説明
- 関連したアプリケーション名
- SCA モジュール・インポート:
 - 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。
 - インポート・バインディングは、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する具象定義です。例えば、SOAP/HTTP を使用します。
- SCA モジュール・エクスポート:
 - 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。
 - エクスポート・バインディングは、サービス要求元が SCA モジュールにアクセスする (それによって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する具象定義です。
- SCA モジュール・プロパティー

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。

このタスクについて

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、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る Service Component Architecture (SCA) モジュールです。

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・プロバイダー間で伝送中のメッセージを操作します。メッセージを異なるサービス・プロバイダーにルーティングでき、またメッセージの内容または形式を修正することもできます。メディエーション・モジュールは、ユーザーの要件に合わせて調整したメッセージ・ロギングやエラー処理などの機能を提供できます。

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・モジュールには、以下のアイテムが含まれています。

- **インポート:** SCA モジュールとサービス・プロバイダーの間の対話を定義します。これにより、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。WebSphere Process Server を使用してメディエーション・モジュール・インポートを表示し、バインディングを変更できます。
- **エクスポート:** SCA モジュールとサービス要求元の間の対話を定義します。これにより、SCA モジュールでサービスを提供し、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) を定義することが可能になります。メディエーション・モジュール・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:** メディエーション・モジュールなどの SCA モジュールを構築するための基本単位です。SCA モジュールとコンポーネントは、WebSphere Integration Developer を使用してグラフィカルに作成およびカスタマイズできます。メディエーション・モジュールのデプロイ後に、WebSphere Process Server の管理コンソールから、モジュールを再デプロイせずにその属性を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 コンポーネントを含みます。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。1 つのメディエーション・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 1 つのみです。

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・プリミティブを 1 つまたは複数含めることができますが、まったく含めないことも可能です。WebSphere Process Server では、メッセージ・ルーティングおよび変換のための機能を提供する、あらかじめ用意された 1 組のメディエーション・プリミティブがサポートされます。メディエーション・プリミティブをより柔軟に使用する必要がある場合は、カスタム・メディエーション・プリミティブを使用して、カスタム・ロジックを呼び出すことができます。

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、サービス要求を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: WebSphere Process Server から、メディエーション・モジュールを表示して、一定の変更を加えることができます。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。

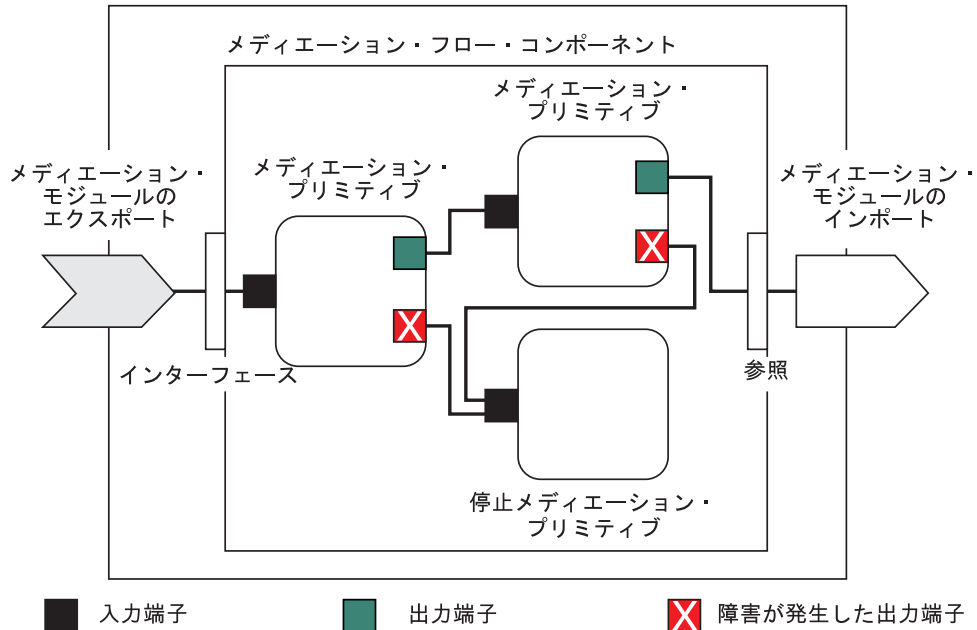


図26. メディエーション・モジュールの簡単な例： メディエーション・モジュールには、メディエーション・プリミティブが含まれた 1 つのメディエーション・フロー・コンポーネントが含まれています。

- 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・プリミティブのプロパティを WebSphere Process Server 管理コンソールで表示するには、統合開発者はこれらの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。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。他のプロパティは、変更するとメディエーション・フローに影響が生じ、メディエーション・モジュールの再デプロイが必要になるおそれがあるため、管理構成には適しません。

WebSphere Integration Developer 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、プロモート対象として選択できるプロパティのリストを示します。

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。

プロパティの変更がデプロイメント・マネージャー・セル内で行われた場合を除き、新たに呼び出されたメディエーション・フローでは、この変更内容がただちに使用されます。デプロイメント・マネージャー・セル内に変更が生じた場合、各ノードで同期化が行われるとそのノード上で変更が有効になります。処理中のメディエーション・フローでは、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、メディエーション・プリミティブのプロパティ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 WebSphere Process Server のエンタープライズ・アーカイブ (EAR) ファイル内にデプロイされます。

プロモートされるプロパティの値は、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すると、WebSphere Integration Developer にメディエーション・モジュールを Java アーカイブ (JAR) ファイル内にパッケージ化させ、その JAR ファイルを EAR ファイル内にパッケージ化させることができます。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

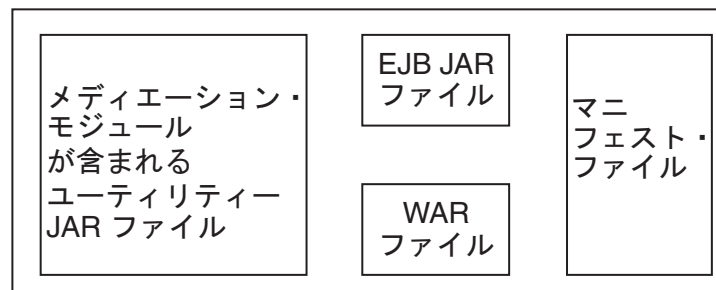


図 27. メディエーション・モジュールが含まれる EAR ファイルの簡単な例: EAR ファイルには JAR が含まれます。ユーティリティ JAR ファイルにはメディエーション・モジュールが含まれます。

関連情報

メディエーション・モジュールの使用

WebSphere Process Server にデプロイされているメディエーション・モジュールを表示することができます。メディエーション・モジュールを始動または停止したり、メディエーション・モジュールやそのアプリケーションの詳細を表示したりすることができます。

301 ページの『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。

302 ページの『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。

プロモート可能なプロパティ

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。SCA モジュールは、インポートを使用することにより、コンポーネントがローカルの表記を使用して外部サービス (SCA モジュールの外部にあるサービス) にアクセスできるようにします。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。

SCA モジュールは、外部サービスにアクセスする必要がある場合は、インポートを持つ必要はありません。メディエーション・モジュールには、通常、意図するターゲットにメッセージまたは要求を渡すために使用するインポートが 1 つ以上あります。

インターフェースとバインディング

SCA モジュール・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、1 つの SCA モジュール・インポートには単一のバインディングがあります。

- インポート・インターフェースは、Web サービスを記述するための XML 言語である Web サービス記述言語 (WSDL) を使用して、操作設定を定義する抽象定義です。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。
- インポート・バインディングとは、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メカニズムを指定する具象定義です。

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、以下のインポート・バインディングをサポートしています。

- Web サービス・バインディングにより、コンポーネントが Web サービスを起動できるようになります。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 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、TCP/IP や HTTP または HTTPS などのさまざまなトランスポート・タイプを活用します。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。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 を JMS プロバイダーとして使用する際に、WebSphere MQ JMS バインディングが必要になる場合があります。
- 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。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。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。

- 汎用 JMS バインディングにより、JMS Application Server Facility (ASF) を使用して WebSphere Application Server を統合するサード・パーティーの JMS プロバイダーとの相互運用が可能になります。
- WebSphere Adapter バインディングによって、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。
- HTTP バインディングでは、HTTP プロトコルを使用して アプリケーションにアクセスできます。

関連情報

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、インポート・バインディングの詳細を変更したりすることもできます。

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。SCA モジュールはエクスポートを使用して、他のモジュールにサービスを提供します。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。

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。

- 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ための XML 言語である Web サービス記述言語 (WSDL) を使用して、操作設定を定義する抽象定義です。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。
- エクスポート・バインディングとは、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メカニズムを指定する具象定義です。通常、SCA モジュールのエクスポートには、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、SCA バインディングを持つエクスポートとしてランタイムによって解釈されます。

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、以下のエクスポート・バインディングをサポートしています。

- Web サービス・バインディングにより、エクスポートを Web サービスとして起動できるようになります。サポートされるプロトコルは、SOAP/HTTP と SOAP/JMS です。
- 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。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 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。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 を JMS プロバイダーとして使用する際に、WebSphere MQ JMS バインディングが必要になる場合があります。
- 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。リモート (つまりクライアント) 接続とは、リモートのマシンの MQ キュー・マネージャーに接続する必要がある接続タイプのことです。ローカル (つまりバインディング) 接続とは、WebSphere MQ への直接接続のことです。これは同じマシン上の MQ キュー・マネージャーへの接続にのみ使用できます。WebSphere MQ では、両方のタイプの接続が許可されますが、MQ バインディングでは、「リモート」(つまり「クライアント」) 接続のみがサポートされます。
- 汎用 JMS バインディングにより、JMS Application Server Facility (ASF) を使用して WebSphere Application Server を統合するサード・パーティーの JMS プロバイダーとの相互運用が可能になります。
- WebSphere Adapter バインディングによって、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。
- HTTP バインディングにより、HTTP プロトコルを使用してエクスポートにアクセスできるようになります。

関連情報

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。

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

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含まれます。

- 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

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。

- SCA タイプのインポート・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。
- Web サービス (WS) タイプのインポート・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。
- JMS、WebSphere MQ JMS、汎用 JMS、WebSphere MQ、および HTTP の各タイプのエクスポート・バインディングおよびインポート・バインディングの変更できる属性
- メディエーション・モジュール・プロパティ
 - メディエーション・モジュールのプロパティは、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の一部は SCA モジュールの追加プロパティとして表示されます。WebSphere Process Server からメディエーション・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。

