



お願い

本書および本書で紹介する製品をご使用になる前に、387ページの『特記事項』に記載されている情報をお読みください。

本書は、WebSphere Process Server for z/OS バージョン 6、リリース 0、モディフィケーション 1 (製品番号 5655-N53) および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

本マニュアルに関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。

<http://www.ibm.com/jp/manuals/main/mail.html>

なお、日本 IBM 発行のマニュアルはインターネット経由でもご購入いただけます。詳しくは

<http://www.ibm.com/jp/manuals/> の「ご注文について」をご覧ください。

(URL は、変更になる場合があります)

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

原 典： WebSphere® Process Server for z/OS
Administering WebSphere Process Server
Version 6.0.1

発 行： 日本アイ・ピー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

第1刷 2006.6

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

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

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

© Copyright IBM Japan 2006

目次

WebSphere Process Server の管理	1
WebSphere Process Server の管理の概要	1
WebSphere Process Server の管理コンソール	3
WebSphere Process Server 管理環境ロードマップのセットアップ	12
管理アーキテクチャーのセットアップ	13
セル全体に対する設定を構成	14
WebSphere Process Server サーバー・ロードマップの管理	14
WebSphere Process Server リソース・ロードマップの構成	18
WebSphere Process Server 設定ロードマップの構成	19
イベント・サービスの管理	19
リレーションシップ・サービスの管理	20
クラスター環境の作成	23
アプリケーションをサポートするサーバーまたはクラスターの準備	28
Business Process Choreographer の管理	37
管理コンソールによる Business Process Choreographer の管理	37
スクリプトによる Business Process Choreographer の管理	45
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	54
ビジネス・プロセスおよびヒューマン・タスクの管理	54
ビジネス・ルール	73
エンタープライズ・アプリケーションの管理	102
Application Scheduler	104
WebSphere Process Server リソースの管理	109
アダプターの管理	109
セレクター・コンポーネントの管理の概要	160
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	165
ターゲットの概要	167
拡張メッセージング・リソースの管理: 概要	172
WebSphere ESB の管理	180
バス環境の管理	182
サービス・アプリケーションの管理	227
リレーションシップの管理	268
リレーションシップ・タイプの表示	269
リレーションシップ詳細の表示	269
リレーションシップ・インスタンスの照会	270
リレーションシップ・インスタンス詳細の表示	271
リレーションシップ・インスタンスの詳細の編集	271
リレーションシップ・インスタンスの新規作成	272
リレーションシップ・インスタンスの削除	273
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	273
リレーションシップに関連したロール・タイプの表示	274
ロールを基にしたリレーションシップ・インスタンスの照会	274
ロール詳細の表示	275
ロール・インスタンス詳細の表示	276
ロール・インスタンスの新規作成	276
ロール・インスタンスの削除	277
ロール・インスタンス・プロパティの編集	278
チュートリアル: Relationship manager の管理	278
WebSphere Process Server の失敗したイベントの管理	281
Failed Event Manager のロール・ベースのアクセス	282
失敗したイベントの検索	282

失敗したイベントのデータの操作	288
失敗したイベントの再サブミット	292
失敗したイベントの削除	293
Common Event Infrastructure の使用	293
Common Event Infrastructure の管理	294
WebSphere Process Server 管理のトラブルシューティング	356
Failed Event Manager のトラブルシューティング	356
Business Process Choreographer 構成のトラブルシューティング	357
ビジネス・プロセスとヒューマン・タスクのトラブルシューティング	362
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	384
Common Base Event ブラウザーのトラブルシューティング	385
特記事項	387
プログラミング・インターフェース情報	389
商標	389

WebSphere Process Server の管理

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

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

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

管理インターフェース

WebSphere Process Server ではランタイム環境を管理するための以下のような複数のインターフェースが提供されています。

管理コンソール

管理コンソールは、ブラウザ・ベースのインターフェースで、これによりユーザーは、さまざまな種類のアプリケーション、サービス、およびリソースを監視、更新、停止、および始動することができます。また管理コンソールは、リレーションシップを操作し、失敗した WebSphere Process Server イベントの場所を探索して解決するためにも使用できます。

管理コンソールには、管理対象を拡張して、WebSphere Application Server およびその他のユーザー定義製品を管理する能力が備えられています。

詳しくは、3 ページの『WebSphere Process Server の管理コンソール』を参照してください。

Business Process Choreographer Explorer

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

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、実行用のスクリプト言語プログラムを実行依頼できます。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 ツールは、実稼働環境および不在操作向けです。

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

コマンド行ツール

コマンド行ツールは、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あり、特定のタスクを実行するものです。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、ノードの追加と除去、および他のタスクを実行することができます。

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、このコマンドを使用すると 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、それらをプロダクション・サーバーにインストールする準備をすることができます。

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

管理プログラム

Java Management Extensions (JMX) 仕様の下の一連の Java クラスおよびメソッドは、Service Component Architecture (SCA) およびビジネス・オブジェクトを管理するためのサポートを提供します。各プログラミング・インターフェースにはその目的の説明、インターフェースまたはクラスを使用する方法を説明する例、個々のメソッド説明への参照が含まれています。

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

構成情報

WebSphere Process Server 用のほとんどの構成データは、構成リポジトリ・ツリー (マスター・リポジトリ) のディレクトリ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、つまりデータが適用される範囲が決定されます。

- 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。
- 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	Adaptive Entity Service、拡張メッセージング・サービス、および WebSphere Business Integration Adapter Service 構成を含む、プロセス・サーバーおよびそのコンポーネントを識別します。
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、ノード、およびサーバー・スコープに存在します。これには、拡張メッセージング・プロバイダーおよび WebSphere Business Integration Adapter が含まれています。
cell-wbi.xml	セルを識別します。このファイルは、リレーションシップ・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 Flow Manager、Human Task Manager、Staff Service、および Service Reference Service を含む、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	Staff Plugin Provider 用の構成情報を含む、Business Process Choreographer コンテナ用のオペレーティング環境リソースを定義します。このファイルは、セル、ノード、およびサーバー・スコープに存在します。
deployment-bpc.xml	ビジネス・プロセス・コンテナ用のアプリケーション・デプロイメント設定を構成します。
server-core.xml	Artifact Loader Service、Events Service、および Business Context Data Service を含む、コア WebSphere Process Server 構成用の構成情報を識別します。

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

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

WebSphere Process Server の管理コンソール

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

注: WebSphere Process Server の管理コンソールは、より大きな WebSphere Application Server 管理コンソールの一部です。その結果、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 の両方で同じになります。これらのタスクについては、WebSphere Application Server for z/OS インフォメーション・センターに記載されています。

コンソールに関連付けされている WebSphere Process Server タスクの理解

コンソールで実行される 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・サーバーの構成およびその設定
- 新規アプリケーションのサーバーへの配置
- 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、およびセクターなどのサーバー・リソースの管理
- Business Process Choreographer の管理
- プロセス・サーバーで失敗したイベントの管理
- 製品セキュリティーの構成
- 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・インターフェースの理解

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、コンソールのログアウト、製品情報のアクセス、およびサポートのアクセスに関するオプションを提供します。

ナビゲーション・ツリー

ナビゲーション・ツリーは、コンソールの左側にあります。ナビゲーション・ツリーは、サーバー、アプリケーション、およびその他リソースの作成および管理で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 (+) をクリックして拡張するか、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、拡張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。これには、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で使用するページが

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することによって、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の説明については、10 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、現在のページ上のそれぞれのフィールドに関する概要、およびヘルプ・ブラウザー内のより詳しい情報が提供されています。

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。ナビゲーション・ツリーを使用して、これらのリソースを以下のように配置します。

- 「**統合アプリケーション**」：以下へのアクセスを提供します。
 - 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 Browser
- 「**リソース**」：以下へのアクセスを提供します。
 - WebSphere Business Integration Adapters
 - Common Event Infrastructure プロバイダー
 - スタッフ・プラグイン・プロバイダー
 - 拡張メッセージング・プロバイダー
- 「**サーバー**」 > 「**アプリケーション・サーバー**」 > *server_name*: 以下へのアクセスを提供します。
 - ビジネス・プロセスおよびヒューマン・タスクに関するコンテナ設定
 - Application Scheduler
 - ビジネス・ルール
 - イベント・サービス
 - 拡張メッセージング・サービス
 - セレクター
 - スタッフ・サービス
 - WebSphere Business Integration Adapter サービス
 - Web サービス参照サービス

管理コンソールからオンライン・ヘルプのアクセス

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。ヘルプの利用は、以下のいずれかの方法で行います。

- 新規 Web ブラウザーでオンライン・ヘルプを表示するには、コンソール・タスクバーから「ヘルプ」をクリックします。

ヘルプ・ブラウザーから以下を行うことができます。

- 「索引」タブに表示するトピックをブラウズします。ブラウザーの右パネルにオープンするには、そのトピックのリンクをクリックします。
- 「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。表示するには、トピック・リンクをクリックします。
- フィールドの吹き出しヘルプを表示するには、そのフィールド上にカーソルを移動します。
- 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して、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を表示します。

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「このページについての詳細情報」リンクをクリックします。

関連概念

『管理コンソール入門』

11 ページの『管理コンソールのボタン』

管理コンソール入門

タスクの以下のリストは、初めて管理コンソールを使用して WebSphere Process Server リソースを管理する場合に役立ちます。

- **管理コンソールのサーバーを始動します。**

startServer コマンドは、指定されたアプリケーション・サーバーの構成ファイルを読み取り、そのサーバーを始動します。startServer 使用の詳細については、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

- **管理コンソールを開始します。**

詳しくは、7 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、ワークスペースがどのように振る舞うかと同様に、管理コンソールでデータをどのように表示するかを制御します。8 ページの『管理コンソール設定の変更』を参照してください。

- **コンソール・スコープを設定します。**

スコープは、管理コンソールでリソースが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、またはサーバー・スコープのコンソール・コレクション表に表示されます。スコープ設定の詳細については、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

- **情報を表示するために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。9 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、非アクティブの時間が 30 分を超えるとコンソール・セッションがタイムアウトになります。WebSphere Application Server for z/OS インフォメーション・センターに説明されているように、`deployment.xml` 構成ファイルを編集すれば、この値は変更できます。

- **自分の作業をマスター・リポジトリに保管します。**

変更内容をマスター・リポジトリに保管するまで、コンソールは、ローカル・ワークスペースを使用して変更内容を追跡します。変更内容を保管するには、「システム管理」>「変更をマスター・リポジトリに保管」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

管理コンソールにアクセスするには、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。コンソールで作業が終了したら、作業を保管してログアウトします。

以下のステップを実行して、コンソールを開始および停止します。

1. 管理コンソールを開始します。

- a. 分散プラットフォーム: WebSphere ベース製品用の `server1` アプリケーション・サーバー上で管理コンソールが実行されていることを確認してください。
- b. 管理コンソールにアクセスするのに使用する Web ブラウザーで Cookie を使用可能にします。
- c. 分散プラットフォーム: Cookie を使用可能にした Web ブラウザーで、以下を入力します。 `http://your_fully_qualified_server_name:9060/ibm/console`

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を指定します。管理コンソールがローカル・マシン上にある場合、セキュリティーが使用可能でないと、`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。

セキュリティーが使用可能の場合、要求は

`https://your_fully_qualified_server_name:9043/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です。

注: コンソールは、ポート 9060 および 9043 をデフォルトで使用します。実際のインストール用のポート番号は変わることがあります。

管理コンソールはブラウザーでロードされ、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**」フィールドで、ユーザー名またはユーザー ID を入力します。

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

- 既存のユーザーをセッションから強制ログアウトします。既存のユーザー ID 用の構成ファイルは、一時領域に保管されます。
- 既存ユーザー ID のログアウトを待つか、またはセッションのタイムアウトを待ちます。
- 別のユーザー ID を指定します。

1 つのユーザー ID は、それをログインで使用したセッションの期間中のみ使われます。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。

- また、コンソールがグローバル・セキュリティーを使用する場合は、「パスワード」フィールドにパスワードを入力する必要があります。
- 「OK」をクリックします。

管理コンソールには、ここで「ようこそ」ページが表示されます。

3. コンソールを停止します。

- 「システム管理」>「変更内容をマスター・リポジトリに保管」>「保管」をクリックして、このセッション間に行った作業をすべて保管します。
- 「OK」をクリックします。

変更はマスター・リポジトリに保管され、管理コンソールは閉じます。

管理コンソール設定の変更

管理コンソール設定を使用してコレクション・ページ (テーブルのデータまたはリソースのコレクションをリストするページ) 上のデータの表示をカスタマイズすることができます。設定はユーザー・レベルで設定され、通常、管理コンソールの各領域に対して別個に設定する必要があります。

コレクション・ページに対して以下の表示設定を設定できます。

- 「**最大行数**」: コレクションが大きいときに表示される行の最大数を指定します。指定した最大値より多くの行がある場合、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。これが使用可能になっている場合、コンソール・コレクション・ページでは、最初に、保存されているフィルター基準を使用して、設定の下を表にデータを表示します。詳しくは、9 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。
- 「**最大結果セットサイズ**」: 検索が戻ることができ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設定します。

1. 任意のコレクション・ページから「設定」をクリックします。

このページは設定フィールドを表示するために拡大されます。

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

コレクション・テーブルは更新されて、指定した値どおりに表示されます。

ワークスペースを自動的に更新するかどうか、およびデフォルトでどのスコープを使用するかなど、グローバル管理コンソール設定を設定することもできます。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。これらの設定の設定方法に関する資料については、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は、WebSphere Process Server のデータまたはリソースのリストを表示します。フィルターを使用して、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 つの列のみにフィルターを設定できます。

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

「フィルター」ダイアログ・ボックスがテーブル上部の行の上に開きます。

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

基準は、それが表示されるためにテーブル項目の名前で見つかる必要のあるストリングです。ストリングは、ワイルドカード文字として % 記号 (%), アスタリスク (*), または疑問符 (?) 記号を含むことができます。例えば、「Resource アダプター」ページで、「名前」列用のフィルター基準として *JMS* を入力することで、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、検索基準と正確に一致するように、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 (¥) を付けます。

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「検索項目」フィールドに指定します。

¥(XA¥)

4. 「実行」をクリックします。

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

管理コンソール・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 つのいずれかでフォーマットされています。各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、その効果的な使用に役立ちます。

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) の収集を管理します。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テーブルに表示するか、およびその表示方法をどうするかの設定が容易になります。

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

標準的ボタンは 11 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、コレクション表から 1 つ以上のオブジェクトを選択して、ボタンをクリックする必要があります。アクションは、すべての選択されたオブジェクトに適用されます。

トグル・ボタンのソート

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。これには、通常次のエレメントが 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般プロパティ・セットがあります。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。

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、ほとんどの場合、読み取り専用です。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。

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を表

示されます。異なるレベルのトポロジーを拡張および縮小して、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で構成される構成プロセスを実行するのに役立ちます。ウィザードは、構成中であ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。

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 (Check all)」	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 (Uncheck all)」	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なにも実行されなくなります。
「フィルター・ビューの表示 (Show the filter view)」	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。 9 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・ビューの非表示 (Hide the filter view)」	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。
「フィルター値のクリア (Clear filter value)」	フィルターに対して行ったすべての変更をクリアし、最後に保管した値に復元します。

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、ページに変更を保管します。

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」ボタンおよび「進む」ボタンの使用をサポート

していないため、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」ボタンおよび「進む」ボタンの使用をサポートしていないため、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。

選択の消去

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。

クローズ

ダイアログを閉じます。

削除 選択したインスタンスを除去します。

OK 変更を保管して、ページを終了します。

リセット

タブまたはページ上の変更を消去し、最後に保管された値を復元します。

保管 ローカル構成の変更をマスター構成に保管します。

管理コンソールで使用するボタン (WebSphere Application Server と WebSphere Process Server の両方のリソースを管理する目的で使用) の完全なリストについては、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

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

WebSphere Process Server 管理環境が正しくセットアップされていることを確認するには、特定のタスクが必要です。

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

表 2. WebSphere Process Server 管理環境のセットアップに関連したタスク

タスク	説明箇所
製品の構成	インストール後の製品の構成
ポートの構成	ポート競合の回避およびポート番号設定
管理アーキテクチャーのセットアップ	管理アーキテクチャーのセットアップ
セル全体に対する設定を構成	セル全体に対する設定を構成
サーバー構成ファイルの変更	サーバー構成ファイルの使用
サーバーおよびネットワーク・デプロイメント環境の管理	プロセス・サーバーの管理
WebSphere Process Server リソースの構成	WebSphere Process Server リソースの構成

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

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

管理アーキテクチャーのセットアップ

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

このタスクでは以下の点を完了していることが前提となります。

1. テープからの製品のアンロード (Unloaded the product from the tape)
2. インストール・スクリプトの実行による製品定義の作成 (Created the product definitions by running the install script)
3. 製品の構成
4. ポートの構成 (『ポート競合の回避』および『ポート番号設定』を参照)

始める前に、WebSphere Application Server for z/OS インフォメーション・センターの管理アーキテクチャーのセットアップで説明されているタスクを確認する必要があります。

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

1. 管理サービス設定を使用して管理サービスを構成します。
2. データベース製品のインスタンスをインストールします。
3. データベース・インスタンス内に WebSphere Process Server データベースを作成します。
4. WebSphere Process Server データベース内にテーブルをセットアップします。
5. セルを構成します。
6. デプロイメント・マネージャーを構成します。
7. 管理コンソールで、「環境」>「WebSphere 変数」>(ノードのスコープ)>「DatabaseName_JDBC_DRIVER_PATH」と選択して、WebSphere Process Server データベースがある場所を定義します。

以下の変数のいずれかを使用して、データベースを定義します。

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

詳しくは、『WebSphere 変数の構成』を参照してください。

8. ノードを管理します。
9. ノード・エージェントを管理します。
10. ノード・グループを管理します。

11. リモート・ファイル・サービスを構成します。

ご使用の WebSphere Process Server 環境は、作業を処理する準備ができました。

アプリケーションをインストールしてください。

セル全体に対する設定を構成

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

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

ネットワーク・デプロイメント環境を確立する場合、またはクラスターへのアプリケーションのインストールを計画している場合も、セル全体に特定の設定を構成する必要があります。

- 仮想ホストの構成
- 変数の構成

WebSphere Process Server データベースを定義する変数をノードまたはセルの有効範囲で構成する必要があります。変数 `DatabaseName_JDBC_DRIVER_PATH` の場合以下のいずれかを使用します:

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

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

WebSphere Process Server サーバー構成によって、エンタープライズ・アプリケーションおよびそのコンポーネントを実行するためのサービスをサーバーが提供する方法を制御する設定が提供されます。管理者は、既存アプリケーション・サーバー環境で、サーバーを作成および構成できます。

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

表 3. サーバー管理タスク

タスク	説明箇所
サーバーの作成	サーバーの作成
アプリケーション・サーバーの管理	アプリケーション・サーバーの管理
トランスポート・チェーンの構成	トランスポート・チェーンの構成
カスタム・サービスの開発	カスタム・サービスの開発
アプリケーション・サーバー用プロセスの定義	アプリケーション・サーバー・プロセスの定義
Java™ 仮想マシンの構成	JVM の構成

サーバーの作成

WebSphere Process Server の場合、wsadmin createApplicationServer コマンド、また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**」ウィザードのいずれかを使用して、サーバーを作成することができます。

このタスクは、製品がインストール済みおよび構成済みであることを前提にしています。

WebSphere Process Server アプリケーション用の処理環境を提供するためにサーバーを作成します。

メソッドを選択して、そのメソッドのステップを完了します。

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

アプリケーション・サーバー・ロードマップの管理

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

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

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

表 4 に、アプリケーション・サーバーを管理するタスクをリストします。

表 4. 管理コンソールで使用可能なサーバー管理タスク

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

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

インストール後に、ユーザーの WebSphere Process Server 環境用に、Application Scheduler、ビジネス・プロセス、Common Event Infrastructure、ビジネス・ルール、セレクトター、およびリレーションシップなどの、キー・リソースを構成する必要があります。

管理環境を設定するためのインストール後タスクを完了する際に重要なステップは、ユーザーのキー・リソースの構成です。表 5 を使用して、リソースの構成に必要な情報を見つけます。

表 5. リソース構成のステップ

リソース	構成方法
アプリケーション・スケジューラー	Application Scheduler の構成
Business Process Choreographer	Business Process Choreographer の構成
Common Event Infrastructure	Common Event Infrastructure の構成
ビジネス・ルール	ビジネス・ルールの構成
セレクトター・コンポーネント	セレクトター・コンポーネントの管理
リレーションシップ・サービス	リレーションシップ・サービスの管理

サーバー構成ファイルの使用

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

WebSphere Process Server のサーバー構成ファイルを処理するには、WebSphere Application Server Network Deployment バージョン 6.0.x インフォメーション・センターの「サーバー構成ファイルの使用」トピックの手順に従ってください。

アプリケーションのインストールと管理

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

インストール後に、タスクのカスタマイズ、アプリケーションのデプロイおよび管理環境の管理などの、アプリケーションおよび環境の管理を行う必要があります。

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

アプリケーションおよびアプリケーション・サービスの管理

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

WebSphere Process Server 設定ロードマップの構成

ソフトウェアのインストール後に構成する WebSphere Process Server のサービスをリストします。

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

表 6. 構成するサービス

サービス	説明箇所
アプリケーション・スケジューラー	スタンドアロン・サーバー用 Application Scheduler の構成
イベント・サービス	イベント・サービスの管理
拡張メッセージング・サービス	拡張メッセージング・サービスの使用可能化
WebSphere Business Integration Adapter サービス	WebSphere Business Integration Adapter の操作
リレーションシップ・サービス	リレーションシップ・サービスの管理

イベント・サービスの管理

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

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

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

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

1. 構成するアプリケーション・サーバーを選択します。

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

サーバー・プロパティーが表示されます。

2. イベント・サービス構成パネルを表示します。

Business Integration プロパティーから「イベント・サービス」を選択します。

「イベント・サービス」構成パネルが表示されます。

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

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

4. Java Naming and Directory Interface (JNDI) 名が正しいかどうか確認します。

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

5. 必要に応じてカスタム・プロパティーを選択します。

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

6. 構成を保管します。

「適用」をクリックして構成を保管し、「イベント・サービス構成」パネルに留まります。「OK」をクリックして構成を保管し、Business Integration パネルに戻ります。

7. 変更内容をサーバーに適用します。

アプリケーション・サーバーを停止および再始動して、変更内容を有効にします。

リレーションシップ・サービスの管理

リレーションシップ・サービスは、システム内で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、リレーションシップの定義の指定と、定義から派生するインスタンスの操作の要件を可能にします。

リレーションシップは、異なる環境からの同じデータの ID を相関させます。リレーションシップの参加者は、リレーションシップで担うロールによって識別されます。

リレーションシップおよびロールは、WebSphere Integration Developer のリレーションシップ・エディター・ツールのグラフィカル・インターフェースで設計した定義で記述されます。リレーションシップの定義は、リレーションシップの内容を記述し、各ロールを識別し、およびロールの関連性を指定するテンプレートです。ロールの定義により、参加者についての構造および制約要件を指定します。それぞれのリレーションシップの定義には、その構成要素として、ロールの定義があります。

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、特定のサーバーに対して J2EE アプリケーションの一部としてデプロイされます。

リレーションシップの作成、リレーションシップのタイプ、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およびタスク情報について詳しくは、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。

実行時に、マップまたはその他 WebSphere Process Server コンポーネントが実行していて、リレーションシップ・インスタンスが必要になったとき、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、作成されるかまたは取得されます。リレーションシップおよびロールのランタイム・インスタンス・データの作成および取り扱いという要件に対応するために、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 WebSphere Process Server のリレーションシップ・マネージャー・ツールに公開します。リレーションシップ・マネージャーのグラフィカル・インターフェースを使用して、リレーションシップおよびロールと対話します。

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、デフォルトのデータ・ソース内の特定サーバー上に保管されるリレーションシップ・テーブルに保管されます。デフォルトのデータ・ソースは、リレーションシップ・サービスを構成したときに指定したものです。

リレーションシップ・サービスは各サーバー上でセル・レベルで実行します。

「**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・インスタンスについて作業する前に、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 の使用についての詳細な情報は、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の管理』および『WebSphere Process Server の管理コンソール』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 環境を対象にしたリレーションシップ・サービスのために実行する、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、リレーションシップ・サービス用のパラメーターを構成する必要があります。

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

リレーションシップ・サービス用のデータ・ソース・パラメーターおよびページ・サイズ・パラメーターを設定するには、以下のタスクを実行します。

1. 管理コンソールが稼働していることを確認します。

2. ナビゲーション・ペインで、「統合アプリケーション」>「**Relationship Manager**」をクリックします。
3. 「**リレーションシップ・サービス構成**」をクリックします。

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

4. 「**データ・ソース**」フィールドで、セル・レベルで定義されたデータ・ソースの JNDI 名を入力して、リレーションシップ・サービスのデフォルトのデータ・ソースを指定します。これは、リレーションシップ・サービス用のテーブルが保管される場所です。デフォルトでは、リレーションシップに関連するすべてのスキーマがこのデータ・ソース内に作成されます。
5. 「**ページ・サイズ**」フィールドに、**Relationship Manager** のリレーションシップ・インスタンスの検索用のページごとに戻される結果の最大数を指定します。デフォルト値は 500 です。
6. 以下のオプションがあります。
 - 変更内容を保管し、そのページに留まる場合は、「**適用**」をクリックします。
 - 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、最後に保管された値に戻す場合は、「**リセット**」をクリックします。
 - そのページで保管されていない変更内容をすべて破棄し、前のページに戻る場合は、「**キャンセル**」をクリックします。

リレーションシップ・サービスで管理されるリレーションシップの表示

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

必要なセキュリティ・ロール: どの WebSphere セキュリティ・ロールでもこの構成を表示することができます。

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

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

4. このコレクション・ページの表示設定を設定するには、「**設定**」をクリックします。フィールド値を必要な値に変更し、「**適用**」をクリックします。
5. リレーションシップの構成プロパティを表示するには、リレーションシップ・コレクション・テーブルのリレーションシップ名をクリックします。

リレーションシップ・プロパティの表示

リレーションシップ・サービスにより、リレーションシップ・サービス・レベル (リレーションシップ・サービスに対して適用される時) と個々のリレーションシ

ップ・レベル (個々のリレーションシップに適用されるとき) の両方のレベルで管理される構成プロパティを表示するには、このタスクを実行します。

必要なセキュリティ・ロール: どの WebSphere セキュリティ・ロールでもこの構成を表示することができます。

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

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

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

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

クラスター環境の作成

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

クラスター環境を作成するには、事前に必要な作業がいくつかあります。

- クラスタリングを正常にインプリメントするための十分なリソースがあることを確認します。
- クラスター環境にデプロイするサービス・アプリケーションを分析します。実行するオプションの手順の一部は、サービス・アプリケーションの必要性に応じて異なります。
- アプリケーション・ロジックがクラスター環境に耐えられることを確認します。例えば、以下のことを確認します。
 - 孤立した要求や不適切に処理される要求に備える場合、アプリケーションが分割されたキューを許容すること。アプリケーション障害が発生した場合、これらのいずれかまたは両方の状況が生成されることがあります。
 - ローカルに維持するシステム規模の値がないこと。
- WebSphere Application Server for z/OS インフォメーション・センターの Network Deployment とクラスタリングの説明を確認しておきます。
- 手順を実行する前に、ここで記載する指示を確認しておきます。手順全体の概念を把握しておく、効率よく手順に移ることができます。

- WebSphere Application Server for z/OS がインストール済みであることを確認します。

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

- WebSphere Process Server for z/OS クラスタ環境をサポートするために、WebSphere Application Server for z/OS を構成します。

このためには、以下の作業を行う必要があります。

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

注: デプロイメント・マネージャーには統合しないでください。これは、BBOWMNAN というジョブを、空の管理対象ノードを作成するためのプロセスの一部として実行しないでくださいという意味です。

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

この作業の説明については、「インストール・スクリプトの実行」を参照してください。

- データベースおよびストレージ・グループを作成します。

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

この作業の説明については、『データベースおよびストレージ・グループの作成 (Creating databases and storage groups)』を参照してください。

- デプロイメント・マネージャーの応答ファイルを編集します。

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

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

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

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

この作業の説明については、『構成スクリプトの実行 (Run the configuration script)』を参照してください。

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

この作業の説明については、『インストール・スクリプトの実行』参照してください。

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

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

この作業の説明については、『構成スクリプトの実行 (Run the configuration script)』を参照してください。

- 管理対象ノードをデプロイメント・マネージャーに統合します。

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

アプリケーションが必要とする容量と可用性が単一サーバー提供のものでは足りない場合に、クラスター環境を作成します。クラスター環境には、このほかに以下の利点があります。

ワークロード・バランシング

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

アプリケーションの処理能力

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

アプリケーションの可用性

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

保守容易性

計画した保守のために、アプリケーションの処理を停止することなく、サーバーを停止することができます。

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

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

1. クラスター環境を設計します。
 - a. セルのトポロジーを設計します。セルに必要な物理的、論理的リソースを決定します。

- b. セルのさまざまなコンポーネント用にデータベースとスキーマのどちらを使用するかを決定します。
 - c. イベントをモニターする必要があるかどうか、モニター対象のイベントをどのサーバーでホストするかを決定します。
2. セルに必要なその他のデータベースを作成します。例えば、以下のようなデータベースを作成します。

- a. 環境変数を作成します。 以下を作成します。

- DB2UNIVERSAL_JDBC_DRIVER_PATH = /db2810/jcc/classes
- DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH = /db2810/jcc/lib
- JDBC JCC プロパティ・ファイルを DMGR/サーバーの付属カスタム・プロパティおよびサーバント・カスタム・プロパティに追加します。

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

db2.jcc.propertiesFile (例)

/shared/db2810/jcc/properties/DB2JccConfiguration.properties

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

- 3. **オプション:** イベントをモニターする必要がある場合は、モニター対象のイベントを処理するクラスターを作成します。
 - a. デフォルトの WebSphere Process Server テンプレートを使用して、クラスターとクラスター・メンバーを作成します。
 - b. ターゲットとして構成しているクラスターと 23 ページの『クラスター環境の作成』で作成した Common Event Infrastructure (CEI) データベースを使用して、CEI アプリケーションをインストールし、メッセージ駆動型 Bean (MDB) アプリケーションをインストールします。(Common Event Infrastructure の構成)
- 4. **オプション:** 管理アプリケーションにビジネス・ルールやセレクターが含まれていて、アプリケーションのデプロイ後に、アプリケーションでそのルールやセレクターに変更を加える必要がある場合は、管理アプリケーションを処理するクラスターを作成します。

注: ビジネス・ルールやセレクターの管理を、ほかの WebSphere Process Server コンポーネントを処理するクラスターにデプロイすることを決定した場合は、このステップをステップ 3 またはステップ 6 (27 ページ) と組み合わせることで実行することができます。

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

注: ほかの WebSphere Process Server コンポーネントを処理するクラスターが管理メッセージングに対応することを決定した場合は、このステップをステップ 3 (26 ページ)、ステップ 4 (26 ページ)、またはステップ 6 と組み合わせて実行することができます。

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

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

ステップ 23 ページの『クラスター環境の作成』で作成したデータ・ソースを使用します。ステップ 1b (26 ページ) で、1 つのデータベースで複数のスキーマを使用することを決定した場合は、メッセージング・エンジン・データ・ソースの構成時に適切なスキーマを選択します。

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

ステップ 23 ページの『クラスター環境の作成』で作成したデータ・ソースを使用します。ステップ 1b (26 ページ) で、1 つのデータベースで複数のスキーマを使用することを決定した場合は、メッセージング・エンジン・データ・ソースの構成時に適切なスキーマを選択します。

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

このステップをステップ 5 (26 ページ) と組み合わせて実行する場合は、そのステップで指定したデフォルトのメッセージング・クラスターを選択してください。そうでない場合は、適切なオプションを選択して、ステップ 5 (26 ページ) で作成したクラスターを選択します。

- c. **オプション:** ステップ 23 ページの『クラスター環境の作成』で作成したデータベースまたはスキーマを使用して Business Process Choreographer のサポートを構成します。(Business Process Choreographer の構成)

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

- d. **オプション:** 「拡張構成」パネルを使用して、エミッター・プロファイルの Java Naming and Directory Interface (JNDI) 名を選択し、クラスターとステップ 23 ページの『クラスター環境の作成』で作成したエミッター・ファクトリー・プロファイルを関連付けます。

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

これで、アプリケーションをクラスターにインストールできるようになりました。

関連概念

WebSphere Process Server 管理環境のセットアップ

サービス・アプリケーションをサポートするサーバーまたはクラスターの準備

WebSphere Process Server 管理環境のセットアップ

サービス・アプリケーションをサポートするサーバーまたはクラスターの準備

関連タスク

インストール・メディアからの製品コードのアンロード

インストール・スクリプトの実行

製品の構成

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

Common Event Infrastructure の構成

Business Process Choreographer の構成

実動サーバーへのモジュールのインストール

製品コードのアンロード

インストール・スクリプトの実行

製品の構成

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

Business Process Choreographer の構成

実動サーバーへのモジュールのインストール

関連情報

サーバー構成ファイルの使用

メッセージング・エンジンについて

データ・ストアについて

バス宛先の学習

デフォルトのメッセージング・プロバイダーについて

サーバー構成ファイルの使用

メッセージング・エンジンについて

データ・ストアについて

バス宛先の学習

デフォルトのメッセージング・プロバイダーについて

Common Event Infrastructure の構成

アプリケーションをサポートするサーバーまたはクラスターの準備

サーバーまたはクラスターは、WebSphere Process Server Service Component Architecture (SCA) アプリケーションのデプロイメント・ターゲットとして構成し、別のロケーションまたは両方にデプロイされている WebSphere Process Server アプリケーションをサポートできます。管理コンソールの拡張構成ページを使用して、必要な構成タスクを実行します。拡張構成ページは、サーバー・スコープとクラスター・スコープの両方で使用できます。

WebSphere Process Server アプリケーションをサポートするサーバーまたはクラスターの構成には、3 つの基本的なシナリオがあります。

- サーバーまたはクラスターは、別のサーバーまたはクラスターにデプロイされたアプリケーションをサポートします (この構成でのサーバーまたはクラスターは、WebSphere Process Server アプリケーションをホストしません。ただし、ビジネス・ルール・マネージャーまたはサービス統合バス・メンバーはホストできます)。デフォルトの構成オプションでは、サービス統合バス・メンバーの構成なしでこのシナリオが選択されます。この構成の特定の情報について詳しくは、30 ページの『別のサーバーまたはクラスターにデプロイされた WebSphere Process Server アプリケーションのサポート』を参照してください。
- サーバーまたはクラスターは、メッセージング・エンジンが既に構成されているリモート・サービス統合バス・メンバーに依存します。サーバーまたはクラスターは、WebSphere Process Server アプリケーションをホストします。宛先は、ローカルまたはリモートでホスト可能です。また、ビジネス・ルール・マネージャーもホストできます。この構成の特定の情報について詳しくは、32 ページの『バス・メンバー・サポートなしの WebSphere Process Server アプリケーションのサポート』を参照してください。
- サーバーまたはクラスターには、サービス統合バス・メンバーが組み込まれています。サーバーまたはクラスターは、WebSphere Process Server アプリケーションをホストし、オプションで、それらのアプリケーションで必要とされるメッセージング・エンジンと宛先もホストします。これらのメッセージング・エンジンは、デプロイメント・ターゲットが異なるアプリケーションで必要とされる宛先もホストできます。最終的に、サーバーまたはクラスターはビジネス・ルール・マネージャーもホストできます。この構成の特定の情報について詳しくは、34 ページの『WebSphere Process Server アプリケーションとバス・メンバーのサポート』を参照してください。

上記のシナリオのいずれかを構成する場合は、アプリケーションによって使用される宛先 (Java Message Service (JMS) キュー) の配置場所を考慮する必要があります。宛先は、アプリケーションのインストール先と同じサーバーまたはクラスターでホストするか、リモートのサーバーまたはクラスターでホストすることが可能です。拡張構成ページには、これらの宛先をホストする場所を指定するオプションがあります。

構成プロセスの一部として、次のオプションも指定できます。

- ビジネス・ルール・マネージャーのインストール

ビジネス・ルール・マネージャーは、クラスター内の 1 つのサーバーまたは全サーバーでビジネス・ルールを変更および管理するために使用する、Web ベースのツールです。

注: このオプションは、WebSphere Process Server の一部としてこのツールをインストールしている場合にのみ使用可能です。

- Common Event Infrastructure (CEI) 生成イベントの送付

WebSphere Process Server モニター機能を使用して、実行時に生成されたすべての Common Base Event を正しい CEI サーバー・アプリケーションに送付して処理することができます。CEI 生成イベントを送付するため、イベント・エミッタ

ー・ファクトリー・プロファイルの Java Naming and Directory Interface (JNDI) 名を指定する必要があります。クラスターまたはサーバーにデプロイされるすべてのアプリケーションは、同じエミッター・プロファイルを使用します。既に CEI を構成済みの場合は、インストールされたイベント・サーバー・アプリケーションを指すエミッター・ファクトリー・プロファイルが存在しており、拡張構成ページに表示されます。

別のサーバーまたはクラスターにデプロイされた WebSphere Process Server アプリケーションのサポート

クラスターまたはサーバーは、別のクラスターまたはサーバーにデプロイされている WebSphere Process Server サービス・アプリケーションをサポートするように構成できます。これは、デフォルトの構成シナリオです。

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。

このシナリオでは、クラスターまたはサーバーは、WebSphere Process Server アプリケーションをホストしません。ただし、リモートのサーバーまたはクラスターで実行されているアプリケーションが使用するビジネス・ルール・マネージャーおよび JMS キュー (宛先) はホストできます。JMS 宛先は、サーバーまたはクラスターで実行されるメッセージング・エンジンでホストされます。拡張構成ページを使用し、各システム統合バスのメンバーとしてサーバーまたはクラスターが追加されるときに、デフォルトで 1 つのメッセージング・エンジンが作成されます。

1. 管理コンソール内で、該当するスコープに応じて以下のいずれかをクリックします。
 - サーバーを構成する場合: 「サーバー」 > 「アプリケーション・サーバー」 > *server_name* > 「拡張構成」
 - クラスターを構成する場合: 「サーバー」 > 「クラスター」 > *cluster_name* > 「拡張構成」
2. このサーバーまたはクラスターにビジネス・ルール・マネージャーをインストールする場合は、「**Business Rules Manager のインストール**」をクリックします。
3. 「**エミッター・ファクトリー・プロファイル JNDI 名**」フィールドの値を検証します。
 - 生成イベントの送付のデフォルト構成を変更する場合は、フィールドから適切なエミッター・ファクトリー・プロファイル JNDI 名を選択します。注意すべき点は、サーバーまたはクラスターにデプロイされるすべてのアプリケーションは、同じエミッター・ファクトリー・プロファイルを使用することです。
 - デフォルトのエミッター・ファクトリー・プロファイルを使用する場合は、メニューから「なし」を選択します。
4. 「**SCA アプリケーションのホストを構成しない (Do not configure to host SCA applications)**」オプションが選択されていることを確認します。
5. サーバーまたはクラスターによりアプリケーションの宛先をホストする場合は、以下のタスクのいずれかを実行します。

構成されるリソース	実行するステップ
サーバーの構成	<ol style="list-style-type: none"> 1. 「デフォルト宛先ロケーション」をクリックします。 2. SI Bus サービスにより、Cloudscape を使用してデータ・ソースとデータベースを自動的に作成させる場合は、「デフォルト・データ・ソース値を使用」チェック・ボックスをクリックし、ステップ 6 (32 ページ) に進みます。 3. 新規のデータ・ソースとデータベースを作成する場合は、「JDBC プロバイダー」メニューを使用して、適切な JDBC プロバイダー・テンプレートを指定します。 4. オプションとして、「アプリケーション・バス・スキーマ名」フィールドに、アプリケーション・バス・データ・ソースのテーブルを含めるために使用されるデータベース・スキーマの名前を入力します。 5. オプションとして、「システム・バス・スキーマ名」フィールドに、システム・バス・データ・ソースのテーブルを含めるために使用されるデータベース・スキーマの名前を入力します。 6. 「データ・ソースのユーザー名」フィールドに、データベースにアクセスするとき使用するユーザー名を入力します。 7. 「データ・ソースのパスワード」フィールドに、上記のユーザー名に関連付けられたパスワードを入力します。 8. 「アプリケーション・バス・データ・ソース・プロパティ」フィールドで、SCA アプリケーション・バスが使用するデータ・ソースで必要とされるカスタム・プロパティを入力します。 9. 「システム・バス・データ・ソース・プロパティ」フィールドで、SCA システム・バスが使用するデータ・ソースで必要とされるカスタム・プロパティを入力します。 10. メッセージング・エンジンにデータ・ソースのデータベース表を作成させる場合は、「テーブルの作成」を選択します。メッセージング・エンジンをホストするサーバーの始動時に、テーブルが作成されます。

構成されるリソース	実行するステップ
クラスタの構成	<ol style="list-style-type: none"> 1. 「デフォルト宛先ロケーション」をクリックします。 2. 「JDBC プロバイダー」メニューを使用して、適切な JDBC プロバイダー・テンプレートを指定します。 3. オプションとして、「アプリケーション・バス・スキーマ名」フィールドに、アプリケーション・バス・データ・ソースのテーブルを含めるために使用されるデータベース・スキーマの名前を入力します。 4. オプションとして、「システム・バス・スキーマ名」フィールドに、システム・バス・データ・ソースのテーブルを含めるために使用されるデータベース・スキーマの名前を入力します。 5. 「データ・ソースのユーザー名」フィールドに、データベースにアクセスするときに使用するユーザー名を入力します。 6. 「データ・ソースのパスワード」フィールドに、上記のユーザー名に関連付けられたパスワードを入力します。 7. 「アプリケーション・バス・データ・ソース・プロパティ」フィールドで、SCA アプリケーション・バスが使用するデータ・ソースで必要とされるカスタム・プロパティを入力します。 8. 「システム・バス・データ・ソース・プロパティ」フィールドで、SCA システム・バスが使用するデータ・ソースで必要とされるカスタム・プロパティを入力します。 9. メッセージング・エンジンにデータ・ソースのデータベース表を作成させる場合は、「テーブルの作成」を選択します。メッセージング・エンジンをホストするサーバーの始動時に、テーブルが作成されます。

6. 「OK」をクリックします。

バス・メンバー・サポートなしの WebSphere Process Server アプリケーションのサポート

クラスタまたはサーバーは、WebSphere Process Server アプリケーションのデプロイメント・ターゲットとして構成できます。アプリケーションの宛先は、アプリケーションと同じクラスタかサーバー、またはリモートのクラスタかサーバーでホストできます。クラスタまたはサーバーでアプリケーションをホストするよ

うにセットアップすると、構成の一部として SchedulerCalendar アプリケーションが自動的にインストールされる点に注意してください。

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

このトピックでは、リモートのサーバーまたはクラスターでアプリケーションの宛先をホストすると同時に、1つのサーバーまたはクラスターで WebSphere Process Server アプリケーションをホストする方法を説明します。同じクラスターまたはサーバーで宛先をホストする方法については、34ページの『WebSphere Process Server アプリケーションとバス・メンバーのサポート』を参照してください。

1. 管理コンソール内で、該当するスコープに応じて以下のいずれかをクリックします。
 - サーバーを構成する場合: 「サーバー」 > 「アプリケーション・サーバー」 > *server_name* > 「拡張構成」
 - クラスターを構成する場合: 「サーバー」 > 「クラスター」 > *cluster_name* > 「拡張構成」
2. このサーバーまたはクラスターにビジネス・ルール・マネージャーをインストールする場合は、「**Business Rules Manager のインストール**」をクリックします。
3. 「**エミッター・ファクトリー・プロファイル JNDI 名**」フィールドの値を検証します。
 - 生成イベントの送付のデフォルト構成を変更する場合は、フィールドから適切なエミッター・ファクトリー・プロファイル JNDI 名を選択します。注意すべき点は、サーバーまたはクラスターにデプロイされるすべてのアプリケーションは、同じエミッター・ファクトリー・プロファイルを使用することです。
 - デフォルトのエミッター・ファクトリー・プロファイルを使用する場合は、メニューから「なし」を選択します。
4. 「**SCA アプリケーションのホストを構成しない (Do not configure to host SCA applications)**」オプションの選択を解除します。
5. 次のようにして、宛先のロケーションを指定します。
 - a. 「**リモート宛先ロケーション (Remote Destination Location)**」をクリックします。
 - b. 関連するメニューを使用して、宛先をホストすることになるリモート・サーバーまたはクラスターの名前を指定します。
6. 「**OK**」をクリックします。
 - このクラスターまたはサーバー上で Business Process Execution Language (BPEL) によって定義されたビジネス・プロセスを使用する計画の場合は、ビジネス・プロセス・コンテナ・ウィザードを使用して必要な構成を実行する必要もあります。
 - ヒューマン・タスクが含まれるアプリケーションを使用する計画、あるいはこのクラスターまたはサーバーで Business Process Choreographer Explorer を使用する計画の場合は、ヒューマン・タスク・コンテナ・ウィザードを使用して必要な構成を実行する必要もあります。

WebSphere Process Server アプリケーションとバス・メンバーのサポート

クラスターまたはサーバーは、WebSphere Process Server アプリケーションをホストし、リモートのクラスターまたはサーバーにインストールされた WebSphere Process Server アプリケーションをサポートするように構成できます。この構成シナリオでは、アプリケーション・バスが現在のクラスターまたはサーバー用に構成されます。加えて、WebSphere Process Server アプリケーションをホストするために、SchedulerCalendar アプリケーションが自動的にインストールされます。このシナリオでは、アプリケーションの宛先を、アプリケーション自体と同じクラスターかサーバー、またはリモート・ロケーションでホストできます。このトピックでは、サーバーまたはクラスターをセットアップして宛先をホストする方法を説明します。

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

1. 管理コンソール内で、該当するスコープに応じて以下のいずれかをクリックします。
 - サーバーを構成する場合: 「サーバー」 > 「アプリケーション・サーバー」 > *server_name* > 「拡張構成」
 - クラスターを構成する場合: 「サーバー」 > 「クラスター」 > *cluster_name* > 「拡張構成」
2. このサーバーまたはクラスターにビジネス・ルール・マネージャーをインストールする場合は、「**Business Rules Manager のインストール**」をクリックします。
3. 「**エミッター・ファクトリー・プロファイル JNDI 名**」フィールドの値を検証します。
 - 生成イベントの送付のデフォルト構成を変更する場合は、フィールドから適切なエミッター・ファクトリー・プロファイル JNDI 名を選択します。注意すべき点は、サーバーまたはクラスターにデプロイされるすべてのアプリケーションは、同じエミッター・ファクトリー・プロファイルを使用することです。
 - デフォルトのエミッター・ファクトリー・プロファイルを使用する場合は、メニューから「なし」を選択します。
4. 「**SCA アプリケーションのホストを構成しない (Do not configure to host SCA applications)**」オプションの選択を解除します。
5. アプリケーションの宛先をホストするには、以下のタスクのいずれかを実行します。

構成されるリソース	実行するステップ
サーバーの構成	<ol style="list-style-type: none"> 1. 「デフォルト宛先ロケーション」をクリックします。 2. SI Bus サービスにより、Cloudscape を使用してデータ・ソースとデータベースを自動的に作成させる場合は、「デフォルト・データ・ソース値を使用」チェック・ボックスをクリックし、ステップ 6 (36 ページ) に進みます。 3. 新規のデータ・ソースとデータベースを作成する場合は、「JDBC プロバイダー」メニューを使用して、適切な JDBC プロバイダー・テンプレートを指定します。 4. オプションとして、「アプリケーション・バス・スキーマ名」フィールドに、アプリケーション・バス・データ・ソースのテーブルを含めるために使用されるデータベース・スキーマの名前を入力します。 5. オプションとして、「システム・バス・スキーマ名」フィールドに、システム・バス・データ・ソースのテーブルを含めるために使用されるデータベース・スキーマの名前を入力します。 6. 「データ・ソースのユーザー名」フィールドに、データベースにアクセスするときに使用するユーザー名を入力します。 7. 「データ・ソースのパスワード」フィールドに、上記のユーザー名に関連付けられたパスワードを入力します。 8. 「アプリケーション・バス・データ・ソース・プロパティ」フィールドで、SCA アプリケーション・バスが使用するデータ・ソースで必要とされるカスタム・プロパティを入力します。 9. 「システム・バス・データ・ソース・プロパティ」フィールドで、SCA システム・バスが使用するデータ・ソースで必要とされるカスタム・プロパティを入力します。 10. メッセージング・エンジンにデータ・ソースのデータベース表を作成させる場合は、「テーブルの作成」を選択します。メッセージング・エンジンをホストするサーバーの始動時に、テーブルが作成されます。

構成されるリソース	実行するステップ
<p>クラスタの構成</p>	<ol style="list-style-type: none"> 1. 「デフォルト宛先ロケーション」をクリックします。 2. 「JDBC プロバイダー」メニューを使用して、適切な JDBC プロバイダー・テンプレートを指定します。 3. オプションとして、「アプリケーション・バス・スキーマ名」フィールドに、アプリケーション・バス・データ・ソースのテーブルを含めるために使用されるデータベース・スキーマの名前を入力します。 4. オプションとして、「システム・バス・スキーマ名」フィールドに、システム・バス・データ・ソースのテーブルを含めるために使用されるデータベース・スキーマの名前を入力します。 5. 「データ・ソースのユーザー名」フィールドに、データベースにアクセスするときに使用するユーザー名を入力します。 6. 「データ・ソースのパスワード」フィールドに、上記のユーザー名に関連付けられたパスワードを入力します。 7. 「アプリケーション・バス・データ・ソース・プロパティ」フィールドで、SCA アプリケーション・バスが使用するデータ・ソースで必要とされるカスタム・プロパティを入力します。 8. 「システム・バス・データ・ソース・プロパティ」フィールドで、SCA システム・バスが使用するデータ・ソースで必要とされるカスタム・プロパティを入力します。 9. メッセージング・エンジンにデータ・ソースのデータベース表を作成させる場合は、「テーブルの作成」を選択します。メッセージング・エンジンをホストするサーバーの始動時に、テーブルが作成されます。

6. 「OK」をクリックします。

- このクラスタまたはサーバー上で Business Process Execution Language (BPEL) によって定義されたビジネス・プロセスを使用する計画の場合は、ビジネス・プロセス・コンテナ・ウィザードを使用して必要な構成を実行する必要もあります。

- ヒューマン・タスクが含まれるアプリケーションを使用する計画、あるいはこのクラスターまたはサーバーで Business Process Choreographer Explorer を使用する計画の場合は、ヒューマン・タスク・コンテナ・ウィザードを使用して必要な構成を実行する必要もあります。

Business Process Choreographer の管理

Business Process Choreographer は、管理コンソールまたはスクリプトを使用して管理できます。

管理コンソールによる Business Process Choreographer の管理

管理コンソールを使用して実行可能な管理操作について説明します。

サーバーの補正サービスの管理

管理コンソールを使用して、アプリケーションの始動時に、自動的に補正サービスを開始し、リカバリー・ログの場所および最大サイズを指定します。

ビジネス・プロセスがアプリケーション・サーバーで実行されるときは、そのサーバーに補正サービスが開始されている必要があります。プロセスが完了する前に、いくつかのトランザクションで行われた更新を管理するため、補正サービスが使用されます。

管理コンソールを使用して、アプリケーション・サーバーの補正サービスのプロパティを表示および変更することができます。

1. 管理コンソールを表示します。
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」 → *server_name* をクリックします。
3. 「構成」タブの「コンテナの設定」の下で、「コンテナ・サービス」 → 「補正サービス (Compensation service)」をクリックします。このアクションにより、パネルに補正サービス・プロパティが表示されます。

サーバー始動時にサービスを使用可能にする

アプリケーション・サーバーの始動時は必ず、補正サービスの開始を試行するように指定します。

このチェック・ボックスが選択されていることを確認します。ビジネス・プロセスを使用する際は、補正サービスが使用可能になっている必要があります。クラスターでビジネス・プロセスを実行する場合、クラスターの各サーバーで補正サービスを使用可能にする必要があります。

リカバリー・ログ・ディレクトリー

補正サービスがリカバリー用のログ・ファイルを保管する、このサーバーでのディレクトリー名を指定します。補正を使用すると、WebSphere 製品が、補正に必要な情報を保管します。

リカバリー・ログ・ファイル・サイズ

このアプリケーション・サーバーでの補正ログ・ファイルの最大サイズを MB 単位で指定します。

4. オプション: 必要に応じて、補正サービス・プロパティを変更します。

5. 「OK」をクリックします。
6. 構成を保管するには、管理コンソール・ウィンドウのメッセージ・ボックスで「保管」をクリックします。次に「アプリケーション・サーバーの保管 (Application Servers Save)」ペインの「保管」をクリックします。

管理コンソールを使用した、失敗したメッセージの照会と再生

ここでは、処理できなかったビジネス・プロセスまたはヒューマン・タスクのメッセージの有無を確認し、メッセージが存在する場合には応答する方法について説明します。

メッセージの処理中に問題が発生すると、そのメッセージは保存キューまたは保留キューに移されます。このタスクでは、失敗したメッセージが存在するかどうかを判別する方法と、それらのメッセージを内部キューへ再び送信する方法を説明します。

1. 保留キューおよび保存キューにメッセージがいくつかあるかを確認します。
 - a. 「サーバー」 → 「アプリケーション・サーバー」 → `server_name` をクリックします。
 - b. 「構成」タブの「コンテナの設定」セクションで、以下の手順の 1 つをクリックします。
 - ビジネス・プロセスの場合: 「ビジネス・プロセス・コンテナの設定」 → 「ランタイム構成」
 - ヒューマン・タスクの場合: 「ヒューマン・タスク・コンテナの設定」 → 「ランタイム構成」

保留キューおよび保存キュー上のメッセージの数が、「一般プロパティ」に表示されます。

2. 保留キューまたは保存キューにメッセージが含まれている場合は、内部作業キューへメッセージを移動できます。

以下のいずれかのオプションをクリックします。

- ビジネス・プロセスの場合: 「保留キューの再生」および「保存キューの再生」
- ヒューマン・タスクの場合: 「保留キューの再生」

注: セキュリティーが使用可能になっている場合は、オペレーター権限を持つユーザーに対してのみ再生ボタンが表示されます。

Business Process Choreographer は、再生された全メッセージを再び保守しようとします。

失敗したメッセージ数の最新表示:

管理コンソールを使用して、ビジネス・プロセスまたはヒューマン・タスクの失敗したメッセージの数を最新表示します。

保留キューおよび保存キュー上に表示されたメッセージの数、およびメッセージ例外の数は、最新表示されるまで静的なままになっています。このタスクでは、これらのキュー上のメッセージ数、およびメッセージ例外の数の更新および表示方法について説明します。

1. 該当するアプリケーション・サーバーを選択します。

「サーバー」 → 「アプリケーション・サーバー」 → *server_name* をクリックします。

2. メッセージ数を最新表示します。

「構成」タブの「コンテナの設定」セクションで、以下の手順の 1 つをクリックします。

- ビジネス・プロセスの場合: 「ビジネス・プロセス・コンテナの設定」 → 「ランタイム構成」 → 「メッセージ・カウントの更新」
- ヒューマン・タスクの場合: 「ヒューマン・タスク・コンテナの設定」 → 「ランタイム構成」 → 「メッセージ・カウントの更新」

以下の更新済み値が、「一般プロパティ」に表示されます。

- ビジネス・プロセスの場合: 保留キューおよび保存キュー上のメッセージの数
- ヒューマン・タスクの場合: 保留キュー上のメッセージの数
- キューへのアクセス中に何らかの例外が発生すると、「メッセージ例外」フィールドにメッセージ・テキストが表示されます。

メッセージ処理の失敗と静止モード:

Business Process Choreographer は、一時的なインフラストラクチャー障害を処理するための機能を提供します。

このセクションでは、失敗したメッセージをビジネス・プロセス・コンテナによって処理する仕組みを説明します。この仕組みは、42 ページの『ヒューマン・タスクでの失敗したメッセージ処理』で説明するヒューマン・タスク・コンテナによって使用される簡略化された仕組みとは対照的です。

長期実行プロセスは、一連のトランザクションで構成されています。トランザクションは、Java Message Service (JMS) メッセージによって分離されます。サーバーは、メッセージ駆動型 Bean に送信します。この Bean の処理は、着信メッセージをプロセス・サーバーに受け渡すことによって行います。それぞれのトランザクションは、以下のアクションで構成されています。

- メッセージを受信します。
- メッセージを代行してナビゲートします。
- 後続のトランザクションを起動するメッセージを送信します。

サーバーは、以下の理由のいずれかで、メッセージ駆動型 Bean が受信したメッセージの処理に失敗することがあります。

- 指定された数の連続メッセージは処理されませんでした。この場合、インフラストラクチャーが使用不可になっていると考えられます。
- 一部のメッセージしか処理できません。処理できなかったすべての単一メッセージは、破損していると考えられます。

これらの原因に対する対応は、以下のとおりです。

原因	応答
インフラストラクチャーが使用不可	メッセージ駆動型 Bean は、指定した期間、この状態からのリカバリーを試行します。この Bean は、サーバーが再び作動可能になるまで、すべてのメッセージを使用可能な状態に保持するように試行します。例えばこの問題は、データベース障害によって発生している場合があります。
メッセージの破損	指定した回数の試行後、メッセージは保留キューに書き込まれます。この保留キューで、メッセージの操作および検討を行います。保留キューから入力キューに戻して、トランザクションを再試行することもできます。

ビジネス・プロセス用のメッセージのインプリメンテーションは以下のとおりです。

- メッセージの処理が失敗した場合は、サーバーはそのメッセージを保存キューに書き込みます。この保存キューで、使用可能な状態が保持されます。これは、指定した時間内でインフラストラクチャーの問題が解決される場合です。
- メッセージが保存キューにある場合、オプションは以下のとおりです。
 - 後続のメッセージが正常に処理できる場合、保存キューからのすべてのメッセージは、メッセージ駆動型 Bean の入力キューに戻されます。それぞれのメッセージごとに、メッセージが保存キューに送信された頻度数が保持されます。この数が所定のメッセージの再試行限度を超えた場合、メッセージは保留キューに書き込まれます。
 - 次のメッセージの処理に失敗した場合、そのメッセージも保存キューに書き込まれます。この処理は、保存キューの最大メッセージしきい値に到達するまで継続します。このしきい値に到達すると、メッセージ駆動型 Bean は、保存キューからすべてのメッセージを入力キューに戻し、静止モードにスイッチします。

メッセージ駆動型 Bean が静止モードで作動している場合、この Bean は、定期的にメッセージの処理を試行します。処理に失敗したメッセージは、配信回数を増加させたり、保存キューの巡回数を増加させたりすることなく、入力キューに書き戻されます。メッセージの処理を正常に行うことが可能になったら、即時にメッセージ駆動型 Bean は標準処理モードにスイッチバックします。

この機能は、2 つの数値限度、2 つのキュー、静止モード、およびメッセージ再試行の動作から構成されています。

再試行限度

再試行限度は、保留キューに書き込む前に保存キューでメッセージを転送できる最大回数を定義しています。

保存キューに書き込まれるには、メッセージの処理が 3 回失敗する必要があります。

例えば、再試行限度が 5 である場合は、メッセージが保存キューを 5 回通過してから (3 * 5 = 15 回失敗してから) 最後の再試行ループが開始されます。最後の再

試行ループでさらに 2 回失敗すると、メッセージは保留キューに入れられます。つまり、メッセージは、 $(3 * \text{RetryLimit}) + 2$ 回失敗してから保留キューに入れられません。

信頼できるインフラストラクチャーで実行中の、パフォーマンスが重要なアプリケーションでは、再試行限度を少なく、例えば 1 または 2 にしておく必要があります。このパラメーターは、「ビジネス・プロセス・コンテナ」構成ページの管理コンソールにあります。

保存キュー・メッセージ限度

保存キュー・メッセージ限度は、保存キューに入れることができるメッセージの最大数を定義します。保存キューがオーバーフローすると、システムは静止モードに入ります。1 つのメッセージが失敗したら即時にシステムが静止モードに入るようにするには、値をゼロに設定します。ビジネス・プロセス・コンテナのインフラストラクチャー障害に対する耐性を強化するには、値を増やします。

このパラメーターは、「ビジネス・プロセス・コンテナ」ページの管理コンソールにあります。(このパラメーターを見つけるには、「サーバー」→「アプリケーション・サーバー」→ *server_name* をクリックします。次に、「ビジネス・プロセス・コンテナ設定 (Business Process Container Settings)」見出しで、「ビジネス・プロセス・コンテナ」をクリックします。「再試行限度」フィールドは、「一般プロパティ」見出しにあります。)

保存キュー

保存キューは、障害を起こしたメッセージを保存します。これらは、ビジネス・プロセス・コンテナの内部作業キューへ戻ることによって再生されます。メッセージは、3 回失敗したら保存キューに書き込まれます。メッセージが $(3 * \text{RetryLimit}) + 2$ 回失敗した場合は、保留キューに書き込まれます。(再試行限度について詳しくは、40 ページの『再試行限度』を参照してください。) 保存キューが保存キュー・メッセージ限度で定義された限度に達してから別のメッセージが失敗すると、キューはオーバーフローし、システムは静止モードに入ります。管理者は、失敗したメッセージの照会と再生のタスクを実行して、このキュー内のメッセージを内部キューに戻すことができます。

保留キュー

保留キューには、 $(3 * \text{RetryLimit}) + 2$ 回失敗したメッセージが含まれます。(再試行限度について詳しくは、40 ページの『再試行限度』を参照してください。) 管理者は、失敗したメッセージの照会と再生のタスクを実行して、このキュー内のメッセージを内部キューに戻すことができます。

メッセージの再生

管理者は、保留キューまたは保存キューから内部キューへメッセージを戻すことができます。この操作は、管理コンソールか管理コマンドを使用して実行できます。

静止モード

保存キューがオーバーフローすると、静止モードに入ります。これが起こると、場合によっては一時的だが重大なインフラストラクチャー障害が発生したと想定されます。静止モードの目的は、システムが多数のリソースを使用しないようにすることですが、インフラストラクチャー障害は、ほとんどのメッセージがいずれにせよ失敗する可能性があることを意味します。静止モードでは、次のメッセージの処理を試行する前に、2 秒間のシステム・スリープがあります。メッセージが正常に処理されると即時に、システムは標準メッセージ処理を再開します。

ヒューマン・タスクでの失敗したメッセージ処理

ヒューマン・タスク・コンテナには、保存キューや再試行限度がありません。保留キューがあるだけで、失敗したメッセージはこのキューに置かれ、このキューから再生することができます。

管理コンソールを使用したスタッフ照会の最新表示

スタッフ照会の結果は静的です。管理コンソールを使用して、スタッフ照会を最新表示します。

Business Process Choreographer は、ランタイム・データベースで、Lightweight Directory Access Protocol (LDAP) サーバーなど、スタッフ・ディレクトリーに対して評価されたスタッフ割り当ての結果をキャッシュに入れます。スタッフ・ディレクトリーを変更する場合は、スタッフ割り当てを再評価するよう強制できます。

スタッフ照会を最新表示するには、以下の手順を実行します。

1. 「サーバー」 → 「アプリケーション・サーバー」 → *server_name* をクリックします。
2. 「構成」タブの「コンテナの設定」セクションで、「ヒューマン・タスク・コンテナの設定」 → 「ランタイム構成」 → 「スタッフ照会の更新」をクリックします。

注: セキュリティーが使用可能になっている場合は、オペレーター権限を持つユーザーに対してのみ更新ボタンが表示されます。

すべてのスタッフ照会は、更新されます。

Common Base Event および監査証跡の使用可能化

このタスクを使用して、Business Process Choreographer イベントを Common Event Infrastructure に Common Base Event として放出するか、監査証跡に保管できるようにします。

ビジネス・プロセス・コンテナまたはヒューマン・タスク・コンテナの状態監視者設定は、「構成」タブで永続的に、または「ランタイム」タブで一時的に変更することができます。これらの「構成」タブまたは「ランタイム」タブでの選択は、該当するコンテナで実行されるすべてのアプリケーションに影響します。変更内容をビジネス・プロセス・コンテナとヒューマン・タスク・コンテナの両方に反映させるには、それぞれの設定を別々に変更する必要があります。

構成済みロギング・インフラストラクチャーの変更:

このタスクを使用して、監査ログの状態監視者ロギング、または構成の Common Event Infrastructure ロギングを変更します。

「構成」タブで行われた選択は、次にサーバーを開始したときにアクティブになります。選択した設定は、サーバーを開始するたびに適用されます。

以下のように構成を変更します。

1. 「ビジネス・プロセス・コンテナ」または「ヒューマン・タスク・コンテナ」ペインを表示します。

「サーバー」 → 「アプリケーション・サーバー」 → *server_name* をクリックします。次に、「コンテナ設定」で、以下のいずれかの手順をクリックします。

- ビジネス・プロセスの場合: 「ビジネス・プロセス・コンテナの設定」 → 「ビジネス・プロセス・コンテナ」
 - ヒューマン・タスクの場合: 「ヒューマン・タスク・コンテナの設定」 → 「ヒューマン・タスク・コンテナ」
2. 「一般プロパティ」セクションで、インプリメントするロギングを選択します。状態監視者は、それぞれ独立しています。どちらかいずれか、またはその両方を使用可能にしたり、使用不可にすることができます。

Common Event Infrastructure ロギングの使用可能化

このチェック・ボックスを選択して、Common Event Infrastructure に基づくイベント放出を使用可能にします。

監査ロギングの使用可能化

このチェック・ボックスを選択して、リレーションシップ・データベースの監査証跡テーブルに、監査ログ・イベントを保管します。

3. 変更を受け入れます。
 - a. 「適用」をクリックします。
 - b. メッセージ・ボックスで「保管」をクリックします。
 - c. 「アプリケーション・サーバー」ペインで「保管」をクリックします。

状態監視者は、必要に応じて設定します。変更は、サーバーの再始動後に有効になります。

変更を有効にするには、コンテナを再始動します。

セッション用ロギング・インフラストラクチャーの構成:

このタスクを使用して、監査ログの状態監視者ロギング、またはセッションの Common Event Infrastructure ロギングを変更します。

「ランタイム」タブで行った選択は、即時に有効になります。選択した設定は、次にサーバーを開始するまで有効です。

以下のように、セッション・インフラストラクチャーを変更します。

1. 「メッセージの再生」ペインを表示します。

「サーバー」 → 「アプリケーション・サーバー」 → *server_name* をクリックします。次に、「コンテナ設定」で、以下のいずれかの手順をクリックします。

- ビジネス・プロセスの場合: 「ビジネス・プロセス・コンテナの設定」 → 「ランタイム構成」
 - ヒューマン・タスクの場合: 「ヒューマン・タスク・コンテナの設定」 → 「ランタイム構成」
2. 「状態監視者ロギング」フィールドで、インプリメントされるロギングを選択します。状態監視者は、それぞれ独立しています。どちらかいずれか、またはその両方を使用可能にしたり、使用不可にすることができます。

Common Event Infrastructure ロギングの使用可能化

このチェック・ボックスを選択して、Common Event Infrastructure に基づくイベント放出を使用可能にします。

監査ロギングの使用可能化

このチェック・ボックスを選択して、リレーションシップ・データベースの監査証跡テーブルに、監査ログ・イベントを保管します。

3. 変更を受け入れます。

「状態監視者の更新」をクリックします。

状態監視者は、必要に応じて設定します。

イベントの放出および保管:

状態変更のイベントは、ビジネス・プロセス、ヒューマン・タスク、またはその両方を実行する場合に生成される可能性があります。

イベントがアプリケーションによって検索されるように、2 つのインフラストラクチャーは、イベントの放出または保管を行います。アプリケーションでは、イベントを使用して、ビジネス・プロセスをモニターし、ビジネス・プロセスまたはヒューマン・タスクのヒストリーを分析することができます。

例えばタスク・イベントは、ビジネス・プロセスが関係しなくても発生する可能性があります。これらのイベントは、監査証跡および Common Event Infrastructure (CEI) によってコンシュームされることもあります。これは、スタンドアロン・タスク、全くのヒューマン・タスク、およびビジネス・プロセス以外のアプリケーション・コンポーネントによって呼び出されるタスクに当てはまります。

イベントの生成はシステム・パフォーマンスに影響を与えるため、イベントの保管および放出を行うインフラストラクチャーを選択できます。

Common Event Infrastructure

イベントは、サブスクリプト・アプリケーションに保管および公表することができます。このイベント・インフラストラクチャーを使用するために、Common Event Infrastructure がインストールされ、構成されていることを確認してください。

Common Event Infrastructure に基づくイベントの放出を使用し、Common Event Infrastructure 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) を経由して、Common Base Event 形式でイベントを検索します。サブスクリプション、または Common Event Infrastructure の照会指向インターフェースを使用して、利用中のアプリケーションに接続できます。

Common Event Infrastructure に基づくイベントの放出は、監査ログ・イベントよりも、システム・パフォーマンスに重大な影響を与えます。ただし、利用中のアプリケーションには、非常に大きな柔軟性が付与されます。

監査証跡

イベントは、リレーションシップ・データベース内にあるテーブルのレコードとして保管されます。

これは、パフォーマンスにほとんど影響を与えない高速イベント保管インフラストラクチャーです。利用中のアプリケーションがデータベースからイベントを検索するには、Structured Query Language (SQL) 照会が必要です。

インフラストラクチャーのいずれか、あるいは両方を選択することも、またはどちらも選択しないこともできます。インフラストラクチャーの選択では、イベントの保管または放出は必ずしも暗黙指定されません。選択によって、インフラストラクチャーが使用可能になります。後でメカニズムを追加することによって、実際のイベントの生成を制御できます。ただし、インフラストラクチャーを使用可能にすると、システム・パフォーマンスに影響する基本的なオーバーヘッドが発生します。

スクリプトによる Business Process Choreographer の管理

スクリプトを使用して実行可能な管理操作について説明します。

管理コマンドを使用した、監査ログ・エントリーの削除

管理コマンドを使用して、監査ログ・エントリーの一部またはすべてを削除します。

この手順を始める前に、次の条件が満たされている必要があります。

- 使用しているユーザー ID に管理権限がある必要があります。
- 監査ログ・エントリーの削除に使用するアプリケーション・サーバーが稼働している必要があります。つまり、サーバー接続が必要であるため、wsadmin の `-conntype none` オプションは使用できません。
- セキュリティーが使用可能に設定されている場合は、オペレーター権限を持っている必要があります。

`deleteAuditLog.jacl` スクリプトを使用すると、データベースから監査ログ・エントリーを削除できます。

1. スクリプトがある Business Process Choreographer ユーティリティー・ディレクトリーに移動します。

以下のコマンドを入力します。

```
cd install_root/ProcessChoreographer/util
```

2. 監査ログ・テーブルでエントリーを削除します。

次のコマンドを 1 つ以上入力します。コマンド間の相違を強調して表示しています。

```
install_root/bin/wsadmin -f deleteAuditLog.jacl  
                        -server serverName  
                        [-profile profileName]  
                        [options]
```



```
install_root/bin/wsadmin -f deleteAuditLog.jacl
                        -node nodeName
                        -server serverName
                        [-profile profileName]
                        [options]
```

```
install_root/bin/wsadmin -f deleteAuditLog.jacl
                        -cluster clusterName
                        [-profile profileName]
                        [options]
```

各部の意味は、次のとおりです。

cluster *clusterName*

クラスターの名前。ビジネス・プロセス・コンテナが WebSphere クラスター用に構成されている場合は必須です。

node *nodeName*

サーバー名を指定する場合のオプション。この名前は、ノードを示します。デフォルトはローカル・ノードです。

server *serverName*

サーバーの名前。クラスター名が指定されていない場合は必須です。

profileName *profileName*

ユーザー定義プロファイルの名前。デフォルト・プロファイルで作業していない場合、このオプションを指定します。

使用可能なオプションは以下のとおりです。

-all

データベース内の監査ログ・エントリーをすべて削除します。複数のトランザクションで削除が実行されます。各トランザクションでは、`slice` パラメーターで指定されたエントリー数またはデフォルト数が削除されます。

-time *timestamp*

timestamp に指定した時刻より古い監査ログ・エントリーすべてが削除されます。使用される時刻は、協定世界時 (UTC) です。その時刻形式は、YYYY-MM-DD[*T*HH:MM:SS] となっている必要があります。年月日だけを指定する場合、時間と分と秒は 00:00:00 に設定されます。

`-time` および `-processtime` オプションは、相互に排他的です。

-processtime *timestamp*

timestamp で指定した時刻より前に完了したプロセスに属す監査ログ・エントリーがすべて削除されます。`-time` パラメーターの場合と同じ時刻形式を使用します。

`-time` および `-processtime` オプションは、相互に排他的です。

-slice *size*

`-all` パラメーターとともに使用され、*size* は各トランザクションに含まれるエントリー数を指定します。最適値は、データベース・システムで使用可能なログ・サイズによって決まります。値が高いほど、必要なトランザクションは少なくなりますが、データベース・ロギング・スペースを超過する可能性があります。値が低い場合は、スクリプトでの削除の完了に時間がかかる可能性があります。`slice` パラメーターのデフォルトのサイズは 250 です。

無効になったプロセス・テンプレートおよびタスク・テンプレートの削除

管理コマンドを使用して、データベースから、無効になったプロセス・テンプレートか無効になったタスク・テンプレートのいずれか、またはその両方を削除します。

この手順を始める前に、削除されるテンプレートが置かれているアプリケーション・サーバーが稼働している必要があります。つまり、サーバー接続が必要であるため、`wsadmin` の `-conntype none` オプションは使用できません。セキュリティーが使用可能になっている場合でも、このコマンドを実行するための特殊権限は不要です。

ここで説明するメソッドを使用して、データベースから、テンプレート、およびそのテンプレートに属するすべてのオブジェクト (WebSphere 構成リポジトリ内に、これらのオブジェクトを含んでいる有効な対応するアプリケーションがない場合) を除去します。この状態は、ユーザーによってアプリケーションのインストールが取り消されたか、Configuration Repository に保管されていなかった場合に発生します。これらのテンプレートは、通常影響はありません。これらのテンプレートは、Business Process Choreographer Explorer では表示されません。

これらのテンプレートがフィルタリングされない状況がまれに発生します。この場合には、以下のスクリプトでデータベースから除去される必要があります。

スクリプトを使用して、データベースから有効なアプリケーションのテンプレートを除去することはできません。対応するアプリケーションが有効な場合、この状態は検査され、ConfigurationError 例外がスローされます。

- スクリプトがある Business Process Choreographer ユーティリティー・ディレクトリに移動します。

以下のコマンドを入力します。

```
cd install_root/ProcessChoreographer/util
```

- データベースから、無効になったビジネス・プロセス・テンプレート、またはヒューマン・タスク・テンプレートを削除します。

無効になったビジネス・プロセス・テンプレートを削除するには、次のコマンドのいずれかを入力します。コマンド間の相違を強調して表示しています。

```
install_root/bin/wsadmin.sh -f deleteInvalidProcessTemplate.jacl  
-server serverName  
-template templateName  
-validFrom validFromString  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -f deleteInvalidProcessTemplate.jacl  
-server serverName  
-node nodeName  
-template templateName  
-validFrom validFromString  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -f deleteInvalidProcessTemplate.jacl  
-cluster clusterName
```

```
-template templateName  
-validFrom validFromString  
[-profileName profileName]
```

無効になったヒューマン・タスク・テンプレートを削除するには、次のコマンドのいずれかを入力します。コマンド間の相違を強調して表示しています。

```
install_root/bin/wsadmin.sh -f deleteInvalidTaskTemplate.jacl  
-server serverName  
-template templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -f deleteInvalidTaskTemplate.jacl  
-server serverName  
-node nodeName  
-template templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