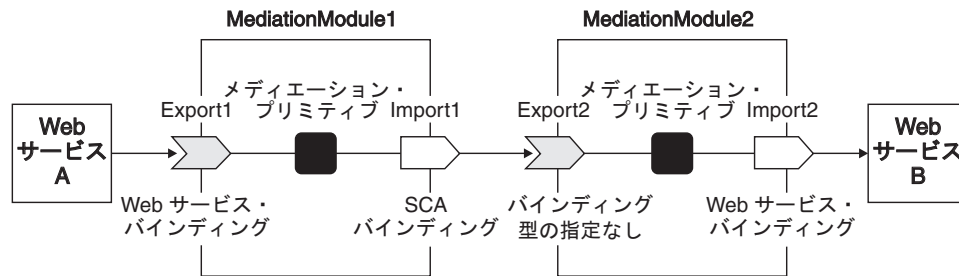


図 28. 別のメディエーション・モジュールと対話するメディエーション・モジュールの例: Mediation Module1 は Mediation Module2 に接続します。

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。

始める前に

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センブ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 Network Deployment のインフォメーション・センターを参照してください。

プロシージャ

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。

結果

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。

次のタスク

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。

関連概念

[WebSphere JMS バインディング](#)

[WebSphere MQ バインディング](#)

関連タスク

 [Common Event Infrastructure の管理](#)

以下のトピックでは、Common Event Infrastructure コンポーネントの実行時の動作を制御するために実行できる管理タスクをいくつか説明します。

336 ページの『[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。

関連情報

 [serviceDeploy コマンド](#)

serviceDeploy コマンドを使用して、Service Component Architecture (SCA) 対応モジュールを、サーバーにインストール可能な Java アプリケーションとしてパッケージします。このコマンドは、wsadmin を使用してバッチ・インストールを実行する場合に役立ちます。

 [プロモート可能なプロパティ](#)

 [メディエーション・モジュール用リソースの管理](#)

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が提供す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。

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・モジュールの実行を開始する前に、メディエーション・モジュールをサーバーまたはクラスターにデプロイする必要があります。このデプロイメントでは、インストール可能な EAR ファイルを作成し、この 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ます。

始める前に

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、JAR ファイルからインストール可能な EAR ファイルを作成します。詳しくは、294 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。

このタスクについて

メディエーション・モジュールの実行を開始する前に、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おく必要があります。

管理コンソールを使用する代わりに、他の方法 (`wsadmin` ツールの `AdminApp.install` コマンドまたは `AdminApp.installinteractive` コマンドなど) で EAR ファイルをインストールすることもできます。

重要: ステップの実行開始後に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」をクリックする必要があります。そのまま他の管理コンソール・ページには移動しないでください。

プロシージャ

1. 管理コンソールで EAR ファイルをインストールします。そのためには、コンソールのナビゲーション・ペインで「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 ページで構成される「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. 「アプリケーション・インストールの準備」の最初のページで、次のステップを実行します。
 - a. EAR ファイルの絶対パス名を指定します。詳しくは、『コンソールでのアプリケーションのインストール (Installing applications with the console)』を参照してください。
 - b. デフォルト値を使用するか、一部の値を自分で指定するかを以下から選択します。

追加情報が必要な場合のみプロンプトを出す

モジュール・マッピング・ステップ、および情報を指定する必要があるその他のステップのみを表示します。

すべてのインストール・オプションおよびパラメータを表示する

すべてのインストール・ステップを表示します。不完全なバインディングに対してデフォルト値を指定する「デフォルト・バインディングの生成」を使用するには、このオプションを選択します。

- c. 「次へ」をクリックします。

3. 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方法は、WebSphere Application Server へのその他のエンタープライズ・アプリケーション EAR ファイルのインストール方法に似ています。2 番目の「**アプリケーション・インストールの準備**」ページの入力方法と、ウィザードの残りのステップでのオプションの指定方法の詳細については、『**コンソールでのアプリケーションのインストール (Installing applications with the console)**』を参照してください。
4. メディエーション・モジュールの EAR ファイルについては、実行できるオプションのステップがもう 1 つあります。「**モジュール・プロパティの編集**」パネルで、モジュールのプロパティ値を編集できます。

結果

これで、メディエーション・モジュールを開始することができます。

Apache 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、Apache™ 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。Apache 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。

始める前に

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。
- Apache 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 Network Deployment 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。Apache 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。

このタスクについて

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 Apache 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、Apache 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。

プロシージャ

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 Apache 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 Apache Ant タスクを作成します。
6. ServiceDeploy Apache 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 Apache 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。

結果

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 Apache 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、Apache 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。

次のタスク

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し、個々のメディエーション・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ができます。

このタスクについて

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 SCA モジュールのタイプです。

メディエーション・モジュールの使用

WebSphere Process Server にデプロイされているメディエーション・モジュールを表示することができます。メディエーション・モジュールを始動または停止したり、メディエーション・モジュールやそのアプリケーションの詳細を表示したりすることができます。

このタスクについて

デプロイ済みのメディエーション・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 モジュールのリストを表示するには、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。

結果

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。

関連情報

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、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る Service Component Architecture (SCA) モジュールです。

 プロモート可能なプロパティ

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。

このタスクについて

デプロイ済みのメディエーション・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。

プロシージャ

1. 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」を展開して、SCA モジュールを表示します。
2. コンテンツ・ペインで SCA モジュールをクリックして、SCA モジュールを選択します。

結果

コンテンツ・ペインには、SCA モジュールの名前と説明、関連するエンタープライズ・アプリケーションの名前、インポートおよびエクスポートの展開可能なリスト、およびモジュール・プロパティ・リンクが表示されます。

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。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールを表示します。
2. 「アプリケーション」というラベルの付いた列で、アプリケーション名をクリックして SCA モジュールを選択します。

結果

コンテンツ・ペインにはアプリケーションの詳細ページが表示され、このページに、アプリケーションの構成プロパティとローカル・トポロジー (該当する場合) が記載されています。

『アプリケーションの構成』で説明されているように、このページから、アプリケーション用の構成プロパティを確認し、必要に応じて変更したり、その他のコンソール・ページにリンクしたりすることができます。

メディエーション・モジュールの開始と停止:

停止状況のメディエーション・モジュールを開始したり、実行中 (開始済み状況) 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・モジュールの状況を変更するには、モジュールをデプロイするために使用されるアプリケーションを開始または停止します。

始める前に

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、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。

このタスクについて

メディエーション・モジュールおよび関連す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。

手動でアプリケーションを開始および停止するには、以下を使用します。

- 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、以下のステップを実行する管理コンソールを使用します。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
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 (<i>partially started</i>)」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。

4. 「停止」をクリックするか、再始動するアプリケーションを選択し、次に「開始」をクリックして実行中のアプリケーションを再始動します。

結果

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。

このタスクについて

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。

デプロイ済みメディエーション・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。

プロシージャ

1. 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」を展開して、SCA モジュールを表示します。
2. SCA モジュールを選択するには、コンテンツ・ペインで必要な SCA モジュールをクリックします。
3. コンテンツ・ペインの「追加プロパティ」で「モジュール・プロパティ」をクリックして、SCA モジュール・プロパティのリストを表示します。

結果

コンテンツ・ペインで、SCA モジュールの更新可能なプロパティが、プロパティの名前、タイプ、および値を記載したテーブルに表示されます。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変更するには、WebSphere Integration Developer を使用します。更新できるプロパティがない場合はメッセージが表示されます。

関連情報

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、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る Service Component Architecture (SCA) モジュールです。

 プロモート可能なプロパティ

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。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
2. SCA モジュールを選択するには、コンテンツ・ペインで SCA モジュールをクリックします。
3. 「追加プロパティ」で、コンテンツ・ペインの「モジュール・プロパティ」を選択し、SCA モジュール・プロパティを表示します。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。管理コンソールからはプロパティ値の更新しかできません。プロパティ名とプロパティ・タイプを変更するには、WebSphere Integration Developer を使用してください。

4. プロパティ値を選択するには、「プロパティ」テーブルでプロパティ値をクリックします。
5. プロパティ・タイプに準拠する値を入力して、プロパティ値を変更します。
6. 「OK」をクリックして変更を保存します。次に、変更をマスター構成に保管します。

結果

プロパティ値が変更されました。プロパティの変更がデプロイメント・マネージャー・セル内で行われた場合を除き、新たに呼び出されたメディエーション・フローでは、この変更内容がただちに使用されます。デプロイメント・マネージャー・セル内に変更が生じた場合、セル内の各ノードで同期化が行われるとそのノード上で変更が有効になります。プロパティ値の変更時点で処理中のメディエーション・フローは、前の値を引き続き使用します。

注: プロパティ値ではなく、プロパティ名およびタイプを変更したい場合は、WebSphere Integration Developer を使用してください。

関連情報

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、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る Service Component Architecture (SCA) モジュールです。

 プロモート可能なプロパティ

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、インポート・バインディングの詳細を変更したりすることもできます。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
2. 使用する SCA モジュールを選択します。
3. コンテンツ・ペインのモジュール・コンポーネントの下で「インポート」を展開して、SCA モジュール・インポートをリストします。

結果

インポートのリストがコンテンツ・ペインに表示されます。インポートがない場合、情報メッセージが表示されます。

関連概念

221 ページの『HTTP バインディング』

HTTP バインディングは、Service Component Architecture (SCA) と HTTP の接続を提供する目的で設計されています。これにより、既存のまたは新規作成された HTTP アプリケーションをサービス指向アーキテクチャー (SOA) 環境内に組み込むことができます。

関連情報

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。SCA モジュールは、インポートを使用することにより、コンポーネントがローカルの表記を使用して外部サービス (SCA モジュールの外部にあるサービス) にアクセスできるようにします。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。

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。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
2. SCA モジュールを選択します。
3. コンテンツ・ペインで、モジュール・コンポーネントの「インポート」を展開し、SCA モジュール・インポートのリストを表示します。
4. コンテンツ・ペインのモジュール・コンポーネントで、必要なインポートを展開してインポートの詳細を表示します。
5. 「インターフェース」を展開して、インポート・インターフェースを表示します。
6. インターフェースを選択します。

結果

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 にデプロイ後は、Web サービス、SCA、JMS、アダプターの各タイプのインポート・バインディングを表示できます。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
2. SCA モジュールを選択します。
3. コンテンツ・ペインで、モジュール・コンポーネントの「インポート」を展開し、SCA モジュール・インポートのリストを表示します。
4. コンテンツ・ペインのモジュール・コンポーネントで、必要なインポートを展開してインポートの詳細を表示します。
5. 「バインディング」を展開して、インポート・バインディングを表示します。
6. バインディングを選択します。

結果

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。

異なるターゲット・モジュールを選択する場合、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更されます。

関連概念

173 ページの『バインディング』

バインディングにより、さまざまなプロトコルとトランスポートを使用して対話できるように SCA インポートとエクスポートが構成されます。SCA アプリケーションは、選択されたバインディングのトランスポートとプロトコルを使用して、外部クライアントとサービスに接続します。バインディングは、特定のインターフェースのメッセージのフォーマットとプロトコルの詳細を指定します。

SCA インポート・バインディングの変更:

管理コンソールで、SCA インポート・バインディングの詳細を変更できます。その他のタイプのバインディングの詳細の変更方法については、『バインディング』のセクションを参照してください。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示をします。
2. SCA モジュールを選択します。
3. モジュール・コンポーネントのコンテンツ・ペインで「インポート」を展開し、SCA モジュール・インポートのリストを表示します。
4. モジュール・コンポーネントのコンテンツ・ペインで、必要なインポートを展開し、インポートの詳細を表示します。
5. 「バインディング」を展開して、インポート・バインディングを表示します。
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」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。

結果

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。

関連概念

173 ページの『バインディング』

バインディングにより、さまざまなプロトコルとトランスポートを使用して対話できるように SCA インポートとエクスポートが構成されます。SCA アプリケーションは、選択されたバインディングのトランスポートとプロトコルを使用して、外部クライアントとサービスに接続します。バインディングは、特定のインターフェースのメッセージのフォーマットとプロトコルの詳細を指定します。

Web サービス・インポート・バインディングの変更:

管理コンソールを使用して、Web サービス・インポート・バインディングのエンドポイント URL を変更できます。その他のタイプのバインディングの詳細の変更方法については、『バインディング』のセクションを参照してください。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
2. SCA モジュールを選択します。
3. モジュール・コンポーネントのコンテンツ・ペインで「インポート」を展開し、SCA モジュール・インポートのリストを表示します。
4. モジュール・コンポーネントのコンテンツ・ペインで、必要なインポートを展開し、インポートの詳細を表示します。
5. 「バインディング」を展開して、インポート・バインディングを表示します。

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。

結果

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。

変更内容は、マスター構成を更新し SCA モジュールを再始動した後で有効になります。

関連概念

173 ページの『バインディング』

バインディングにより、さまざまなプロトコルとトランスポートを使用して対話できるように SCA インポートとエクスポートが構成されます。SCA アプリケーションは、選択されたバインディングのトランスポートとプロトコルを使用して、外部クライアントとサービスに接続します。バインディングは、特定のインターフェースのメッセージのフォーマットとプロトコルの詳細を指定します。

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。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
2. 参照したい特定の SCA モジュールを選択します。
3. コンテンツ・ペインの モジュール・コンポーネントの下で「エクスポート」を展開し、SCA モジュール・エクスポートのリストを表示します。

結果

エクスポートのリストがコンテンツ・ペインに表示されます。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。

関連概念

221 ページの『HTTP バインディング』

HTTP バインディングは、Service Component Architecture (SCA) と HTTP の接続を提供する目的で設計されています。これにより、既存のまたは新規作成された HTTP アプリケーションをサービス指向アーキテクチャー (SOA) 環境内に組み込むことができます。

関連情報

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。SCA モジュールはエクスポートを使用して、他のモジュールにサービスを提供します。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。

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。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
2. SCA モジュール を選択します。
3. コンテンツ・ペインのモジュール・コンポーネントで「エクスポート」を展開し、SCA モジュール・エクスポートのリストを表示します。
4. コンテンツ・ペインのモジュール・コンポーネントで、必要なエクスポートを展開し、エクスポートの詳細を表示します。
5. 「インターフェース」を展開して、エクスポート・インターフェースを表示します。
6. インターフェースを選択します。

結果

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。

このタスクについて

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。

プロシージャ

1. 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、SCA モジュールのリストを表示します。
2. SCA モジュール を選択します。

3. コンテンツ・ペインで、モジュール・コンポーネントの「エクスポート」を展開し、SCA モジュール・エクスポートのリストを表示します。
4. コンテンツ・ペインのモジュール・コンポーネントで、必要なエクスポートを展開し、エクスポートの詳細を表示します。
5. 「バインディング」を展開して、エクスポート・バインディングを表示します。
6. バインディングを選択します。

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。

始める前に

WSRR インフォメーション・センター (http://publib.boulder.ibm.com/infocenter/sr/v6r1/topic/com.ibm.sr.doc/cwsr_overview_overview.html) の説明に従って、WSRR をインストールおよび構成します。

このタスクについて

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・インターフェースを提供します。この Web サービス・インターフェースにより、ユーザーはエンドポイント・ルックアップ・メディエーション・プリミティブを (メディエーション・フローの一部である) レジストリー・インスタンスに接続して、呼び出すサービスを検索することができます。WSRR 定義およびその接続プロパティは、レジストリー・インスタンスへの接続、および呼び出す Web サービスの検索に使用されるメカニズムです。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。

プロシージャ

ナビゲーション・ペインで、「サービス統合」 → 「WSRR 定義」を展開します。

結果

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・セットが収められます。

次のタスク

WSRR 定義をクリックして、その定義の一般的なプロパティを表示します。コンテンツ・ペインから詳細な接続プロパティにアクセスすることができます。

関連情報



WebSphere Service Registry and Repository

WebSphere Service Registry and Repository (WSRR) 製品では、サービスに関する情報を保管および管理して、この情報にアクセスすることができます。この情報を使用して、サービスの選択、呼び出し、および再利用を行えます。

WSRR 管理コマンドWSRR

管理パネルを使用する代わりに、WebSphere Services Registry and Repository (WSRR) 管理コマンドを使用して、WSRR 定義を構成することができます。

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。

このタスクについて

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・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。

プロシージャ

1. ナビゲーション・ペインで、「サービス統合」→「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・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 定義の説明を入力します (オプション)。

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

「Web サービス」のみを選択できます。

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. 表示されるコンテンツ・ペインの「追加プロパティ」で「接続プロパティ」をクリックします。「接続プロパティ」構成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。

SSL 構成

Secure Sockets Layer (SSL) 構成は、セキュア WSRR インスタンスと通信を行うときに使用します。

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。
9. 「保管」をクリックします。

結果

新しい WSRR 定義が作成されます。

関連タスク

312 ページの『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。

このタスクについて

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。

プロシージャ

1. ナビゲーション・ペインで、「サービス統合」 → 「WSRR 定義」を展開します。コンテンツ・ペインに、すべての WSRR 定義のリストを含む「WSRR 定義」ページが表示されます。
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。

結果

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)。現在のデフォルト定義を削除したい場合、まず、別の定義をデフォルトに設定する必要があります。

関連タスク

『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。

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。

このタスクについて

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。

プロシージャ

1. ナビゲーション・ペインで、「サービス統合」 → 「WSRR 定義」を展開します。コンテンツ・ペインに、すべての WSRR 定義のリストを含む「WSRR 定義」ページが表示されます。
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。以前別の WSRR 定義がデフォルトに設定されていた場合、その WSRR 定義の「デフォルト」列には「いいえ」が表示されています。
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。
4. 「保管」をクリックします。

結果

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。

関連タスク

311 ページの『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。

310 ページの『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。

SCA 要求のスループットの管理

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たり、適宜、何らかの措置を取って SCA 要求のスループットを管理したりすることができます。

このタスクについて

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)

特別な事情により、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・ポイントにある 1 つ以上のメッセージを削除する必要がある場合があります。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。

サービス・アプリケーションのその他の用途

WebSphere 管理コンソールを使用すると、メディエーション・モジュール本体だけでなく、メディエーション・モジュールが使用するリソースや、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まで管理できます。このタスクはコマンドでも実行できます。

このタスクについて

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、298 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が提供す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`

これは、非同期要求をモジュールに送信するためにエクスポートが使用する宛先です。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。

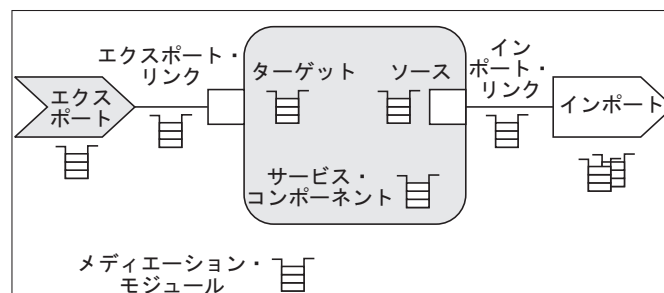
Queue `sca/module_name/importlink/import_name`

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、該当するバス・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ング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。使用される JMS サポートは、モジュールの JMS バインディングによって異なります。例えば、JMS バインディングを含むモジュールは、基盤となる WebSphere Application Server により提供されるデフォルトのメッセージング・プロバイダー上で構成されている JMS 接続ファクトリーを使用し、WebSphere MQ JMS バインディングを含むモジュールは、WebSphere MQ 上で JMS プロバイダーとして構成されている JMS 接続ファクトリーを使用します。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。

関連概念

WebSphere JMS バインディング

WebSphere MQ バインディング

関連タスク

 Common Event Infrastructure の管理

以下のトピックでは、Common Event Infrastructure コンポーネントの実行時の動作を制御するために実行できる管理タスクをいくつか説明します。

336 ページの『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。

関連情報

 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。

 serviceDeploy コマンド

serviceDeploy コマンドを使用して、Service Component Architecture (SCA) 対応モジュールを、サーバーにインストール可能な Java アプリケーションとしてパッケージします。このコマンドは、wsadmin を使用してバッチ・インストールを実行する場合に役立ちます。

 プロモート可能なプロパティ

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。コマンドはスクリプト内で使用できます。

始める前に

wsadmin ツールを使用して、サービス・アプリケーション・コマンドを実行します。

このタスクについて

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している場合は、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 モジュールとそのインポートおよびエクスポートを表示し、インポート・バインディングとエクスポート・バインディングの詳細を変更するコマンドがあります。

注: wsadmin スクリプトの Jacl 構文は使用すべきではありません。今後のリリースでサポートが廃止される予定です。Jython が wsadmin スクリプトの戦略的構文です。

プロシージャ

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
```

関連情報



Jacl



Jython

コマンドを使用したメディエーション・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。

始める前に

wsadmin ツールを使用して WebSphere Process Server コマンドを実行します。

このタスクについて

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。複数のホストを管理する場合や、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。

コマンドを使用したメディエーション・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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

wsadmin ツールを使用してデプロイ済みメディエーション・モジュールをリストするには、以下のステップに従います。

メディエーション・モジュール は Service Component Architecture (SCA) モジュールのタイプであるため、デプロイ済みのメディエーション・モジュールを表示するためには、SCA モジュールをリストします。

プロシージャ

デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`

結果

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。

コマンドを使用したメディエーション・モジュールの詳細の表示:

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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

wsadmin ツールを使用してメディエーション・モジュールの説明を表示するには、以下のステップに従います。

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・モジュール名を知る必要があります。

`listSCAModules` コマンドにより、WebSphere Process Server にデプロイされている SCA モジュールがリストされます。

プロシージャ

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。

結果

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}
```

コマンドを使用したメディエーション・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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

wsadmin ツールを使用してメディエーション・モジュールのプロパティを表示するには、以下のステップに従います。

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・モジュール名を知る必要があります。

listSCAModules コマンドにより、WebSphere Process Server にデプロイされている SCA モジュールがリストされます。

プロシージャ

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。

結果

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}
```

コマンドを使用したメディエーション・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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

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。

プロシージャ

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。

結果

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}
```