```
install_root/bin/wsadmin.sh -f deleteInvalidTaskTemplate.jacl  
-cluster clusterName  
-template templateName  
-validFrom validFromString  
-nameSpace nameSpace  
[-profileName profileName]
```

各部の意味は、次のとおりです。

cluster *clusterName*

クラスターの名前。ビジネス・プロセス・コンテナが WebSphere クラスター用に構成されている場合は必須です。クラスター名、またはサーバー名およびノード名を指定できます。

node *nodeName*

サーバー名を指定する場合のオプション。この名前は、ノードを示します。デフォルトはローカル・ノードです。サーバー名およびノード名、またはクラスター名を指定できます。

server *serverName*

サーバーの名前。クラスター名が指定されていない場合は必須です。サーバー名およびノード名、またはクラスター名を指定できます。

template *templateName*

除去するプロセス・テンプレートまたはタスク・テンプレートの名前。

validFrom *validFromString*

管理コンソールの表示どおりにテンプレートが有効になった日付 (UTC 形式)。ストリングの形式は、「yyyy-MM-ddThh:mm:ss」(年、月、日、T、時、分、秒) です。例えば、2005-01-31T13:40:50 のようになります。

nameSpace *nameSpace*

タスク・テンプレートのターゲット・ネーム・スペース。

profileName *profileName*

ユーザー定義プロファイルの名前。デフォルト・プロファイルで作業していない場合、このオプションを指定します。

管理コマンドを使用した、失敗したメッセージの照会と再生

管理コマンドを使用して、ビジネス・プロセスまたはヒューマン・タスクの失敗したメッセージが存在するかどうかを判別し、失敗したメッセージが存在する場合は、そのメッセージの処理を再試行します。

この手順を始める前に、次の条件が満たされている必要があります。

- 使用しているユーザー ID に管理権限がある必要があります。
- メッセージが照会または再生されるアプリケーション・サーバーが稼働している必要があります。つまり、サーバー接続が必要であるため、wsadmin スクリプトの `-conntype none` オプションは使用できません。
- セキュリティーが使用可能に設定されている場合は、オペレーター権限を持っている必要があります。

内部メッセージの処理中に問題が発生すると、このメッセージが最終的に保存キューまたは保留キューに入れられます。失敗したメッセージが存在するかどうかを判別する方法、およびそれらのメッセージを内部キューへ再び送信する方法は、以下のとおりです。

1. スクリプトがある Business Process Choreographer ユーティリティー・ディレクトリに移動します。以下を入力します。

```
cd install_root/ProcessChoreographer/util
```

2. 保存キューと保留キューの両方で、失敗したメッセージの数を照会します。以下のいずれかのコマンドを入力します。コマンド間の相違を強調して表示しています。

```
install_root/bin/wsadmin -f queryNumberOfFailedMessages.jacl  
-cluster clusterName  
[ -bfm | -htm ]  
[-profile profileName]
```

```
install_root/bin/wsadmin -f queryNumberOfFailedMessages.jacl  
-node nodeName  
-server serverName  
[ -bfm | -htm ]  
[-profile profileName]
```

各部の意味は、次のとおりです。

cluster *clusterName*

クラスターの名前。ビジネス・プロセス・コンテナが WebSphere クラスター用に構成されている場合は必須です。

node *nodeName*

サーバー名を指定する場合のオプション。この名前は、ノードを示します。デフォルトはローカル・ノードです。

server *serverName*

サーバーの名前。クラスター名が指定されていない場合は必須です。

bfm|htm

これらのキーワードはオプションです。デフォルトでは、どちらのオプションも指定されていないければ、ビジネス・プロセスとヒューマン・タスク両方の失敗したメッセージすべてが表示されます。ビジネス・プロセス・コンテナの保留キューと保存キューにあるメッセージ数のみを表示する場合は、

bfm を指定します。ヒューマン・タスク・コンテナの保留キューにあるメッセージ数のみを表示する場合は、htm を指定します。

profile *profileName*

ユーザー定義プロファイルの名前。デフォルト・プロファイルで作業していない場合、このオプションを指定します。

3. 保留キュー、保存キュー、または両方のキューにある失敗したメッセージをすべて再生します。以下のいずれかのコマンドを入力します。

```
install_root/bin/wsadmin -f replayFailedMessages.jacl -cluster clusterName
                        -queue replayQueue profile profileName
install_root/bin/wsadmin -f replayFailedMessages.jacl -node nodeName
                        -server serverName -queue replayQueue profile profileName
install_root/bin/wsadmin -f replayFailedMessages.jacl -server serverName
                        -queue replayQueue profile profileName
```

各部の意味は、次のとおりです。

queue *replayQueue*

以下のいずれか 1 つの値を持っている必要があります。

holdQueue
retentionQueue
両方

cluster *clusterName*

クラスターの名前。ビジネス・プロセス・コンテナが WebSphere クラスター用に構成されている場合は必須です。

node *nodeName*

サーバー名を指定する場合のオプション。この名前は、ノードを示します。デフォルトはローカル・ノードです。

server *serverName*

サーバーの名前。クラスター名が指定されていない場合は必須です。

bfm|htm

これらのキーワードはオプションであり、相互に排他的です。デフォルトでは、どちらのオプションも指定されていなければ、ビジネス・プロセスとヒューマン・タスク両方の失敗したメッセージが再生されます。ビジネス・プロセスのメッセージのみを再生する場合は、bfm を指定します。ヒューマン・タスクのメッセージのみを再生する場合は、htm を指定します。

profile *profileName*

ユーザー定義プロファイルの名前。デフォルト・プロファイルで作業していない場合、このオプションを指定します。

管理コマンドを使用した、スタッフ照会の最新表示

スタッフ照会の結果は静的です。管理コマンドを使用して、スタッフ照会を最新表示します。

この手順を始める前に、次の条件が満たされている必要があります。

- 使用しているユーザー ID に管理権限がある必要があります。

- メッセージが照会または再生されるアプリケーション・サーバーが稼働している必要があります。つまり、サーバー接続が必要であるため、wsadmin の `-conntype none` オプションは使用できません。
- セキュリティーが使用可能に設定されている場合は、オペレーター権限を持っている必要があります。

Business Process Choreographer は、ランタイム・データベースで、Lightweight Directory Access Protocol (LDAP) サーバーなど、スタッフ・ディレクトリーに対して評価されたスタッフ割り当ての結果をキャッシュに入れます。スタッフ・ディレクトリーを変更する場合は、スタッフ割り当てを再評価するよう強制できます。

1. スクリプトがある Business Process Choreographer ユーティリティー・ディレクトリーに移動します。

以下のコマンドを入力します。

```
cd install_root/ProcessChoreographer/util
```

2. スタッフ割り当てを再評価するよう強制します。

以下のいずれかのコマンドを入力します。コマンド間の相違を強調して表示しています。

```
install_root/bin/wsadmin -f refreshStaffQuery.jacl
  -server serverName
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userList username{,username}...]
  [-profile profileName]
```

```
install_root/bin/wsadmin -f refreshStaffQuery.jacl
  -node nodeName
  -server serverName
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userList username{,username}...]
  [-profile profileName]
```

```
install_root/bin/wsadmin -f refreshStaffQuery.jacl
  -cluster clusterName
  [-processTemplate templateName |
  (-taskTemplate templateName [-nameSpace nameSpace]) |
  -userList username{,username}...]
  [-profile profileName]
```

各部の意味は、次のとおりです。

cluster *clusterName*

クラスターの名前。ビジネス・プロセス・コンテナが WebSphere クラスター用に構成されている場合は必須です。

node *nodeName*

サーバー名を指定する場合のオプション。この名前は、ノードを示します。デフォルトはローカル・ノードです。

server *serverName*

サーバーの名前。クラスター名が指定されていない場合は必須です。

processTemplate *templateName*

プロセス・テンプレートの名前。このプロセス・テンプレートに属すスタッフ割り当てが更新されます。

taskTemplate *templateName*

タスク・テンプレートの名前。このプロセス・テンプレートに属すスタッフ割り当てが更新されます。

nameSpace *nameSpace*

タスク・テンプレートのネーム・スペース。

userList *userName*

コンマで区切られたユーザー名のリスト。指定された名前を含むスタッフ割り当てが更新されます。

profileName *profileName*

ユーザー定義プロファイルの名前。デフォルト・プロファイルで作業していない場合、このオプションを指定します。

注: *templateName* も *userList* も指定しない場合は、データベースに保管されているすべてのスタッフ照会が更新されます。パフォーマンス上の理由により、この処理は避けた方がよいでしょう。

スタッフ照会を最新表示するときの間隔の設定:

データベースがキャッシュ済みのスタッフ照会を最新表示するときの間隔を設定します。

スタッフ照会は、指定されたスタッフ・リポジトリによって解決されます。結果は Business Process Choreographer データベースに保管されます。スタッフ照会の解決性能を最適化するため、取得された照会の結果はキャッシュされます。キャッシュの内容の現行性は、新規プロセス・インスタンスの作成時や、対応するスタッフ・アクティビティーのスケジュールが行われるときにチェックされます。デフォルトで、共用スタッフ照会結果の有効期限は 1 時間です。スタッフ・リポジトリが頻繁に変化する場合は、このプロパティーに大きな値を使用し、要求時に `refreshStaffQuery.jacl` スクリプトを使用してスタッフ照会を最新表示することを考慮してください。

スタッフ照会結果の有効期限のデフォルト値は、管理コンソールで変更できます。

1. ヒューマン・タスク・コンテナのカスタム・プロパティー・ページに移動します。

「サーバー」 → 「アプリケーション・サーバー」 → *Server_Name* → 「ヒューマン・タスク・コンテナ」 → 「カスタム・プロパティー」をクリックします。

2. **StaffQueryResultValidTimeSeconds** を選択し、秒単位で新しい値を入力します。
3. 「OK」をクリックします。
4. 変更を保管し、アプリケーション・サーバーを再起動して変更を有効にします。

新規の有効期限の値は、新規のスタッフ照会にのみ適用され、既存のスタッフ照会には適用されません。

管理コマンドを使用した、未使用のスタッフ照会の除去

管理コマンドを使用して、データベースから未使用のスタッフ照会を削除します。

この手順を始める前に、次の条件が満たされている必要があります。

- 使用しているユーザー ID に管理権限がある必要があります。
- 未使用のスタッフ照会の削除に使用するアプリケーション・サーバーが稼働している必要があります。つまり、サーバー接続が必要であるため、wsadmin の `-conntype none` オプションは使用できません。
- セキュリティーが使用可能に設定されている場合は、オペレーター権限を持っている必要があります。

Business Process Choreographer は、評価されたスタッフ式のランタイム・データベースでユーザー名の名前を維持します。これらのスタッフ式を使用したプロセス・インスタンスは完了していますが、ユーザー名のリストは、対応するビジネス・プロセス・アプリケーションがアンインストールされるまでデータベース内に維持されます。

データベースのサイズがパフォーマンスに影響する場合は、データベース表にキャッシュされた未使用のスタッフ・リストを削除できます。

1. スクリプトがある Business Process Choreographer ユーティリティー・ディレクトリーに移動します。

以下のコマンドを入力します。

```
cd install_root/ProcessChoreographer/util
```

2. 未使用のスタッフ・リストを除去します。

以下のいずれかのコマンドを入力します。コマンド間の相違を強調して表示しています。

```
install_root/bin/wsadmin -f cleanupUnusedStaffQueryInstances.jacl
                        -server serverName
                        [-profile profileName]
```

```
install_root/bin/wsadmin -f cleanupUnusedStaffQueryInstances.jacl
                        -node nodeName
                        -server serverName
                        [-profile profileName]
```

```
install_root/bin/wsadmin -f cleanupUnusedStaffQueryInstances.jacl
                        -cluster clusterName
                        [-profile profileName]
```

各部の意味は、次のとおりです。

cluster *clusterName*

クラスターの名前。ビジネス・プロセス・コンテナが WebSphere クラスター用に構成されている場合は必須です。

node *nodeName*

サーバー名を指定する場合のオプション。この名前は、ノードを示します。デフォルトはローカル・ノードです。

server *serverName*

サーバーの名前。クラスター名が指定されていない場合は必須です。

profileName *profileName*

ユーザー定義プロファイルの名前。デフォルト・プロファイルで作業していない場合、このオプションを指定します。

データベースから削除されるエントリーの数が表示されます。

アプリケーションおよびアプリケーション・サービスの管理

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

ビジネス・プロセスおよびヒューマン・タスクの管理

ビジネス・プロセスおよびヒューマン・タスクは、エンタープライズ・アプリケーションの一部として配置およびインストールされます。管理コンソールまたは管理コマンドを使用してプロセス・テンプレートおよびタスク・テンプレートを管理し、Business Process Choreographer Explorer を使用してプロセス・インスタンスおよびタスク・インスタンスを操作できます。

プロセス・テンプレートおよびプロセス・インスタンスの管理

管理コンソールおよび管理コマンドを使用して、プロセス・テンプレートを管理します。プロセス・インスタンスの処理には Business Process Choreographer Explorer を使用します。

プロセス・テンプレートは、エンタープライズ・アプリケーション内でビジネス・プロセスを定義します。プロセス・テンプレートを含むエンタープライズ・アプリケーションがインストールされ、配置され、開始されると、プロセス・テンプレートが開始状態になります。管理コンソールまたは管理コマンドを使用して、プロセス・テンプレートの停止および開始を行います。

プロセス・テンプレートが Business Process Choreographer Explorer に表示されません。プロセス・インスタンスは、長期実行プロセスの場合と、Microflow の場合があります。Business Process Choreographer Explorer を使用すると、プロセス・テンプレートやプロセス・インスタンスに関する情報を表示したり、プロセス・インスタンスに対してアクションを実行したりできます。例えば、アクションにはプロセス・インスタンスの開始、および、長期実行プロセスの場合、プロセス・インスタンスの中断や再開、あるいは終了などその他のプロセス・ライフ・サイクル・アクション、アクティビティーの修復などがあります。

ビジネス・プロセスのための許可のロール:

ビジネス・プロセスで実行できるアクションは、ご使用の許可のロールに依存します。このロールは、J2EE のロールまたはインスタンス・ベースのロールとすることができます。

ロールとは、同じ権限レベルを共有する従業員のグループです。Java 2 Platform, Enterprise Edition (J2EE) のロールは、ビジネス・プロセス・コンテナが構成されたときにセットアップされます。インスタンス・ベースのロールは、プロセスがモデル化されたときにプロセスおよびアクティビティーに割り当てられます。ロール・ベースの許可については、グローバル・セキュリティが WebSphere Application Server で使用可能になっていることが必要です。

J2EE のロール

以下の J2EE のロールがサポートされています。

- J2EE BPESystemAdministrator。このロールを割り当てられたユーザーは、すべての特権を持ちます。
- J2EE BPESystemMonitor。このロールを割り当てられたユーザーは、すべてのビジネス・プロセス・オブジェクトのプロパティを表示できます。

これらのロールへのユーザーおよびグループの割り当てを変更するには、管理コンソールを使用できます。

RACF セキュリティーによるロールの設定: この RACF 許可は、以下のセキュリティー・フィールドを指定した場合に適用されます。

- **com.ibm.security.SAF.authorization= true**
RDEFINE EJBROLE BPESystemAdministrator UACC(NONE)
PERMIT BPESystemAdministrator CLASS(EJBROLE) ID(userid) ACCESS(READ)

RDEFINE EJBROLE BPESystemMonitor UACC(NONE)
PERMIT BPESystemMonitor CLASS(EJBROLE) ID(userid) ACCESS(READ)
- **com.ibm.security.SAF.delegation= true**
RDEFINE EJBROLE JMSAPIUser UACC(NONE) APPLDATA('userid')

Security Authorization Facility (SAF) ベースの許可 (RACF EJBROLE プロファイルの使用など) を使用して、ビジネス・プロセス・コンテナなど、EJB およびエンタープライズ・アプリケーションにおける Java 2 Platform、Enterprise Edition (J2EE) ロールに対するクライアントのアクセスを制御できます。SAF の使用についての詳細は、WebSphere Application Server for z/OS インフォメーション・センターの『役割ベースの許可の System Authorization Facility』を参照してください。

インスタンス・ベースのロール

プロセス・インスタンスまたはアクティビティーはプロセス・モデルではスタッフ・メンバーには直接割り当てられておらず、その代わりに、使用可能なロールの 1 つに割り当てられます。インスタンス・ベースのロールに割り当てられたスタッフ・メンバーは、そのロールに応じたアクションを実行できます。インスタンス・ベースのロールとユーザーの関連付けは、スタッフの解決を使用して実行時に決定されます。

以下のインスタンス・ベースのロールがサポートされています。

- プロセスに対して: 読者、スターター、管理者
- アクティビティーに対して: 読者、編集者、可能な所有者、所有者、管理者

これらのロールは、以下のアクションの実行が許可されています。

ロール	許可されたアクション
アクティビティー・リーダー	関連したアクティビティー・インスタンスのプロパティおよび入出力メッセージを表示します。
アクティビティー編集者	アクティビティー・リーダーに許可されたアクションと、アクティビティーに関連したメッセージおよびその他のデータへの書き込みアクセス権限。

ロール	許可されたアクション
可能なアクティビティ所有者	アクティビティ・リーダーに許可されたアクション。このロールのメンバーはアクティビティを要求でき、receive アクティビティまたは pick アクティビティにメッセージを送信できます。
アクティビティ所有者	アクティビティを処理し、完了します。このロールのメンバーは、所有された作業項目を管理者または可能な所有者に転送できます。
アクティビティ管理者	予期しないエラーで停止したアクティビティを修理し、長期実行アクティビティを強制終了します。
プロセス・スターター	関連したプロセス・インスタンスのプロパティおよび入出力メッセージを表示します。
プロセス・リーダー	関連したプロセス・インスタンスのプロパティ、入出力メッセージ、および含まれるアクティビティすべてに対してアクティビティ・リーダーがサポートするものはみな表示しますが、サブプロセスのものについては対象外です。
プロセス管理者	このロールのメンバーは、プロセス・インスタンスを管理し、開始済みプロセスに介入して、作業項目の作成、削除、および転送ができます。このロールのメンバーには、アクティビティ管理者権限もあります。

プロセス・スターターのユーザー ID は、プロセス・インスタンスが存在している場合、ユーザー・レジストリーから削除しないでください。これを行うと、このプロセスのナビゲーションが継続できません。システム・ログ・ファイルから以下の例外を受け取ります。

no unique ID for: <user ID>

ビジネス・プロセス管理 - よくある質問:

ビジネス・プロセスの管理に関するよくある質問への一連の回答。

- 57 ページの『プロセス・テンプレートが開始済み状態で、それが属しているアプリケーションが停止状態である場合は、どうなりますか?』
- 57 ページの『プロセス・インスタンスの作成を停止するにはどうすればよいでしょうか?』
- 57 ページの『より新しいプロセス・テンプレートが有効になった場合、実行中のインスタンスはどうなりますか?』
- 57 ページの『作成に使用されたテンプレートが停止されると、実行中のインスタンスはどうなりますか?』
- 57 ページの『プロセス・インスタンスがまだ実行中であることは、どのようにして分かりますか?』
- 57 ページの『ビジネス・プロセス・アプリケーションにプロセス・インスタンスがある場合、アプリケーションを停止できないのはなぜですか?』

プロセス・テンプレートが開始済み状態で、それが属しているアプリケーションが停止状態である場合は、どうなりますか？

現在有効なプロセス・テンプレートが開始済み状態で、アプリケーションが停止状態である場合、新規プロセス・インスタンスはテンプレートから作成されません。既存プロセス・インスタンスは、アプリケーションが停止状態の場合にはナビゲートできません。

プロセス・インスタンスの作成を停止するにはどうすればよいでしょうか？

管理コンソールを使用して、プロセス・テンプレートを選択し、「停止」をクリックします。このアクションは、プロセス・テンプレートを停止状態にし、テンプレートからはこれ以上のインスタンスは作成されません。テンプレートの停止後は、テンプレートからプロセス・インスタンスを作成するすべての試行は、`EngineProcessModelStoppedException` エラーになります。

より新しいプロセス・テンプレートが有効になった場合、実行中のインスタンスはどうなりますか？

プロセス・テンプレートが無効な場合、この状態は、テンプレートからインスタンス化されたインスタンスの実行に影響を与えません。既存プロセス・インスタンスの実行は、完了するまで続行します。古いインスタンスと新規インスタンスは、古いインスタンスがすべて完了するか、強制終了されるまで、並行して実行します。

作成に使用されたテンプレートが停止されると、実行中のインスタンスはどうなりますか？

プロセス・テンプレートの状態を「stopped」に変更した場合、停止されるのは作成中の新規インスタンスだけです。既存のプロセス・インスタンスは、通常どおり、完了するまで実行を継続します。

プロセス・インスタンスがまだ実行中であることは、どのようにしてわかりますか？

Business Process Choreographer Explorer にプロセス管理者としてログオンし、「自分で管理するプロセス・インスタンス」ページに移動すると、実行中のプロセス・インスタンスがすべて表示されます。必要な場合は、これらのプロセス・インスタンスを強制終了して削除することができます。

ビジネス・プロセス・アプリケーションにプロセス・インスタンスがある場合、アプリケーションを停止できないのはなぜですか？

プロセス・インスタンスを実行させるには、対応するアプリケーションも稼働している必要があります。アプリケーションが停止している場合、プロセス・インスタンスのナビゲーションは継続できません。そのため、ビジネス・プロセス・アプリケーションは、プロセス・インスタンスがない場合にしか停止できません。

サブプロセスのライフ・サイクル管理とバージョン管理の振る舞い:

別のプロセスによって開始されたプロセスをサブプロセス といいます。サブプロセスのライフ・サイクルの管理が可能な方法と、サブプロセスのバージョン管理の振る舞いは、プロセスのモデル化方法によって異なります。

モジュール性と再利用性を高めるため、多くの場合、カプセル化のプログラミング概念をビジネス・プロセス・モデリングに適用することには意味があります。すなわち、ビジネス・ロジックの 1 つ以上のステップを個別のプロセスとしてインプリメントし、このプロセスをメイン・プロセスから呼び出すのです。サブプロセスもまた、別のプロセスを開始することができます。これにより、不特定な深さのプロセス・インスタンスの階層が生じます。これらのプロセスをデプロイする場合は、プロセス間リレーションシップのプロセス・テンプレートのすべてを、同じ Business Process Choreographer データベースにデプロイする必要があります。

ライフ・サイクル管理

サブプロセスは、呼び出しプロセスと、対等リレーションシップか親子リレーションシップにあります。このリレーションシップにより、呼び出しプロセスのプロセス・ライフ・サイクルを管理するアクションが呼び出されたときのサブプロセスの振る舞いが決まります。ライフ・サイクルのアクションは、中断、再開、終了、削除、および補正で構成されます。プロセスのライフ・サイクルを管理するアクションは、トップレベルのプロセス・インスタンスでのみ実行可能です。

呼び出しプロセス対サブプロセスのリレーションシップは、サブプロセスの `autonomy` 属性によって判別されます。この属性の値は、次の値のいずれかになります。

peer 対等プロセスは、トップレベル・プロセスと見なされます。トップレベル・プロセスは、別のプロセス・インスタンスによって呼び出されていないか、別のプロセス・インスタンスによって呼び出されたプロセス・インスタンスですが、`autonomy` 属性の値は `peer` です。サブプロセスが対等リレーションシップの一部になっている場合、呼び出しプロセス・インスタンスのライフ・サイクル・アクションは、サブプロセス・インスタンスに適用されません。

ただし、片方向インターフェースをインプリメントする作成オペレーションを伴う長期実行プロセスの場合は、実行時に `autonomy` 属性の値が自動的に `peer` に設定されます。`autonomy` 属性を `child` に設定すると、この値は実行時に無視されます。

child サブプロセスが親子リレーションシップの一部になっている場合、親プロセス・インスタンスのライフ・サイクル・アクションは、サブプロセス・インスタンスに適用されます。例えば、親プロセス・インスタンスが中断されると、`autonomy` 属性が `child` のサブプロセス・インスタンスもすべて中断されます。

`Microflow` はいつでも子プロセスとして実行されます。ただし、2 つのプロセスの間に別のコンポーネントがある場合、例えば、2 つのプロセス・コンポーネントの間を接続しているインターフェース・マップ・コンポーネントなどは、親子関係の確立を妨げることがあります。

バージョン管理の振る舞い

使用されるプロセスのバージョンは、早期バインディングのシナリオと実行時バインディングのシナリオのどちらでプロセスが使用されるかによって決まります。

早期バインディング

早期バインディングのシナリオでは、呼び出されるサブプロセスのバージョ

ンはデプロイメント中に決定されます。呼び出しプロセスは、Service Component Architecture (SCA) ワイヤリングに従って、静的にバインドされた専用のサブプロセスを呼び出します。プロセスのバージョン管理は無視されます。

早期バインディングの例は SCA ワイヤヤーです。例えば、スタンドアロン参照をプロセス・コンポーネントに結びつけると、この参照を使用するプロセスの呼び出しはすべて、プロセス・コンポーネントによって表される特定のバージョンを対象とします。

実行時バインディング

実行時バインディングのシナリオでは、呼び出されるサブプロセス・テンプレートの決定は、呼び出しプロセス・インスタンスでサブプロセスを呼び出す必要が生じたときに行われます。この場合は、現行で有効なサブプロセスのバージョンが使用されます。プロセスの新しいバージョンが、そのプロセスの以前のバージョンのどれよりも優先されます。既存のプロセス・インスタンスは、開始時に関連付けられたプロセス・テンプレートを使って、継続して実行されます。これにより、プロセス・テンプレートは次のカテゴリに分けられます。

- 既に最新ではなくても、既存の長期実行プロセス・インスタンスにとっては有効なプロセス・テンプレート
- 新規プロセス・インスタンスで使用される現行のプロセス・テンプレート
- 有効開始日時に従って将来的に有効になるプロセス・テンプレート

サブプロセスの呼び出し時に実行時バインディングを適用するため、親プロセスは、参照パートナーで有効なサブプロセスが選択されるときは選択元となるサブプロセス・テンプレートの名前を指定する必要があります。プロセスの有効開始属性を使用して、現行で有効なサブプロセス・テンプレートが判別されます。SCA ワイヤリングはどれも無視されます。

実行時バインディングの例は、Business Process Choreographer Explorer で新規プロセスが呼び出される場合です。作成されるインスタンスはいつでも、有効開始日付が将来ではないプロセス・テンプレートの最新バージョンに基づいています。

新規バージョンのプロセス・モデルが作成されて、既存のプロセス・モデルが実行時バインディング・シナリオで使用されるとき、変更を行うことは避けてください。変更を行うと、新規バージョンのプロセスが有効になり、例えば、親プロセスが新規バージョンのサブプロセスのインスタンスを起動したときに、互換性の問題をもたらすことがあります。以下は、避ける必要のある非互換の変更です。

- 関連セットの変更
- 親プロセスがサブプロセスと通信するために使用するインターフェースの変更

管理コンソールによるプロセス・テンプレートの停止および開始:

管理コンソールを使用して、それぞれのプロセス・テンプレートを個々に開始および停止することができます。

グローバル・セキュリティーが使用可能に設定されている場合は、使用するユーザー ID にオペレーター権限があることを確認します。アプリケーションがインストールされているサーバーが稼働している必要があります。

例えば、プロセス・テンプレートが属しているビジネス・プロセス・アプリケーションをアンインストールするには、まずそのビジネス・テンプレートを停止する必要があります。以下のステップでは、管理コンソールを使用して、プロセス・テンプレートを管理する方法について説明します。

1. 管理するアプリケーションを選択します。

管理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**」 → 「**エンタープライズ・アプリケーション**」をクリックしてから、管理するアプリケーションをクリックします。

2. EJB モジュールを選択します。

「関連項目」の下の「**EJB モジュール**」をクリックしてから、EJB モジュールをクリックします。

3. 管理するプロセス・テンプレートを選択します。

「追加プロパティ」の下で、「**ビジネス・プロセス**」 をクリックしてからプロセス・テンプレートをクリックします。

4. プロセス・テンプレートを停止します。

プロセス・テンプレートの既存のインスタンスは、正常に終了するまで稼働を続けます。ただし、停止したテンプレートからプロセス・インスタンスを作成することはできません。

5. 停止状態のプロセス・テンプレートを開始します。

管理コマンドによるプロセス・テンプレートの停止および開始:

管理コマンドは、プロセス・テンプレートを停止および開始するため管理コンソールの代わりに提供します。

グローバル・セキュリティーが使用可能に設定されている場合は、使用するユーザー ID にオペレーター権限があることを確認します。

例えば、ビジネス・テンプレートが属しているビジネス・プロセス・アプリケーションをアンインストールするには、まずそのビジネス・テンプレートを停止する必要があります。以下のステップでは、管理コマンドを使用して、プロセス・テンプレートを管理する方法について説明します。

1. Business Process Choreographer サンプル・ディレクトリに変更します。 以下を入力します。

```
cd install_root/ProcessChoreographer/sample
```

2. プロセス・テンプレートを停止します。

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

ここで *application_name* は、そのテンプレートが所属するアプリケーションの名前です。

プロセス・テンプレートの既存のインスタンスは、正常に終了するまで稼働を続けます。アプリケーションが停止している場合、停止したテンプレートからプロセス・インスタンスを作成することはできません。

3. プロセス・テンプレートを開始します。

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

プロセス・テンプレートが開始されます。Business Process Choreographer Explorer を使用して、プロセス・テンプレートからプロセス・インスタンスを開始することができます。

新規プロセス・インスタンスの開始:

新規プロセス・インスタンスは、使用を許可されている任意のプロセス・テンプレートから開始できます。

インストールされたプロセス・テンプレートは、すべて Business Process Explorer でプロセス・テンプレートのリストに表示されます。新規プロセス・インスタンスを開始するには、次のステップを実行します。

1. 使用を許可されているプロセス・テンプレートを表示します。

ナビゲーション・ペインの「プロセス・テンプレート」の下で、「自分のプロセス・テンプレート」をクリックします。

2. リストからプロセス・テンプレートを選択し、「インスタンスの開始」をクリックします。

このアクションにより、「プロセス入力メッセージ」ページが表示されます。ここで、このページからビジネス・プロセスのインスタンスを開始するために必要な入力データを提供することができます。

プロセスに複数の操作が存在する場合、このアクションによって、すべての使用可能な操作を含むページが表示されます。プロセス・インスタンスを開始するための操作を選択します。

3. プロセス・インスタンスを開始するための入力データを提供します。

プロセスが長期実行プロセスである場合は、プロセス・インスタンス名に入力できます。名前を指定しない場合、新規プロセス・インスタンスには、システムによって生成された名前が割り当てられます。

プロセス入力メッセージに対する入力を完了します。

4. プロセスを開始するには、「実行依頼」をクリックします。

プロセス・インスタンスが開始されます。ビジネス・プロセスに、人間による対話を必要とするアクティビティーが含まれている場合は、潜在的な所有者すべてに対してタスクが生成されます。潜在的な所有者である場合は、このタスクが「自分のタスク」ページのリストに表示されます。

プロセスが長期実行プロセスである場合は、そのプロセスが終了すると即時にプロセス出力メッセージが表示されます。すべてのプロセスに出力メッセージがあるわけではありません。例えば、プロセスが片方向操作を実装している場合、出力メッセージは表示されません。

プロセス・インスタンスの中断と再開:

実行中のプロセス・インスタンスを中断して、後で再び再開することができます。

プロセス・インスタンスを中断して再開するには、プロセス管理者権限が必要です。

プロセス・インスタンスを中断するには、プロセス・インスタンスが実行中または失敗している状態である必要があります。プロセスを再開するには、プロセス・インスタンスが中断された状態である必要があります。

長期間にわたって実行するトップレベルのプロセス・インスタンスを中断することができます。これは、例えば、プロセスの後半で使用するバックエンド・システムへのアクセスの構成、あるいはプロセス・インスタンスの失敗の原因となっている問題の修正を行えるよう実行することができます。プロセスの前提条件が満たされている場合は、プロセス・インスタンスの実行を再開できます。

プロセス・インスタンスを中断または再開するには、**Business Process Choreographer Explorer** で次のステップを完了します。

1. プロセス・インスタンスのリストを表示します。

例えば、ナビゲーション・ペインの「プロセス・インスタンス」の下で、「**自分で管理**」をクリックします。

2. プロセスを中断します。

プロセス・インスタンスを選択し、「**中断**」をクリックします。

このアクションにより、指定されたトップレベルのプロセス・インスタンスが中断されます。プロセス・インスタンスは中断状態になります。autonomy 属性が child に設定されたサブプロセスも、状態が実行中、失敗、終了、または補正になっていれば中断されます。ただし、その場合も、プロセス・インスタンスに属するアクティブなアクティビティおよびタスクは完了できます。

3. プロセス・インスタンスを再開します。

中断状態のプロセス・インスタンスを選択し、「**再開**」をクリックします。プロセス・インスタンスとそのサブプロセスは、中断される前の状態 (例えば、実行中) になります。プロセス・インスタンスおよびそのサブプロセスが再開します。

Microflow の補正の管理:

Microflow に対する補正が失敗した場合に実行できる管理アクションはいくつかあります。

Microflow を補正するには、管理コンソールで補正サービスが開始されている必要があります。

Microflow の実行時に、問題が発生する場合があります。そのような場合、プロセス・モデルでプロセスに対して補正が定義されている可能性があります。補正によって、直前に完了したステップを元に戻すことができます。例えば、データおよび状態をリセットして、これらの問題からリカバリーできます。

ただし、補正処理も失敗する場合があります。Microflow の補正が失敗した場合は、プロセス管理者が問題を解決するために介入する必要があります。

失敗した補正アクションを管理するには、Business Process Choreographer Explorer で次のステップを実行します。

1. 失敗した補正アクションのリストを表示します。

ナビゲーション・ペインの「プロセス・インスタンス」の下で、「失敗した補正」をクリックします。

「失敗した補正」ページが表示されます。このページには、名前付き補正アクションが失敗した理由に関する情報が含まれています。この情報は、失敗した補正を訂正するために実行すべきアクションを判別するうえで役立ちます。

2. アクティビティーを選択してから、使用可能なアクションのいずれかをクリックします。

次の管理アクションが使用可能です。

スキップ

現在の補正アクションをスキップし、Microflow の補正を続行します。補正されていないアクティビティーが発生することになります。

再試行 失敗した補正アクションを訂正するためのアクションを実行するには、「再試行」をクリックして、補正アクションを再試行します。

停止 補正処理を停止します。

ビジネス・プロセスの補正:

補正は、正常に完了したプロセス内の操作を元に戻すための手段です。

補正処理は、プロセス・モデルで補正が定義された実行中のプロセス・インスタンスでエラーが発生した場合に開始されます。補正は、エラーが発生した時点までにコミットされた操作の影響をリバースし、整合した状態に戻します。

プロセス・モデルで、長期実行プロセスおよび Microflow に対する補正を定義できます。

長期実行プロセスの補正

長期実行プロセスに対する補正は、ビジネス・レベル補正とも呼ばれます。このタイプの補正は、スコープ・レベルで定義されています。これは、プロセスの一部、またはプロセス全体が補正可能ということです。

補正は、障害ハンドラー、スコープまたはプロセスの補正ハンドラーによって起動されます。補正は、プロセスのもう一つのナビゲーション・パスです。

長期実行プロセスは、子プロセスを囲む親スコープが補正されたときに、正常終了した子プロセスを自動的に補正します。プロセス内では、正常に完了した起動およびスコープ・アクティビティーのみ補正されます。

Microflow の補正

Microflow の補正は、テクニカル補正としても知られています。このタイプの補正は、Microflow が含まれている作業単位（トランザクションまたはアクティビティ

ー・セッション) がロールバックされた場合に起動されます。したがって、通常、やり直しアクションは、作業単位のロールバックによってリバースすることのできないアクティビティーに対して指定されます。プロセス・インスタンスが実行されると、補正可能アクティビティーに対するやり直しアクションが、それを囲む作業単元に登録されます。この作業単位 (ロールバックまたはコミット) の結果によって、補正が開始されます。

Microflow が補正可能な長期実行プロセスの子である場合は、Microflow の完了時に、親プロセスに対して Microflow の取り直しアクションが使用可能になります。したがって、Microflow は親プロセスの補正に参加できる可能性があります。このようなタイプの Microflow では、プロセス・モデルを定義する際にプロセス内のすべてのアクティビティーに対して取り直しアクションを指定することをお勧めします。

補正処理中にエラーが発生した場合、補正アクションでは、エラーを手動で解決する必要があります。Business Process Choreographer Explorer を使用して、これらの補正アクションを修復することができます。

プロセス・インスタンスの終了:

プロセス・インスタンスを終了するには、プロセス管理者権限が必要です。

例えば、プロセス・インスタンスが示している作業または文書が必要なくなった場合や、プロセス・インスタンスを完了できるユーザーがいない場合、プロセス・テンプレートに問題が発生して再設計が必要な場合などは、プロセス・インスタンスを終了することができます。

ビジネス・プロセス・モデルに補正が定義されている場合、補正を持つプロセス・インスタンスの終了を選択できます。

プロセス・インスタンスを終了するには、Business Process Choreographer Explorer で次のステップを実行します。

1. 管理できるプロセス・インスタンスを表示します。

ナビゲーション・ペインの「プロセス・インスタンス」の下で、「自分で管理」をクリックします。

2. 停止するプロセス・インスタンスを選択します。

- 補正を持つプロセス・インスタンスを終了させるには、「補正」をクリックします。

このアクションで、プロセス・インスタンスは終了し、補正処理が開始します。

- 補正を持たないプロセス・インスタンスを終了させるには、「終了」をクリックします。

このアクションにより、プロセス・インスタンスは未解決のアクティビティーまたはタスクを待たず、即時に停止します。終了したプロセス・インスタンスは補正されません。

プロセス・インスタンスの削除:

すべてのプロセス・インスタンスが完了時に自動的に削除されるわけではありません。完了したプロセス・インスタンスは明示的に削除できます。

プロセス・インスタンスを削除するには、プロセス管理者権限が必要です。プロセス・インスタンスは、終了または強制終了の状態になっている必要があります。

対応するプロパティが、プロセス・モデルのプロセス・テンプレート用に設定されている場合、完了したプロセス・インスタンスは自動的に Business Process Choreographer データベースから削除されます。

データベースにプロセス・インスタンスを保持する必要がある場合があります。例えば、監査ログに書き込まれていないプロセス・インスタンスからのデータを照会する場合、またはオフピーク時に、プロセスの削除を延期したい場合です。ただし、不要になった以前のプロセス・インスタンスのデータが、ディスク・スペースおよびパフォーマンスに影響を与える可能性があります。したがって、不要になった、または保持する必要がなくなったプロセス・インスタンスのデータは、定期的に削除する必要があります。この保守タスクは、オフピーク時に実行するようにしてください。

プロセス・インスタンスを削除するには、Business Process Choreographer Explorer で次のステップを実行します。

1. 管理するプロセス・インスタンスを表示します。

ナビゲーション・ペインの「プロセス・インスタンス」の下で、「自分で管理」をクリックします。

2. 削除するプロセス・インスタンスを選択し、「削除」をクリックします。

このアクションによって、選択したプロセス・インスタンスがデータベースから削除されます。

タスク・テンプレートとタスク・インスタンスの管理

管理コンソールおよび管理コマンドを使用して、タスク・テンプレートを管理します。タスク・インスタンスの処理には Business Process Choreographer Explorer を使用します。

ヒューマン・タスクのための許可のロール:

ヒューマン・タスクで実行できるアクションは、ご使用の許可のロールに依存します。このロールは、J2EE のロールまたはインスタンス・ベースのロールとすることができます。

ロールとは、同じ権限レベルを共有する従業員のグループです。Java 2 Platform, Enterprise Edition (J2EE) のロールは、ヒューマン・タスク・コンテナが構成されたときにセットアップされます。インスタンス・ベースのロールは、タスクがモデル化されたときにヒューマン・タスクおよびエスカレーションに割り当てられます。ロール・ベースの許可については、グローバル・セキュリティーが WebSphere Application Server で使用可能になっていることが必要です。

J2EE のロール

以下の J2EE のロールがサポートされています。

- J2EE TaskSystemAdministrator。このロールを割り当てられたユーザーは、すべての特権を持ちます。
- J2EE TaskSystemMonitor。このロールを割り当てられたユーザーは、すべてのタスク・オブジェクトのプロパティを表示できます。

これらのロールへのユーザーおよびグループの割り当てを変更するには、管理コンソールを使用できます。

RACF セキュリティーによるロールの設定: この RACF 許可は、以下のセキュリティー・フィールドを指定した場合に適用されます。

- **com.ibm.security.SAF.authorization= true**

```
RDEFINE EJBROLE TaskSystemAdministrator UACC(NONE)
PERMIT TaskSystemAdministrator CLASS(EJBROLE) ID(userid) ACCESS(READ)
```

```
RDEFINE EJBROLE TaskSystemMonitor UACC(NONE)
PERMIT TaskSystemMonitor CLASS(EJBROLE) ID(userid) ACCESS(READ)
```

- **com.ibm.security.SAF.delegation= true**

```
RDEFINE EJBROLE JMSAPIUser UACC(NONE) APPLDATA(' userid')
```

Security Authorization Facility (SAF) ベースの許可 (RACF EJBROLE プロファイルの使用など) を使用して、WebSphere Application Server 管理コンソール・アプリケーションなど、EJB および Web アプリケーションにおける Java 2 Platform, Enterprise Edition (J2EE) ロールに対するクライアントのアクセスを制御できます。詳細については、WebSphere Application Server for z/OS インフォメーション・センターの『役割ベースの許可の System Authorization Facility』を参照してください。

インスタンス・ベースのロール

タスク・インスタンスまたはエスカレーション・インスタンスはタスク・モデルではスタッフ・メンバーには直接割り当てられておらず、その代わり、使用可能なロールの 1 つに割り当てられます。インスタンス・ベースのロールに割り当てられたスタッフ・メンバーは、そのロールに応じたアクションを実行できます。インスタンス・ベースのロールとユーザーの関連付けは、スタッフの解決を使用して実行時に決定されます。

以下のインスタンス・ベースのロールがサポートされています。

- タスクに対して: 可能なインスタンス作成者、オリジネーター、可能なスターター、スターター、可能な所有者、所有者、読者、編集者、管理者
- エスカレーションに対して: エスカレーション受信者

これらのロールは、以下のアクションの実行が許可されています。

ロール	許可されたアクション
潜在的インスタンス作成者	このロールのメンバーは、タスクのインスタンスを作成できません。タスク・テンプレートまたはアプリケーション・コンポーネントに対して可能なインスタンス作成者が定義されていない場合、すべてのユーザーがこのロールのメンバーとして考慮されません。

ロール	許可されたアクション
オリジネーター	このロールのメンバーは、タスクが開始されるまで管理権限を持ちます。タスクが開始されると、オリジネーターは読者の権限を持ち、タスクの中断や再開、および作業項目の転送などの一部の管理アクションを実行できます。
可能なスターター	このロールのメンバーは、既存のタスク・インスタンスを開始できます。可能なスターターが指定されない場合、オリジネーターが可能なスターターになります。可能なスターターがない場合のインライン・タスクについては、デフォルトは全員となります。
スターター	このロールのメンバーは読者の権限を持ち、作業項目の転送などの一部の管理アクションを実行できます。
可能な所有者	このロールのメンバーはタスクを要求できます。タスク・テンプレートまたはアプリケーション・コンポーネントに対して可能な所有者が定義されていない場合、すべてのユーザーがこのロールのメンバーとして考慮されます。
所有者	タスクを処理し、完了します。
読者	すべてのタスク・オブジェクトのプロパティを表示できますが、操作はできません。
編集者	このロールのメンバーはタスクの内容を扱うことができますが、要求または完了することはできません。
管理者	このロールのメンバーは、タスク、タスク・テンプレート、およびエスカレーションを管理できます。
エスカレーション受信者	このロールのメンバーは、読者権限を持ちます。

管理コンソールによるタスク・テンプレートの停止および開始:

管理コンソールを使用して、タスク・テンプレートを開始および停止します。

グローバル・セキュリティーが使用可能に設定されている場合は、ユーザー ID にオペレーター権限があることを確認します。

タスク・テンプレートは、エンタープライズ・アプリケーション内でスタンドアロン・タスクと表される Service Component Architecture (SCA) サービスを定義します。タスク・テンプレートを含むエンタープライズ・アプリケーションがインストールされ、配置され、開始されると、タスク・テンプレートが開始状態になります。

1. 管理するアプリケーションを選択します。

管理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**」 → 「**エンタープライズ・アプリケーション**」をクリックしてから、管理するアプリケーションをクリックします。

2. EJB モジュールを選択します。

「関連項目」の下の「**EJB モジュール**」をクリックしてから、EJB モジュールをクリックします。

3. 管理するタスク・テンプレートを選択します。

「追加プロパティ」の下の「ヒューマン・タスク」をクリックしてから、タスク・テンプレートをクリックします。

4. タスク・テンプレートを停止するには、「停止」をクリックします。
5. タスク・テンプレートを開始するには、「開始」をクリックします。

管理コマンドによるタスク・テンプレートの停止および開始:

管理コマンドは、タスク・テンプレートを停止および開始するため管理コンソールの代わりに提供します。

グローバル・セキュリティーが使用可能に設定されている場合は、オペレーター権限を持つユーザー ID でログインしていることを確認します。

タスク・テンプレートは、エンタープライズ・アプリケーション内でスタンドアロン・タスクと表される Service Component Architecture (SCA) サービスを定義します。タスク・テンプレートを含むエンタープライズ・アプリケーションがインストールされ、配置され、開始されると、タスク・テンプレートが開始状態になります。

1. Business Process Choreographer サンプル・ディレクトリーに変更します。以下を入力します。

```
cd install_root/ProcessChoreographer/sample
```

2. タスク・テンプレートを停止します。

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -stop application_name
```

ここで *application_name* は、そのテンプレートが所属するアプリケーションの名前です。タスク・テンプレートの既存のインスタンスは、正常に終了するまで稼働を続けます。

3. タスク・テンプレートを開始します。

```
install_root/bin/wsadmin -f bpcTemplates.jacl  
                        -start application_name
```

タスク・テンプレートが開始されます。Business Process Choreographer Explorer を使用して、タスク・テンプレートに関連付けられているタスク・インスタンスを処理できます。

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タスク・テンプレートのどれからでも作成して開始することができます。

インストールされたタスク・テンプレートは、すべて Business Process Choreographer Explorer でタスク・テンプレートのリストに表示されます。タスク・テンプレートからタスク・インスタンスを作成して開始するには、次のステップを実行します。

1. 使用を許可されているタスク・テンプレートを表示します。

ナビゲーション・ペインの「タスク・テンプレート」の下で、「自分のタスク・テンプレート」をクリックします。

2. リストからタスク・テンプレートを選択し、「**インスタンスの作成**」をクリックします。
3. タスク・インスタンスを開始します。

ナビゲーション・ペインの「タスク・インスタンス」の下にある「**自分で開始したタスク・インスタンス (Task Instances Initiated By Me)**」をクリックし、タスク・インスタンスを選択して、「**インスタンスの開始**」をクリックします。

このアクションにより、「タスク入力メッセージ」ページが表示されます。ここで、このページからタスク・テンプレートのインスタンスを開始するために必要な入力データを提供することができます。

4. タスク・インスタンスを開始するための入力データを提供します。
5. タスク・インスタンスを開始するには、「**実行依頼**」をクリックします。

タスク・インスタンスで作業する準備が整いました。

タスクの操作:

タスクを操作するには、タスクを要求してから、それを完了するために必要なアクションを実行します。

タスクが作動可能状態である場合、タスクの潜在的な所有者または管理者は、そのタスクを要求できます。タスクを要求したユーザーは、そのタスクの所有者になり、タスクの完了に責任を負います。

タスクのリストには、ユーザーが読者または編集者のロールを持っているタスクも表示されます。

Business Process Choreographer Explorer を使用してタスクを要求し完了するには、次のステップを実行します。

1. 自分に割り当てられているタスクを表示します。

「**タスク・インスタンス**」 → 「**自分のタスク**」をクリックします。

このアクションにより、「自分のタスク」ページが表示され、割り当てられているタスクがリストされます。

2. 操作するタスクを要求します。

タスクの横にあるチェック・ボックスを選択し、「**処理**」をクリックします。

このアクションにより、「タスク・メッセージ」ページが表示されます。

3. タスクを完了する情報を提供します。

例えばタスクを完了するために同僚からの情報が必要な場合など、操作を中断する必要がある場合は、「**保管**」をクリックして変更を保管します。

4. 「**完了**」をクリックして、提供した情報でタスクを完了します。

完了したタスクは、完了状態になります。タスクを完了しないと、そのタスクは要求済み状態のままです。

タスク割り当ての管理:

タスクの開始後に、そのタスクに対するタスク割り当てを管理する必要がある場合があります。

作業項目は、特定の理由でのユーザーまたはユーザー・グループに対する、タスクまたはプロセス・インスタンスなどのビジネス・エンティティの割り当てです。割り当ての理由により、ユーザーは、可能な所有者、編集者、または管理者など、ビジネス・プロセス・シナリオでのさまざまなロールを演じることができます。

さまざまなユーザーがさまざまなロールを持つことができるので、タスク・インスタンスには幾つかの作業項目を関連付けることができます。例えば、John、Sarah、Mike はすべてタスク・インスタンスの潜在的な所有者であり、Anne は管理者です。作業項目は、この 4 人全員に対して生成されます。John、Sarah、Mike のタスク・リストには、それぞれ自分の作業項目だけがタスクとして表示されます。Anne は管理者なので、自分のタスクの作業項目を取得し、さらに John、Sarah、Mike に対して生成された作業項目を管理できます。

場合によっては、タスクの開始後にタスク割り当てを変更する必要があります。例えば、元の所有者から別のユーザーへ作業項目を転送します。追加の作業項目の作成や、必要なくなった作業項目の削除を実行しなければならない場合もあります。

作業項目の転送:

例えば、作業項目の所有者が休暇中で、その所有者が出社する前に作業項目を完了する必要がある場合などに、作業項目を別のユーザーに転送することができます。

作業項目を転送するには、次のいずれかのロールを持っている必要があります。

ロール	タスク状態	作業項目は、次のユーザー・ロールに転送可能です。
所有者	要求	潜在的な所有者、管理者。
スターター	強制終了、期限切れ、終了、失敗、または実行中	潜在的なスターター、管理者。
オリジネーター	任意のタスク状態	潜在的なインスタンス作成者、管理者。タスクがアクティブ状態の場合は、任意のユーザー・ロールに転送できます。
管理者	作動可能、要求済み、強制終了、期限切れ、終了、失敗、または実行中	任意のユーザー・ロール。

作業項目を転送するには、Business Process Choreographer Explorer で次のステップを実行します。

1. 管理するタスク・インスタンスを表示します。

ナビゲーション・ペインの「タスク・インスタンス」の下で、「自分で管理」をクリックします。

2. タスク・インスタンスの作業項目を表示します。

「自分で管理するタスク・インスタンス」ページで、タスク・インスタンスを選択して「作業項目」をクリックします。

3. 作業項目を転送します。
 - a. 「新規所有者」フィールドで、新規作業項目所有者のユーザー ID を指定します。

タスク管理者であれば、作業項目をだれにでも転送可能です。作業項目の現行所有者は、作業項目の別の潜在的所有者またはタスク管理者にのみ作業項目を転送できます。

- b. 「理由」リストから 1 つ以上のロールを選択します。これらのロールにより、割り当てられたユーザーが転送済みの作業項目で実行できるアクションが決まります。
 - c. 「転送」をクリックします。

転送済みの作業項目が、新規作業項目所有者に属すタスクのリストに表示されません。

作業項目の作成:

例えば現在の潜在的な所有者が追加作業を受け入れられない場合などに、新規の潜在的な所有者の作業項目を作成できます。また、スタッフ・リポジトリに対する照会が潜在的な所有者を戻さない場合に、作業項目を作成することもできます。これは、例えば、プロセスの開始以降に組織が変更した場合に、長期実行プロセスで発生します。

タスク・インスタンスの作業項目を作成するには、タスクに対する適切なロールを持っている必要があります。タスク管理者は、タスク・インスタンスの状態が、作動可能、要求済み、実行中、終了、または失敗のいずれかになっていれば、そのタスク・インスタンスの作業項目を作成できます。タスク・インスタンスがタスク・テンプレートから派生している場合は、タスクが強制終了または期限切れ状態のときにも作業項目を作成できます。

作業項目を作成するには、Business Process Choreographer Explorer で次のステップを実行します。

1. 管理するタスク・インスタンスを表示します。

ナビゲーション・ペインの「タスク・インスタンス」の下で、「自分で管理」をクリックします。
2. 作業項目を作成する対象のタスク・インスタンスを選択し、「作業項目の作成」をクリックします。「作業項目の作成」ページが表示されます。
3. 作業項目を作成します。
 - a. 「新規所有者」フィールドで、新規作業項目所有者のユーザー ID を指定します。
 - b. 「理由」リストから 1 つ以上のロールを選択します。

これらのロールにより、割り当てられたユーザーが新規作業項目で実行できるアクションが決まります。
 - c. 「作成」をクリックします。

作業項目は、新規所有者に対して指定する各ロールに対して作成されます。新規タスクは、このユーザーに割り当てられるタスクのリストに表示されます。

作業項目の削除:

例えば、エラーの作業項目を作成した場合や、もう会社で働いていない人に対して作業項目が生成されている場合は、作業項目を削除することができます。

タスク・インスタンスの作業項目を削除するには、タスクに対する適切なロールを持っている必要があります。タスク管理者は、タスク・インスタンスの状態が、作動可能、要求済み、実行中、終了、または失敗のいずれかになっていれば、そのタスク・インスタンスを削除できます。タスク・インスタンスがタスク・テンプレートから派生している場合は、強制終了または期限切れ状態のタスクも削除できます。

作業項目を削除するには、**Business Process Choreographer Explorer** で次のステップを実行します。

1. 管理するタスク・インスタンスを表示します。

ナビゲーション・ペインの「タスク・インスタンス」の下で、「自分で管理」をクリックします。

2. タスク・インスタンスの作業項目を表示します。

「自分で管理するタスク・インスタンス」ページで、タスク・インスタンスを選択して「作業項目」をクリックします。

3. 作業項目を削除します。

作業項目を選択し、「削除」をクリックします。

作業項目が削除されます。

タスク・エスカレーションの表示:

エスカレーションは、割り当てられたタスクをユーザーが時間どおりに完了できない可能性があることをエスカレーション受信者に通知します。

タスクが期限切れになると、エスカレーションが発生する場合があります。エスカレーションの結果として、次のアクションが発生します。

- 例えば、マネージャーが問題の解決をサポートするための処置をとれるように、新規の作業項目が作成されます。
- ヒューマン・タスク・コンテナの構成時に E メール設定を指定した場合は、エスカレートされたタスクについて知らせるために、指定された担当者に E メールが送信されます。
- イベント通知ハンドラーが呼び出されます。

エスカレーションを表示するには、「タスク・インスタンス」の下にある「自分のエスカレーション」をクリックします。

- エスカレーションに関する情報を表示するには、エスカレーション ID をクリックします。

- エスカレートされたタスクに関する情報を表示するには、タスク名をクリックします。

ビジネス・ルール

ビジネス・ルールを使用して、ビジネス・プラクティスの振る舞いを制御します。(ビジネス・ルールの構築、配置についての詳細は、WebSphere Integration Developer インフォメーション・センターを参照してください。)

ビジネス・ルールとは

ビジネス・ルールは、ビジネス・プラクティスの振る舞いに構造を与えたり振る舞い制御したりするすべてのものです。ルールは、ビジネス・ポリシーの施行、組織内での共通のガイドラインを確立、ビジネス環境でのアクセスの制御などを実行できます。(ビジネス・ルールの構築、配置についての詳細は、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、ビジネス内からのビジネス・プラクティスまたは規制機関などビジネス外から要求されるビジネス・プラクティス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・ルールの構築、配置についての詳細は、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・ルールの使用法

Eclipse ベースの WebSphere Integration Developer ツールを使用して、ビジネス・ルールを作成および変更します。(ビジネス・ルールの構築、配置についての詳細は、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。) Web ベースのビジネス・ルール・マネージャー・ツールを使用して、ビジネス・ルール値を管理および変更できます。

スタンドアロン・サーバー用ビジネス・ルール動的リポジトリのインストール

インストール・スクリプトおよび構成スクリプトの実行時に、動的リポジトリがスタンドアロン・サーバーにインストールされます。

ビジネス・ルールをサポートしないデータベースを選択すると、システムはデフォルトのデータベースである Cloudscape を使用します。

DB2 for z/OS バージョン 7 では、基本キーのサイズが 255 バイトに制限されます。ビジネス・ルールやセクターの成果物用の動的な成果物リポジトリでは、ターゲットのネーム・スペース、名前、タイプのいずれかを使用して基本キーを生成します。DB2 for z/OS 版を使用するようにシステムを構成した場合は、名前を以下のように制限する必要があります。

- ターゲットのネーム・スペース = 170 バイト
- 名前の最大長 = 75 バイト
- タイプの最大長 = 10 バイト (この値はランタイム・システムにより設定されます)

注: DB2 for z/OS バージョン 8 では、この制限は適用されません。

ビジネス・ルール動的リポジトリを手動でインストールする場合、DB2 用に提供されている `configureArtifactRepository` コマンドを使用します。

```
wsadmin -f configureArtifactRepository.jaclproperties file -profile default wsadmin
-f configureArtifactRepository.jacl -profile default
```

各部の意味は、次のとおりです。

properties file データベース・プロパティーを含むファイル

プロパティー・ファイルで指定するデータベースのプロパティーは次のとおりです。

cellName

セルの名前

例: `cellName=T40Cell04`

nodeName

ノードの名前

例: `nodeName=T40CellManager04`

profilePath

プロファイルのルート名

例: `profilePath=install_root/AppServer/profiles/profile_name`

profileName

プロファイル名

例: `profileName=default`

WBI_HOME

install_root = WebSphere Process Server のインストール・ディレクトリー

例: `/usr/lpp/zWPS/V6R0`

WAS_HOME

install_root = WebSphere Application Server for z/OS のインストール・ディレクトリー

例: `/WebSphere/V6R0M0/AppServer`

dbName

データベース名

例: `dbName=WPSDB`

dbType

DB2_Universal、DB2UDBOS390_V7_1、DB2UDBOS390_V8_1、DB2_CLI または CLOUDSCAPE

Cloudscape が dbtype 用に指定されている場合、以下のプロパティも提供される必要があります。

dbServerPort

Cloudscape サーバー・ポート

例: dbServerPort=50000

dbServerName

Cloudscape サーバー名

例: dbServerName=dbserver

dbHostName

Cloudscape ホスト名

例: dbHostName=dbserver

dbPassword

Cloudscape サーバー・パスワード

例: db2passwd

dbClassPath

Cloudscape JDBC ドライバーへのパス

例: dbClassPath=Install_root/IBM/SQLLIB/java

注: WebSphere Process Server に付属する Cloudscape JDBC ドライバーは、*WPS_INSTALL_ROOT/Cloudscape/lib* にあります。ここで、*WPS_INSTALL_ROOT* は WebSphere Process Server をインストールしたディレクトリーです。

configureArtifactRepository コマンドの例を次に示します。

```
wsadmin -f configureArtifactRepository.jacl install_root/AppServer/profiles/default
```

Network Deployment 用ビジネス・ルール動的リポジトリーのインストール

ビジネス・ルールを中に含むアプリケーションをインストールするには、ビジネス・ルール用動的リポジトリーがインストールされている必要があります。

動的リポジトリーでは、中央構成がサポートされています。この構成では、同一のリポジトリーを使用するようにすべてのサーバーを構成できるので、アプリケーションはすべて同じデータを使用することができます。これは、ビジネス・ルール・マネージャー・ツールを使用してこのデータを動的に更新するユーザーにとって重要です。中央リポジトリーにより、以下を実行できるようになります。

- 1 度だけ行われる変更
- すべてのサーバー・インストールで有効になる変更

DB2 for z/OS バージョン 7 では、基本キーのサイズが 256 バイトに制限されます。ビジネス・ルールやセクターの成果物用の動的な成果物リポジトリーでは、ターゲットのネーム・スペース、名前、タイプのいずれかを使用して基本キーを生成します。DB2 for z/OS 版を使用するようにシステムを構成した場合は、名前を以下のように制限する必要があります。

- ターゲットのネーム・スペース = 171 バイト

- 名前の最大長 = 75 バイト
- タイプの最大長 = 10 バイト (この値はランタイム・システムにより設定されます)

注: DB2 for z/OS バージョン 8 では、この制限は適用されません。

製品構成の一部として、WPSDB データベースを非 DB2 データベースとなるように構成する場合は、`configureArtifactRepository` コマンドを使用して動的リポジトリを手動で構成する必要があります。手動構成は DB2 に限定されます。

`configureArtifactRepository` コマンドの構文を次に示します。

```
wsadmin -f configureArtifactRepository.jacl property file -profile {default | dmgr}
```

各部の意味は、次のとおりです。

properties file = データベース・プロパティを含むファイル
-profile {default | dmgr} = プロファイル・タイプ

プロパティ・ファイルで指定するデータベースのプロパティは次のとおりです。

cellName

セルの名前

例: `cellName=T40Cell04`

nodeName

ノードの名前

例: `nodeName=T40CellManager04`

profilePath

プロファイルのルート名

例: `profilePath=install_root/AppServer/profiles/profile_name`

profileName

プロファイル名

例: `profileName=default`

WBI_HOME

install_root = WebSphere Process Server のインストール・ディレクトリー

例: `/usr/lpp/zWPS/V6R0`

WAS_HOME

install_root = WebSphere Application Server Network Deployment のインストール・ディレクトリー

例: `/WebSphere/V6R0M0/AppServer`

dbName

データベース名

例: `dbName=dynamicDB`

dbType

DB2_Universal、DB2UDBOS390_V7_1、DB2UDBOS390_V8_1、または
DB2_CLI

configureArtifactRepository コマンドの例を次に示します。

```
wsadmin -f configureArtifactRepository.jacl install_root/AppServer/profiles/default
```

ビジネス・ルール・コンポーネントへのアクセス

ビジネス・ルール・コンポーネントを表示することは、ビジネス・ルール・グループを管理する際の最初のステップです。この表示から、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できます。

WebSphere Process Server でこのタスクを実行するには、ユーザーは管理コンソールにいる必要があります。

このタスクを実行して、サーバー上に存在するビジネス・ルール・グループを判別します。

1. 管理コンソールで「サーバー」>「アプリケーション・サーバー」を選択します。
2. 「サーバー名」をクリックして、ビジネス・ルールを表示するサーバー・リストからサーバーを選択します。
3. 「Business Integration」の下の「ビジネス・ルール」をクリックします。

コンソールに、各グループの説明で定義されたすべてのビジネス・ルール・コンポーネントのリストが表示されます。

管理コンソールを使用したビジネス・ルールのエクスポート:

ビジネス・ルール・コンポーネントをエクスポートすると、開発環境にインポートするファイルが作成されます。それによって、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。

このタスクを開始するには、『ビジネス・ルール・コンポーネントへのアクセス』に説明されているように、ビジネス・ルール・コンポーネントが既に表示されている必要があります。

ビジネス・ルール・テーブルが変更されていて、開発環境と実稼働環境を同期する場合、ビジネス・ルール・コンポーネントをエクスポートします。このタスクは、ビジネス・ルール・コンポーネント画面から開始されます。

1. エクスポートするビジネス・ルール・グループを選択します。

ビジネス・ルール・グループの隣のチェック・ボックスをクリックしてから、「エクスポート」をクリックします。ブラウザーには、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「エクスポートするビジネス・ルール」パネルです。) 各ビジネス・ルール・グループにはファイル拡張子の .zip があります。

2. ファイルをダウンロードします。

各ファイル名をクリックすると、プロンプトが出てファイルを保管するように要求されます。プロンプトが出たら、「OK」をクリックしてファイル・システムにファイルを配置します。

注: 選択すると、ダウンロードするときにファイルを名前変更することができます。

3. ビジネス・ルール表示パネルに戻ります。

「戻る」をクリックして、ビジネス・ルール・グループのリストに戻ります。

指定した場所にファイルが保管されます。その後、そのファイルをテスト・システムにコピーできます。

このファイルは WebSphere Integration Developer 環境にインポートする必要があります。WebSphere Integration Developer について詳しくは、インフォメーション・センターを参照してください。

コマンド行を使用したビジネス・ルールのエクスポート:

ビジネス・ルールのコンポーネントは、`exportBusinessRuleArtifacts` コマンドを使用してエクスポートすることもできます。

目的

`exportBusinessRuleArtifacts` コマンドを使用してコマンド行からビジネス・ルール・コンポーネントをエクスポートします。

構文

```
wsadmin -f exportBusinessRuleArtifacts.jacl target namespace business rule  
group component name<user> -zipf
```

パラメーター

ここで、引数は以下のとおりです。

target namespace = エクスポートするビジネス・ルール・グループ・コンポーネントのターゲット・ネーム・スペースの名前

business rule group component name = エクスポートするビジネス・ルール・グループ・コンポーネントの名前

user = ユーザー

-zipf (オプション) = ビジネス・ルール・グループ・コンポーネントをエクスポートするための ZIP ファイルの名前。指定されない場合、*business rule group component name.zip* にデフォルト設定される

例

```
wsadmin -f exportBusinessRuleArtifacts.jacl  
http://test.oo.brules OnlineOrder admin -zipf c:/artifacts/onlineorder.zip
```

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・ルール値を変更する際にビジネス・アナリストを支援する Web ベースのツールです。このツールは、WebSphere Process Server の最初のインストール後にインストールする、このサーバーのオプションです。ビジネス・ルール・マネージャーは、Web ベースのインターフェースを使用してビジネス・ルールをブラウズおよび編集します。

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・コンポーネントは、ビジネスを営むときのルールを管理するために、ビジネス・アナリストが使用する主要な WebSphere Process Server ツールです。

ビジネス・ルール・マネージャーを使用して、以下のタスクを実行します。

- リポジトリからビジネス・ルールのコピーを開く
- ビジネス・ルールを参照および編集する
- リポジトリへビジネス・ルールを公開する

図 1 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。

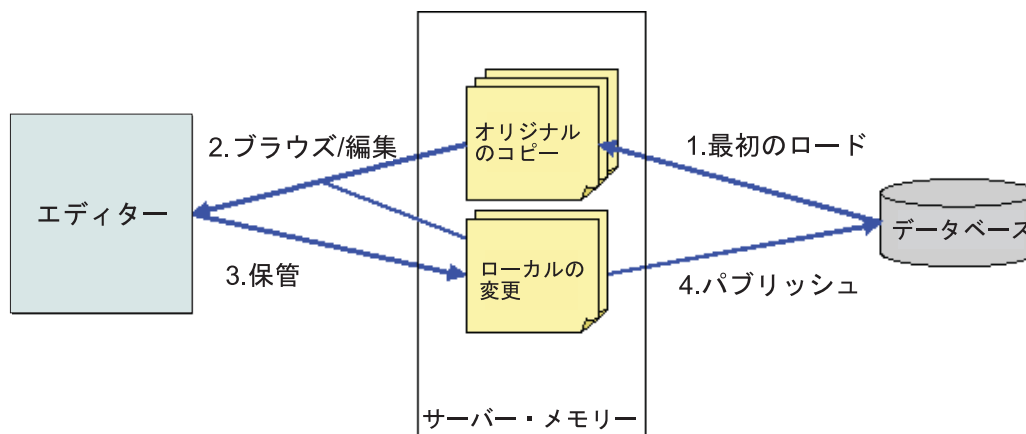


図 1. ビジネス・ルール・マネージャーのイベント・シーケンス

ビジネス・ルール・マネージャーにログオンした後に、ビジネス・ルールを変更すると、以下のイベントが発生します。

1. ビジネス・ルールを選択すると、ビジネス・ルール・マネージャーは、データベースからルールにアクセスし、それをオリジナルのコピーとしてサーバー・メモリーに保管します。
2. これで、ルールをブラウズまたは編集できるようになります。
3. ビジネス・ルールをコピーとしてローカル・サーバー・メモリーに保管します。
4. ローカル・コピーを再びデータベースに公開します。

ビジネス・ルール・テンプレート:

テンプレートは、WebSphere Process Server でオーサリングする Web ベースのルールのメカニズムを提供します。ビジネス・ルール・テンプレートは、ビジネス・ア

ナリストがアプリケーション・サーバー上でリアルタイムにルールをオーサリングするための、制限付きの方法を提供します。 WebSphere Integration Developer でテンプレートを作成し、ビジネス・アナリストがビジネス・ルール・マネージャーを使用してビジネス・ルールを編集できるようにします。(ビジネス・ルールの構築、配置についての詳細は、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。) テンプレートは、デシジョン・テーブル内のテーブル・セル、またはルール・セット内のルール用に作成されます。すべてのルール、条件ケース、またはアクションをテンプレートに配置する必要があります。

テンプレートを使用して以下を行います。

- ビジネス・ルールの値 (ルールの制約内で) を変更します。
- デシジョン・テーブルまたはルール・セット内でルールを作成します。

ビジネス・ルール・マネージャーのインストール:

ビジネス・ルール・マネージャーは、WebSphere Process Server 用のプラグインとしてインストールされます。インストールには管理コンソールを使用するか、の .jac1 スクリプトを使用します。

ビジネス・ルール・マネージャーをコマンド行から手動でインストールするには、Process Server を始動した後に、WAS_HOME/bin 内のインストール・スクリプトを実行する必要があります。

以下のコマンドを使用します。

```
- run shell/command prompt and change directory to WAS_HOME/bin  
  
    /wsadmin.sh -f ./installBRManager.jac1 -s servername  
               -n nodename -c cellname -r rootname  
               -a <applicationname
```

各部の意味は、次のとおりです。

servername

アプリケーション・サーバーの名前。

このペアの引数は、 Network Deployment 構成で必要です。

nodename

インストール・ノードの名前。

このペアの引数は、 Network Deployment 構成で必要です。

cellname

インストール・セルの名前。

このペアの引数はオプションです。

rootname

アプリケーションのルートの名前。

このペアの引数はオプションです。欠落している場合、rootname のデフォルト値は、「br/webclient」です。

applicationname

アプリケーションの名前。

欠落している場合、*applicationname* のデフォルト値は、「BusinessRulesManager」です。

注: WebSphere Process Server がシングル・サーバー環境で構成されている場合、これらのペアの引数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、必要なただ 2 つの引数ペアは以下のとおりです。

- -s *servername*
- -n *nodename*

他の引数ペアはオプションのオプションです。

注: 引数が欠落している場合は、以下のデフォルト値をとります。

- 「server1」 = *servername*
- 「br/webclient」 = *rootname*
- 「BusinessRulesManager」 = *applicationname*

サーバーの構成:

ビジネス・ルール・マネージャーを使用するサーバーを構成する必要があります。

サーバーの構成では、次の設定を行います。

注: 管理コンソールからビジネス・ルール・マネージャーをインストールする場合のみ、次の 3 つの項目を設定する必要があります。

- ユーザー ID の作成

ユーザー ID を作成する場合は、グローバル・セキュリティーを使用可能に設定する必要があります。ユーザー ID を作成して、それらを BusinessRuleUser またはビジネス・アナリストのロールにマップします。各ユーザー ID にロールを割り当てる必要があります。

- セッション追跡機構の設定

Cookie を使用してセッションを追跡します。

- 他のパラメーターの設定

少なくとも、適切なセッション・タイムアウトの値を設定してください。

- **オプション:** サーバー上でセキュリティーを使用可能に設定

異なるロールまたはユーザー ID を使用する場合、サーバーの構成時に、グローバル・セキュリティーを使用可能にする必要があります。

クライアントの構成:

ビジネス・ルール・マネージャーのインストール中に、サーバーによってクライアントが自動的に構成されます。

この 2 つの領域はクライアント・ワークステーションで構成されます。

ビジネス・ルール・マネージャーの使用時に問題が発生した場合は、以下の点を確認してください。

- スクリプトが Web ブラウザーで使用可能になっている

ビジネス・ルール・マネージャーを機能させるにはスクリプト機能が必要です。

- Cookie が使用可能になっている

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 を使用可能に設定します。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・マネージャーの起動:

サーバーおよびクライアントが、ビジネス・ルール・マネージャー用に正しく構成されていることを確認してください。

ビジネス・ルール・マネージャーの開始に必要な URL をシステム管理者にお尋ねください。

例えば、`https://hostname:9443/br/webclient` のようになります。ここで、`hostname` はサーバーの名前です。例えば、`server1` です。

ビジネス・ルール・マネージャーのアクセス:

ビジネス・ルール・マネージャーには、Web ブラウザーを使用してアクセスします。

サーバーとクライアントの両方を正しく構成しているか確認してください。詳しくは、80 ページの『ビジネス・ルール・マネージャーのインストール』を参照してください。

ビジネス・ルール・マネージャーにアクセスできるデフォルトの URL:

- `https://hostname.9443/br/webclient` (セキュリティーが使用可能な場合)
- `https://hostname.9080/br/webclient` (セキュリティーが使用不可の場合)

ここで、`hostname` はホストの名前です。

注: URL は環境によって異なる場合があります。

注: サーバーのグローバル・セキュリティーが使用可能な場合に限り、ログイン・ページが開きます。詳しくは、81 ページの『サーバーの構成』を参照してください。グローバル・セキュリティーが使用不可の場合に、ビジネス・ルール・マネージャーにアクセスすると、Rule books 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手順に従ってください。

1. ログイン・ページで、**ユーザー ID** を入力します。
2. **パスワード** を入力します。
3. 「**ログイン**」をクリックします。ビジネス・ルール・マネージャーの初期ページが開きます。

初期ページが開くと同時に、ナビゲーション領域に既存のルール・ブックがリスト表示されます。リストされているどのビジネス・ルールにも変更を加えることができるようになります。

「**Business Rules Manager**」ページのレイアウト: 以下は、ビジネス・ルール・マネージャーのページ・レイアウトに関する説明です。

ツールバー

ツールバーの内容は次のとおりです。

ようこそ

ユーザー名が表示されます。

ユーザー識別

「**ウェルカム・ユーザー名 (Welcome User Name)**」で始まる現在のユーザー名を示します。

ログアウト

グローバル・セキュリティーが使用可能である場合に、「ログイン」ページを開きます。

注: 公開せずにログアウトすると、非公開の変更がすべて失われます。

ヘルプ Business Tools Editor のインフォメーション・センターへのアクセスを提供します。(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。)

左側ペイン

ページの左側ペインには、ナビゲーション・ツリーがあります。このツリーを使用すると、必要なルール・レベルへドリルダウンできます。リソースの横の正符号 (+) または負符号 (-) をクリックして、ルール・ブックを展開または縮小します。

注: この領域は、編集モードのページには表示されません。

公開および復帰

「公開および復帰」ページを開きます。このページでは、データベースへの変更の公開やデータベース上にあった元のコピーへの復帰を実行できます。

ルール・ブック

「ルール・ブック」ページを開きます。

注: ルール・ブックの下には、ビジネス・ルール・グループのリストが表示されます。これは、ブラウズのトップレベルです。

右側ペイン

ページの右側ペインは、上と下のセクションに分割されます。

上のセクション

上のセクションには、次のエレメントが含まれています。

パス情報

この例で示されているルール・ブックとルール・ページの名前を提供します。

BusinessRuleGroup01 > Table1_operation1

ルール・タイトル

この例で示されているビジネス・ルールのリソースとタイプの名前を示します。

Ruleset1 12 - Ruleset

ボタン ページに表示されるボタンは、その特定のページの機能によって異なります。このテーブルには、ページに表示できる可能なプッシュボタンすべてのリストが表示されます。