コマンドを使用した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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

wsadmin ツールを使用してメディエーション・モジュールのインポートをリストするには、以下のステップに従います。

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・モジュールの名前を知る必要があります。

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。

プロシージャ

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。

結果

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}
```

コマンドを使用したインポートの詳細の表示:

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。

始める前に

`wsadmin` ツールを使用して、以下のコマンドを実行します。

このタスクについて

`wsadmin` ツールを使用して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、以下のステップに従います。

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・モジュール名とインポート名を知る必要があります。

`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。

プロシージャ

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}`

`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。

注: *applicationName* を指定することにより、パフォーマンスが向上します。

結果

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}
```

コマンドを使用したインポート・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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

wsadmin ツールを使用して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、以下のステップに従います。

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・モジュール名とインポート名を知る必要があります。

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。

プロシージャ

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}`

moduleName の指定に加えて、オプションで *applicationName* も指定できます。

注: *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}`

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。

注: *applicationName* を指定することにより、パフォーマンスが向上します。

結果

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}
```

コマンドを使用した 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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

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、各種のメディエーション・モジュールがあります。

特定のメディエーション・モジュールで異なるメディエーション・モジュールを呼び出す場合、SCA インポート・バインディングを変更する場合があります。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。

プロシージャ

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`

オプションで、*applicationName* および *targetApplicationName* も指定できます。

注: *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。

結果

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}
```

コマンドを使用したインポート 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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

特定のメディエーション・モジュールで異なる Web サービスを呼び出す場合、インポート Web サービス・バインディングを変更する場合があります。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。

プロシージャ

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}**

moduleName の指定に加えて、オプションで *applicationName* も指定できます。

注: *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`

結果

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}
```

コマンドを使用した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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

wsadmin ツールを使用してメディエーション・モジュールのエクスポートをリストするには、以下のステップに従います。

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・モジュール名を知る必要があります。

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。

プロシージャ

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}`

moduleName の指定に加えて、オプションで *applicationName* も指定できます。

注: *applicationName* を指定することにより、パフォーマンスが向上します。

結果

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}
```

コマンドを使用したエクスポートの詳細の表示:

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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

wsadmin ツールを使用してメディエーション・モジュールのエクスポートの詳細を表示するには、以下のステップに従います。

特定のメディエーション・モジュールのエクスポートの詳細を表示するには、メディエーション・モジュール名とエクスポート名を知る必要があります。

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。

プロシージャ

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。

3. 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`

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。

注: *applicationName* を指定することにより、パフォーマンスが向上します。

結果

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}
```

コマンドを使用したエクスポート・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。

始める前に

wsadmin ツールを使用して、以下のコマンドを実行します。

このタスクについて

特定のメディエーション・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、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・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。

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。

プロシージャ

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}`

moduleName の指定に加えて、オプションで *applicationName* も指定できます。

注: *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。

結果

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}
```

デプロイされた SCA モジュールの WebSphere MQ JMS 宛先の変更

ビジネス環境の発展に応じて、Service Component Architecture (SCA) モジュールが使用する WebSphere MQ 宛先を変更する必要があることがあります。

始める前に

変更する WebSphere MQ JMS 宛先を、どの 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。
- チャンネル圧縮は使用できません。
- データを正しく処理するには、必要なチャンネル出口をすべて設定する必要があります。これを行うには、「WebSphere MQ メッセージング・プロバイダー」 → 「WebSphere MQ 接続ファクトリー・オブジェクト (WebSphere MQ connection factory objects)」 → 「カスタム・プロパティ」をクリックして、以下のカスタム・プロパティを設定します。
 - **SENDEXIT**。このプロパティには、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl` を設定します。
 - **RECEXIT**。このプロパティには、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl` を設定します。
 - **SENDEXITINIT**。このプロパティには、選択したテキスト・ストリングを設定します。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。
 - **RECEXITINIT**。このプロパティには、選択したテキスト・ストリングを設定します。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。

このタスクについて

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 JMS 宛先の変更が必要になることがあります。

プロシージャ

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 JMS 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。

次のタスク

ステップ 1 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

SCA モジュールと WebSphere MQ キューを接続して、相互にサービスを提供することができます。

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。モジュールを作成するときは、SCA モジュールと WebSphere MQ キュー間の接続を定義します。

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。
- データを正しく処理するには、必要なチャネル出口をすべて設定する必要があります。これを行うには、「WebSphere MQ メッセージング・プロバイダー」 → 「WebSphere MQ 接続ファクトリー・オブジェクト (WebSphere MQ connection factory objects)」 → 「カスタム・プロパティ」をクリックして、以下のカスタム・プロパティを設定します。
 - **SENDEXIT**。このプロパティには、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalSendExitImpl` を設定します。
 - **RECEXIT**。このプロパティには、値 `com.ibm.ws.sca.internal.mq.exit.MQInternalReceiveExitImpl` を設定します。
 - **SENDEXITINIT**。このプロパティには、選択したテキスト・ストリングを設定します。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。
 - **RECEXITINIT**。このプロパティには、選択したテキスト・ストリングを設定します。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。

ターゲットの使用

ターゲットを使用すると、参照用に構成されたターゲットを変更して処理を変更できるため、柔軟性が向上します。

コンポーネントは、別のモジュール内のコンポーネントを呼び出すことができます。これにより、アプリケーション作成の時間とコストを最小限に抑えることができます。また、管理コンソールを使用することにより、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、モジュール間呼び出しのエンドポイントを変更できます。これにより、インストール済みのアプリケーションは、処理の進歩やその他の変更からの利点を得られます。ターゲットを使用すると、こうしたさらなる柔軟性が得られます。

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。

始める前に

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。

このタスクについて

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。

プロシージャ

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">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」>「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。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。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。4. 変更するインポートをリストから選択します。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。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。7. 「OK」をクリックして変更を保管します。

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"> 1. 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」>「バス」を使用してパネルにナビゲートします。 2. SCA.System.cellname.Bus を選択します。 3. 「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。 4. 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。 5. 「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。 6. 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。 7. 「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。 8. 「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。 9. 「OK」をクリックして変更を保管します。

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。

次のタスク

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、この仕様を削除する必要がある場合があります。

始める前に

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を削除する場合、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。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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、その仕様を削除します。

プロシージャ

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、またはデプロイメント環境の「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。

結果

システムは、選択された仕様を表示から削除します。

次のタスク

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。

始める前に

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。

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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)。

プロシージャ

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。

結果

パネルには残りの宛先のみが表示されます。

次のタスク

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。

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 25. 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"> • 開始 - アプリケーションは、すべてのデプロイメント・ターゲットで開始されました。 • 一部開始 - 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。
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"> • 停止 - アプリケーションは、すべてのデプロイメント・ターゲットで停止されました。 • 一部停止 - 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。
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>
更新	<p>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せん。</p>
ファイルの除去	<p>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。</p>
エクスポート	<p>「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nding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。</p>

表 25. 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
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 インフォメーション・センターを参照してください。

関連情報



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。



serviceDeploy コマンド

serviceDeploy コマンドを使用して、Service Component Architecture (SCA) 対応モジュールを、サーバーにインストール可能な Java アプリケーションとしてパッケージします。このコマンドは、wsadmin を使用してバッチ・インストールを実行する場合に役立ちます。



プロモート可能なプロパティ



メディエーション・モジュール用リソースの管理

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が提供す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。

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。また、スケジューラー・エントリーは、WebSphere InterChange Server スケジューラー・エントリーを含む WebSphere InterChange Server リポジトリのマイグレーション中に生成できます。(WebSphere InterChange Server からのマイグレーション、および reposMigrate コマンドに関するトピックを参照してください)。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。

スタンドアロン・サーバー環境では、Application Scheduler は自動的にインストールされます。スタンドアロン・サーバー・プロファイルを作成した場合、Application Scheduler は、そのサーバー上にインストールされて構成されます。

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、関連するタスクを参照してください。

WebSphere InterChange Server では、コラボレーション・オブジェクトまたはコネクタを含むアプリケーションを、コンポーネント・レベルで始動、一時停止、および停止できます。つまり、アプリケーションの残りの部分が続行を許可されても、コンポーネントを停止できます。WebSphere Process Server の場合、イベントのスケジュールは、Application Scheduler を介して行います。Application Scheduler では、プロセスをアプリケーション・レベルで開始および停止できます。

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詳しくは、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』
- 341 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。