表7. ボタン・リスト

ボタン名	機能
戻る	前のページに戻ります。
編集	「ルール」ページ、デシジョン・テーブル、またはルール・セットの編集を使用可能にします。
保管	変更を保管して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。
公開	サーバーへの「ルール」ページの公開です。
復帰	すべての変更を消去し、データベースにある下のコピーに戻します。

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または発生したエラーを表示します。これは、状況メッセージの例です。

"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することができます。

下のセクション

下のセクションには、ビジネス・ルール・マネージャーのメインの表示編集域があります。このセクションには、次の情報があります。

注: 情報は、表示しているルールのレベルによって異なります。

ルール・グループ

ルール・ブック内にあるルールをすべてリストします。

「ルール」ページ

ルール、および特定のビジネス・ルール・レコードに関するすべての情報をテーブル形式で表示します。ルール・ページの列見出しのリストを次に示します。

- 発効日時 (Effective date and time)
- アクション
- 説明
- ルール・セットまたはデシジョン・テーブル (Rule sets or decision tables)

公開および復帰

「公開」ページへのリンクです。このページでは、データベースへのルールの公開や元の状態へのルールの復帰を実行できます。詳しくは、79 ページの『ビジネス・ルール・マネージャーが機能する仕組み』を参照してください。

「ルール・ブック」ページ:

「ルール・ブック」ページには、特定のアプリケーション用に作成されたルール・グループがリストされます。

左側のナビゲーション・ツリーには、ルール・グループのリストが表示されています。ルールをすべて表示するには、ルール・ブックの横にある正符号 (+) をクリックしてツリーを展開します。左のペインのナビゲーション・ツリーでルール・ブックを選択すると、右のペインにすべての子ルールが表示されます。

右のペインの内容はテーブル形式で、次の列見出しがあります。

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

ビジネス・ルール、デシジョン・テーブル、およびルール・セットの名前がリストされています。

説明 ルール、デシジョン・テーブル、およびルール・セットの短い説明または名前のいずれかが表示されます。

アクション

最初は空になっています。ただし、グループの拡張時に、「編集」ボタンがそれぞれのルールの横に表示されます。

「ルール・ブック」ページ:

ルール・ブックが選択されている場合、「ルール・ブック」ページが開き、ルール・グループ内のすべてのルールがリストされます。

左側のナビゲーション・ツリーには、ルール・ブックのリストが表示されています。ルールをすべて表示するには、ルール・ブックの横にある正符号 (+) をクリックしてツリーを展開します。ナビゲーション・ツリーでルール・ブックを選択すると、右のペインにすべての子ルールが表示されます。

右のペインの内容はテーブル形式で、次の列見出しがあります。

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

ビジネス・ルール、デシジョン・テーブル、およびルール・セットの名前がリストされています。

説明 ルール、デシジョン・テーブル、およびルール・セットの短い説明または名前のいずれかが表示されます。

アクション

最初は空になっています。ただし、グループの拡張時に、「編集」ボタンがそれぞれのルールの上に表示されます。

「ルール」 ページ:

「ルール」 ページには、各ルールに属するテーブル形式の一連のルールおよびビジネス・ルール・レコード (デシジョン・テーブルおよびルール・セット) のセットがリストされます。

右側のペインの内容には、以下のようなエレメントが含まれています。

「タイトル」 セクション

ルール・タイトル

ルール名-ルール・ ページの形式でルール名が表示されます。ルール・タイトルの例を次に示します。

calculateDiscount-Rule Page

機能ボタン

そのページに対して特定の機能を提供します。ボタンのリストを次に示します。

- 戻る - 前のページに戻ります。
- 編集 - 選択したルールに対して編集モードを開きます。

メッセージ

誤った日付や時刻などの特定のエラーを示すメッセージ、または次に示すようにルールの状況に関するメッセージが表示されます。

"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することができます。

「一般情報」 セクション

このセクションには、次のフィールドがあります。

「最後に公開」

ルールが最後に公開された日付と状況を表示します。

状況 ルールが公開されているかどうかの状況をリストします。

説明 ルールに関する短い説明を提供します。

「ビジネス・ルール選択レコード (Business Rule Selection Records)」 セクション

ルールの構築ブロックである有効なビジネス・ルールのリストを提供します。次の情報が含まれています。

「開始日」

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」

特定の日付または「終了日無し (no end date)」のいずれかを選択できます。

注: 「終了日無し (no end date)」は、ルール・ロジックが開始日およびそれ以降のすべての日に対して有効であることを意味します。

「有効なビジネス・ルール (Effective Business Rule)」

ルール・ロジック・レコードのデシジョン・テーブルまたはルール・セットを表示します。

「デフォルト・ビジネス・ルール (Default Business Rule)」

他のルール・ロジックが適用できない場合、ビジネス・ルールにデフォルト値を提供します。これは、その他の選択レコードのいずれとも日付が一致しない場合に選択されます。

注: ルール・ロジック選択に使用される日付は、アプリケーション・プログラムによって提供されるか、サーバーの現在日付が使用されます。後者の場合、日付は使用前に **UTC** に変換されます。サーバーの時間帯は関係ありません。

「デシジョン・テーブル」 ページ:

「デシジョン・テーブル」 ページは、「ルール」 ページから開きます。

「デシジョン・テーブル」 ページの右側のペインの内容には、テーブル形式の以下のようなエレメントが含まれています。

「タイトル」 セクション

テーブル・タイトル

次の形式、「デシジョン・テーブル名-デシジョン・テーブル (テーブル)」の形式でルール名が表示されます。テーブル・タイトルの例を次に示します。

Table2212-Decision Table

機能ボタン

そのページに対して特定の機能を提供します。ボタンは以下のとおりです。

- **戻る** - 前のページに戻ります。
- **編集** - 選択したルールに対して編集モードを開きます。
- **コピー** - 新規ルールを作成するために、ルールのコピーを作成します。

注: ビジネス・ルール・マネージャーでは新規ルールは作成できません。

新規ルールを作成するには、既存ルールをコピーして、次に値を変更する必要があります。詳しくは、「デシジョン・テーブルの作成」を参照してください。

「一般情報」 セクション

このセクションには、次のフィールドがあります。

「最後に公開」

テーブルが最後に公開された日付と状況を表示します。

状況 テーブルが公開されているかどうかの状況をリストします。

説明 テーブルに関する短い説明を提供します。

「デシジョン・テーブル」セクション

このセクションでは、ルールに応じて変化するテーブル形式で、ルールのリストを提供します。

列には、方向を水平から垂直に切り替える「**方向 (orientation)**」アイコンを付けることができます。 91 ページの『デシジョン・テーブル』を参照してください。

「ルール・セット」ページ:

「ルール・セット」ページでは、ビジネス・ルール用のルール・リストが提供されています。

「ルール・セット」ページの右側のペインの内容には、テーブル形式の以下のようなエレメントが含まれています。

上のセクション

このセクションには、次のフィールドと情報があります。

ルール・タイトル

ルール名が「ルール・セット名-ルール・セット」の形式で表示されます。
ルール・セット・タイトルの例を次に示します。

Ruleset112-Ruleset

機能ボタン

次の機能があります。

- **戻る** - 前のページに戻ります。
- **編集** - 選択したルールに対して編集モードを開きます。
- **コピー** - 新規ルールを作成するために、ルールのコピーを作成します。

注: ビジネス・ルール・マネージャーでは新規ルールは作成できません。
新規ルールを作成するには、既存ルールをコピーして、次に値を変更する必要があります。詳しくは、「ルール・セットの作成」を参照してください。

「一般情報」セクション

このセクションには、次のフィールドがあります。

「最後に公開」

ルール・セットが最後に公開された日付と状況を表示します。

状況 ルール・セットが公開されているかどうかの状況をリストします。

説明 ルール・セットに関する短い説明を提供します。

「ルール」セクション

ルールの構築ブロックであるルールのリストを提供します。次の情報が含まれています。

名前 ルールの名前を提供します。

ルール ルールを定義する変数、制約、範囲、および列挙をリストします。

ビジネス・ルール・レコードの操作

ビジネス・ルール・マネージャーを使用して、ビジネス・ルール・レコードの作成、変更、または削除を行います。

ビジネス・ルール・レコードの作成:

既存のレコードからビジネス・ルール・レコードを作成する必要があります。

ルールを作成する場合、**編集**モードを使用する必要があります。

新規ビジネス・ルール・レコードを作成するには、以下のようになります。

1. 「**選択レコードの追加**」ボタンをクリックします。「**開始日/時刻**」フィールドが「**1 月 1 日**」に設定された新規ビジネス・ルール・レコードが、リストの下部に表示されます。また、日付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/時刻の設定。
 - a. 「**月**」を選択します。
 - b. **日**を選択または入力します。
 - c. **年**を入力します。
 - d. **時**を入力します (24 時間形式)。
3. 終了日/時刻の設定。
 - a. **日**を選択または入力します。
 - b. **年**を入力します。
 - c. **時**を入力します (24 時間形式)。

注: 任意の 1 時点で 1 つのルールのみが有効です。複数のルールの日付が、重複する日付/時刻を持つことはできません。

注: 日付/時刻範囲内のギャップは許可されています。デフォルトのビジネス・ルールが宣言されている場合、ギャップの間にそれが使用されます。

4. 「**有効なルール・ロジック**」を選択します。
5. 「**保管**」ボタンをクリックします。

レコードが一時的に保管されたことを知らせるメッセージがメッセージ・フィールドに表示されます。行った変更は「**公開および復帰**」ページで公開できます。(詳しくは、97 ページの『**ビジネス・ルールの公開**』を参照してください。)

関連タスク

96 ページの『**ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードの削除**』

ビジネス・ルール・レコードの変更:

ルールの「**編集**」ボタンをクリックして、既存のビジネス・ルール選択レコードの日付および時刻値を変更します。

アクション列の「編集」ボタンをクリックして、変更しようとするルール「編集」モードにいることを確認してください。ビジネス・ルール・レコードを追加、変更、または削除する方法について、詳しくは、91 ページの『デシジョン・テーブル』または 94 ページの『ルール・セット』を参照してください。

ビジネス・ルールを変更するには、以下のようにします。

1. 選択レコードの「開始日/時刻」の編集。
 - a. 「月」を選択します。
 - b. 日を選択または入力します。
 - c. 年を入力します。
 - d. 時を入力します (24 時間形式)。
2. 選択レコードの「終了日/時刻」の編集。
 - a. 「終了日/時刻」で日を選択または入力します。
 - b. 「終了日/時刻」で年を入力します。
 - c. 時を入力します (24 時間形式)。

注: 任意の 1 時点で 1 つのルールのみが有効です。複数のルールの日付が、重複する日付/時刻を持つことはできません。

注: 日付/時刻範囲内のギャップは許可されています。デフォルトのビジネス・ルールが宣言されている場合、ギャップの間にそれが使用されます。

3. 「保管」ボタンをクリックします。

注: 「時刻/日付」フィールドが無効な場合、そのフィールドは赤色に反転し、さらに時刻/日付が無効なことを知らせるメッセージがメッセージ・フィールドに表示されます。

ローカル側でレコードが保管されます。レコードをデータベースに公開する準備が整います。(詳しくは、97 ページの『ビジネス・ルールの公開』を参照してください。)

ビジネス・ルールの日付の設定について詳しくは、91 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

96 ページの『ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードの削除』

日付/時刻の選択:

使用できる日付/時刻の選択は 4 つあります。

注: 継続的な日付選択は、「終了日/時刻」フィールドでのみ可能です。この選択を使用すると、終了日は、選択レコードよりも後の一番早い開始日に自動的に設定されます。

「日付/時刻の指定」

この選択は、手動で日付を指定します。

「連続」

この選択は、終了日を選択レコードよりも後の一番早い開始日に自動的に設定する自動日付計算を使用します。

注: 連続選択は、2 つのビジネス・ルール選択レコードの日付範囲が連続している場合に使用します。連続属性は、最初の選択レコードの終了日に設定されます。この属性が設定されている場合は、2 番目の選択レコードの開始日は、最初の選択レコードの終了日に設定されます。そのため、両方の日付を指定する必要はありません。

「開始日なし」または「終了日なし」

この選択では、選択内容に応じて、開始日や終了日は設定されません。

クローズ

この選択は、「終了日/時刻」メニューを閉じます。

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・ルールを別の目的で簡単に変更できるようになります。

変更するビジネス・ルールを選択して、**編集モード**になっているか確認してください。

ビジネス・ルール・レコードを分割するには、以下のようにします。

1. 変更するレコードの「分割」ボタンをクリックします。開始日が Jan 1 の新規レコードが作成されます。またフィールドが赤色に反転し、無効な日付/時刻フィールドが存在することを示すメッセージがメッセージ・フィールドに表示されます。
2. 新規レコードの「開始日/時刻」を選択します。元のレコードの「終了日/時刻」を「連続」から前のレコードの「開始日」に変更し、新規レコードの「終了日/時刻」を次のレコードの「開始日/時刻」に変更します。
3. 元のレコードに準じたレコードの「日付/時刻」を変更します。
4. 新規ルールのニーズに合うように「有効なルール・ロジック」を変更します。

ビジネス・ルール日付は変更されています。

デシジョン・テーブル

ビジネス・ルール・マネージャーを使用して、デシジョン・テーブルの値を変更します。

デシジョン・テーブルは、次の図に示すテーブル・フォーマットのビジネス・ルール・レコードです。

Purchase amount →	<= 1000 dollars	> 1000 dollars
Member Type ↓	Discount ↓	Discount ↓
Gold	8%	10%
Silver	3%	5%

図 2. デシジョン・テーブル

ビジネス・ルール・マネージャーは、テーブル形式のデシジョン・テーブルを示す Web インターフェースを提供しています。このデシジョン・テーブルでは、各変数は太字で示されます。条件およびアクションの方向が、図 2 のように矢印で示されます。デシジョン・テーブルは、対応するアクションに対する条件として機能します。条件に適合している場合、対応するアクションが実行されます。

デシジョン・テーブル・レコードの作成:

新規のデシジョン・テーブル・レコードを作成するには、最初に既存のデシジョン・テーブルをコピーする必要があります。


1. デシジョン・テーブルを選択してコピーします。
 - a. 選択したビジネス・ルール・レコードで「**コピー**」をクリックします。
 - b. 「**OK**」をクリックしてレコードをコピーします。「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「**編集**」画面が開きます。
 - c. 「**OK**」をクリックしてレコードをコピーします。
2. 「**タイトル・フィールド (Title field)**」に新規のビジネス・ルール・レコードの**名前**を入力します。
3. 新規レコードの**簡単な説明**を入力します。
4. 各条件の**値**を変更します。注: 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循環メッセージが表示されます。
5. **上矢印**をクリックして、ルールを正しいシーケンスに配置します。
6. 「**保管**」をクリックします。

レコードが一時的に保管されたことを知らせるメッセージがメッセージ・フィールドに表示されます。行った変更は「公開および復帰」ページで公開できます。詳しくは、97 ページの『ビジネス・ルールの公開』を参照してください。

関連タスク

96 ページの『ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードの削除』

デシジョン・テーブル・レコードの変更:

デシジョン・テーブルを編集するには、入力フィールドに値を直接入力するか、フィールドのリスト・ボックスから値を選択します。また、**ページ・アイコン**  をクリックすると表示される特別な「**編集**」メニューもあります。

注: 列または行の再配列は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		条件	アクション
下に追加	新規の条件値をセルの下に追加します (垂直方向に追加)	可能	
右に追加	新規の条件値をセルの右側に追加します (水平方向に追加)	可能	
テンプレートの変更	セルのテンプレートを変更します	可能	可能
上へ移動	条件値を 1 行上に移動させます (垂直方向に移動)	可能	
下へ移動	条件値を下に移動させます (垂直方向に移動)	可能	
左へ移動	条件値を左に移動させます (水平方向に移動)	可能	
右へ移動	条件値を右に移動させます (水平方向に移動)	可能	
削除	条件値を削除します	可能	
メニューを閉じる	メニューをクローズします	可能	可能

デシジョン・テーブルの値を変更するには、以下のようにします。

1. 「ナビゲーション」または「コンテンツ」セクションで、変更するテーブルを選択します。
2. 「編集」ボタンをクリックします。
3. 所要の値を変更します。

フィールドのリスト・ボックスを使用して値を直接入力するか、フィールドの隣のページ・アイコンをクリックします。

4. 「保管」ボタンをクリックします。


ローカル側でルールが変更されます。ルールをサーバーに公開する準備が整います。詳しくは、97 ページの『ビジネス・ルールの公開』を参照してください。

関連タスク

96 ページの『ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードの削除』

デシジョン・テーブル・テンプレートでの特別なアクション・メニューの使用:


「デシジョン・テーブル」ページには、テンプレート構造および値を変更するための特別なアクション・メニューがあります。

特別なアクション・メニューを表示するには、デシジョン・テーブル用の編集モードで、「特別なアクション (Special Actions)」アイコン () をクリックする必要があります。特別なアクション・アイコンを持つ変数はすべて変更可能なので、テンプレートに複数の変更を加えることができます。

「特別なアクション」メニューには以下のオプションがあります。

選択	アクション
下に追加	現在行の下に新規行を作成します
テンプレートの変更	セル値を変更することができます
下へ移動	変数を下の行に移動します
削除	変数を削除します
メニューを閉じる	メニューを閉じて編集モードに戻ります

デシジョン・テーブルのテンプレート値の変更:

特別なアクション・メニューを表示するには、デシジョン・テーブル用の編集モードで、特別なアクション・アイコン () をクリックする必要があります。変更可能な特別なアクション・アイコンの変数は、いずれも変更が可能で、テンプレートに複数の変更を加えることができます。

テンプレート値を変更するには、以下を行います。

1. テンプレート値を変更します。
 - a. メニューから「テンプレートの変更」を選択します。
 - b. そのテンプレート用の新規の値を入力します。
2. 「閉じる」をクリックします。

注: 1 つの値のみ変更する場合は、「アクション」列の「変更」をクリックすることもできます。

3. 「保管」をクリックします。

デシジョン・テーブル・テンプレートは変更され、公開する準備ができました。詳しくは、97 ページの『ビジネス・ルールの公開』を参照してください。

ルール・セット

ビジネス・ルール・マネージャーでルールを作成および変更します。

ルール・セットは、サーバーが順番に実行する if/then ルールのグループです。順序は、セットの先頭から下部になります。このため、ルールを変更または追加する場合、ルールが正しい順序で配置してあることを確認します。

ルール・セットはテンプレートを使用して作成します。テンプレートは、ルール・セットの仕組みを決定する構造体です。

ルール・セット・レコードの作成:

新規のルール・セットを作成するには、最初に既存のルール・セットをコピーする必要があります。

1. 選択したルール・セットをコピーします。
 - a. ルール・ロジック・レコードで「コピー」をクリックします。
 - b. 「OK」をクリックしてレコードをコピーします。「編集モード:Copy_of_TableName-Ruleset (Edit Mode:Copy_of_TableName-Ruleset)」というタイトルの「編集」画面が開きます。
 - c. 「OK」をクリックしてレコードをコピーします。
2. 「タイトル・フィールド (title field)」に新規のビジネス・ルール・レコードの**名前**を入力します。
3. 新規レコードの簡単な**説明**を入力します。
4. 各条件の**値**を変更します。

注: 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循環メッセージが表示されます。

5. 上または下矢印をクリックして、ルールを正しいシーケンスに配置します。
6. 「保管」をクリックします。

レコードが一時的に保管されたことを知らせるメッセージがメッセージ・フィールドに表示されます。行った変更は「公開および復帰」ページで公開できます。

ルール・セットを公開する準備ができました。詳しくは、97 ページの『ビジネス・ルールの公開』を参照してください。

関連タスク

96 ページの『ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードの削除』

テンプレートからルール・セット内のルールの作成:

ルール・セットに関連したルール・テンプレートを使用して、ルール・セット内に新規ルールを作成します。

編集モードでルール・セットを開きます。

テンプレートから新規のルールを作成するには、以下を行います。

1. 「**テンプレートからの新規ルール**」 ボタンをクリックします。テンプレートのリストがページの下部に表示されます。
2. 既存のテンプレートを使用して新規ルールを作成します。
 - a. 「名前」フィールドに新規ルールの**名前**を入力します。
 - b. ルール用の適切な**変数または制約**を入力、または選択します。
 - c. 「**追加**」をクリックします。
3. 上または下矢印をクリックして、ルールを適切な順序に配置します。

注: ルール・セットが正常に機能するためには、すべてのルールが正しい順序になければなりません。

4. 「保管」をクリックします。

ルール・セットを公開する準備ができました。詳しくは、97 ページの『ビジネス・ルールの公開』を参照してください。

関連タスク

『ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードの削除』

テンプレートを使用してルール・セット内のルールの変更:

ルール・セットに関連するテンプレートを使用して、そのルール・セット内のルールを変更します。

ルール・セットを開きます。

既存のテンプレートを使用してルールを編集するには、以下のようにします。

1. **ルール・セット**を編集します。
2. 「**編集**」ボタンをクリックします。
3. 既存の値を上書きするか、プルダウン・リストから適切な値を選択して、ルール内の値を変更します。
4. 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・セットが正常に機能するためには、すべてのルールが正しい順序になければなりません。

5. 「**保管**」をクリックします。

変更されたルール・セットを公開する準備ができました。詳しくは、97 ページの『ビジネス・ルールの公開』を参照してください。

関連タスク

『ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードの削除』

ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードの削除

編集モードで行う必要があります。

注: ビジネス・ルール・グループの各操作は、関連するビジネス・ルールを少なくとも 1 つ持つ必要があります。すべてのビジネス・ルールを削除しようとすると、エラーになります。

ビジネス・ルール、デシジョン・テーブル、またはルール・セット・レコードを削除する必要がある場合があります。レコードを削除するには、以下のようにします。

1. ビジネス・ルール・リストから「**レコード**」を選択します。
2. 「**削除**」ボタンをクリックします。レコードを削除しようとしていることを警告するメッセージが表示されます。
3. 「**OK**」をクリックします。
4. 「**保管**」ボタンをクリックします。

ローカル側でルールが変更されます。ルールをサーバーに公開する準備が整います。詳しくは、97 ページの『ビジネス・ルールの公開』を参照してください。

ビジネス・ルールの公開

ビジネス・ルール・グループの任意の部分を保管すると、変更はローカル側で保管されただけであり、データベースにその変更を保管する必要があるということが通知されます。サーバーはルール・ページ・レベルで変更を公開します。複数のルール・ページを単一トランザクションとして同時に公開できます。別のユーザーがルール・ロジックまたはルール・ブックを更新している間に、この同じルール・ロジックまたはルール・ブックを更新すると、エラーが発生する場合があります。詳しくは、98 ページの『アクセス競合エラー』を参照してください。

ビジネス・ルール・グループ・エディターからビジネス・ルールを公開するには、以下のようにします。

1. 「ナビゲーション」領域のある画面またはページで「公開および復帰」オプションをクリックします。「公開および復帰」ページが開きます。
2. 「コンテンツ」領域の左端の列にあるチェック・ボックスをクリックして、データベースに送信するページを選択します。選択したルール・ページがデータベースに書き込まれたことを示す画面をリフレッシュすると、選択したルール・ページは表示されなくなります。

公開した編集済みのルールは、アプリケーション・サーバー内にあるデータベースに保管されます。ビジネス・ルールはサーバーに即時にエクスポートできます。(詳しくは、77 ページの『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』を参照してください。)

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・マネージャーに問題が生じた場合には、ログイン・エラー、ログインの競合、アクセスの競合などについて検証します。

ログイン・エラー:

ログイン時に、ログイン・エラー・メッセージを受信します。

以下はログイン・エラー・メッセージです。

ログインを処理できません。ユーザー ID とパスワードを調べて、やり直してください。

このエラーは、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発生します。

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る場合にのみ発生します。

1. エラー・メッセージで「OK」をクリックします。

ログイン・ページに戻ります。

2. 有効なユーザー ID とパスワードを入力してください。

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か確認してください。

ユーザー ID とパスワードのスペルが正しいか確認してください。

ユーザー ID とパスワードが正しいかどうかシステム管理者に確認してください。

3. 「ログイン」 ボタンをクリックします。

ログイン・エラーが解決したら、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。エラーが解決できない場合は、システム管理者にお問い合わせください。

ログイン競合エラー:

このイベントは、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。

以下はログイン競合メッセージです。

現在、別のユーザーが同じユーザー ID でログインしています。次のオプション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のみ発生します。

選択できるオプションは 3 つあります。

- 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て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非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。

進行中の作業を継続して行う場合、このオプションを使用します。前のセッションの保管していない非公開のローカル変更はすべて失われません。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:

一人のユーザーがデータベース内のビジネス・ルールを更新している時に、別のユーザーが同時に同じルールを更新しようとするアクセス競合が発生します。

ローカル側の変更をデータベースに公開すると、このエラーが報告されます。

以下はアクセス競合エラーを訂正する操作です。

- 「ルール」 ページを公開します。

- エラーの原因となっているビジネス・ルールのソースを検出し、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、自身の変更は必要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継続する場合、データベースからエラーが発生したルール・ページを再ロードする必要があります。これは、エラーが発生したルール・ページのローカル変更が使用できなくなるためです。エラー状態でない他のルール・ページのローカル変更は引き続き使用できます。
- エラーが報告されたルールの「公開および復帰」ページで「再ロード」をクリックし、ルール・ページを再ロードします。

ビジネス・ルール・マネージャーを使用したアプリケーションのアンインストールおよび再インストール

ビジネス・ルールおよびセレクターによって、モジュールに柔軟性が付与されます。付与された柔軟性によって、モジュールのインストールまたは削除方法が変わります。これは、サーバーが中央リポジトリに、ビジネス・ルールおよびセレクターを保管しているためです。

ビジネス・ルール・マネージャーは、管理コンソールからアクセスし、ビジネス・ルールおよびその有効開始日付を動的に変更する機能を提供します。管理コンソールを使用して、リポジトリからファイル・システムにビジネス・ルール・データをエクスポートすることもできます。

ビジネス・ルール・マネージャーの動的機能だけでなく、アプリケーションのインストールと削除によっても、リポジトリ内のデータは影響を受けます。ビジネス・ルールおよびセレクターを使用するアプリケーションのインストールおよび削除について詳しくは、関連するトピックを参照してください。

関連概念

『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』

本トピックでは、ビジネス・ルールおよびセレクターを含んでいるモジュールをインストールまたは削除する場合に考慮すべき情報について説明します。

関連タスク

リポジトリからのビジネス・ルールおよびセレクター・データの除去

ビジネス・ルールやセレ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は、サーバーはリポジトリからこの成果物を除去しません。このタスクによって、リポジトリから不要のビジネス・ルールおよびセレクター成果物を除去します。

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:

本トピックでは、ビジネス・ルールおよびセレクターを含んでいるモジュールをインストールまたは削除する場合に考慮すべき情報について説明します。

ビジネス・ルールおよびセレクターによって、モジュールに柔軟性が付与されます。付与された柔軟性によって、モジュールのインストールまたは削除方法が変わります。これは、サーバーが中央リポジトリに、ビジネス・ルールおよびセレクターを保管しているためです。

ビジネス・ルールまたはセレクターの変更に関する考慮事項

ビジネス・ルールおよびセレクターは、影響があるモジュールを再組み立て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・ルールまたはセレクターを含むファイルにはコピーされません。ビジネス・ルールまたはセレクターを変更後、そのビジネス・ルールまたはセレクターをエクスポートし、開発環境に再度インポートしてください。ビジネス・ルールおよびセレクターのエクスポートに習熟していない場合は、このインフォメーション・センター内のこれらのタスクが説明されているトピックを参照してください。

ビジネス・ルールまたはセレクターが含まれているモジュールの置換に関する考慮事項

ビジネス・ルールまたはセレクターを含むモジュールを置換する場合は、サーバーはリポジトリのビジネス・ルールおよびセレクターのコピーを上書きします。モジュールの置換時に、動的に行ったすべての変更は失われます。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・ルールおよびセレ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ます。

あるモジュールによってインプリメントされているビジネス・ルールまたはセレ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・ルールまたはセレ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に、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・ルールまたはセレ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・ルールまたはセレクターを含むモジュールをサーバーから削除する時に、サーバーはビジネス・ルールおよびセレ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・ルールまたはセレクターが必要ないと判断した場合、リポジトリから除去します。『リポジトリからのビジネス・ルールおよびセレクター・データの除去』では、不要なビジネス・ルールまたはセレクターの消去方法について説明します。

データベース構成に関する考慮事項

ビジネス・ルールやセレクターの成果物用の動的な成果物リポジトリでは、ターゲットのネーム・スペース、名前、タイプのいずれかを使用して基本キーを生成します。DB2 for z/OS バージョン 7 では、基本キーのサイズが 255 バイトに制限されます。

DB2 for z/OS 版を使用するようにシステムを構成した場合は、名前を以下のように制限する必要があります。

- ターゲットのネーム・スペース = 170 バイト

- 名前の最大長 = 75 バイト
- タイプの最大長 = 10 バイト

注: DB2 for z/OS バージョン 8 では、この制限は適用されません。

『リポジトリからのビジネス・ルールおよびセクター・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は、サーバーはリポジトリからこの成果物を除去しません。このタスクによって、リポジトリから不要のビジネス・ルールおよびセクター成果物を除去します。

リポジトリからのビジネス・ルールおよびセクター・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は、サーバーはリポジトリからこの成果物を除去しません。このタスクによって、リポジトリから不要のビジネス・ルールおよびセクター成果物を除去します。

除去しようとしているビジネス・ルールまたはセクターを使用するアプリケーションのすべてのコピーを、すべてのサーバーから間違いなくアンインストールしてください。

ビジネス・ルールまたはセクター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は、サーバーはこれらの成果物をデータベース表に格納するため、ユーザーはアプリケーションを変更することなくその成果物をダイナミックに更新できます。また、これによって、他のサーバーはこの成果物を共有することができます。アプリケーションをアンインストールする場合は、サーバーはデータベース表からこ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みる際に、それらのアプリケーションは失敗します。

未使用成果物をデータベースから削除するには、その成果物を使用するすべてのアプリケーションをアンインストールした後で、手動で削除する必要があります。リポジトリのデータベース・プラットフォームで提供されるツールを使用して成果物を除去します。

1. データベースの場所を探します。

データベースの場所はデータベース・プラットフォームに依存します。

データベース・プラットフォーム
Cloudscape™

その他のデータベース

ロケーション

WASHOME¥cloudscape¥ /

databases¥RepositoryDB

サーバーのインストールおよび構成時に構成される場所により異なります。例えば、サーバーを自動的に構成し、デフォルト・データベース名を選択した場合、データベースの名前は WPSDB となります。

2. 行を削除する以下のデータベース表の場所を探します。

BYTESTORE

ビジネス・ルールおよびセレクター成果物を含むメイン・テーブル

BYTESTOREOVERFLOW

メイン・テーブル用のオーバーフロー・テーブル

APPTIMESTAMP

ビジネス・ルールおよびセレクター成果物を含むインストール済みアプリケーション

3. アプリケーション用の成果物を削除します。

ご使用のデータベース・プラットフォーム用のツールを使用して、所定のアプリケーション用のすべてのビジネス・ルールおよびセレクター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 表で、**APPNAME** 列の値がアプリケーション名と同じ行をすべて探します。
- b. 見つかったすべての行の基本キー列の値を記録します。BYTESTORE 表の基本キー列は、**ARTIFACTNS**、**ARTIFACTNAME**、および **ARTIFACTTYPE** です。
- c. BYTESTORE 表からステップ 3a で見つかった行を削除します。
- d. ステップ 3b で記録した基本キー値の各セットに対して、対応する列に同じ値を持つ BYTESTOREOVERFLOW 表の行を検索します。

注: 基本キー値の各セットに対して、BYTESTOREOVERFLOW 表に 0、1 つまたは複数の対応する行がある可能性があります。

- e. BYTESTOREOVERFLOW 表からステップ 3d で見つかった行を削除します。
- f. **APPNAME** 列がアプリケーションの名前に等しい APPTIMESTAMP 表の行を削除します。

データベース表から不要なビジネス・ルールおよびセレクター成果物を除去しました。

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・トポロジーを表示します。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、次に以下のボタンのいずれかを使用します。

表 8. 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始動すると、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"> • 開始済み: アプリケーションは、すべてのデプロイメント・ターゲットで始動しました。 • 部分開始: アプリケーションは、1 つ以上のデプロイメント・ターゲットでまだ開始中です。
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"> • 停止済み: アプリケーションは、すべてのデプロイメント・ターゲットで停止しました。 • 部分停止: アプリケーションは、1 つ以上のデプロイメント・ターゲットでまだ停止中です。
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) をサーバーに配置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、構成が保管された後にアプリケーション・モジュールがインストールされる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、サーバー上に配置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、その新規ファイルまたはモジュールが既存のファイルまたはモジュールと置き換えられます。同じ名前のファイルまたはモジュールがない場合は、配置されたアプリケーションに追加されます。</p>
ファイルの除去	<p>配置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。配置されたアプリケーションをバックアップしたり、そのインデント情報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>
DDL のエクスポート	<p>「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。</p>

アプリケーションの管理について詳しくは、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

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ルリングを制御します。管理コンソールの「Application Scheduler」パネルを使用して、これらのマイグレーション済みスケジューラー・エントリーを管理することもできます。

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

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

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

Application Scheduler を使用するには、それがインストールされていることを確認する必要があります。Application Scheduler の構成については、『応答ファイルでの作業 (Working with response files)』を参照してください。

初めにプロファイルを拡張する必要があります。

これはオプション・コンポーネントであり、WebSphere InterChange Server スケジュール・エントリーを WebSphere Process Server に移行するには、Application Scheduler を構成する必要があります。スタンドアロン・サーバー用にアプリケーション・サーバーをインストールするには、このステップに従ってください。

応答ファイル内の Application Scheduler の値を参照し、Application Scheduler を使用するために必要なプロパティ値を設定します。Application Scheduler に関するプロパティについては、『サンプル応答ファイル』を参照してください。

これでインストールは完了しました。

Application Scheduler が使用可能になります。

Application Scheduler へのアクセス

Application Scheduler には、Application Scheduler Mbean インターフェースを使用してプログラマチックに、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。

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

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

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

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

1. クラス `com.ibm.wbiserver.migration.ics.Parameters` にプロパティー `SOAP_HOSTNAME` および `SOAP_PORT` を設定します。このクラスは、`WAS_HOME¥lib` ディレクトリーの `migration-wbi-ics.jar` ファイルにあります。`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・ファイル内のユーザー 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・ファイル名は、ここに示した Parameter 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` への呼び出しをインプリメントするクラス `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、正しい `Mbean` を `name`、`type`、`server name` および `node name` に基づいて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 = "WBISchedulerMB1";
protected static final String TYPE_QUERY_CONSTANT = ",type=";
protected static final String WBI_SCHED_MB_TYPE = "WBIScheduler";
protected static final String SERVER_QUERY_CONSTANT = ",process=";
protected static final String NODE_QUERY_CONSTANT = ",node=";
```

```
serverName = "server1";
nodeName = "myNode";
```

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

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

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

ここに示した例は、`Scheduler Mbean` の `createSchedulerEntry` メソッドの起動です。最初のステップは `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することです。

```
try
{
//First we set up the Schedule entry

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

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

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

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

最後に、すべての Schedule エントリーを、readAllScheduleEntries メソッドを呼び出して追加したエントリーを含めて、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。

サーバーでこのタスクを実行するには、ユーザーは管理コンソールにいる必要があります。

このパネルを表示したり、既存のスケジューラー・イベントを表示するには、以下のステップに従ってください。

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

スケジューラー・イベントを編集したり、新規のスケジューラー・イベントを作成したり、既存のイベントを削除することができるようになりました。

スケジュールされているイベントの作成

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

新規のスケジュール済みイベントを作成するには、サーバー用の管理コンソールの Application Scheduler コレクション・パネルにユーザーが存在する必要があります。詳しくは、106 ページの『管理コンソールを使用したスケジューラー・エントリーの表示』を参照してください。

特定のニーズに適合するようにイベントを作成する必要がある場合があります。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: パネルの『*』が付いたフィールドは、必須フィールドです。

1. 「新規」をクリックします。「追加」パネルが開きます。
2. スケジュールされている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. 初期日付をフォーマット省略形 *month, dd, yyyy* で入力します。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 時間フォーマット *hh:mm* を使用して入力します。

注: **am** または **pm** のいずれであるか、およびタイム・ゾーンを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せん。

- e. 「アクション」を選択します。

オプション: Recurrence パラメーターに記入することもできます。

- 開始期間
 - スケジュール・エントリーを指定した時間で繰り返す必要があるかどうか。
 - 分、時間、日、月または年の 1 つ以上の時間。
 - 1 つ以上の月ごとの特定の週 (第 1、第 2、第 3、第 4 または最後) の特定の日 (日曜から土曜まで)。
 - 1 つ以上の月ごとの最後の日。
3. 「適用」または「OK」をクリックして、イベントを設定します。

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

Application Scheduler で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説明:

各イベントは、状況とアクションを持つ必要があります。

状況

状況フィールドは、モニター目的でイベントがどんな状況にあるかを示します。このテーブルは各状況をリストします。

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付、時刻、および間隔で実行されます。各後続の実行時刻は計算されます。
中断済み	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、ページすることはできます。
無効	タスクの状況が無効になる理由は通常、タスクがパージされているか、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。イベントが実行している非常に短い時間のイベントをモニターするからです。

アクション

各イベントは関連するアクションを持つ必要があります。アクションは、イベントへの対処法を指示します。イベントに使用可能なアクションが 2 つだけあります。

- 「**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。

スケジュールされているイベントの変更

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

スケジュール済みイベントを変更するには、サーバー用の管理コンソールの Application Scheduler コレクション・パネルにユーザーが存在する必要があります。詳しくは、106 ページの『管理コンソールを使用したスケジューラー・エントリーの表示』を参照してください。

ニーズに適合するようにイベントを変更する必要がある場合があります。イベントを変更するには、以下の手順に従います。

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

注: サーバー上のすべてのアプリケーションがリストされているので、既存のイベントの状況を変更するときに注意が必要です。サーバーで実行しているアプリケーションを停止することができます。

- グループ・アプリケーション
- 状況
- 初期日付 フォーマット省略形 *month, dd, yyyy*を持つ
- 初期時間 12 時間フォーマット *hh:mm* を使用する
- 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 Schedule Entry 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、新規イベントを新規 ID でスケジュールに入れます。

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

スケジュールされているイベントの削除

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

スケジュール済みイベントを削除するには、サーバー用の管理コンソールの Application Scheduler コレクション・パネルにユーザーが存在する必要があります。詳しくは、106 ページの『管理コンソールを使用したスケジューラー・エントリーの表示』を参照してください。

イベントが古くなると、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、以下のステップに従ってください。

1. 「選択」列から削除するイベントを選択します。
2. 「削除」をクリックします。
3. プロンプトで「OK」をクリックします。

イベントは削除されます。

WebSphere Process Server リソースの管理

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

アダプターの管理

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

WebSphere アダプター (JCA アダプターまたは J2C アダプターと呼ばれることもあります) は、J2EE Connector Architecture (JCA 1.5) に準拠しています。JCA は、EIS 接続に関する J2EE 標準です。EIS インポートおよび EIS エクスポートは、SCA アプリケーション・モジュールにモジュール外部でのこれらのサービスの一貫性のあるビューを提供します。これにより、コンポーネントは一貫性のある SCA プログラミング・モデルを使用して、さまざまな EIS システムと通信できます。

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

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

WebSphere Business Integration Adapter は、一連のソフトウェア、アプリケーション・プログラミング・インターフェース (API)、およびツールで構成され、これによって、アプリケーションは、統合ブローカー経由でビジネス・データを交換できます。それぞれのビジネス・アプリケーションには、ビジネス・インテグレーション・プロセスに参加するために、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し、成果物を生成して、WebSphere Process Server のためのソリューションをアセンブルできます。WebSphere Business Integration Adapter 用の操作コマンドは、WebSphere Process Server 管理コンソールの一部です。これらのアダプターおよび WebSphere Process Server での作業について詳しくは、WebSphere Business Integration Adapter インフォメーション・センターを参照してください。

WebSphere アダプターの操作

WebSphere アダプターは、管理コンソールを使用してインストールおよび管理できます。

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

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

1. WebSphere アダプターをインストールします。

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

オプション	説明
スタンドアロン・アダプター	管理コンソールを使用して、アダプターをインストールします。 注: スタンドアロンの WebSphere アダプターは、WebSphere Process Server v6.0 ではサポートされません。
組み込みアダプター	このアダプターは、アプリケーションのインストールの一環としてインストールされます。

2. アダプターを管理します。

3. J2EE 接続ファクトリーを構成します。

アダプターの概要

WebSphere Process Server で使用できるアダプターのタイプは、WebSphere Business Integration Adapter および WebSphere アダプターの 2 種類があります。これらのアダプターの機能の概要について説明します。

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

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

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

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

WebSphere Business Integration Adapter によって、複数のアプリケーションの通信が可能になり、システムが完全に統合されます。ユーザーは、Java 2 Enterprise

Edition (J2EE) Connector Architecture (JCA) バージョン 1.5 に完全に準拠する WebSphere アダプターを使用することが推奨されています。WebSphere Business Integration Adapter は、JCA に準拠せず、アプリケーション・サーバーの外部で稼働します。

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

アダプターを使用したアプリケーションの開発についての詳細は、WebSphere Integration Developer インフォメーション・センターを参照してください。

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

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

WebSphere アダプターの概要: WebSphere アダプターは、Java 2 Enterprise Edition (J2EE) Connector Architecture (JCA) バージョン 1.5 をインプリメントします。これらのアダプターは、WebSphere アダプター、または Resource Adapter と呼ばれます。これらのアダプターは、Enterprise Information systems (EIS) および WebSphere Process Server がサポートする J2EE コンポーネント間の管理対象双方向接続を使用可能にします。

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

- 任意の J2EE 準拠アプリケーション・サーバーにプラグインする。
- そのサーバー上で EIS とともに稼働中のアプリケーションに接続する。
- J2EE アプリケーションおよび EIS 間のデータ交換を使用可能にする。

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

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

それぞれの WebSphere アダプターは、以下で構成されています。

- **ファウンデーション・クラス** このクラスは、WebSphere Process Server が、J2EE アプリケーションおよびすべての WebSphere アダプター間の相互作用を管理するために使用する、一連の汎用契約をインプリメントします。これらのサービス品質、およびシステム契約とも呼ばれるライフ・サイクル管理契約は、サービス・

プロバイダー・インターフェース (SPI) を定義します。例えば、システム契約は、セキュリティー信任状管理、接続プール、およびトランザクション管理パラメーターを指定します。

- **EIS サブクラス** これらの汎用で EIS 固有のサブクラスは、Common Client Interface (CCI) および EIS API 契約を定義します。例えば、活動化および接続仕様によって、WebSphere Process Server は、WebSphere Adapter の入力イベントおよび出力イベントを管理できます。
- **Enterprise Metadata Discovery** このユーティリティは、EIS をイントロスペクトし、サービス・データ・オブジェクト (SDO) および、標準 Enterprise Application Archive (EAR) ファイルにコンパイルされているその他の成果物を生成します。

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

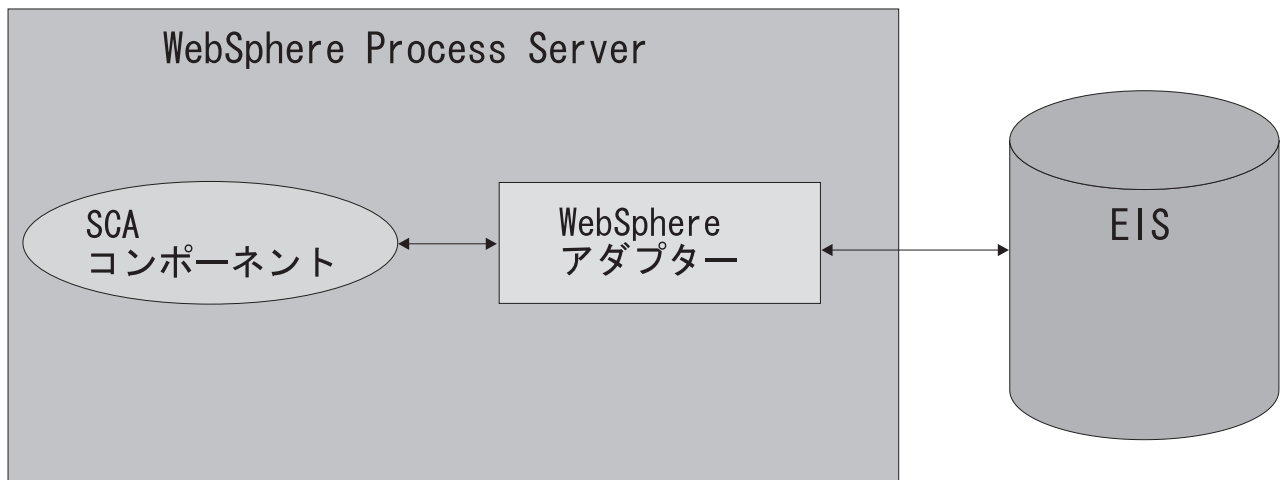


図 3. WebSphere アダプターの単純化した図解

WebSphere アダプターおよび WebSphere Business Integration Adapter 間の差異
: WebSphere アダプターおよび WebSphere Business Integration Adapter 間には、いくつかの差異があります。これらの区別は、アプリケーションの開発に最も重要です。アプリケーションを実行中のサーバーに配置する場合、使用中のアダプターの性質は、実行される必要があるいくつかのステップに影響を与えます。

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 3 および図 4 では、WebSphere Process Server と EIS 間の、2 つのアダプター・タイプの通信について詳しく説明します。

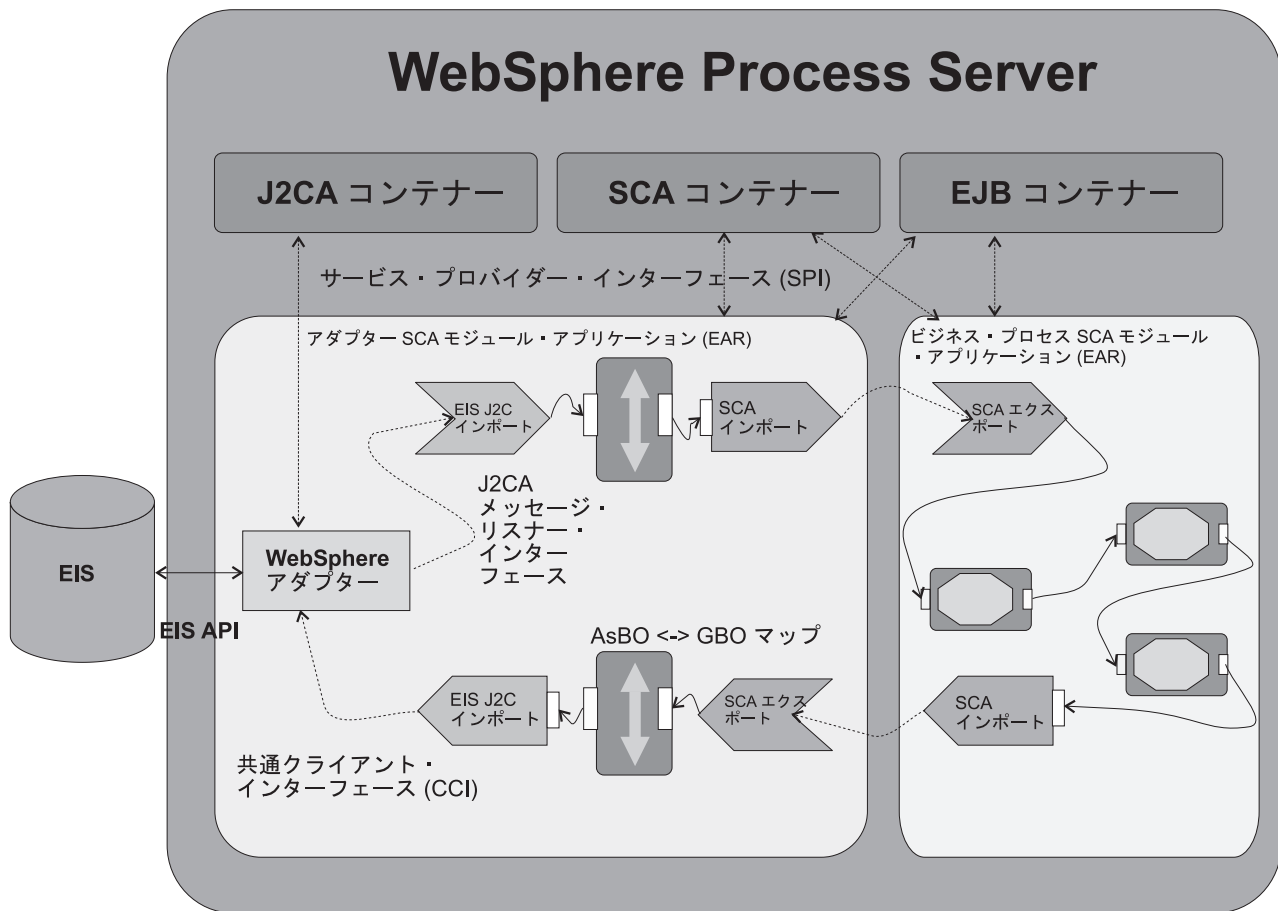


図 4. WebSphere アダプターの詳細な図式

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

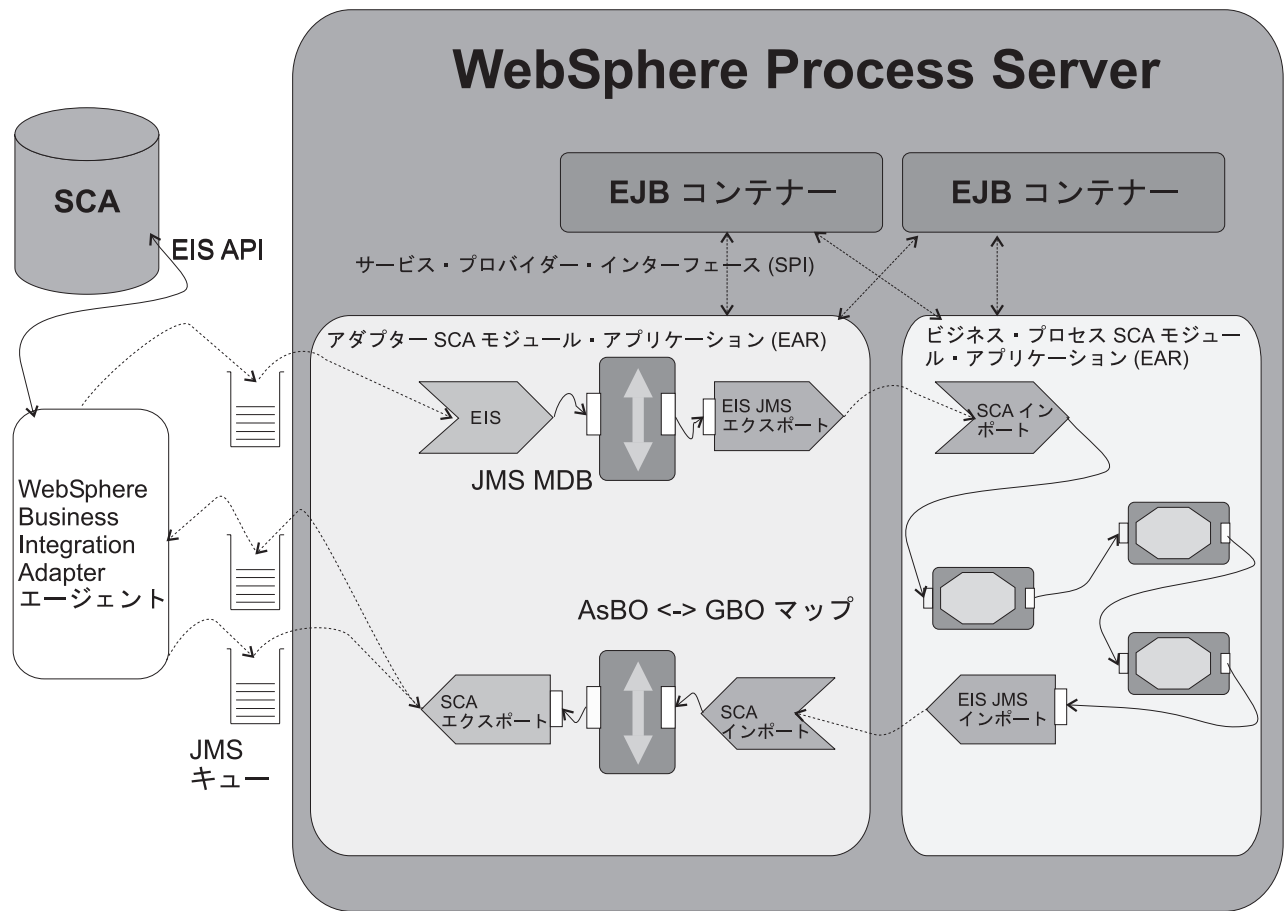


図 5. WebSphere Business Integration Adapter の詳細な図式

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

2 つのアダプター・タイプ間の差異は、以下のとおりです。

表 9. WebSphere アダプターおよび WebSphere Business Integration Adapter 間の差異

フィーチャー	WebSphere アダプター	WebSphere Business Integration Adapters
JCA Compliance	JCA 完全準拠 (バージョン 1.5)。	JCA 非準拠。
Connectivity Manager	標準 JCA 契約を利用し、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 からのイベントを検索します。	pollFor Events メソッドを使用してイベント通知を管理します。

表9. WebSphere アダプターおよび WebSphere Business Integration Adapter 間の差異 (続き)

フィーチャー	WebSphere アダプター	WebSphere Business Integration Adapters
Request Processing	クライアントは、いくつかの相互作用契約の 1 つを直接起動し、EIS のデータを照会または変更します。	組み込みサーバーおよび WebSphere Adapter Framework を利用して、処理要求を開始し、支援します。
Data Models	Enterprise Metadata Discovery (EMD) ユーティリティを使用し、EIS を解析し、Service Data Object (SDO) およびその他有用な成果物を開発します。EMD は、WebSphere アダプターのインプリメンテーションの一部です。	分離した Object Discovery Agent (ODA) を使用して、EIS をイントロスペクトし、ビジネス・オブジェクト定義スキーマを生成します。
Integration	WebSphere Process Server で実行します。	アプリケーション・サーバーの外側にあります。サーバーまたは統合ブローカーは、Java Message Service (JMS) トランSPORT・レイヤーを介してアダプターと通信します。

WebSphere アダプターは推奨ツールです。

WebSphere アダプターを使用する利点: WebSphere アダプターは、WebSphere Business Integration Adapter にいくつかの利点を提供します。以下はこれらの利点の要約です。

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

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

WebSphere アダプターが組み込まれたアプリケーションのインストール

アプリケーションに WebSphere アダプターが組み込まれて作成されている場合は、アダプターはアプリケーションでデプロイされます。アダプターを別個にインストールする必要はありません。ここでは、組み込みアダプター付きアプリケーションのインストール手順を説明します。

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

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

注: サーバーに RAR ファイルをインストールする場合、WebSphere Application Server はコネクター・モジュールのマニフェスト (MANIFEST.MF) を探します。最初、RAR ファイルの connectorModule.jar ファイルで探し、_connectorModule.jar ファイルからマニフェストをロードします。クラスパス・エントリーが connectorModule.jar ファイルからのマニフェストにある場合は、RAR はそのクラスパスを使用します。インストール済みコネクター・モジュールが必要なクラスおよびリソースを見付けることを確認するには、コンソールを使用して、RAR の「クラスパス」設定を確認します。詳しくは、Resource Adapter の設定および WebSphere 関連のリソース・アダプターの設定を参照してください。

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

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

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

ネイティブ・パス要素を含むアダプターをインストールする場合、以下の点を考慮する必要があります。複数のネイティブ・パス要素があり、ネイティブ・ライブラリーの 1 つ (ネイティブ・ライブラリー A) が別のライブラリー (ネイティブ・ライブラリー B) に依存している場合、ネイティブ・ライブラリー B をシステム・ディレクトリーにコピーする必要があります。ほとんどの UNIX システムに制限があるため、ネイティブ・ライブラリーをロードしようとする際に現行ディレクトリーは調べません。

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

注:

所定のネイティブ・ライブラリーは、Java 仮想マシン (JVM) のインスタンスごとに 1 度だけロードできます。アプリケーションごとに独自のクラス・ローダーがあるため、組み込み RAR ファイルを持つ異なるアプリケーション同士が同じネイティブ・ライブラリーを使用することはできません。2 番目のアプリケーションは、ライブラリーをロードしようとする例外を受け取ります。

アプリケーション・サーバーにデプロイされているアプリケーションが、ネイティブ・パス・エレメントを含む組み込み RAR ファイルを使用する場合、未解決のトランザクションがない状態で、アプリケーション・サーバーが完全にシャットダウンしたことを必ず確認する必要があります。アプリケーション・サーバーが完全にシャットダウンしない場合は、サーバーの再始動時にリカバリーを実行し、必要な RAR ファイルおよびネイティブ・ライブラリーをロードします。リカバリーが完了したら、アプリケーションに関する作業は行わないようにしてください。サーバーをシャットダウンして、再始動します。この再始動時にアプリケーション・サーバーによってこれ以上リカバリーが行われることはなく、通常のアプリケーション処理を続行できます。

WebSphere アダプター:

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

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

アプリケーション・サーバーのベンダーは、J2EE コネクタ・アーキテクチャ (JCA) をサポートするように一度そのシステムを拡張すれば、その後は複数の EIS へのシームレスな接続を保証されます。同様に、EIS のベンダーは、コネクタ・アーキテクチャをサポートする、アプリケーション・サーバーにプラグインする機能を備えた 1 つの標準 WebSphere アダプターを提供します。

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

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

IBM は、WebSphere Process Server パッケージとは別に、エンタープライズ・システム向けにリソース・アダプターを用意しています。これには顧客情報管理システム (CICS)、Host On-Demand (HOD)、情報管理システム (IMS)、および Systems, Applications, and Products (SAP) R/3 が含まれます (ただし限定はされていません)。

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

WebSphere アダプターのデプロイメントの考慮事項:

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

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

- スタンドアロン - アダプターはノード・レベルでインストールされ、特定のアプリケーションには関連付けられません。

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

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

組み込み WebSphere アダプター:

- RAR ファイルは、SCA モジュール内に (EIS インポートまたはエクスポートを使用して) アプリケーション・スコープすることができます。
- RAR ファイルは、非 SCA モジュール内にアプリケーション・スコープできます。EIS インポートおよびエクスポートを含むアプリケーション自体は、分離した SCA モジュールです。

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

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

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

スタンドアロン WebSphere アダプターのインストール:

WebSphere アダプターは、アプリケーションに組み込む必要があります。スタンドアロン WebSphere アダプターは、WebSphere Process Server v6.0 ではサポートされていません。これらの説明は参照のみを目的としています。スタンドアロン WebSphere アダプターを使用する場合、ここで説明する手順に従い、WebSphere アダプターをインストールする必要があります。また、組み込みアダプターを使用することもできます。組み込みアダプターは、関連アプリケーションをインストールすると自動的にインストールされます。

アダプターをインストールする前に、データベースを構成する必要があります。

このタスクを実行するには、管理コンソールにアクセスがあり、必要なセキュリティ・ロールの一部である必要があります。

1. 「RAR のインストール」ダイアログ・ウィンドウを開きます。

管理コンソールでの作業:

- a. 「リソース」を展開します。
- b. 「リソース・アダプター」をクリックします。
- c. このリソース・アダプターを定義する有効範囲を選択します。(この有効範囲は、ご使用の接続ファクトリーの有効範囲になります。) セル、ノード、クラスター、またはサーバーを選択することができます。
- d. 「RAR のインストール」をクリックします。

JCA コネクターのインストール、および JCA コネクター用の WebSphere アダプターの作成が可能なウィンドウが開きます。「新規」ボタンを使用することも

できますが、「新規」ボタンによって作成されるのは、新規リソース・アダプターだけです (JCA コネクターがシステムにインストール済みである必要があります)。

注: このダイアログを使用して RAR ファイルをインストールすると、「リソース・アダプター」ページで定義する有効範囲は、RAR ファイルのインストール場所では無効になります。RAR ファイルは、ノード・レベルでしかインストールできません。ファイルをインストールできるノードは、「RAR のインストール」ページの有効範囲により決定されます。(「リソース・アダプター」ページで設定する有効範囲により、新規リソース・アダプターの有効範囲が決まります。新規リソース・アダプターは、サーバー、ノード、またはセルの各レベルでインストールすることができます。)

2. RAR ファイルをインストールします。

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

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

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

WebSphere アダプター・モジュール・アプリケーションは、特定の条件でクラスターのメンバーとしてクローンすることができます。

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

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

クラスターは、Network Deployment 環境内のアプリケーションに、スケーラビリティおよび可用性を提供するために使用します。

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

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

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

WebSphere Business Integration Adapter モジュール・アプリケーションは、特定の条件でクラスターのメンバーとしてクローンすることができます。

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

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

クラスターは、Network Deployment 環境内のアプリケーションに、スケーラビリティおよび可用性を提供するために使用します。

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

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

管理コンソールを使用した WebSphere アダプターの管理

管理コンソールを使用すると、WebSphere アダプターを管理および構成できます。

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

WebSphere アダプターをインストールしている場合、管理コンソールを使用してそれを管理できます。

1. 管理するアダプター用のリソース・アダプター・コンソールを開きます。

管理コンソールでの作業:

- a. 「リソース」を展開します。
 - b. 「リソース・アダプター」をクリックします。
 - c. 管理する WebSphere アダプターを選択します。
2. 「一般プロパティ」または「追加プロパティ」の見出し下の必要なプロパティを変更します。

変更が完了した後、「適用」ボタンをクリックします。

変更は、マスター構成に保管するまでローカル構成にのみに適用されます。

注: 変更が有効になるためには、サーバーの再始動が必要となる場合があります。

WebSphere アダプター管理コンソールの設定:

ここでは、WebSphere アダプターの管理コンソールの設定およびそれらのデフォルト値について説明します。

目的

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

管理コンソールを表示するには、次のようにします。

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

各種のフィールドを以下に説明します。

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

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

プロパティ
データ型

値
ストリング

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

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

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

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

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

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

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

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

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

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

接続ファクトリーの管理

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

接続ファクトリーは、接続インスタンスにアクセスするために、アプリケーション・コンポーネントによって使用されます。次にコンポーネントは、この接続インスタンスを使用して、基盤となる EIS に接続されます。

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

管理コンソールにおける J2EE コネクター接続ファクトリーの構成:

接続ファクトリーは、アプリケーションおよびエンタープライズ情報システム (EIS) 間の通信の仲介で使われます。管理コンソールを接続ファクトリーを管理するために使用することができます。

スタンドアロン・リソース・アダプターを持っている場合、このタスクを実行する必要があります。組み込み WebSphere アダプターは少し異なる処理を必要とします。ユーザーは、管理コンソールを編集するセキュリティー権限も持つ必要があります。

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

1. 管理するリソース・アダプターを選択します。

管理コンソールの最上位から、以下のステップを実行します。

- a. 「リソース」を展開します。
- b. 「リソース・アダプター」を選択します。
- c. 管理する WebSphere アダプターを選択します。

2. このアダプターの新規接続ファクトリーを作成します。

以下のように、新規接続ファクトリーを作成します。

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

別名が使用できない場合、あるいは別のものを定義しようとする場合は、以下を実行します。

- 1) 「適用」をクリックします。
 - 2) 「関連項目」の下にある「J2C 認証データ・エントリ」をクリックします。
 - 3) 「新規」をクリックします。
 - 4) 一般的なプロパティを指定します。
 - 5) 「OK」をクリックします。
- f. 「OK」をクリックします。

3. 新規に作成した接続ファクトリーの接続プール・プロパティを変更し、接続プール・マネージャーの動作を最適化します。

以下のように、接続プール値を変更します。

- a. 新規の接続ファクトリーを選択します。
- b. 「追加プロパティ」の見出しの下の「接続プール・プロパティ」を選択します。

- c. プロパティ名をクリックして、必要な値をすべて変更します。
- d. 「OK」をクリックします。
- e. 「追加プロパティ」の下の「カスタム・プロパティ」をクリックします。
- f. 任意のプロパティ名をクリックして値を変更します。
- g. 「OK」をクリックします。

接続プール用の一般プロパティの設定:

管理コンソールを使用して接続プールに一般設定を割り当てることができます。

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

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

1. 管理コンソールの「接続プール・プロパティ」パネルを開きます。

管理コンソールの最上位から、以下を実行します。

- a. 「リソース」を展開します。
 - b. 「リソース・アダプター」をクリックします。
 - c. 管理する接続ファクトリーを持っている WebSphere アダプターをクリックします。
 - d. 「追加プロパティ」の見出しの下の「J2C 接続ファクトリー」をクリックします。
 - e. 管理する接続ファクトリーを選択します。
 - f. 「追加プロパティ」の見出しの下の「接続プール・プロパティ」をクリックします。
2. タイムアウト、最大および最小接続およびページ・ポリシーなどの一般接続プール・プロパティをこのパネルで変更することができます。デフォルト値が提供されています。
 3. 希望するプロパティを変更後に、「OK」をクリックします。

接続プール設定:

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

目的

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

「構成」タブ

接続タイムアウト

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

この値は、通常、特定接続プールの接続数が最大値に達したために、空きプール内に使用可能な接続がなく、また新規接続が作成できない時に、接続要求が待機する秒数を示します。例えば、接続タイムアウトが 300 に設定されていて、最大数の接続がすべて使用中の場合は、プール・マネージャーは、物理接続が使用可能になるのを 300 秒待機します。物理接続がこの時間内に使用可能にならない場合は、プール・マネージャーが `ConnectionWaitTimeout` 例外を開始します。`getConnection()` メソッドを再試行しても、通常は効果がありません。待機時間を長くする必要がある場合、接続タイムアウトの設定値を増加させる必要があります。`ConnectionWaitTimeout` 例外がアプリケーションによってキャッチされた場合、管理者は予期されるアプリケーションの接続プールの使用法を確認し、それに従って接続プールとデータベースを調整する必要があります。

接続タイムアウトが 0 に設定されている場合、プール・マネージャーは、接続が使用可能になるまで必要なだけ待機します。これは、アプリケーションがランザクションを完了して接続をプールに戻す時、または接続数が最大接続値を下回り、新規物理接続の作成が許可された時に起こります。

最大接続数が 0 に設定されていると、物理接続の数が無制限になるので、接続タイムアウト値は無視されます。

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

最大接続数

このプールに構築できる物理接続の最大数を指定します。

これらは、バックエンド・リソースへの物理接続です。一度この数値に到達すると、新規の物理接続は作成されず、要求側は、現在使用中の物理接続がプールに戻されるか、`ConnectionWaitTimeout` が発行されるまで待機します。

例えば、最大接続数が 5 に設定されていて、5 つの物理接続が使用中の場合、プール・マネージャーは、接続タイムアウトに指定された時間、物理接続が空くのを待機します。

最大接続数が 0 に設定されていると、接続プールは無限に増加できます。これはまた、接続タイムアウト値が無視されるという副次作用を持ちます。

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

Tivoli Performance Viewer を使用して、最適なプール内の接続数を見つけます。同時待機数が 0 より大きいにもかかわらず、CPU 負荷が 100% 近くに達しない場合は、接続プール・サイズを大きくすることを検討します。通常のワークロード下で使用パーセントが常に低い場合は、プール内の接続数を減らすことを検討します。

プロパティ	値
データ型	整数
デフォルト	10
範囲	0 から最大の整数

最小接続数

維持する物理接続の最小数を指定します。

接続プールのサイズが最小接続プール・サイズと同じかまたはそれより小さい場合、「未使用タイムアウト」スレッドは物理接続を破棄しません。しかし、プールは単独では接続を作成して最小接続プール・サイズを維持することを保障しません。また、経過タイムアウトの値を設定すると、最小プール・サイズ設定にかかわらず、経過時間の有効期限が切れた接続は破棄されます。

例えば、最小接続数の値が 3 に設定されていて、1 つの物理接続が作成される場合、その接続が「未使用タイムアウト」スレッドによって廃棄されることはありません。同じトークンによって、スレッドが自動的に 2 つの追加の物理接続を作成し、最小接続数の設定値に達することはありません。

プロパティ	値
データ型	整数
デフォルト	1
範囲	0 から最大の整数

リープ時間

プール維持スレッドの実行とその次の実行との間隔 (秒) を指定します。

例えば、リープ時間が 60 に設定されていると、プール維持スレッドは 60 秒ごとに実行されます。リープ時間の間隔は、未使用タイムアウトと経過タイムアウトの設定値の精度に影響を与えます。間隔が短いほど精度は高まります。プール維持スレッドが使用可能である場合は、リープ時間値を、未使用タイムアウトや経過タイムアウトの値より少なく設定してください。プール維持スレッドが実行されると、未使用タイムアウトで指定された値より長い時間使用されていない接続をすべて廃棄します。廃棄は、最小接続数で指定された接続数になるまで行われます。プール維持スレッドは、経過タイムアウトで指定された時間値よりも長い間アクティブである接続も、すべて廃棄します。

リープ時間間隔も、パフォーマンスに影響を与えます。間隔が短いということは、プール維持スレッドの実行回数が増え、パフォーマンスが低下することを意味します。

プール維持スレッドを使用不可にするには、リープ時間を 0 に設定するか、または未使用タイムアウトと経過タイムアウトの両方を 0 に設定します。プール維持スレッドを使用不可にするためには、リープ時間を 0 に設定する方法をお勧めします。この場合、未使用タイムアウトと経過タイムアウトは無視されます。ただし、未使用タイムアウトと経過タイムアウトが 0 に設定されている場合は、プール維持スレッドは実行されますが、タイムアウト値が非ゼロであるためにタイムアウトになる物理接続だけは廃棄されます。

プロパティ	値
データ型	整数
単位	秒

プロパティ	値
デフォルト	180
範囲	0 から最大の整数

未使用タイムアウト

未使用またはアイドルの接続が廃棄されるまでの時間 (秒) を指定します。

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

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

経過タイムアウト

物理接続が廃棄されるまでの時間 (秒) を指定します。

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

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

ページ・ポリシー

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

有効な値は以下のとおりです。

- EntirePool: プール内の接続は、すべて失効とマークされます。使用されていない接続は、直ちに閉じられます。使用中の接続は閉じられ、その接続に次回操作が行われたときに不整合接続の例外が発行されます。アプリケーションからの後続の `getConnection()` 要求の結果、開かれているデータベースへの新規接続が発生します。このページ・ポリシーを使用しているときは、プール内の一部の接続が、失効していない場合に不必要に閉じられる可能性があります。

ります。ただし、これはめったに起こりません。たいていの場合は、ページ・ポリシーを EntirePool に設定するのが最善の選択です。

- **FailingConnectionOnly**: 不整合接続例外の発生原因である接続だけがクローズされます。この設定により、有効な接続が不必要に閉じられなくなる一方で、アプリケーション・パースペクティブからのリカバリーがより複雑になります。現在失敗している接続だけが閉じられるので、アプリケーションからの次の `getConnection()` 要求により、プールからやはり失効した接続が戻され、結果的により多くの失効した接続の例外が発生する可能性が極めて高くなります。接続の事前テスト機能は、無効なプール済み接続からアプリケーションを分離させます。データベースなどのバックエンド・リソースが停止した場合、無効なプール済み接続が、空きプールに存在していることがあります。これは、ページ・ポリシーが `failingConnectionOnly` の場合、特に発生します。この場合には、障害のある接続は、そのプールから除去されます。障害によっては、プール内の残りの接続が、無効になっている場合があります。

プロパティ	値
データ型	ストリング
デフォルト	EntirePool

接続プール用の拡張プロパティの設定:

管理コンソールを使用して接続プールに拡張設定を割り当てることができます。

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

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

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

管理コンソールの最上位から、以下を実行します。

- a. 「リソース」を展開します。
- b. 「リソース・アダプター」をクリックします。
- c. 管理する接続ファクトリーを持っている WebSphere アダプターをクリックします。
- d. 「追加プロパティ」の見出しの下の「J2C 接続ファクトリー」をクリックします。
- e. 管理する接続ファクトリーを選択します。
- f. 「追加プロパティ」の見出しの下の「接続プール・プロパティ」をクリックします。
- g. 「追加プロパティ」の見出しの下の「拡張接続プール・プロパティ」をクリックします。

2. 分割化、接続過負荷からの保護および応答していない接続の取り扱いなどの拡張接続プール設定。デフォルト値が提供されていますが、これらのプロパティの調整は、改善されたパフォーマンスをもたらす可能性があり、お勧めします。

3. 希望するすべてのプロパティを変更後に、「OK」をクリックします。

接続プール拡張設定:

管理コンソールの「拡張接続プール設定 (Advanced Connection Pool Settings)」パネル上の接続プール向けの拡張設定を変更できます。

目的

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

「構成」タブ

共用区画の数

各共用プールに作成されている区画の数を指定します。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

空きプール区画の数

各空きプールに作成されている区画の数を指定します。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

空きプール分配テーブル・サイズ

空きプール分配テーブル・サイズは、ハッシュ・テーブル内の Subject および CRI ハッシュ値の分配を改善するために使用されて、一致する空き接続の高速検索に対する衝突を最小化します。

クリデンシャルが異なる着信要求が多数ある場合、この値によってそのユーザーの接続用の空きプール分配を検索できます。通常、リソースにアクセスするときには多数の異なるクリデンシャルのあるインストールの場合、値を大きくします。リソースに対するすべての着信要求に同一のクリデンシャルを使用する場合は、小さい値 (1) を使用します。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

サージしきい値

サージ保護が活動化される前に作成された接続の数を指定します。

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

サージしきい値プロパティでは、サージ保護が活動化される前に作成される接続の数を指定します。指定した接続数に到達した後、サージ・モードに入ります。

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

例えば、以下のように設定するとします。

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

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

サージしきい値が > -1 であり、サージ作成間隔が > 0 である場合、サージ接続サポートが開始されます。サージしきい値プロパティのデフォルト値は -1 です。つまり、このプロパティはオフになっています。

wsadmin 例

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

プロパティ	値
データ型	整数
デフォルト値	-1
範囲	-1 から最大の整数

サージ作成間隔

サージ保護モード内での、接続作成間の時間の量を指定します。

「サージしきい値」プロパティで指定された数の接続が行われると、新規接続に対する各要求は、サージ作成間隔で作成されるのを待たなければなりません。このプロパティのデフォルト値は 20 です。つまり、各接続が作成されるまでに少なくとも 20 秒経過する必要があります。このプロパティの有効な値は、任意の正整数です。

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

滞留タイマー時間

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

wsadmin スクリプト・ツールを使用して滞留時間、滞留タイマー時間、または滞留しきい値プロパティを変更することに失敗した場合は、**IllegalState** 例外が発生します。この要求の間、プールにはアクティブな要求またはアクティブな接続を含めることができません。滞留接続サポートを開始するには、3 つのすべての滞留プロパティ値が 0 よりも大きく、最大接続数が 0 よりも大きくなっている必要があります。

また、滞留タイマー時間が設定されている場合、この値は滞留時間の値よりも小さくなっている必要があります。滞留タイマー時間は、滞留時間の 4 分の 1 から 6 分の 1 の値にすることをお勧めします。これにより、接続プールは、接続が滞留であると宣言されるまで、滞留接続を 4 回から 6 回チェックします。これにより、誤って正しい結果が出る可能性が削減されます。

wsadmin 例

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

プロパティ	値
データ型	整数
デフォルト値	5
範囲	0 から最大の整数

滞留時間

滞留接続は、接続プールに応答したり戻されたりすることのないアクティブな接続です。プールが滞留状態 (滞留しきい値に達した) になると、プールの滞留が解除されるまで、すべての新規接続要求にリソース例外が出されます。「滞留時間」プロパティは、単一アクティブ接続が、滞留していると考えられるまでの、バックエンド・リソースで使用される間隔 (秒単位) です。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