このタスクについて

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

プロシージャ

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 WEBSPHERE_MB_QUERY_CONSTANT = "WebSphere:*";  
protected static final String NAME_QUERY_CONSTANT = ",name=";  
protected static final String WBI_SCHED_MB_NAME = "Scheduler_AppScheduler";  
protected static final String TYPE_QUERY_CONSTANT = ",type=";  
protected static final String WBI_SCHED_MB_TYPE = "WASScheduler";  
protected static final String SERVER_QUERY_CONSTANT = ",process=";  
serverName = "<server1>";  
protected static final String NODE_QUERY_CONSTANT = ",node=";  
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。

```
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。

始める前に

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。

このタスクについて

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. 副見出し「**Business Integration**」の下で「**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。その有効範囲で既存のスケジュール済みイベントがリストされます。通常、スケジューラー・エントリーは、サーバーの有効範囲で定義されます。

結果

以上で、新しいスケジューラー・イベントの作成、既存のスケジューラー・イベントの編集、または既存のスケジューラー・イベントの削除を行うことができます。

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。

始める前に

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、『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。

このタスクについて

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。

プロシージャ

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。

イベントを実行するようにスケジュールされた時刻に Application Scheduler または Process Server が稼働していない場合は、開始期間パラメーターによって、スケジュールされたイベント実行時刻に開始する時間またはウィンドウの長さ (分) が定義されます。この期間に、Application Scheduler または Process Server が操作を再開した場合は、イベントが実行されます。ただし、開始期間の期限が切れるまでにその操作が再開されない場合は、次の実行時刻が計算され、その時刻にイベントが実行されます。

例えば、夜の 12 時に実行するようにスケジュールされたイベントに対して開始期間を 60 分に設定したが、その時刻にサーバーが偶然ダウンしたとします。この場合は、サーバーが午前 1:00 前にオンラインに復帰すれば、イベントが実行されます。

- 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。

結果

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。

始める前に

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、341 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。

このタスクについて

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。

プロシージャ

1. 変更するイベントの「スケジュール・エントリー ID」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。

結果

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。

始める前に

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、341 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。

このタスクについて

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。

プロシージャ

1. 「選択」列で、削除するスケジュール・エントリーを選択します。
2. 「削除」をクリックします。

結果

スケジュール・エントリーが削除されます。

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 が機能する仕組み

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 の使用

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の表示

リレーションシップ名、表示名、静的属性、識別属性など、システム内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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

システム内のリレーションシップのリストを表示するには、以下のステップを実行します。

プロシージャ

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の属性、プロパティ値、静的属性、識別属性など、選択したリレーションシップの詳細情報を表示するには、このタスクを実行します。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

選択したリレーションシップの詳細情報を表示するには、以下のステップを実行します。

プロシージャ

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. リレーションシップ名の隣にあるラジオ・ボタンを選択して、「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のリストに戻るには、ページ上部のパスから「リレーションシップ」をクリックするか、「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、選択された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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

選択したロールの詳細情報を表示するには、以下のステップを実行します。

プロシージャ

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。

次のタスク

「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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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。

リレーションシップを照会するには、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して、「**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。1 つのフィールドを空白のままにしておいた場合は、照会によって、インスタンスが 1 つのみ戻されます。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。
「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・インスタンスの表示

リレーションシップ照会に一致するリレーションシップ・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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

リレーションシップ照会に一致するリレーションシップ・インスタンスのリストを表示するには、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して、「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、348 ページの『リレーションシップの照会』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」ページを開きます。

結果

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。

ページをナビゲートするには、以下の操作を行います。

- 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。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。

サーバーから一度に読み取るインスタンス数をカスタマイズできるように照会ブロック・サイズの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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

選択したリレーションシップ・インスタンスの詳細情報を表示するには、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して、「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」）をクリックして、検索基準を指定します。次に、「OK」をクリックして「リレーションシップ・インスタンス」ページを開きます。
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。
 - リレーションシップ・インスタンス ID の隣にあるラジオ・ボタンを選択して、「詳細」をクリックします。

リレーションシップ・インスタンスのリストに戻るには、ページ上部のパスから「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値を編集するには、このタスクを実行します。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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値を編集するには、以下のステップを実行します。

プロシージャ

1. 管理コンソールが稼働していることを確認します。

- 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。
- 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名の隣にあるラジオ・ボタンを選択して、「照会」をクリックします。
- 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」）をクリックして、検索基準を指定します。次に、「OK」をクリックして「リレーションシップ・インスタンス」ページを開きます。
- 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。
 - リレーションシップ・インスタンス ID の隣にあるラジオ・ボタンを選択して、「詳細」をクリックします。
- 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。

- 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」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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

リレーションシップ・インスタンスを新規作成するには、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して「作成」をクリックすると、「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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して、「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」）をクリックして、検索基準を指定します。次に、「OK」をクリックして「リレーションシップ・インスタンス」ページを開きます。

6. 削除するリレーションシップ・インスタンス ID の隣にあるラジオ・ボタンを選択します。
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

このタスクを実行すると、リレーションシップ・インスタンス・データが指定の日時にロールバックされます。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

ロールバック中に以下のアクションが実行されます。

- 特定の期間内に作成されたリレーションシップ・インスタンスがデータベースから削除される (物理的に削除される)。
- アクティブにされたリレーションシップ・インスタンスがデータベースから削除される (物理的に削除される)。
- 特定の期間内に非アクティブにされたリレーションシップ・インスタンスがアクティブにされる。

リレーションシップ・インスタンス・データをロールバックするには、以下のステップを実行します。

プロシージャ

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 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の隣にあるラジオ・ボタンを選択して、「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。

重要: WebSphere Process Server サーバーとデータベース・サーバーが、同じ時間帯に設定されていることを確認してください。同じ時間帯に設定されていない場合は、ロールバックが失敗します。

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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

選択したロール・インスタンスの詳細情報を表示するには、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して、「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」）をクリックして、検索基準を指定します。次に、「OK」をクリックして「リレーションシップ・インスタンス」ページを開きます。
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。
 - リレーションシップ・インスタンス ID の隣にあるラジオ・ボタンを選択して、「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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して、「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」）をクリックして、検索基準を指定します。次に、「OK」をクリックして「リレーションシップ・インスタンス」ページを開きます。
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。
 - リレーションシップ・インスタンス ID の隣にあるラジオ・ボタンを選択して、「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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

リレーションシップのロール・インスタンスを新規作成するには、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して、「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」）をクリックして、検索基準を指定します。次に、「OK」をクリックして「リレーションシップ・インスタンス」ページを開きます。
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。
 - リレーションシップ・インスタンス ID の隣にあるラジオ・ボタンを選択して、「詳細」をクリックします。
7. 新規インスタンスを作成するロールを指定して、ロール・テーブルの下にある「作成」をクリックすると、「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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

このタスクについて

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、以下のステップを実行します。

プロシージャ

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の隣にあるラジオ・ボタンを選択して、「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」）をクリックして、検索基準を指定します。次に、「OK」をクリックして「リレーションシップ・インスタンス」ページを開きます。
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。
 - リレーションシップ・インスタンス ID の隣にあるラジオ・ボタンを選択して、「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. 削除するロール・インスタンスの隣にあるラジオ・ボタンをクリックして、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。

リポジトリからのリレーションシップ・インスタンス・データの除去

リレーションシップを使用するアプリケーションは、リポジトリ内のリレーションシップ・スキーマとデータを関連付けています。リポジトリは、リレーションシップ・インスタンス・データを保持するように構成されたデータベースです。このようなアプリケーションを実動サーバーからアンインストールしても、サーバーは、リレーションシップ・スキーマとデータをリポジトリから除去しません。これらを除去するには、既存のリレーションシップ・スキーマを手動で除去する必要があります。

始める前に

リレーションシップ・スキーマを使用するアプリケーションが、そのスキーマにアクセスするすべてのサーバーからアンインストールされていることを確認します。

このタスクについて

リレーションシップを含むアプリケーションをインストールすると、サーバーは、テーブル、索引、シーケンス、およびストアド・プロシージャを含む、対応するデータベース・スキーマ・オブジェクトを作成します。実行時、このテーブルには、リレーションシップ・インスタンス・データが設定されます。リレーションシップを含むアプリケーションをアンインストールする場合、テーブルおよびインスタンス・データは、データベースから除去されません。この設計は、リレーションシップまたはロール定義の変更後にアプリケーションの再インストールを試みる場合に問題を起こすことがあります。

注: WebSphere Integration Developer (WID) の Unit Test Environment (UTE) テスト・サーバーを使用する場合、アプリケーション・プロジェクトを除去すると、リレーションシップ・スキーマとデータがリポジトリから除去されます。

同じリレーションシップを持つアプリケーションを再インストールする場合、古いスキーマは再利用されます。ただし、リレーションシップまたはロール定義を変更して、既存のスキーマと互換性がなくなるようにした場合、リレーションシップ・

サービスは、例外をスローして、リレーションシップのインストールを終了します。ログには、以下の例外とメッセージが示されます。