滞留しきい値

滞留接続は、接続プールに応答したり戻されたりすることのないアクティブな接続です。プールが滞留状態 (滞留しきい値に達した) になると、プールの滞留が解除されるまで、すべての新規接続要求にリソース例外が出されます。アプリケーションは、この例外を明示的にキャッチして処理を続けることができます。滞留接続数がしきい値を超えると、プールは滞留接続の定期的なチェックを続けます。滞留接続数が滞留しきい値よりも小さくなると、プールは、定期チェック時

にこのことを検出し、プールでの要求処理の再開が可能となります。滞留しきい値は、プールが滞留モードになるために、滞留していると見なされる必要がある接続の数です。

プロパティ	値
データ型	整数
デフォルト値	0
範囲	0 から最大の整数

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

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

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

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

管理コンソールの最上位から、以下のステップを実行します。

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

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

- d. 「次へ」をクリックします。
2. J2C 接続ファクトリーにマップすることを示すアプリケーションをインストールします。

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

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

管理コンソールの最上位から、以下のステップを実行します。

- a. 「アプリケーション」を展開します。
- b. 「エンタープライズ・アプリケーション」を選択します。
- c. 新規にインストールされたアプリケーションを選択します。
- d. 「関連項目」の下にある「コネクター・モジュール」をクリックします。
- e. RAR ファイルを選択します。
- f. 「追加プロパティ」の下にある「リソース・アダプター」を選択します。
- g. 「追加プロパティ」の下にある「J2C 接続ファクトリー」をクリックします。

- h. 「**新規**」をクリックします。
- i. すべての必要な一般プロパティを指定します。
- j. **オプション:** XA リカバリー用の認証別名を指定するか、コンポーネント管理の認証別名を使用します。 XA トランザクションをサポートするリソース用にのみこのフィールドは表示されます。
- k. **オプション:** コンポーネント管理の認証別名を選択します。

希望の別名が使用できない場合、あるいは別のものを定義しようとする場合は、以下を実行します。

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

アプリケーションのコンポーネントが空の引数 `getConnection()` メソッドを持ち、アプリケーションまたは接続ファクトリーごとの認証をリソース参照で指定したこの接続ファクトリーから接続を取得する場合は、別名を選択または定義します。

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

「**J2C 接続ファクトリー・コレクション**」パネルから、以下を実行します。

- a. 作成した接続ファクトリーを選択します。
 - b. 「**接続プール設定 (Connection Pool settings)**」パネルを開きます。「**追加プロパティ**」の見出しの下の「**接続プール・プロパティ**」をクリックします。
 - c. プロパティ名をクリックして、すべてのプロパティの値を変更します。
 - d. 接続プール設定へのこれらの変更を確認します。「**OK**」をクリックします。
5. **オプション:** すべてのカスタム・プロパティの変更

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

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

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

- c. 「**保管**」をクリックします。

J2C 接続ファクトリー・コレクション:

「J2C 接続ファクトリー・コレクション (J2C connection Factory Collection)」パネルでは、接続ファクトリーの選択可能なリストが提供されます。

目的

このパネルを使用して、接続構成値の 1 セットを表す接続ファクトリーを選択します。

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

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

名前

接続ファクトリー表示名のリストを指定します。

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

JNDI 名

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

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

説明

この接続ファクトリーを説明するテキストを指定します。

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

カテゴリー

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

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

J2C 接続ファクトリーの設定:

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

目的

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

「構成」タブ

接続ファクトリー・インターフェース

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

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

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

XA リカバリーの認証別名

このオプション・フィールドは、XA リカバリー処理時に使用される認証別名の指定に使用されます。

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

コンポーネント管理認証別名の使用

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

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

指定:

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

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

名前

接続ファクトリー表示名のリストを指定します。

これは必要プロパティです。

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

JNDI 名

この接続ファクトリーの JNDI 名を指定します。

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

この値を設定した後、値を保管してサーバーを再起動します。この文字列は、dumpNameSpace ツールを実行すると表示されます。これは必要プロパティです。JNDI 名を指定しない場合は、「名前」フィールドを使用してデフォルトで入力されます。

プロパティ データ型 デフォルト	値 文字列 <i>eis/display name</i>
------------------------	-------------------------------------

説明

この接続ファクトリーを説明するテキストを指定します。

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

カテゴリ

この接続ファクトリーの分類またはグループ化に使用できる文字列を指定します。

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

コンポーネント管理認証別名

リソースに対するコンポーネント管理サインオンの認証データを指定します。

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

ピック・リストに表示されたことのない新規別名を定義します。

- 「適用」をクリックして「関連項目」を表示する。
- 「J2C 認証データ・エントリ」をクリックする。
- 別名を定義する。
- 「J2C 認証データ・エントリ」ページの最上部にある接続ファクトリー名をクリックして、接続ファクトリーのページに戻る。
- 別名を選択する。

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

コンテナ管理認証別名

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

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

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

まだリストに組み込まれていない新規別名を定義します。

- 「適用」 をクリックして 「関連項目」 を表示する。
- 「J2C 認証データ・エントリー」 をクリックする。
- 別名を定義する。
- 「J2C 認証データ・エントリー」 ページの最上部にある接続ファクトリー名をクリックして、接続ファクトリーのページに戻る。
- 別名を選択する。

プロパティ
データ型

値
ピック・リスト

認証設定

この接続ファクトリーに定義される認証メカニズムを指定します。

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

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

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

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

- <authentication-mechanism-type>BasicPassword</authentication-mechanism-type>
- <authentication-mechanism-type>Kerbv5</authentication-mechanism-type>

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

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

マッピング構成別名

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

注: マッピング構成別名は、res-auth=Container を持つコンポーネントのために、デプロイメント時のリソース参照マッピングでのログイン構成の指定によって置き換えられます。

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

プロパティ	値
データ型	ピック・リスト

J2C 接続ファクトリーの拡張設定:

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

目的

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

「構成」タブ

キャッシュ・ハンドルの管理

チェックされている場合、キャッシュ・ハンドル (Bean の inst vars に保持されているハンドル) がコンテナによって追跡されます。

プロパティ	値
データ型	チェック・ボックス

トランザクション・コンテキストの欠落をログに記録

チェックされている場合、コンテナは、接続の確立時にトランザクション・コンテキストの欠落をログに記録します。

プロパティ	値
データ型	チェック・ボックス

WebSphere Business Integration Adapter の構成

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

1. アプリケーション EAR ファイルをインストールします。

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

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

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

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

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

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

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

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

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

WebSphere Process Server で WebSphere Business Integration Adapter を使用するために必要な成果物を作成するには、『アプリケーションのインストール』の説明に従う必要があります。

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

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

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

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

1. キュー接続ファクトリーを作成します。

管理コンソールの最上位から、以下のステップを実行します。

- a. 「リソース」を展開します。
- b. 「JMS プロバイダー」を展開します。
- c. 「デフォルトのメッセージング」を選択します。

d. 「**JMS キュー接続ファクトリー**」を選択します。

JMS 副見出しから、「**JMS キュー接続ファクトリー**」を選択します。

e. 新規 JMS キュー接続ファクトリーを作成します。

「**新規**」をクリックします。

f. 以下を例外として、すべてのデフォルト値を受け入れます。

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

g. 新規 JMS キュー接続ファクトリーの作成を完了します。

「**OK**」をクリックします。

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

h. ローカル構成レベルで行った変更をマスター構成に適用します。

メッセージ・ウィンドウで「**保管**」をクリックします。

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

管理コンソールの最上位から、以下のステップを実行します。

a. 「**リソース**」を展開します。

b. 「WebSphere Business Integration Adapter」パネルを開きます。

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

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

「**新規**」をクリックします。

d. 以下を例外として、すべてのデフォルト値を受け入れます。

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

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

「**OK**」をクリックします。

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

f. ローカル構成レベルで行った変更をマスター構成に適用します。

メッセージ・ウィンドウで「**保管**」をクリックします。

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

管理コンソールの最上位から、以下のステップを実行します。

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。

該当の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「Business Integration」の副見出しから「**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」チェック・ボックスが選択されていることを確認します。
- f. 「**OK**」をクリックします。

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

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

メッセージ・ウィンドウで「保管」をクリックします。

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

WebSphere Business Integration Adapter の管理:

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

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

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

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

1. 管理するリソース (1 つまたは複数) を選択します。

管理コンソールの最上位から、以下のステップを実行します。

- 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。

該当の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「Business Integration」の副見出しから「**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作用するコマンド・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。

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

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

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

注: このタスクは WebSphere Integration Developer で実行され、ここに参照の目的でのみ説明します。詳しくは、WebSphere Integration Developer のインフォメーション・センターを参照してください。

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

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

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

一般プロバイダーの JMS リソースの使用

このトピックは、一般メッセージング・プロバイダーによって提供される JMS リソースを WebSphere アプリケーションが使用できるようにするための方法に関する一連のトピックへの入り口です。「一般メッセージング・プロバイダー」という用語は、WebSphere のデフォルトのメッセージング・プロバイダーおよび WebSphere MQ 以外のメッセージング・プロバイダーを意味します。WebSphere MQ は個別の製品であり、インストールされている場合はメッセージング・プロバイダーとしてのみ使用できることに注意してください。

デフォルトのメッセージング・プロバイダー以外のメッセージング・プロバイダーをインストールすることができます。メッセージング・プロバイダーとして WebSphere MQ を選択することは可能ですが、この製品は WebSphere Process Server の一部としてはインストールされません。WebSphere アプリケーションは JMS 1.1 インターフェースまたは JMS 1.0.2 インターフェースを使用して、デフォルトのメッセージング・プロバイダーや WebSphere MQ (インストール済みの場合) によって提供される JMS リソースに加えて、一般メッセージング・プロバイダーによって提供される JMS リソースにアクセスできます。

WebSphere Process Server 管理コンソールを使用して、一般メッセージング・プロバイダーによって提供される JMS 接続ファクトリーおよび宛先を管理できます。

一般メッセージング・プロバイダーの定義:

このタスクを使用して、新規メッセージング・プロバイダーを、デフォルトのメッセージング・プロバイダーまたは WebSphere MQ (インストールしている場合) の代わりとして使用するために、WebSphere Process Server に対して定義します。

このタスクをインストールする場合は、メッセージング・プロバイダーの提供するツールおよび情報を使用して、メッセージング・プロバイダーおよびそのリソースのインストールと構成が事前に完了していなければなりません。

WebSphere Process Server に対して新規の一般メッセージング・プロバイダーを定義するには、管理コンソールを使用して、以下のステップに従います。

1. ナビゲーション・ペインで、「**JMS プロバイダー**」>「**汎用**」をクリックします。これにより、コンテンツ・ペインに既存の一般メッセージング・プロバイダーが表示されます。
2. 新規の一般メッセージング・プロバイダーを定義するには、コンテンツ・ペインで「**新規**」をクリックします。または、既存のメッセージング・プロバイダーの定義を変更するために、プロバイダー名をクリックします。そうすると、コンテンツ・ペイン内に、メッセージング・プロバイダーの定義に使用するプロパティが表示されます。
3. 以下の必要なプロパティを指定します。後のステップで説明するように、他のプロパティを指定することができます。

- 名前。IBM WebSphere Application Server 内で、このメッセージング・プロバイダーを管理目的で使用する際の名前。
 - 外部初期コンテキスト・ファクトリー。JMS プロバイダーの初期コンテキスト・ファクトリーの Java クラス名。
 - 外部プロバイダー URL。外部 JNDI ルックアップを行うための JMS プロバイダー URL。
4. **オプション:** 「適用」をクリックします。これにより、ユーザーは追加プロパティを指定することができます。
 5. **オプション:** 他のメッセージング・プロバイダーのプロパティを指定します。「追加プロパティ」の下で「カスタム・プロパティ」リンクを使用すると、標準の `javax.naming` プロパティの形式で、初期コンテキスト・ファクトリーのカスタム・プロパティを指定することができます。
 6. 「OK」をクリックします。
 7. 変更をマスター構成に保管します。
 8. 変更した構成を有効にするには、アプリケーション・サーバーを停止してから再始動してください。

これで、汎用メッセージング・プロバイダー用の JMS リソースを構成できるようになりました。

一般メッセージング・リソースの管理リストの表示:

WebSphere Process Server 管理コンソールを使用して、デフォルトのメッセージング・プロバイダーまたはインストールしている場合は WebSphere MQ 以外のメッセージング・プロバイダーによって提供される JMS リソースの管理リストを表示します。

WebSphere Process Server 管理コンソールを使用して、一般メッセージング・プロバイダーによって提供される以下の JMS リソース・タイプのリストを表示することができます。表示されたパネルを使用して、管理対象の JMS リソースの選択や、JMS リソースの作成または削除 (必要に応じて) を行うことができます。

一般メッセージング・プロバイダーの JMS リソースの管理リストを表示するには、以下の一般的なステップを完了してください。

1. WebSphere 管理コンソールを開始します。
2. ナビゲーション・ペインで、「リソース」>「JMS プロバイダー」>「汎用」をクリックします。
3. コンテンツ・ペインで、一般メッセージング・プロバイダーの有効範囲を変更します (適切な場合)。
4. コンテンツ・ペインの「追加リソース (Additional Resources)」の下にある、JMS リソースのタイプへのリンクをクリックします。これにより、選択したタイプの既存のリソースのリストが表示されます。リソースのために表示された設定パネルの詳細については、関連する解説トピックを参照してください。

JMS プロバイダー・コレクション:

このパネルを使用して、JMS プロバイダーをリストしたり、JMS プロバイダーを選択して、その構成プロパティを表示または変更したりします。

この管理コンソール・ページを表示するには、「リソース」>「JMS プロバイダー」>「一般」をクリックします。

JMS プロバイダーのプロパティあるいはそのリソースを表示または変更するには、表示されたリストでその名前を選択します。

新規の一般 JMS プロバイダーを定義するには、「新規」をクリックします。

リストされている 1 つ以上の JMS プロバイダーを操作するには、操作するオブジェクト名の隣にあるチェック・ボックスをクリックして表示されるボタンを使用します。

名前 管理目的でこの JMS プロバイダーを使用する際の名前。

説明 管理目的で使用する、この JMS プロバイダーの説明。

一般 JMS 接続ファクトリー・コレクション:

関連する一般メッセージング・プロバイダーで構成される、Point-to-Point および Publish/Subscribe メッセージングの両方についての JMS 接続ファクトリー。このパネルを使用して、JMS 接続ファクトリーを作成または削除したり、接続ファクトリーを選択して、その構成プロパティを表示または変更したりします。

コレクション・パネル

このパネルには、一般 JMS 接続ファクトリーのリストが、その構成プロパティの要約とともに表示されます。

この管理コンソール・ページを表示するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「リソース」>「JMS プロバイダー」>「汎用」の順に展開します。
2. 目次ペインで、JMS 接続ファクトリーをサポートするための一般メッセージング・プロバイダーの名前をクリックします。
3. 「追加プロパティ」の下の「JMS 接続ファクトリー」をクリックします。

新しい JMS 接続ファクトリーを定義するには、「新規」をクリックします。

JMS 接続ファクトリーのプロパティを表示または変更するには、表示されたリストでその名前を選択します。

リストされた JMS 接続ファクトリーを 1 つ以上操作する場合は、操作するオブジェクトの名前の隣にあるチェック・ボックスをクリックし、表示されたボタンを使用します。

一般 JMS 接続ファクトリー設定:

このパネルを使用して、関連する一般 JMS プロバイダーで使用する、選択済みの JMS 接続ファクトリーの構成プロパティを表示または変更します。これらの構成プロパティは、プロバイダー上の JMS 宛先への接続の作成方法を制御します。

目的

JMS 接続ファクトリーは、JMS 宛先への接続を作成するために使用されます。
JMS 接続ファクトリーは、関連する JMS プロバイダーによって作成されます。

このページを表示するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「リソース」>「**JMS プロバイダー**」>「**汎用**」の順に展開します。
2. コンテンツ・ペインで、JMS 接続ファクトリーをサポートするメッセージング・プロバイダーの名前をクリックします。
3. コンテンツ・ペインで、一般メッセージング・プロバイダーの有効範囲を変更します (適切な場合)。
4. 「追加プロパティ」の下の「**JMS 接続ファクトリー**」をクリックします。
5. 作業の対象となる JMS 接続ファクトリーの名前をクリックします。

一般 JMS プロバイダー (デフォルトのメッセージング・プロバイダーまたは JMS プロバイダーとしてインストールしている場合の WebSphere MQ 以外) の JMS 接続ファクトリーには、以下のプロパティがあります。

プロパティ

名前

IBM WebSphere Process Server 内で、この JMS 接続ファクトリーを管理目的で使用する際の名前。この名前は、関連のあるメッセージング・プロバイダー内で固有でなければなりません。

型

この接続ファクトリーで JMS キュー宛先を作成するか、JMS トピック宛先を作成するかを指定します。

以下のオプションのいずれかを選択します。

- **QUEUE** Point-to-Point メッセージングを行うための JMS キュー接続ファクトリー。
- **TOPIC** Publish/Subscribe メッセージングを行うための JMS トピック接続ファクトリー。

JNDI 名

接続ファクトリーを WebSphere Process Server ネーム・スペースにバインドする際に使用される JNDI 名。

規則として、完全修飾の JNDI 名を使用します。例えば、jms/Name の形式です。ここで、Name は、リソースの論理名です。

この名前は、プラットフォームのバインディング情報をリンクするために使用されます。バインディングにより、モジュールのデプロイメント記述子で定義されているリソースと、プラットフォーム別に JNDI にバインドされている実際の (物理的な) リソースとの関連付けが行われます。

説明

WebSphere Process Server 内で管理目的で使用する、この接続ファクトリーの説明。

カテゴリ

WebSphere Process Server 管理レコード用の、この接続ファクトリーの分類またはグループ化に使用されるカテゴリ。

外部 JNDI 名

一般メッセージング・プロバイダーのネーム・スペースに接続ファクトリーをバインドするために使用される JNDI 名。

規則として、完全修飾の JNDI 名を使用します。例えば、jms/Name の形式です。ここで、**Name** は、リソースの論理名です。

この名前は、プラットフォームのバインディング情報をリンクするために使用されます。バインディングにより、モジュールのデプロイメント記述子で定義されているリソースと、プラットフォーム別に JNDI にバインドされている実際の (物理的な) リソースとの関連付けが行われます。

コンポーネント管理認証別名

この別名は、アプリケーション管理認証のための JMS プロバイダーへの接続を認証するのに使用するユーザー ID とパスワードを指定します。

このプロパティでは、WebSphere Process Server に定義されている J2C 認証データ・エントリー別名のリストを提供します。使用するデータ・エントリー別名を選択して、JMS プロバイダーへの新規接続を生成するための認証を受けることができます。

WebSphere Process Server に対するグローバル・セキュリティーを使用可能にしている場合は、JMS プロバイダーへの新規接続の作成の認証を受けるために使用するユーザー ID とパスワードを指定する別名を選択します。この別名の使い方は、アプリケーション・コンポーネントのデプロイメント記述子の接続ファクトリー・リソース参照で宣言されているリソース認証 (res-auth) の設定によって異なります。

コンテナ管理認証別名

この別名は、コンテナ管理認証のための JMS プロバイダーへの接続認証に使用するユーザー ID とパスワードを指定します。

このプロパティでは、WebSphere Process Server に定義されている J2C 認証データ・エントリー別名のリストを提供します。使用するデータ・エントリー別名を選択して、JMS プロバイダーへの新規接続を生成するための認証を受けることができます。

WebSphere Process Server に対するグローバル・セキュリティーを使用可能にしている場合は、JMS プロバイダーへの新規接続の作成の認証を受けるために使用するユーザー ID とパスワードを指定する別名を選択します。この別名の使い方は、アプリケーション・コンポーネントのデプロイメント記述子の接続ファクトリー・リソース参照で宣言されているリソース認証 (res-auth) の設定によって異なります。

接続プール

接続プールの設定のオプション・セットを指定します。

接続プール・プロパティは、すべての J2C コネクタに共通です。

アプリケーション・サーバーは、メッセージング・プロバイダーとの接続とセッションをプールすることで、パフォーマンスを向上させます。アプリケーションの接続とセッションのプール・プロパティを適切に構成する必要があります。構成が適切でないと、接続とセッションが正しく動作しない場合があります。

JMS リソースへのサーバー・サイド・アクセスが並行して行われ、それがデフォルト値を超える場合は、接続プールのサイズを変更してください。接続プールのサイズは、キューごと、あるいはトピックごとに設定されます。

セッション・プール

セッション・プール設定のオプションのセット。

このリンクは、すべての J2C コネクタに共通の、オプションの接続プール・プロパティのパネルを提供します。

アプリケーション・サーバーは、メッセージング・プロバイダーとの接続とセッションをプールすることで、パフォーマンスを向上させます。アプリケーションの接続とセッションのプール・プロパティを適切に構成する必要があります。構成が適切でないと、接続とセッションが正しく動作しない場合があります。

カスタム・プロパティ

メッセージング・プロバイダーに渡されるカスタム・プロパティの名前と値の対のオプション・セットです。

コンテナ管理認証別名

この別名は、コンテナ管理認証のための JMS プロバイダーへの接続認証に使用するユーザー ID とパスワードを指定します。

このプロパティでは、WebSphere Process Server に定義されている J2C 認証データ・エントリ別名のリストを提供します。使用するデータ・エントリ別名を選択して、JMS プロバイダーへの新規接続を生成するための認証を受けることができます。

WebSphere Process Server に対するグローバル・セキュリティを使用可能にしている場合は、JMS プロバイダーへの新規接続の作成の認証を受けるために使用するユーザー ID とパスワードを指定する別名を選択します。この別名の使い方は、アプリケーション・コンポーネントのデプロイメント記述子の接続ファクトリー・リソース参照で宣言されているリソース認証 (res-auth) の設定によって異なります。

コンテナ管理認証別名

この別名は、コンテナ管理認証のための JMS プロバイダーへの接続認証に使用するユーザー ID とパスワードを指定します。

このプロパティでは、WebSphere Process Server に定義されている J2C 認証データ・エントリ別名のリストを提供します。使用するデータ・エントリ別名を選択して、JMS プロバイダーへの新規接続を生成するための認証を受けることができます。

WebSphere Process Server に対するグローバル・セキュリティを使用可能にしている場合は、JMS プロバイダーへの新規接続の作成の認証を受けるために使用するユーザー ID とパスワードを指定する別名を選択します。この別名の使

いは、アプリケーション・コンポーネントのデプロイメント記述子の接続ファクトリー・リソース参照で宣言されているリソース認証 (res-auth) の設定によって異なります。

「構成」タブ

有効範囲

このリソース定義をアプリケーションに対して可視にするレベルを指定します。

メッセージング・プロバイダー、ネーム・スペース・バインディング、共用ライブラリーなどのリソースを複数の有効範囲で定義することが可能です。より限定的な有効範囲で定義されたリソースは、それより広い有効範囲で定義された重複するリソースをオーバーライドします。

表示された有効範囲は情報のためだけです。このパネルで変更することはできません。異なる有効範囲のこのリソース (あるいは他のリソース) を参照するかまたは変更したい場合は、メッセージング・プロバイダー設定パネルの有効範囲を変更し、「適用」をクリックしてから、リソース・タイプのリンクをクリックします。

マッピング構成別名

認証別名のマップに使用されるモジュール。

このフィールドは、「グローバル・セキュリティ」>「JAAS 構成」>「アプリケーション・ログイン構成」プロパティ上で構成されたモジュールのリストを提供します。

一般 JMS 宛先コレクション:

関連するメッセージング・プロバイダーで構成される、Point-to-Point および Publish/Subscribe メッセージングについての JMS 宛先。このパネルを使用して、JMS 宛先を作成または削除したり、JMS 宛先を選択して、その構成プロパティを表示または変更したりします。

コレクション・パネル

この管理コンソール・ページを表示するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「リソース」>「JMS プロバイダー」>「汎用」の順に展開します。
2. 目次ペインで、JMS 宛先をサポートするための一般メッセージング・プロバイダーの名前をクリックします。
3. 「追加プロパティ」の下の「JMS 宛先」をクリックします。

新しい JMS 宛先を定義するには、「新規」をクリックします。

JMS 宛先のプロパティを表示または変更するには、表示されたリストでその名前を選択します。

リストされている 1 つ以上の JMS 宛先を操作するには、操作するオブジェクト名の隣にあるチェック・ボックスをクリックして表示されるボタンを使用します。

一般 JMS 宛先設定:

このパネルを使用して、関連する JMS プロバイダーで使用する、選択済みの JMS 宛先の構成プロパティを表示または変更します。

目的

JMS 宛先は、関連する一般メッセージング・プロバイダー (デフォルトのメッセージング・プロバイダーあるいは WebSphere MQ ではなく) の JMS 宛先のプロパティを構成する際に使用されます。JMS 宛先への接続は、関連する JMS 接続ファクトリーによって作成されます。

このページを表示するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「リソース」>「JMS プロバイダー」>「汎用」の順に展開します。
2. コンテンツ・ペインで、JMS 宛先をサポートするメッセージング・プロバイダーの名前をクリックします。
3. コンテンツ・ペインで、一般メッセージング・プロバイダーの有効範囲を変更します (適切な場合)。
4. 「追加プロパティ」の下で「JMS 宛先」をクリックします。
5. 作業の対象となる JMS 宛先の名前をクリックします。

一般メッセージング・プロバイダーで使用する JMS 宛先には、次のプロパティがあります。

プロパティ

名前

IBM WebSphere Process Server 内で、このキューを管理目的で使用する際の名前。

型

この JMS 宛先がキュー (Point-to-Point の場合) であるかトピック (Publish/Subscribe の場合) であるかを指定します。

以下のオプションのいずれかを選択します。

- **QUEUE** Point-to-Point メッセージング用の JMS キュー宛先。
- **TOPIC** Publish/Subscribe メッセージングを行うための JMS トピック宛先。

JNDI 名

接続ファクトリーを WebSphere Process Server ネーム・スペースにバインドする際に使用される JNDI 名。

規則として、完全修飾の JNDI 名を使用します。例えば、**jms/Name** の形式です。ここで、**Name** は、リソースの論理名です。

この名前は、プラットフォームのバインディング情報をリンクするために使用されます。バインディングにより、モジュールのデプロイメント記述子で定義されているリソースと、プラットフォーム別に JNDI にバインドされている実際の (物理的な) リソースとの関連付けが行われます。

説明

管理を目的とした、キューの説明。

カテゴリ

WebSphere Process Server 管理レコード用の、このキューの分類またはグループ化に使用されるカテゴリ。

外部 JNDI 名

キューを WebSphere Process Server のネーム・スペースにバインドする際に使用される JNDI 名。

規則として、完全修飾の JNDI 名を使用します。例えば、jms/Name の形式です。ここで、**Name** は、リソースの論理名です。

この名前は、プラットフォームのバインディング情報をリンクするために使用されます。バインディングにより、モジュールのデプロイメント記述子で定義されているリソースと、プラットフォーム別に JNDI にバインドされている実際の (物理的な) リソースとの関連付けが行われます。

「構成」タブ

有効範囲

このリソース定義をアプリケーションに対して可視にするレベルを指定します。

メッセージング・プロバイダー、ネーム・スペース・バインディング、共用ライブラリーなどのリソースを複数の有効範囲で定義することが可能です。より限定的な有効範囲で定義されたリソースは、それより広い有効範囲で定義された重複するリソースをオーバーライドします。

表示された有効範囲は情報のためだけです。このパネルで変更することはできません。異なる有効範囲のこのリソース (あるいは他のリソース) を参照するかまたは変更したい場合は、メッセージング・プロバイダー設定パネルの有効範囲を変更し、「適用」をクリックしてから、リソース・タイプのリンクをクリックします。

一般メッセージング・プロバイダーの JMS リソースの構成:

以下のタスクを実行して、一般メッセージング・プロバイダー (デフォルトのメッセージング・プロバイダーまたは WebSphere MQ 以外) の JMS 接続ファクトリーおよび宛先を構成します。

これらのタスクを実行する必要があるのは、WebSphere Process Server 環境がデフォルトのメッセージング・プロバイダーまたは WebSphere MQ 以外のメッセージング・プロバイダーを使用して、JMS を使用するエンタープライズ・アプリケーションをサポートする場合のみです。そのような一般メッセージング・プロバイダーを使用可能にするには、WebSphere Application Server への新規 JMS プロバイダーの定義に記載されているように、メッセージング・プロバイダーのインストールおよび構成が完了していなければなりません。

注: WebSphere MQ をインストールしている場合は、メッセージング・プロバイダーとしてのみ使用可能です。これは WebSphere Process Server の一部ではありません。

一般メッセージング・プロバイダーの JMS リソースを構成するには、サブネット・タスクを完了してください。

JMS 接続ファクトリーの構成、一般 JMS プロバイダー:

このタスクを使用して、デフォルトのメッセージング・プロバイダーまたは WebSphere MQ (インストールしている場合) 以外の一般 JMS プロバイダーで使用する、JMS 接続ファクトリーのプロパティを参照または変更します。

一般 JMS プロバイダーで使用する JMS 接続ファクトリーを構成するには、管理コンソールを使用して、以下のステップを完了します。

1. 一般メッセージング・プロバイダーを表示します。ナビゲーション・ペインで、「リソース」>「**JMS プロバイダー**」>「**汎用**」の順に展開します。
2. **オプション:** 「**有効範囲**」設定を、接続ファクトリーがアプリケーションに対して可視であるレベルに変更します。
3. コンテンツ・ペインで、「追加プロパティ」の下にある「**JMS 接続ファクトリー**」をクリックします。すべての既存の JMS 接続ファクトリーとそのプロパティの概要がリストされた表が表示されます。
4. 既存の JMS 接続ファクトリーをブラウズまたは変更するには、リストのその名前をクリックします。そうしない場合は、新規接続ファクトリーを作成するために、次のステップを行います。
 - a. コンテンツ・ペインで「**新規**」をクリックします。
 - b. 以下の必要なプロパティを指定します。後のステップで説明するように、他のプロパティを指定することができます。
 - **名前** IBM WebSphere Process Server 内で、この JMS 接続ファクトリーを管理目的で使用する際の名前。
 - **タイプ** 接続ファクトリーが JMS キュー (QUEUE) 用か JMS トピック (TOPIC) 用か選択します。
 - **JNDI 名** JMS 接続ファクトリーを WebSphere Application Server のネーム・スペースにバインドする際に使用される JNDI 名。
 - **外部 JNDI 名** メッセージング・プロバイダーのネーム・スペースに JMS 接続ファクトリーをバインドするために使用される JNDI 名。
 - c. 「**適用**」をクリックします。これによって、JMS 接続ファクトリーを WebSphere Process Server に定義し、追加プロパティをブラウズまたは変更することができます。
5. **オプション:** 必要に応じて、JMS 接続ファクトリーのプロパティを変更します。
6. 「**OK**」をクリックします。
7. マスター構成について行った変更をすべて保管します。
8. 変更した構成を有効にするには、アプリケーション・サーバーを停止してから再始動してください。

JMS 宛先の構成、一般 JMS プロバイダー:

このタスクを使用して、デフォルトのメッセージング・プロバイダーまたは WebSphere MQ (インストールしている場合) 以外の一般 JMS プロバイダーで使用する、JMS 宛先のプロパティを参照または変更します。

一般 JMS プロバイダーで使用する JMS 宛先を構成するには、管理コンソールを使用して、以下のステップを完了します。

1. 一般メッセージング・プロバイダーを表示します。ナビゲーション・ペインで、「リソース」>「JMS プロバイダー」>「汎用」の順に展開します。
2. **オプション:** 「有効範囲」設定を、接続ファクトリーがアプリケーションに対して可視であるレベルに変更します。
3. コンテンツ・ペインで、「追加プロパティ」の下にある「JMS 宛先」をクリックします。すべての既存の JMS 宛先とそのプロパティの概要がリストされた表が表示されます。
4. 既存の JMS 宛先をブラウズまたは変更するには、リストのその名前をクリックします。そうしない場合は、新規宛先を作成するために、次のステップを行います。
 - a. コンテンツ・ペインで「新規」をクリックします。
 - b. 以下の必要なプロパティを指定します。後のステップで説明するように、他のプロパティを指定することができます。
 - **名前** IBM WebSphere Process Server 内で、この JMS 宛先を管理目的で使用する際の名前。
 - **タイプ** 宛先が JMS キュー (QUEUE) 用か JMS トピック (TOPIC) 用か選択します。
 - **JNDI 名** JMS 宛先を WebSphere Process Server のネーム・スペースにバインドする際に使用される JNDI 名。
 - **外部 JNDI 名** メッセージング・プロバイダーのネーム・スペースに JMS 宛先をバインドするために使用される JNDI 名。
 - c. 「適用」をクリックします。これにより、WebSphere Process Server に対する JMS 宛先が定義され、追加プロパティをブラウズまたは変更できるようになります。
5. **オプション:** 必要に応じて、JMS 宛先のプロパティを変更します。
6. 「OK」をクリックします。
7. マスター構成について行った変更をすべて保管します。
8. 変更した構成を有効にするには、アプリケーション・サーバーを停止してから再始動してください。

EIS アプリケーションのインストール

EIS アプリケーション・モジュール・パターンに従う、EIS アプリケーション・モジュール、サービス・コンポーネント・アーキテクチャー (SCA) ・モジュールを J2SE プラットフォームまたは J2EE プラットフォームにデプロイすることができます。

EIS モジュールをデプロイするのに必要なステップは、プラットフォームに依存します。

詳しくは、後続のタスクを参照してください。

J2SE プラットフォームへの EIS アプリケーション・モジュールのデプロイ:

EIS モジュールを J2SE プラットフォームにデプロイできますが、EIS インポートのみがサポートされます。

このタスクを開始する前に WebSphere Integration Development 環境に JMS インポート・バインディングを持つ EIS アプリケーション・モジュールを作成する必要があります。

メッセージ・キューの使用を介して EIS システムを非同期的にアクセスする場合は、EIS アプリケーション・モジュールが JMS インポート・バインディングに提供されます。

J2SE プラットフォームへのデプロイは、バインディング・インプリメンテーションを非管理モードで実行できるインスタンスのみです。JMS バインディングは非同期および JNDI サポートを必要とします。そのどちらも基本サービス・コンポーネント・アーキテクチャーまたは J2SE で提供されません。J2EE Connector Architecture は非管理インバウンド通信をサポートせず、そのため EIS エクスポートを除去します。

モジュール依存関係に加えて、EIS インポートを持つ EIS アプリケーションが J2SE にデプロイされる場合、インポートで使用する WebSphere アダプターは依存関係としてマニフェストまたは SCA がサポートする他の任意のフォームで指定される必要があります。

J2EE プラットフォームへの EIS アプリケーション・モジュールのデプロイ:

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

このタスクを開始する前に WebSphere Integration Development 環境に JMS インポート・バインディングを持つ EIS モジュールを作成する必要があります。

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

表 10. バインディングから J2EE 成果物へのマッピング

SCA モジュール内のバインディング	生成された J2EE 成果物	作成された J2EE リソース
EIS インポート	モジュール・セッション EJB で生成されたリソース参照。	ConnectionFactory
EIS エクスポート	リソース・アダプターでサポートされるリスナー・インターフェースに応じて、生成またはデプロイされるメッセージ駆動 Bean。	ActivationSpec

表 10. バインディングから J2EE 成果物へのマッピング (続き)

SCA モジュール内のバインディング	生成された J2EE 成果物	作成された J2EE リソース
JMS インポート	ランタイムが提供するメッセージ駆動 Bean (MDB) はデプロイされ、リソース参照はモジュール・セッション EJB 上で生成されます。インポートで受信宛先がある場合のみ MDB が作成されることに注意してください。	<ul style="list-style-type: none"> • ConnectionFactory • ActivationSpec • Destinations
JMS エクスポート	ランタイムが提供するメッセージ駆動 Bean はデプロイされ、リソース参照はモジュール・セッション EJB 上で生成されます。	<ul style="list-style-type: none"> • ActivationSpec • ConnectionFactory • Destinations

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

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

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

JMS インポートが受信宛先でデプロイされる場合、メッセージ・ドライバー Bean (MDB) がデプロイされます。その MDB は、送信された要求への応答を listen します。MDB は、送信された宛先と JMS メッセージの JMSreplyTo ヘッダー・フィールドの要求に関連付けられて (listen する) います。応答メッセージが到着したときに、MDB はその相関 ID を使用してコールバック宛先に格納されているコールバック情報を検索して、その後コールバック・オブジェクトを起動します。

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

JMS エクスポートがデプロイされている場合、メッセージ駆動 Bean (MDB) (JMS インポートにデプロイした MDB とは異なる MDB) がデプロイされます。その

MDB は受信宛先で着信要求を listen し、次に、SCA が処理する要求をディスパッチします。インストール・タスクは、1 つの応答および 2 つの宛先を送信するために使用する JMS インポート、ActivationSpec、ConnectionFactory 用のリソースと同じリソースのセットを作成します。これらのリソースのすべてのプロパティは、エクスポート・ファイルで指定されます。JMS プロバイダーが SIBus Resource Adapter である場合、JMS 宛先に対応する SIBus 宛先が作成されます。

WebSphere アダプター・アプリケーションのリモート・デプロイメント

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

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

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

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

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

WebSphere アダプター・アプリケーションを介して EIS と通信する必要がある BPEL 処理アプリケーションにとって、WebSphere アダプター・アプリケーションのリモート・デプロイメントは理想的です。WebSphere アダプター・アプリケーションは、通信する EIS の近くの便利な位置にあるセル内のノードに置くことができます。BPEL プロセス・アプリケーションと同じノードにある必要はありません。ベスト・プラクティスのガイドラインに従うと、WebSphere Process Server の Network Deployment 環境は、図 6 のようなアーキテクチャーになります。