```
RelationshipServiceException("table <tablename> already exists, but the table schema is different from current role definition")
```

この問題を解決するには、リポジトリのデータベース・プラットフォームが提供するツールを使用して、既存のリレーションシップ・スキーマ成果物を手動で除去し、アプリケーションを再インストールします。

既存のリレーションシップ・スキーマをリポジトリから除去するには、以下のステップを実行します。

プロシージャ

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って異なります。

オプション	説明
データベース・プラットフォーム	ロケーション
Derby	WASHOME¥derby¥databases¥RepositoryDB
その他のデータベース	<p>ロケーションは、サーバーのインストールおよびプロファイル作成時に構成されます。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 WPRCSDB となります。</p> <p>DB2® for i5/OS の場合、参照先コンテナは、データベースではなくコレクションです。これは、インストールおよびプロファイル作成中に構成されたデータベース名ではなく、コレクション名です。つまり、デフォルトの WPRCSDB という名前のデータベースではなく、コレクションです。</p>

2. リレーションシップを構成するデータベース成果物を削除します。 指定のリレーションシップのデータベース・オブジェクトをすべて削除するには、データベース・プラットフォームのツールを使用して、以下のステップを実行します。
 - a. 以下のステップでデータベースからデータを除去する前に、データベースのバックアップを作成します。

注: DB2 for i5/OS の場合、データを除去する前にコレクションのバックアップを作成してください。
 - b. リレーションシップ・テーブルを検索します。 リレーションシップのインストール時に、以下のテーブルが作成されます。

テーブル	フォーマット
リレーションシップ・プロパティ用に 1 つのテーブル	_ <relname> _P_uniqueidentifier
各リレーションシップのインスタンス ID の生成用に 1 つのテーブル (Derby)	_ <relname> _S_uniqueidentifier

テーブル	フォーマット
アプリケーション固有の各ロールの ロール・プロパティ用に 1 つの テーブル	_ <code><relname></code> _ <code><rolename></code> _P_uniqueidentifier
アプリケーション固有のロールごと に 1 つのテーブル (静的リレーシ ョンシップの場合は、汎用ロール用 にも 1 つのテーブルが作成されま す)	_ <code><relname></code> _ <code><rolename></code> _RT_uniqueidentifier

制約事項: リレーションシップ名は、先頭の 4 文字のみが使用されます。複数のリレーションシップ用のテーブルをデータベースが保持している場合は、リレーションシップ名を先頭の 4 文字以内で区別する必要があります。

- c. ストアード・プロシージャを検索します。 ストアード・プロシージャ・オブジェクトの形式は以下のとおりです。

_`<relname>`_RS_uniqueidentifier または
`<relname>``<rolename>`_RS_uniqueidentifier

- d. シーケンスを検索します。 シーケンス・オブジェクトの形式は、以下のとおりです。

_`<relname>`_S_uniqueidentifier

制約事項: Derby では、シーケンスはサポートされません。

- e. データベース・プラットフォームのツールを使用して、以下を削除します。
- 1) テーブル
 - 2) ストアード・プロシージャ
 - 3) シーケンス (Derby を除く)

結果

リレーションシップ・インスタンス・データが、データベース・リポジトリから除去されます。

次のタスク

以上で、アプリケーションを再インストールできます。

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」と相関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 を相関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

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 を相関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。

関連情報

 サンプル (サンプル・ギャラリー) へのアクセス

リレーションシップ・インスタンスの値の変更

このタスクについて

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 を相関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。

プロシージャ

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. SampleCustID の隣にあるラジオ・ボタンを選択して、「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」ページが開きます。

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」をクリックすると、このリレーションシップ・インスタンスのロール・インスタンスを新規作成するための「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。

結果

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。

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定義

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。

リレーションシップの機能の仕組み

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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、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 セキュリティー・ロールでもこの構成を表示できます。

このタスクについて

リレーションシップ・サービスのデータ・ソース・プロパティと照会ブロック・サイズ・プロパティ (リレーションシップ・インスタンス・カウント) を設定するには、以下のステップを実行します。

プロシージャ

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. 「**照会ブロック・サイズ (リレーションシップ・インスタンスのカウント) (Query block size (relationship instance count))**」フィールドに、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 Java Naming and Directory Interface (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。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、どの WebSphere セキュリティー・ロールでもこの構成を表示できます。

このタスクについて

リレーションシップ・リストを表示するには、以下のステップを実行します。

プロシージャ

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。

次のタスク

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

このタスクは、リレーションシップ・サービスが管理する構成プロパティを、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・サービスに適用されるとき)、および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とき) の両方のレベルで表示する場合に使用します。

始める前に

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、どの WebSphere セキュリティー・ロールでもこの構成を表示できます。

このタスクについて

構成プロパティーを表示するには、以下のステップを実行します。

プロシージャ

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、プロパティー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。

拡張メッセージング・リソースの管理

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するには、管理コンソールを使用します。拡張メッセージング・サービスの使用可能化、遅延応答を処理するためのリスナー・ポート拡張の構成、および拡張メッセージングを使用するアプリケーションの入力および出力ポートの追加または変更が可能です。

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。

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・サービスを使用する前に、以下の制限に注意してください。

- 拡張メッセージング・サービス機能は、WebSphere Process Server 6.x 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ことを検討してください。
- **i5/OS** 拡張メッセージング・サービス機能は、i5/OS システムではサポートされません。

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・サービスを使用可能にすると、コンテナ管理メッセージング (拡張メッセージング) のランタイム・サポートが提供されます。「拡張メッセージング・サービス」ページでは、アプリケーション・サーバーの始動時にこのサービスを自動的に始動するかどうか、またはサービスを手動で始動する必要があるかどうかを指定できます。

このタスクについて

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、この作業を実行するには、管理者またはコンフィギュレーターとしてログインする必要があります。

拡張メッセージング・サービスを使用可能にするには、以下のステップを実行します。

プロシージャ

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。

遅延応答を処理するリスナー・ポート拡張の構成

リスナー・ポートを使用可能にして遅延応答を処理するには、ポートが応答を検査する頻度とポートが応答を待機する時間を指定する拡張を構成します。

このタスクについて

遅延応答は、送信側 Bean が送信したメッセージに対して、メッセージング・インフラストラクチャーが応答を遅延したときに発生します。これにより、アプリケーションは、その応答を受信できなくなります。拡張メッセージングは、これらの遅延応答メッセージを取得して、アプリケーションが提供するメッセージ駆動型 Bean にそのメッセージを渡すことにより、遅延応答を処理します。

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、この作業を実行するには、管理者またはコンフィギュレーターとしてログインする必要があります。

遅延応答を処理するリスナー・ポート拡張を作成して使用可能にするには、以下のステップを実行します。

プロシージャ

1. リスナー・ポートが定義および構成されていること、および「**遅延応答の処理 (Handle late responses)**」オプションが使用可能な状態で送信側 Bean がデプロイされていることを確認します。

注: このオプションを使用可能にして送信側 Bean をデプロイする方法について詳しくは、WebSphere Business Integration Server Foundation インフォメーション・センターを参照してください。

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。
3. 「リスナー・ポートの拡張」ページから、「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。
4. 「新規リスナー・ポート拡張機能」ページから「使用可能」チェック・ボックスを選択して、拡張および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、新しい値を指定して、リスナー・ポートが遅延応答を検査する頻度を設定します。
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、新しい値を指定して、リスナー・ポートが遅延応答を待機する時間を設定します。リスナー・ポートは、指定したタイムアウト値の経過後に受信した応答はすべて破棄します。
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。

次のタスク

リスナー・ポート拡張を作成したあとは、「リスナー・ポートの拡張」の設定ページを使用して、その構成を必要に応じて変更できます。

拡張メッセージング・プロバイダーの選択

管理する拡張メッセージング・プロバイダーを選択するには、「拡張メッセージング・プロバイダー」ページで適切な有効範囲をクリックします。拡張メッセージングを使用するアプリケーションを含む各有効範囲（セル、ノード、およびサーバー）は、リソースを管理するために独自の拡張メッセージング・プロバイダーを所有しています。ユーザーは、各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、または削除できます。

このタスクについて

このタスクに必要なセキュリティ・ロール: セキュリティおよび役割ベースの許可が有効になっている場合、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する必要があります。

管理する拡張メッセージング・プロバイダーを選択するには、以下のステップを実行します。

プロシージャ

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。
2. 「拡張メッセージング・プロバイダー」ページから、管理する拡張メッセージング・プロバイダーの適切な有効範囲を選択します。
 - **セル:** 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:** 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:** 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 インフォメーション・センターを参照してください。

3. 「適用」をクリックします。

結果

管理コンソールは、ページの下部の「有効範囲」、「名前」、および「説明」フィールドを更新して、選択したリソース・プロバイダーの値を反映させます。

次のタスク

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。

入力ポートの構成

セッション Bean から構成された各受信側 Bean について、新しい入力ポートを作成または既存の入力ポートを変更するには、管理コンソールを使用します。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、メッセージの選択および処理方法を指定して、必要な応答の宛先の詳細を提供します。

このタスクについて

注: メッセージ駆動型 Bean から構成された受信側 Bean の入力ポートを作成する必要はありません。必要な詳細は、デPLOYされたメッセージ駆動型 Bean とメッセージ・リスナー・サービスに関連付けられます。

このタスクに必要なセキュリティ・ロール: セキュリティと役割ベースの許可が有効になっている場合、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する必要があります。

入力ポートを追加または変更するには、以下のステップを実行します。

プロシージャ

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。
2. 「拡張メッセージング・プロバイダー」ページから、操作するリソース・プロバイダーの適切な有効範囲を選択します。
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「入力ポート」をクリックします。
5. 「入力ポート」のコレクション・ページから、以下のいずれかを実行します。
 - 新しい入力ポートを作成する場合は、「新規」をクリックします。
 - 既存の入力ポートを変更する場合は、ポート名をクリックします。
6. 「入力ポート設定」ページから、入力ポートの適切なプロパティを指定します。
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:

新しい入力ポートを作成する場合、または既存の入力ポートを変更する場合は、特定のプロパティを指定する必要があります。このトピックの情報を参照すると、プロパティがオプションと必須のどちらであるのかと、プロパティが受け入れるデータ・タイプを確認できます。

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。管理コンソールは、このフィールドに自動的にデータを設定します。値は編集できません。

名前 管理目的で使用する入力ポートの名前。このフィールドには、ストリング値が必要です。

JNDI 名

入力ポートの Java Naming and Directory Interface (JNDI) 名。このフィールドには、ストリング値が必要です。

説明 管理目的で使用する入力ポートの説明。このフィールドはオプションであり、ストリング値を受け入れます。

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。このフィールドはオプションであり、ストリング (最大 30 文字までの ASCII 文字) を受け入れます。

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。このフィールドには、ストリング値が必要です (例: jms/connFactory1)。

JMS 宛先 JNDI 名

入力ポートで使用される JMS 宛先の JNDI 名。このフィールドには、ストリング値が必要です (例: jms/destn1)。

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。このフィールドには、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) に必要です。

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る場合
 - 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る場合
- 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。このフィールドには、以下のいずれかの値が必要です。

- 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。このフィールドは、JMS 宛先タイプがトピックの場合に必要です。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「非永続」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。このフィールドには、ストリング値が必要です (例: jms/connFactory1)。

応答 JMS 宛先 JNDI 名

応答に使用する JMS 宛先の JNDI 名。このフィールドには、ストリング値が必要です (例: jms/destn1)。

出力ポートの構成

送信側 Bean の新しい出力ポートを作成または既存の出力ポートを変更するには、管理コンソールを使用します。出力ポートは、送信側 Bean のプロパティを指定します。また、送信メッセージの宛先を定義する必要があります。応答が期待されるときは、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。

このタスクについて

このタスクに必要なセキュリティ・ロール: セキュリティーと役割ベースの許可が有効になっている場合、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する必要があります。

出力ポートを追加または変更するには、以下のステップを実行します。

プロシージャ

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。
2. 「拡張メッセージング・プロバイダー」ページから、変更する拡張メッセージング・プロバイダーの適切な有効範囲を選択します。
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。

5. 「出力ポート」のコレクション・ページから、以下のいずれかを実行します。
 - 新しい出力ポートを追加する場合は、「**新規**」をクリックします。
 - 既存の出力ポートを変更する場合は、ポート名をクリックします。
6. 「出力ポート設定」ページから、出力ポートの適切なプロパティを指定します。
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:

新しい出力ポートを作成する場合、または既存の出力ポートを変更する場合は、特定のプロパティを指定する必要があります。このトピックの情報を参照すると、プロパティがオプションと必須のどちらであるのかと、プロパティが受け入れるデータ・タイプを確認できます。

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

拡張メッセージング・プロバイダーの有効範囲。この値は、構成ファイルの場所を示します。管理コンソールは、このフィールドに自動的にデータを設定します。値は編集できません。

名前 管理目的で使用する出力ポートの名前。このフィールドには、ストリング値が必要です。

JNDI 名

出力ポートの Java Naming and Directory Interface (JNDI) 名。このフィールドには、ストリング値が必要です。

説明 管理目的で使用する出力ポートの説明。このフィールドはオプションであり、ストリング値を受け入れます。

カテゴリー

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ー・ストリング。このフィールドはオプションです。ストリング値 (最大 30 文字までの ASCII 文字) を受け入れます。

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。このフィールドには、ストリング値が必要です (例: jms/connFactory1)。

JMS 宛先 JNDI 名

出力ポートで使用される JMS 宛先の JNDI 名。このフィールドには、ストリング値が必要です (例: 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。このフィールドでは、以下のいずれかの値を選択する必要があります。

- 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。このフィールドには、0 から 9 の整数値が必要です。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。

このフィールドには、0 から n の整数値が必要です。

- 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。これは、必須フィールドです。以下のいずれかの値を選択します。

- 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。これは、必須フィールドです。以下のいずれかの値を選択します。

- 選択済み: システムは、メッセージ・タイム・スタンプを追加せずにメッセージを送信する。
- クリア: システムは、メッセージ・タイム・スタンプを自動的に追加してメッセージを送信する。

デフォルトでは、システムは、メッセージ・タイム・スタンプを追加してメッセージを送信します。

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。このフィールドには、ストリング値が必要です (例: `jms/connFactory1`)。

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。このフィールドには、ストリング値が必要です (例: `jms/destn1`)。

第 10 章 イベントの使用

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。イベントは特定の順序で処理できます。また、失敗したイベントを表示、破棄、変更、または再サブミットできます。

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

『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。

378 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せん。

関連概念

375 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

375 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

イベ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` エレメントが含まれ、このエレメント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>
```

コンポーネントでイベントの順序付けを宣言する場合は、コンポーネントが管理対象スレッドで呼び出されることを確認します。管理対象スレッドは、イベントを適切に順序付けるために必要なセッション情報を提供します。

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、JMS、WebSphere 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。

関連情報

エクスポート・バインディングとイベント順序付け

 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

 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・バインディングでのイベントの順序付けの有効化

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 26. 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

表 26. 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
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
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、ダウンストリーム・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。

以下のシナリオでは、イベントの順序付けはサポートされていません。

- 非管理対象スレッドまたは非 SCA バインディングを使用して、適切なセッション・コンテキストなしで宛先にイベントを送信する場合。
- 非同期呼び出しを行うコンポーネントに対して同期呼び出しを使用する場合。

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、このコマンドを任意のクラスター上で実行できますが、デプロイメント・マネージャー上では実行しないでください。

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 27. 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。

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。

パフォーマンスの問題

イベント順序付けコンポーネントに使用しているメッセージング・エンジン・サーバーでメモリーの問題が発生する場合は、`install_root/properties/eventsequencing.properties` ファイルのランタイム・イベント順序付けプロパティ `maxActiveMessages` を変更します。

`maxActiveMessages` プロパティは、コンポーネント宛先でロックされる現行メッセージの数を定義します。大きなメッセージが多数ロックされていると、パフォーマンスが低下し、メモリーの問題が発生する可能性があります。値 0 (ゼロ) を指定すると、ロックされるメッセージの数が無制限になる点に注意してください。デフォルトでは、`maxActiveMessages` プロパティは 100 に設定されています。

`maxActiveMessages` プロパティを変更するには、以下のステップを実行します。

1. テキスト・エディターで `eventsequencing.properties` ファイルを開きます。
2. ご使用の環境に合わせて変更します。
3. ファイルを保存して閉じます。
4. 変更を有効にするため、イベント順序付けコンポーネントの一部であるアプリケーションをすべて停止してから再始動します。

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 アプリケーションが受け取った要求を意味します。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、そのイベントは、WebSphere Process Server アプリケーションの適切なビジネス・ロジックによって処理されます。

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、同期呼び出しまたは Business Process Execution Language (BPEL) 非同期要求/応答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、WebSphere Process Server Recovery サービスが処理する以下のタイプの失敗したイベントすべてを参照および管理します。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。

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 内で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、『アプリケーションと環境の保護』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一部分を検索するかを選択できます。

始める前に

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。

このタスクについて

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。

プロシージャ

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」 → 「失敗したすべてのイベントを取得」をクリックします。

結果

「検索結果」ページが開き、セル内にある WebSphere Process Server の失敗したすべてのイベントのリストが表示されます。

次のタスク

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。

失敗したイベントのセッション ID 別検索

「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。

始める前に

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。

このタスクについて

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。

失敗したイベントをセッション ID 別に検索するには、以下の手順を実行します。

プロシージャ

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。

結果

「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。

失敗したイベントの宛先別検索

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するには、「検索」ページの「**宛先による検索**」タブを使用します。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宛先を判断します。

始める前に

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。

このタスクについて

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。

失敗したイベントを宛先別に検索するには、以下の手順を実行します。

プロシージャ

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**: このフィールドは、失敗したイベントの宛先モジュールを指定する場合に使用します。
 - **宛先コンポーネント**: このフィールドは、失敗したイベントの宛先コンポーネントを指定する場合に使用します。
 - **宛先メソッド**: このフィールドは、失敗したイベントの宛先メソッドを指定する場合に使用します。
4. 「**OK**」をクリックして検索を開始します。

結果

「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。

失敗したイベントのソース別検索

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するには、「検索」ページの「**ソースによる検索**」タブを使用します。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ソースを判断します。

始める前に

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。

このタスクについて

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空白のままにすると、空白・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。

プロシージャ

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**: このフィールドは、失敗したイベントの発信元モジュールを指定する場合に使用します。
 - **ソース・コンポーネント**: このフィールドは、失敗したイベントの発信元コンポーネントを指定する場合に使用します。
4. 「OK」をクリックして検索を開始します。

結果

「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。

失敗したイベントの日付別検索

特定の期間に失敗したイベントのみを検索するには、「検索」ページの「日付ごと」タブを使用します。

始める前に

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。

このタスクについて

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。

失敗したイベントの日付別検索を実行するには、以下の手順を実行します。

プロシージャ

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。

失敗したイベントのビジネス・オブジェクト・タイプ別検索

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するには、「検索」ページの「**タイプによる検索**」タブを使用します。

始める前に

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。

このタスクについて

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。

プロシージャ

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. 以下のいずれかの項目を使用して、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):** このドロップダウン・メニューは、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する場合に使用します。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:** このフィールドは、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する場合に使用します。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せん。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。

結果

「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。

失敗したイベントの例外別検索

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するには、「検索」ページの「**例外による検索**」タブを使用します。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。

始める前に

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。

このタスクについて

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。

プロシージャ

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。

結果

「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**」タブを使用すると、他の検索タブにある基準を組み合わせ、失敗したイベントをさらに詳しく検索できます。

始める前に

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。

このタスクについて

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。

プロシージャ

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:** このフィールドは、失敗したイベントの宛先モジュールを指定する場合に使用します。
- **宛先コンポーネント:** このフィールドは、失敗したイベントの宛先コンポーネントを指定する場合に使用します。
- **セッション ID:** このフィールドは、イベントが実行されたセッションを指定する場合に使用します。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:** このフィールドは、失敗したイベントの宛先メソッドを指定する場合に使用します。
- **ソース・モジュール:** このフィールドは、失敗したイベントの発信元モジュールを指定する場合に使用します。
- **ソース・コンポーネント:** このフィールドは、失敗したイベントの発信元コンポーネントを指定する場合に使用します。
- **開始日:** このフィールドは、特定の期間内で検索する場合に、開始日時を指定するために使用します。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:** このフィールドは、特定の期間内で検索する場合に、終了日時を指定するために使用します。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:** このフィールドは、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する場合に使用します。
- **例外テキスト:** このフィールドは、検索する例外に関連付けられたテキストを指定する場合に使用します。

4. 「OK」をクリックして検索を開始します。

結果

「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。

始める前に

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。

このタスクについて

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もかまいません。

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。

プロシージャ

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、データを参照する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。

始める前に

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。

このタスクについて

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。

プロシージャ

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

387 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一部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。

始める前に

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。

このタスクについて

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。

プロシージャ

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

イベントの送信を再試行する場合は、Failed Event Manager からイベントを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。

このタスクについて

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。

プロシージャ

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。

結果

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と、今度はそれが成功したかどうかを確認できます。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。

このタスクについて

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、インフォメーション・センターのモニターに関する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する必要があります。

プロシージャ

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。

次のタスク

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。

始める前に

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。

このタスクについて

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。

プロシージャ

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、387 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。

結果

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。

始める前に

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。

このタスクについて

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。

プロシージャ

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、387 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。

結果

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。

始める前に

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。

このタスクについて

- 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。

プロシージャ

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	『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	403 ページの『「期限切れイベントを削除」機能を使用すると 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	403 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は、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、フィールドの上に表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、各フィールドの上を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」機能を使用すると 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。

第 11 章 WebSphere Process Server のトラブルシューティング管理

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。

Business Process Choreographer または Common Event Infrastructure コンポーネントのトラブルシューティングについては、以下のいずれかを参照してください。

- WebSphere Process Server for Multiplatforms バージョン 6.1 インフォメーション・センター
- *Business Process Choreographer* の PDF
- *Common Event Infrastructure* の PDF

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	402 ページの『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	402 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	403 ページの『「期限切れイベントを削除」機能を使用すると 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	403 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」機能を使用すると Failed Event Manager が中断するようになる

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。

始める前に

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。

このタスクについて

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。

プロシージャ

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。このページへ移動するには、「アプリケーション」→「SCA モジュール」>「*moduleName*」>「インポート/エクスポート」>「*importName*/*exportName*」>「バインディング」>「*bindingName*」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、WebSphere 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。

WebSphere 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 WebSphere 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) キューの両方を必ず調べます。
- d. WebSphere 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、『失敗したイベントの検索』を参照してください。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。

パフォーマンスの問題

イベント順序付けコンポーネントに使用しているメッセージング・エンジン・サーバーでメモリーの問題が発生する場合は、`install_root/properties/eventsequencing.properties` ファイルのランタイム・イベント順序付けプロパティー `maxActiveMessages` を変更します。

`maxActiveMessages` プロパティーは、コンポーネント宛先でロックされる現行メッセージの数を定義します。大きなメッセージが多数ロックされていると、パフォーマンスが低下し、メモリーの問題が発生する可能性があります。値 0 (ゼロ) を指定すると、ロックされるメッセージの数が無制限になる点に注意してください。デフォルトでは、`maxActiveMessages` プロパティーは 100 に設定されています。

`maxActiveMessages` プロパティーを変更するには、以下のステップを実行します。

1. テキスト・エディターで `eventsequencing.properties` ファイルを開きます。
2. ご使用の環境に合わせて変更します。
3. ファイルを保存して閉じます。
4. 変更を有効にするため、イベント順序付けコンポーネントの一部であるアプリケーションをすべて停止してから再始動します。

デプロイメント環境のトラブルシューティング

処理が遅くなっているように見えたり、要求が失敗したりするときは、焦点を絞った方法を使用して、環境の問題の原因を判別します。ここで説明する方法は、非スタンドアロン・サーバー環境向けです。

始める前に

このタスクを実行するには、デプロイメント・マネージャーの管理コンソールにログインする必要があります。

このタスクに必要なセキュリティ・ロール: セキュリティーおよび役割ベースの許可が有効になっている場合、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。

このタスクについて

以下の症状に気付いた場合は、デプロイメント環境の状態を調査してください。

- アプリケーションが使用できない
- アプリケーションの動作が遅い
- アプリケーションが停止する
- スループットが低下する
- パフォーマンスが低下する

プロシージャ

1. このデプロイメント環境を記述したトポロジー・レイアウトを表示して、トポロジーの状況を確認します。
2. トポロジーを表示して、トポロジー内の各種のロールの状態を確認します。以降の調査のために、予期しない状態のロール、または警告状態のロールをメモしておきます。
3. 各ロールのエラー状態の原因となっているノードを探します。
4. すべてのノードが同期化されていることを確認します。

管理コンソールの「ノード」ページで、すべての非同期ノードを選択し、「同期」をクリックします。

5. すべてのバスに関連付けられているメッセージング・エンジンが稼働していることを確認します。

メッセージング・エンジンが稼働していない場合は、停止してから再始動してください。

6. エラー状態のノードに関連付けられているログを探し、そのログを表示してエラー・メッセージがないかどうかを確認します。
7. 訂正に影響するエラー・メッセージで説明されたアクションをすべて実行します。
8. エラーをすべて訂正して、影響を受けるノードを再始動します。

結果

以前エラー状態だったノードが始動して、トポロジーの状況が「実行中」になります。

次のタスク

影響を受けるアプリケーションをすべて再始動します。

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・エラーの解決

ログイン・エラーは、ログイン時に発生します。

このタスクについて

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。

プロシージャ

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。

次のタスク

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の解決

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。

このタスクについて

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。

ログイン競合エラーを解決するには、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の解決

1 人のユーザーがデータ・ソース内のビジネス・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と、アクセス競合エラーが発生します。

始める前に

ローカル側の変更をリポジトリに公開すると、このエラーが報告されます。

このタスクについて

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。

- 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、エラーが発生したビジネス・ルール・グループおよびルール・スケジュールをデータ・ソースから再ロードする必要があります。これは、エラーが発生したビジネス・ルール・グループおよびルール・スケジュールのローカル変更は使用できなくなるからです。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、ビジネス・ルール・グループまたはルール・スケジュールのページを再ロードします。エラー状態でない他のビジネス・ルール・グループおよびルール・スケジュールのローカル変更は引き続き使用できます。

バインディングのトラブルシューティング

バインディングでは、特定のタイプのバインディングに固有の各種エラー状態が発生する可能性があります。

このタスクについて

エラー状態の処理方法は、該当するバインディングのタイプによって異なります。

JMS および WebSphere MQ JMS バインディングのトラブルシューティング

JMS バインディングおよび WebSphere 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。

実装例外

このタスクについて

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。

- **ServiceBusinessException**: 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。
- **ServiceRuntimeException**: 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>
      <element name="price" type="float"></element>
    </sequence>
  </complexType>
</schema>
```

以下のいずれかの例外を受け取る場合:

```
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>

```

Failed Event Manager に表示されない JMS ベースの SCA メッセージ

このタスクについて

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となります。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 JMS ベースの WebSphere MQ SCA メッセージ

このタスクについて

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となります。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。

障害

このタスクについて

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作業している場合は、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。

プロシージャ

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。

誤用例: WebSphere MQ バインディングとの比較

このタスクについて

WebSphere MQ JMS バインディングは、WebSphere MQ に対してデプロイされている JMS アプリケーションと相互協調処理するよう設計されています。これにより、メッセージは JMS メッセージ・モデルに基づいて公開されます。これに対し、WebSphere MQ インポートおよびエクスポートは、ネイティブ WebSphere MQ アプリケーションと相互協調処理することができ、WebSphere MQ メッセージ本体の内容全体をメディエーションに公開するように設計されています。

以下のシナリオでは、WebSphere MQ バインディングではなく WebSphere MQ JMS バインディングを使用して作成する必要があります。

- JMS メッセージ駆動型 Bean (MDB) を SCA モジュールから呼び出す。この MDB は、WebSphere MQ JMS プロバイダーに対してデプロイされています。これは、WebSphere MQ JMS インポートを使用して実装されます。
- SCA モジュールを J2EE コンポーネント・サーブレットまたは EJB から JMS を介して呼び出すことができるようにする。これは、WebSphere MQ JMS エクスポートを使用して実装されます。
- WebSphere MQ 上で転送中の JMS MapMessage の内容のメディエーションを実行する。この場合、WebSphere MQ JMS エクスポートとインポート、および適切なデータ・バインディングを組み合わせて使用します。

WebSphere MQ バインディングと WebSphere MQ JMS バインディングの相互協調処理が予期される状況があります。特に、J2EE WebSphere MQ アプリケーションと非 J2EE WebSphere MQ アプリケーション間をブリッジングする場合は、WebSphere MQ エクスポートと WebSphere MQ JMS インポート (あるいはこの逆) を、適切なデータ・バインディングまたはメディアエーション・モジュール (あるいはこの両方) と組み合わせて使用します。

汎用 JMS バインディングのトラブルシューティング

汎用 JMS バインディングでは特定の失敗状態が生じることがあります。

このタスクについて

汎用 JMS バインディングに関するさまざまなエラー状態が発生する可能性があります。

汎用 JMS 例外のトラブルシューティング

汎用 JMS のインポートおよびエクスポート実装が、さまざまなエラー状態に対応して例外を戻すことがあります。

このタスクについて

汎用 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。

- `ServiceBusinessException` - この例外は、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戻されます。
- `ServiceRuntimeException` - 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。JMS の場合これは `JMSException` です。

汎用 JMS メッセージの有効期限のトラブルシューティング

JMS プロバイダーの要求メッセージには有効期限が設定されます。

このタスクについて

要求有効期限 は、JMS プロバイダーの要求メッセージの `JMSExpiration` の時刻に達した場合の、その要求メッセージの有効期限を指します。その他の JMS バインディングについては、汎用 JMS バインディングは要求有効期限を処理するため、インポートにより配置されたコールバック・メッセージの有効期限を、発信要求と同じ有効期限に設定します。コールバック・メッセージの有効期限の通知により、要求メッセージの有効期限が切れていることが示されます。クライアントには、ビジネス例外により期限切れを通知する必要があります。

コールバック宛先がサード・パーティー・プロバイダーに移動すると、このタイプの要求有効期限はサポートされなくなります。

応答有効期限 は、JMS プロバイダーの応答メッセージの `JMSExpiration` の時刻に達した場合の、その応答メッセージの有効期限を指します。

汎用 JMS バインディングでは応答有効期限はサポートされていません。これは、サード・パーティー JMS プロバイダーの正確な期限切れ動作が定義されていないためです。ただし、応答を受信する場合、応答を受信する時点で応答有効期限が切れていないかどうかを確認できます。

アウトバウンド要求メッセージでは、JMSExpiration 値は待機時間と、asyncHeader の requestExpiration 値 (設定されている場合) から算出されます。

汎用 JMS 接続ファクトリーのエラーのトラブルシューティング

汎用 JMS プロバイダーで特定のタイプの接続ファクトリーを定義すると、アプリケーションの開始時にエラー・メッセージを受け取ることがあります。この問題を回避するには、外部接続ファクトリーを変更します。

このタスクについて

アプリケーションの起動時に、「MDB リスナー・ポート JMSConnectionFactory タイプが JMSDestination タイプと一致しません (MDB Listener Port JMSConnectionFactory type does not match JMSDestination type)」というエラー・メッセージが出力されることがあります。

この問題は、外部接続ファクトリーの定義時に発生します。特に、JMS 1.1 (統合) 接続ファクトリー (Point-to-Point 通信およびパブリッシュ/サブスクライブ通信の両方をサポート可能な接続ファクトリー) ではなく JMS 1.0.2 トピック接続ファクトリーを作成すると、例外がスローされることがあります。

この問題を解決するには、以下のステップを実行します。

プロシージャ

1. 使用している汎用 JMS プロバイダーにアクセスします。
2. 定義されている JMS 1.0.2 トピック接続ファクトリーを JMS 1.1 (統合) 接続ファクトリーに置き換えます。

結果

新規に定義した JMS 1.1 接続ファクトリーを使用してアプリケーションを起動すると、エラー・メッセージは表示されなくなります。

WebSphere MQ バインディングのトラブルシューティング

WebSphere MQ バインディングで発生する障害または失敗の状態を診断し、このような状態を修正できます。

このタスクについて

WebSphere MQ バインディングの主な障害状態は、トランザクションの意味構造、WebSphere MQ 構成、またはその他のコンポーネントの既存の動作への参照に基づいて判別されます。主な障害状態は以下のとおりです。

- WebSphere MQ キュー・マネージャーまたはキューに接続できない: WebSphere MQ に接続できず、メッセージを受信できない場合は、MDB ListenerPort が開始できません。この状態は、WebSphere Application Server ログに記録されます。永続メッセージは、正常に取得されるまで (または WebSphere MQ により期限切れ

となるまで) WebSphere MQ キューに残ります。 WebSphere MQ に接続できず、アウトバウンド・メッセージを送信できないと、送信操作を制御するトランザクションがロールバックされます。

- インバウンド・メッセージを構文解析できない、またはアウトバウンド・メッセージを構成できない: データ・バインディングが失敗すると、作業を制御するトランザクションがロールバックされます。
- アウトバウンド・メッセージを送信できない: メッセージを送信できないと、関連するトランザクションがロールバックされます。
- 複数の応答メッセージまたは予期しない応答メッセージが戻される: インポートでは、要求メッセージごとに応答メッセージが 1 つのみ戻されることが想定されています。応答を受信すると、レコードが削除されます。予期しない状態で受信した応答メッセージは、JMS インポートで破棄されます。

誤用例: WebSphere MQ JMS バインディングとの比較

このタスクについて

通常、WebSphere MQ インポートおよびエクスポートは、ネイティブ WebSphere MQ アプリケーションと相互協調処理し、WebSphere MQ メッセージ本体の内容全体をメディエーションに公開するように設計されています。一方、WebSphere MQ JMS バインディングは、WebSphere MQ に対してデプロイされている JMS アプリケーションと相互協調処理するように設計されています。これにより、メッセージは JMS メッセージ・モデルに基づいて公開されます。

以下のシナリオでは、WebSphere MQ バインディングではなく WebSphere MQ JMS バインディングを使用して作成する必要があります。

- JMS メッセージ駆動型 Bean (MDB) を SCA モジュールから呼び出す。この MDB は、WebSphere MQ JMS プロバイダーに対してデプロイされています。これは、WebSphere MQ JMS インポートを使用して実装されます。
- SCA モジュールを J2EE コンポーネント・サーブレットまたは EJB から JMS を介して呼び出すことができるようにする。これは、WebSphere MQ JMS エクスポートを使用して実装されます。
- WebSphere MQ 上で転送中の JMS MapMessage の内容のメディエーションを実行する。この場合、WebSphere MQ JMS エクスポートとインポート、および適切なデータ・バインディングを組み合わせて使用します。

WebSphere MQ バインディングと WebSphere MQ JMS バインディングの相互協調処理が予期される状況があります。特に、J2EE WebSphere MQ アプリケーションと非 J2EE WebSphere MQ アプリケーションを連携させる場合は、WebSphere MQ エクスポートと WebSphere MQ JMS インポート (あるいはこの逆) を、適切なデータ・バインディングまたはメディエーション・モジュール (あるいはこの両方) と組み合わせて使用します。

一般に構成エラーが原因で WebSphere MQ がメッセージを意図する宛先に配信できない場合、メッセージは指定されている送達不能キューに送信されます。このとき、メッセージ本体の先頭には、送達不能ヘッダーが追加されます。このヘッダーには、失敗の原因、元の宛先、およびその他の情報が含まれています。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

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

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

〒106-8711

東京都港区六本木 3-2-12

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

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

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

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

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

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

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

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

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

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

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

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

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

著作権使用許諾:

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

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。(c) (お客様の会社名) (西暦年)。このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。(c) Copyright IBM Corp. _年を入れる_。 All rights reserved.

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

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、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、IBM logo、DB2、i5/OS、WebSphere、および z/OS は、International Business Machines Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc. の米国およびその他の国における商標です。

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

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

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

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

この製品には、Eclipse Project (<http://www.eclipse.org>) により開発されたソフトウェアが含まれています。



IBM WebSphere Process Server for Multiplatforms バージョン 6.1.0



Printed in Japan