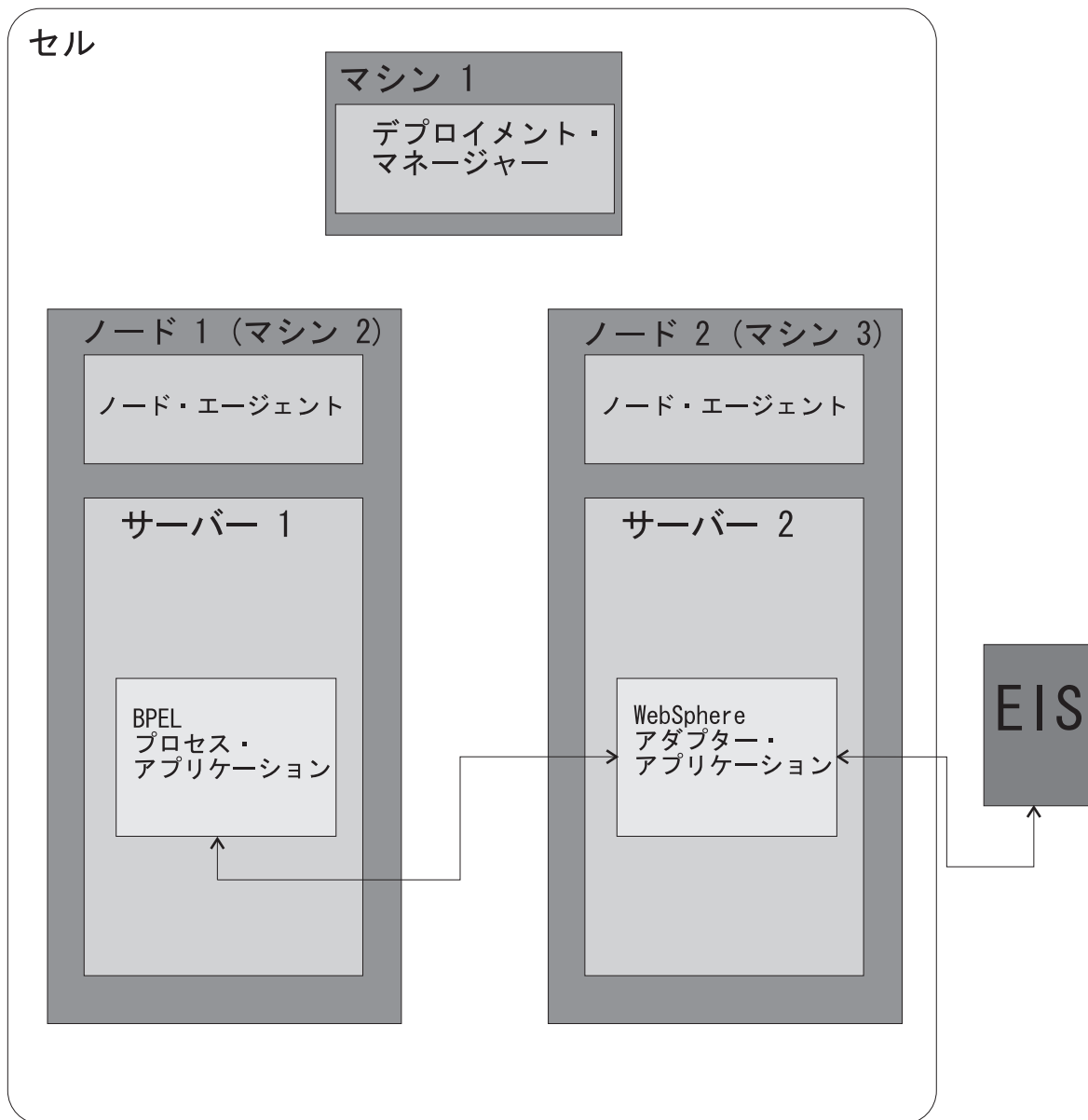


図 6. Network Deployment 環境に WebSphere アダプター・アプリケーションをリモート配置するセル・アーキテクチャ

JMS ヘッダー

JMS ヘッダーは、JMS メッセージ・プロパティのすべてのプロパティを含むサービス・データ・オブジェクト (SDO) です。

JMS ヘッダーは、インバウンド・メッセージからの JMS メッセージ・プロパティか、またはアウトバウンド・メッセージに適用される JMS メッセージ・プロパティのすべてのプロパティを含む、生成されたサービス・データ・オブジェクトです。

JMS ヘッダーは、システム・プログラミング・インターフェースを使用して提供されるため、結果として、サービス・コンポーネント・アーキテクチャー・メッセー

ジにアクセスすることで JMS ヘッダーを設定または取得します。JMS ヘッダーは、クライアント・プログラムから設定または取得できません。ただし、ESB メッセージングを使用して JMS ヘッダー値を取得したり、またはそれらを発信要求に設定できます。

JMS ヘッダーおよび JMS メッセージ・プロパティが使用されるのは、基本サービス・コンポーネント・アーキテクチャー SCDL バインディング・スイッチがオンになっている場合のみです。スイッチがオンになっているとき、コンテキスト情報は伝搬されます。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オンおよびオフにすると、ヘッダー情報がメッセージまたはターゲット・コンポーネントに流れることを許可しますが、これを行うには、eisBinding の **ContextPropagationEnabled** 属性を true または false に指定します。インポートおよびエクスポートの両方についてこれを行います。以下に例を挙げます。

```
<esbBinding xsi:type="eis:JMSImportBinding" ContextPropagationEnabled="true">
```

セレクター・コンポーネントの管理の概要

ビジネスの変更に応じて、それらの駆動機構であるビジネス・プロセスも変更する必要があります。これらの変更の中には、プロセスの設計を変更することなく、特定のプロセスが元の設計とは異なる結果を戻す必要があるものもあります。セレクター・コンポーネントは、その柔軟性の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、サービスへの単一のインターフェースを提供します。セレクター・コンポーネントは、インターフェースおよびセレクター・テーブルで構成されます。セレクター・テーブルは、基準に基づいて、どのコンポーネント (ターゲット・コンポーネントと呼ばれます) が要求を処理するかを決定します。サーバーは、ターゲット・コンポーネントによって提供された処理結果をクライアントに戻します。

ビジネス・プロセスの構築時に、ソリューション設計者はセレクター・コンポーネントの要件を識別し、セレクター・コンポーネントが処理の完了に使用するインターフェースおよびセレクター・テーブルを定義します。セレクター・コンポーネントの開発に含まれるタスクについては、WebSphere Integration Developer インフォメーション・センターを参照してください。

セレクター・コンポーネントの管理は、セレクター・コンポーネントに関連したタスクまたはセレクター・テーブルに関連したタスクから構成されます。

セレクター・コンポーネントの表示

セレクター・コンポーネントの表示は、セレクター・コンポーネントを管理する最初のステップです。この表示から、セレクター・コンポーネントの一部またはすべてをエクスポートしたり、セレクター・コンポーネントを構成するセレクター・テーブルを表示できます。

このタスクを実行するには、ユーザーは WebSphere Process Server の管理コンソールにいる必要があります。

このタスクを実行して、サーバー上に存在するセレクター・コンポーネントを判別します。

1. 「サーバー」をクリックして、さまざまなサーバー・タイプを表示します。
2. 「アプリケーション・サーバー」をクリックして、アプリケーション・サーバー・リストを展開します。
3. **サーバー名**をクリックして、サーバーのリストから表示するセレクターを持つサーバーを選択します。
4. 「**Business Integration**」の下の「**セレクター**」をクリックします。

コンソールに、各コンポーネントの説明と共に定義されたすべてのセレクター・コンポーネントのリストが表示されます。

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。表示されるのはターゲット・コンポーネントのリストで、そこから処理基準の変更、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、またはテーブルからのターゲット・コンポーネントの削除 (それによる基準の削除) などを実行できます。

このタスクを実行するには、ユーザーは WebSphere Process Server の管理コンソールにいる必要があります。

他のセレクター・テーブル関連タスクを実行するには、テーブルを構成する項目の決定時にセレクター・テーブルを表示します。このタスクは、セレクター・コンポーネントの表示を実行した後に開始されます。

1. セレクター・コンポーネント画面からセレクター・コンポーネントをクリックします。ブラウザーに、選択したコンポーネント内のセレクター・テーブルが表示されます。
2. 画面上のセレクター・テーブルの 1 つをクリックします。ブラウザーに、セレクター・テーブルを構成するターゲット・コンポーネントが表示されます。

ターゲット・コンポーネントの変更

ターゲット・コンポーネントを変更すると、セレクター・コンポーネント処理を変更することができます。変更するには、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、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、選択基準とターゲット・コンポーネントの両方を変更するのいずれかを実行します。

このタスクを実行するには、セレクター・テーブルが存在している必要があります。

ターゲット・コンポーネントを変更して、選択基準を変更したり、別のターゲット・コンポーネントのセレクター・テーブルの項目を使用します。

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前に、アプリケーションを停止します。長期実行アプリケーションの処理中は、ターゲット・コンポーネントを変更しないでください。

1. 『セレクター・テーブルの表示』で説明されているように、セレクター・テーブルを表示します。

2. 変更するターゲット・コンポーネントの横にある「**ターゲット ID**」をクリックします。システムによって、ターゲット・コンポーネントの詳細パネルが表示されます。
3. 項目を変更します。

変更する項目の一部 ターゲット宛先

選択基準

ターゲット宛先および選択基準

変更するステップ

1. ターゲット・コンポーネント・リストの横にある矢印をクリックします。システムによって、適格なターゲット・コンポーネントが表示されます。
2. リストからターゲット・コンポーネントを選択します。
3. 「**開始日**」か「**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依存し、ロケール・フォーマットに応じて表示されます。米国英語ロケールの場合、表示されるフォーマットは、以下のとおりです。
 - 月
 - 日
 - 年 (YYYY 形式)
 - 時刻 (HH:MM:SS 形式)
 - 時間帯
4. オプション: 「**デフォルト**」チェック・ボックスをクリックして、これをデフォルトのターゲット・コンポーネントにします。

選択基準が他のターゲット・コンポーネントの範囲内でない場合、セレクター・コンポーネントはこのターゲット・コンポーネントを使用します。
5. 「**適用**」または「**OK**」をクリックします。

この画面で作業を継続するには、「**適用**」をクリックします。ターゲット・コンポーネント画面に戻るには、「**OK**」をクリックします。

6. ターゲット・コンポーネント画面で「**保管**」をクリックして、変更内容をセレクトター・テーブルに保管します。

セレクトター・テーブル・ファイルに更新された選択基準とターゲット・コンポーネントが組み込まれます。セレクトター・コンポーネントは、更新されたセレクトター・テーブルを使用して、受信した次の要求を処理します。

ターゲット・コンポーネントの追加

現在セレクトター・テーブルに存在するものと異なる選択基準に合う追加処理が必要な場合、ターゲット・コンポーネントを追加します。

このタスクを実行するには、セレクトター・テーブルが存在している必要があります。

ビジネス・プロセスにさらに柔軟性が必要な場合、ターゲット・コンポーネントを追加します。セレクトター・コンポーネントがアクティブな場合、新規コンポーネントを追加できます。

1. 『セレクトター・コンポーネントの表示』で説明されているように、セレクトター・コンポーネントを表示します。
2. 『セレクトター・テーブルの表示』に説明されているように、セレクトター・テーブルを表示します。
3. セレクトター・テーブル画面で「**新規**」をクリックします。 ブラウザーに、事前に埋め込まれたターゲット・コンポーネントの詳細パネルが表示されます。
4. 『ターゲット・コンポーネントの変更』で説明されているアプリケーション要件に合うように、ターゲット宛先情報を編集します。
5. 「**OK**」をクリックしてターゲット・コンポーネントを保管し、ターゲット・コンポーネント画面に戻ります。

セレクトター・テーブルに新規ターゲット・コンポーネントが組み込まれます。セレクトター・コンポーネントは、更新されたセレクトター・テーブルを使用して、受信した次の要求を処理します。

ターゲット・コンポーネントの削除

特定の選択基準に合うセレクトター・テーブルの項目を削除することによってターゲット・コンポーネントを削除すると、セレクトター・コンポーネントの処理が変更されます。

このタスクを実行するには、セレクトター・テーブルが存在している必要があります。

ビジネス・プロセスでターゲット・コンポーネントの処理が不要になったら、ターゲット・コンポーネントを削除します。ターゲット・コンポーネントの削除後は、要求が入力されてそれがその他の特定の選択基準と一致しない場合、デフォルトの基準によって要求が処理されます。

1. 『セレクトター・テーブルの表示』に説明されているように、ターゲット・コンポーネントを表示します。 パネルにセレクトター・テーブル画面が表示されます。

2. 削除するターゲット・コンポーネントの横にあるチェック・ボックスをクリックして、「**削除**」をクリックします。システムによってパネルが更新され、残ったターゲット・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。残ったターゲット・コンポーネントを示す項目で更新されたセレクトター・テーブルが、システムによって保管されます。

セレクトター・テーブル・ファイルには残ったターゲット・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。

セレクトター・コンポーネントのエクスポート

セレクトター・コンポーネントをエクスポートすると、開発環境にインポートできるファイルが作成されます。それによって、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。

このタスクを開始するには、『セレクトター・コンポーネントの表示』に説明されているように、セレクトター・コンポーネントが既に表示されている必要があります。

セレクトター・テーブルが変更されていて、開発環境と実稼働環境を同期する必要がある場合、セレクトター・コンポーネントをエクスポートします。このタスクは、セレクトター・コンポーネント画面から開始されます。

1. エクスポートするセレクトターを選択します。

セレクトターの隣のチェック・ボックスをクリックしてから、「**エクスポート**」をクリックします。ブラウザーには、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**」パネルです。) 各セレクトターにはファイル拡張子として .zip が付いています。

2. ファイルをダウンロードします。

各ファイル名をクリックすると、プロンプトが出てファイルを保管するように要求されます。プロンプトが出たら、「**OK**」をクリックしてファイル・システムにファイルを配置します。

注: 選択すると、ダウンロードするときにファイルを名前変更することができます。

3. セレクトター表示に戻ります。

「**戻る**」をクリックして、セレクトターのリストに戻ります。

指定した場所にファイルが保管されます。その後、そのファイルをテスト・システムにコピーできます。

このファイルは WebSphere Integration Developer 環境にインポートする必要があります。WebSphere Integration Developer については、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・ルールおよびセクターを含むモジュールの考慮事項

本トピックでは、ビジネス・ルールおよびセクターを含んでいるモジュールをインストールまたは削除する場合に考慮すべき情報について説明します。

ビジネス・ルールおよびセクターによって、モジュールに柔軟性が付与されます。付与された柔軟性によって、モジュールのインストールまたは削除方法が変わります。これは、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組み立て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、開発環境に再度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に習熟していない場合は、このインフォメーション・センター内のこれらのタスクが説明され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が含まれている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は、サーバーはリポジトリのビジネス・ルールおよびセクターのコピーを上書きします。モジュールの置換時に、動的に行ったすべての変更は失われます。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ます。

あるモジュールによってインプリメント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に、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する時に、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が必要ないと判断した場合、リポジトリから除去します。『リポジトリからのビジネス・ルールおよびセクター・データの除去』では、不要なビジネス・ルールまたはセクターの消去方法について説明します。

データベース構成に関する考慮事項

ビジネス・ルールやセクターの成果物用の動的な成果物リポジトリでは、ターゲットのネーム・スペース、名前、タイプのいずれかを使用して基本キーを生成します。DB2 for z/OS バージョン 7 では、基本キーのサイズが 255 バイトに制限されます。

DB2 for z/OS 版を使用するようにシステムを構成した場合は、名前を以下のように制限する必要があります。

- ターゲットのネーム・スペース = 170 バイト
- 名前の最大長 = 75 バイト
- タイプの最大長 = 10 バイト

注: DB2 for z/OS バージョン 8 では、この制限は適用されません。

101 ページの『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』

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は、サーバーはリポジトリからこの成果物を除去しません。このタスクによって、リポジトリから不要のビジネス・ルールおよびセクター成果物を除去します。

リポジトリからのビジネス・ルールおよびセクター・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は、サーバーはリポジトリからこの成果物を除去しません。このタスクによって、リポジトリから不要のビジネス・ルールおよびセクター成果物を除去します。

除去しようとしているビジネス・ルールまたはセクターを使用するアプリケーションのすべてのコピーを、すべてのサーバーから間違いなくアンインストールしてください。

ビジネス・ルールまたはセクター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は、サーバーはこれらの成果物をデータベース表に格納するため、ユーザーはアプリケーションを変更することなくその成果物をダイナミックに更新できます。また、これによって、他のサーバーはこの成果物を共有することができます。アプリケーションをアンインストールする場合は、サーバーはデータベース表からこ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みる際に、それらのアプリケーションは失敗します。

未使用成果物をデータベースから削除するには、その成果物を使用するすべてのアプリケーションをアンインストールした後で、手動で削除する必要があります。リポジトリのデータベース・プラットフォームで提供されるツールを使用して成果物を除去します。

1. データベースの場所を探します。

データベースの場所はデータベース・プラットフォームに依存します。

データベース・プラットフォーム
Cloudscape

ロケーション
WASHOME¥cloudscape¥ /
databases¥RepositoryDB

その他のデータベース

サーバーのインストールおよび構成時に構成される場所により異なります。例えば、サーバーを自動的に構成し、デフォルト・データベース名を選択した場合、データベースの名前は WPSDB となります。

2. 行を削除する以下のデータベース表の場所を探します。

BYTESTORE

ビジネス・ルールおよびセクター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成果物を含むインストール済みアプリケーション

3. アプリケーション用の成果物を削除します。

ご使用のデータベース・プラットフォーム用のツールを使用して、所定のアプリケーション用のすべてのビジネス・ルールおよびセクター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 表で、**APPNAME** 列の値がアプリケーション名と同じ行をすべて探します。
- b. 見つかったすべての行の基本キー列の値を記録します。BYTESTORE 表の基本キー列は、**ARTIFACTNS**、**ARTIFACTNAME**、および **ARTIFACTTYPE** です。
- c. BYTESTORE 表からステップ 3a (102 ページ) で見つかった行を削除します。
- d. ステップ 3b (102 ページ) 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 表の行を検索します。

注: 基本キー値の各セットに対して、BYTESTOREOVERFLOW 表に 0、1 つまたは複数の対応する行がある可能性があります。

- e. BYTESTOREOVERFLOW 表からステップ 3d (102 ページ) で見つかった行を削除します。
- f. **APPNAME** 列がアプリケーションの名前に等しい APPTIMESTAMP 表の行を削除します。

データベース表から不要なビジネス・ルールおよびセクター成果物を除去しました。

ターゲットの概要

ターゲットは、参照用に構成されたターゲットを変更して処理を変更する機能を提供することによって、柔軟性を追加します。

コンポーネントは、他のモジュールのコンポーネントを呼び出し、それにより既存のロジックを再使用して、アプリケーションを作成する時間と費用を最小化することができます。WebSphere Process Server は、ターゲットによって、柔軟性を追加します。ターゲットは、インストール済みアプリケーションにモジュール間呼び出しのエンドポイントを変更させることにより、管理コンソールを使用して、アプリケーションを再作成または再配置することなくインストール済みアプリケーションの処理を向上させることができます。

提供された柔軟性を利用するには、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起動する方法が元になっています。名前のフォーマットは以下のとおりです。

呼び出しタイプ

名前のフォーマット

同期 Java Naming and Directory Interface (JNDI) フォーマットに準拠する名前。以下はその例です。

folder/export/fullpath_to_target/target_component_name

非同期 次のフォーマットの名前

*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と、アプリケーションの再コンパイルおよび再インストールをしないで起こるコンポーネントの利点を生かす柔軟性がアプリケーションに提供されます。

ターゲットの変更

参照のターゲットを変更すると、アプリケーションの再コンパイルおよび再インストールをしないで起こるコンポーネントの利点を生かす柔軟性がアプリケーションに提供されます。

参照のターゲットを変更する前に、以下のことを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
- モジュールが同期的にまたは非同期的にターゲットを起動しているかを確認する
- 参照が 1 つまたは複数のサービスをターゲットにしているかを確認する

オリジナル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できる新規または改良された機能を提供する場合、モジュールからのインポートのターゲットを変更します。

1. 変更してい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。この表示は更新され、停止したときのアプリケーションが示されます。

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」リストが最新表示された後で、新規ターゲットのため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. モジュールが常駐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用の宛先ターゲットを表示します。 4. 呼び出し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 という語を含んでいます。 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。プロンプトが出たら、「保管」をクリックします。

モジュールを始動し、モジュールが期待される結果を受信することを確認します。

J2C 活動化仕様の削除

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、システムで 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、この仕様を削除しなければならない場合があります。

アプリケーションのインストールに失敗したために仕様を削除する場合、Java Naming and Directory Interface (JNDI) 名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です。例えば、sca/SimpleBOCrsmA/ActivationSpec の場合、**SimpleBOCrsmA** がモジュール名です。

サービスが含まれるアプリケーションをインストールした後、何かの事情で構成を保管したものの、J2C 活動化仕様が不要な場合は、その仕様を削除します。

1. 削除する活動化仕様を見つけます。

仕様は「リソース・アダプター」パネルに含まれています。「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てください。

a. 「Platform Messaging Component SPI Resource Adapter」を見つけます。

このアダプターを見つけるには、スタンドアロン・サーバーの「ノード」スコープまたは、Network Deployment 環境の「サーバー」スコープにいる必要があります。

2. Platform Messaging Component SPI Resource Adapter に関連した J2C 活動化仕様を表示します。

リソース・アダプター名をクリックすると、次のパネルに関連した仕様が表示されます。

3. 削除するモジュール名に一致した「JNDI 名」の仕様をすべて削除します。
 - a. 該当する仕様の横にあるチェック・ボックスをクリックします。
 - b. 「削除」をクリックします。

システムは選択された仕様を表示から削除します。

変更を保管します。

SIBus 宛先の削除

SIBus 宛先とは、アプリケーションでサービスを使用可能にするための関連付けのことです。宛先を削除する必要がある場合があります。

アプリケーションのインストールに失敗したために宛先を削除する場合、宛先のモジュールの名前とインストールできなかったモジュールの名前が一致していることを確認してください。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です。例えば、sca/SimpleBOCrsmA/component/test/sca/cros/simple/cust/Customer の場合、SimpleBOCrsmA がモジュール名です。

サービスが含まれるアプリケーションをインストールした後、何らかの事情で構成を保管したものの、宛先が不要になった場合は、SIBus 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。サービスが含まれる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 活動化仕様の削除』を参照してください)。

1. 管理コンソールにログインします。
2. SCA システム・バスの宛先を表示します。

「サービス統合」>「バス」をクリックしてパネルにナビゲートします。

3. SCA システム・バスの宛先を選択します。

表示された **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. 除去するモジュールに一致するモジュール名を含む宛先を削除します。
 - a. 該当する宛先の横にあるチェック・ボックスをクリックします。
 - b. 「削除」をクリックします。

パネルには残った宛先だけが表示されます。

これらの宛先が作成されたモジュールに関連した J2C 活動化仕様を削除します。

拡張メッセージング・リソースの管理: 概要

拡張メッセージングはコンテナ管理メッセージングを使用可能にします。基本の Java Message Service (JMS) サポート、Enterprise Java Bean (EJB) コンポーネント・モデル、および EJB 2.0 メッセージ駆動型 Bean のサポートを拡張することで、既存のコンテナ管理パーシスタンスおよびトランザクション動作が使用可能になります。

拡張メッセージングは Bean 管理のメッセージング・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 管理の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ので、JMS の使用は単純化されます。

管理コンソール・インターフェースにより、拡張メッセージング・サービス、および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の構成が可能になります。

拡張メッセージングのすべての説明は、WebSphere Business Integration Server Foundation インフォメーション・センターで以下の項目を参照してください。

- 『拡張メッセージング: 概説』
- 『アプリケーションにおける拡張メッセージングの使用』

注: 拡張メッセージング・サービス機能は、WebSphere Process Server 6.0.1 では推奨されません。このリリースにおいて新規および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で置換するか、同等のメッセージング・テクノロジーで置換する必要があります。

関連タスク

『拡張メッセージング・サービスの使用可能化』

拡張メッセージング・サービスでは、コンテナ管理メッセージング (拡張メッセージング) をサポートするランタイム・サービスが提供されます。このサービスは、アプリケーション・サーバーの開始時に自動的に開始することも、手動で開始することもできます。

173 ページの『遅延応答を処理するリスナー・ポート拡張の構成』

174 ページの『拡張メッセージング・プロバイダーの管理』

174 ページの『新規入力ポートの追加』

177 ページの『新規出力ポートの追加』

拡張メッセージング・サービスの使用可能化

拡張メッセージング・サービスでは、コンテナ管理メッセージング (拡張メッセージング) をサポートするランタイム・サービスが提供されます。このサービスは、アプリケーション・サーバーの開始時に自動的に開始することも、手動で開始することもできます。

必要なセキュリティ・ロール: このタスクを完了するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。

2. 「サーバー」>「アプリケーション・サーバー」> *server_name* > 「拡張メッセージング・サービス」をクリックして、「拡張メッセージング・サービス」ページを表示します。
3. サーバーの始動時に拡張メッセージング・サービスが自動的に開始されるようにするには、「サーバー始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ます。
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」をクリックしてマスター・リポジトリに変更を保管します。
6. 変更内容を有効にするために、アプリケーション・サーバーの停止と再始動を実行します。

なお、『遅延応答を処理するリスナー・ポート拡張の構成』に説明されているように、「拡張メッセージング・サービス」ページを使用して、遅延応答を処理するリスナー・ポート拡張を構成することもできます。

遅延応答を処理するリスナー・ポート拡張の構成

アプリケーションのリスナー・ポートを、拡張メッセージング環境で遅延応答を処理するように拡張機能で構成することができます。

必要なセキュリティ・ロール: このタスクを完了するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. リスナー・ポートが定義および構成されていることを確認します。
2. 管理コンソールから、「サーバー」>「アプリケーション・サーバー」> *server_name* > 「拡張メッセージング・サービス」> 「リスナー・ポートの拡張」の順をクリックします。「リスナー・ポートの拡張」ページが開きます。
3. 「新規」をクリックして、新規リスナー・ポート拡張を作成します。「新規リスナー・ポート拡張 (New Listener Port Extension)」ページが開きます。
4. 「使用可能」チェック・ボックスを選択して、遅延応答処理を使用可能にします。
5. 「要求間隔」フィールドで、デフォルト値を受け入れるか、新規の値を指定します。要求間隔では、リスナー・ポートが遅延応答をチェックする頻度を指定します。
6. 「要求タイムアウト」フィールドで、デフォルト値を受け入れるか、新規の値を指定します。要求タイムアウト値では、リスナー・ポートが遅延応答を待つ時間を指定します。その時間後に受信した応答は廃棄されます。
7. 「リスナー・ポート」ドロップダウン・メニューを使用して、この拡張で使用するリスナー・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」をクリックしてマスター・リポジトリに変更を保管します。
10. 変更内容を有効にするために、アプリケーション・サーバーの停止と再始動を実行します。

拡張メッセージング・プロバイダーの管理

拡張メッセージング・プロバイダーは、コンテナ管理メッセージング (拡張メッセージング) と共に使用するために定義されたリソースを管理します。管理コンソールの「拡張メッセージング・プロバイダー」ページを使用して、各リソースの一般プロパティを表示し、編集するリソースを選択します。

拡張メッセージング・リソースは、以下のセル、ノード、またはサーバーの範囲で定義できます。

- セル・範囲: 最も一般的な範囲。セル・範囲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限り、すべてのノードおよびサーバーから参照することができます。
- ノード・範囲: ノード・範囲で定義された拡張メッセージング・リソースは、セル・範囲で定義された重複する定義をオーバーライドします。それらは、ノード内のサーバー・範囲でオーバーライドされていなければ、同一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・範囲: サーバー・範囲で定義された拡張メッセージング・リソースは、セル・範囲または親ノード・範囲で定義された重複する定義をオーバーライドします。これらは特定のサーバーからのみ参照できます。

範囲の詳細については、WebSphere Application Server for z/OS インフォメーション・センターを参照してください。

必要なセキュリティ・ロール: このタスクを完了するには、管理者、オペレーター、コンフィギュレーター、またはモニターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダーの管理」ページが開きます。
2. 適切な範囲の横にあるラジオ・ボタンをクリックします。
3. 「適用」をクリックします。ページの下部の「有効範囲」、「名前」、および「説明」フィールドは更新され、選択されたリソース・プロバイダー用の値を反映します。

これで、選択した拡張メッセージング・プロバイダーの入力ポート、出力ポート、またはその他のカスタム・プロパティを作成、変更、または削除することができます。

関連タスク

『新規入力ポートの追加』

177 ページの『新規出力ポートの追加』

新規入力ポートの追加:

セッション Bean から構成された受信側 Bean は、受信用の Java Message Service (JMS) 宛先のためのプロパティを定義するために入力ポートを必要とします。また、入力ポートでは、メッセージ選択およびメッセージ処理の詳細や、応答宛先の詳細も提供できます (必要に応じて)。

注: メッセージ駆動型 Bean から構成された受信側 Bean には入力ポートは必要ありません。そこに含まれる詳細は、配置済みのメッセージ駆動型 Bean およびメッセージ・リスナー・サービスに関連付けられます。

このタスクでは、入力ポートの初期プロパティを構成します。必要に応じて、ポートのプロパティは後で変更できます。

必要なセキュリティ・ロール: このタスクを完了するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダーの管理」ページが開きます。
2. 新規ポートを追加するリソース・プロバイダーのスコープを選択します。
3. 「適用」をクリックします。
4. 追加プロパティ・テーブルから「入力ポート」をクリックします。「入力ポート」ページが開きます。
5. 「新規」をクリックします。「入力ポート」構成ページが開きます。
6. 新規入力ポートに適切なプロパティを指定します。プロパティの設定について詳しくは、176 ページの『入力ポート設定』を参照してください。
7. 「OK」をクリックします。
8. プロンプトが出されたら、コンソール・タスクバーで、「保管」をクリックしてマスター・リポジトリに変更を保管します。
9. 変更内容を有効にするために、アプリケーション・サーバーの停止と再始動を実行します。

関連タスク

『入力ポートの構成の変更』

入力ポートの構成の変更:

必要に応じて既存の入力ポートの構成を変更できます。入力ポートは、受信用の Java Message Service (JMS) 宛先のプロパティを定義します。また、メッセージ選択およびメッセージ処理の詳細や、応答宛先の詳細も提供されます (必要に応じて)。

必要なセキュリティ・ロール: このタスクを完了するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダーの管理」ページが開きます。
2. 新規ポートを追加するリソース・プロバイダーのスコープを選択します。
3. 「適用」をクリックします。
4. 追加プロパティ・テーブルから「入力ポート」をクリックします。「入力ポート」ページが開きます。
5. 変更する入力ポートを選択します。「入力ポート」構成ページが開きます。ここに、そのポートの現在の構成プロパティが表示されます。

6. 入力ポートのプロパティを変更します。プロパティの設定については、『入力ポート設定』を参照してください。
7. 「OK」をクリックします。
8. プロンプトが出されたら、コンソール・タスクバーで、「保管」をクリックしてマスター・リポジトリに変更を保管します。
9. 変更内容を有効にするために、アプリケーション・サーバーの停止と再始動を実行します。

入力ポート設定: 入力ポートには以下の構成プロパティがあります。

有効範囲

拡張メッセージング・プロバイダーが定義される範囲。値は構成ファイルのロケーションを表します。

名前 管理目的で使用する入力ポートの名前。このフィールドの値はストリングにする必要があります。

JNDI 名

リソースの Java Naming and Directory Interface (JNDI) 名。このフィールドの値はストリングにする必要があります。

説明 管理目的で使用する入力ポートの説明。このフィールドの値はストリングにする必要があります。

このフィールドはオプションです。

カテゴリ

リソースの分類またはグループ化の際に使用するカテゴリ・ストリング。このフィールドの値は、長さが 1 から 30 の間の ASCII 文字によるストリングにする必要があります。

このフィールドはオプションです。

JMS 接続ファクトリー JNDI 名

入力ポートで使用される Java Message Service (JMS) 接続ファクトリーの JNDI 名 (例えば、jms/connFactory1)。このフィールドの値はストリングにする必要があります。

JMS 宛先 JNDI 名

入力ポートで使用される JMS 宛先の JNDI 名 (例えば、jms/destn1)。このフィールドの値はストリングにする必要があります。

JMS 肯定応答モード

メッセージの肯定応答で使用される JMS モード。このプロパティは、Bean 管理トランザクション区分を使用するメッセージ駆動型 Bean (言い換えると、トランザクション・タイプを Bean に設定) の場合にのみ使用されます。

このフィールドの有効な値は以下のとおりです。

- Auto Acknowledge: セッションは、次のどちらかの場合に自動的にメッセージに応答します。
 - セッションが、呼び出しから正常に戻ってメッセージを受信する場合
 - 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った場合

- 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。このため、JMS が失敗すると、重複メッセージが送達されることとなります。

デフォルト・モードは Auto Acknowledge です。

宛先タイプ

JMS リソース・タイプ。このフィールドの有効な値は以下のとおりです。

- QUEUE: 受信側 Bean はキュー宛先からメッセージを受信します。
- 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 QUEUE です。

サブスクリプション耐久性

JMS トピック・サブスクリプションが永続的かどうかを指定します。このフィールドの有効な値は以下のとおりです。

- Durable: サブスクライバーは、JMS によって保存された固有の 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、メッセージを受信するかメッセージが期限切れになるまで、サブスクリプションのメッセージを保管します。
- Nondurable: 非永続的サブスクリプションは、所有するサブスクライバーの存続時間中は持続します。クライアントは、サブスクライバーがアクティブの間だけ、トピックで公開されたメッセージを確認します。サブスクライバーが非アクティブの場合、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逃します。

デフォルト値は Durable です。

このフィールドは、JMS 宛先タイプがトピックの場合にのみ必要です。

応答 JMS 接続ファクトリー JNDI 名

応答で使用する JMS 接続ファクトリーの JNDI 名。このフィールドの値はストリングにする必要があります。

応答 JMS 宛先 JNDI 名

応答で使用する JMS 宛先の JNDI 名。このフィールドの値はストリングにする必要があります。

新規出力ポートの追加:

出力ポートには、送信側 Bean で送信するメッセージの宛先を定義するのに必要なプロパティを指定します。また、応答が期待される場合は、オプションのプロパティも指定します。出力ポートは配置時に送信側 Bean に関連付けられます。

このタスクでは、出力ポートの初期プロパティを構成します。必要に応じてこのプロパティを後で変更できます。

必要なセキュリティ・ロール: このタスクを完了するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダーの管理」ページが開きます。
2. 新規ポートを追加するリソース・プロバイダーのスコープを選択します。
3. 「適用」をクリックします。
4. 追加プロパティ・テーブルから「出力ポート」をクリックします。「出力ポート」ページが開きます。
5. 「新規」をクリックします。「出力ポート」構成ページが開きます。
6. 新規出力ポートに適切なプロパティを指定します。プロパティの設定について詳しくは、『出力ポート設定』を参照してください。
7. 「OK」をクリックします。
8. プロンプトが出されたら、コンソール・タスクバーで、「保管」をクリックしてマスター・リポジトリに変更を保管します。
9. 変更内容を有効にするために、アプリケーション・サーバーの停止と再始動を実行します。

関連タスク

『出力ポートの構成の変更』

出力ポートの構成の変更:

必要に応じて既存の出力ポートの構成を変更できます。出力ポートは、送信されるメッセージの宛先を定義し、応答が予想される場合は宛先を指定します。

必要なセキュリティ・ロール: このタスクを完了するには、管理者またはコンフィギュレーターとしてログインしている必要があります。

1. 管理コンソールから、「リソース」>「拡張メッセージング・プロバイダー」の順にクリックします。「拡張メッセージング・プロバイダーの管理」ページが開きます。
2. 新規ポートを追加するリソース・プロバイダーのスコープを選択します。
3. 「適用」をクリックします。
4. 追加プロパティ・テーブルから「出力ポート」をクリックします。「出力ポート」ページが開きます。
5. 変更する出力ポートを選択します。「出力ポート」構成ページが開きます。ここに、そのポートの現在の構成プロパティが表示されます。
6. 出力ポートのプロパティを変更します。プロパティの設定について詳しくは、『出力ポート設定』を参照してください。
7. 「OK」をクリックします。
8. プロンプトが出されたら、コンソール・タスクバーで、「保管」をクリックしてマスター・リポジトリに変更を保管します。
9. 変更内容を有効にするために、アプリケーション・サーバーの停止と再始動を実行します。

出力ポート設定: 入力ポートには以下の構成プロパティがあります。

有効範囲

このフィールドは拡張メッセージング・プロバイダーの範囲を指定します。値は構成ファイルのロケーションを表します。

このフィールドは編集できません。

名前 管理目的で使用する出力ポートの名前。このフィールドの値はストリングにする必要があります。

JNDI 名

出力ポートの Java Naming and Directory Interface (JNDI) 名。このフィールドの値はストリングにする必要があります。

説明 管理目的で使用する出力ポートの説明。このフィールドの値はストリングにする必要があります。

このフィールドはオプションです。

カテゴリ

リソースの分類またはグループ化の際に使用するカテゴリ・ストリング。このフィールドの値は、最長 30 文字の ASCII 文字によるストリングにする必要があります。

このフィールドはオプションです。

JMS 接続ファクトリー JNDI 名

出力ポートで使用される Java Message Service (JMS) 接続ファクトリーの JNDI 名 (例えば、jms/connFactory1)。このフィールドの値はストリングにする必要があります。

JMS 宛先 JNDI 名

出力ポートで使用される JMS 宛先の JNDI 名。このフィールドの値はストリングにする必要があります (例えば、jms/destn1)。

JMS 送達モード

メッセージの送達で使用される JMS モード。値は、以下のいずれかでなければなりません。

- 永続的: 宛先に書き込まれるメッセージは永続的
- 非永続的: 宛先に書き込まれるメッセージは非永続的

デフォルト値は「永続的」です。

JMS 優先順位

キュー宛先のメッセージ優先順位。値は 0 から 9 までの整数でなければなりません。デフォルト値は 4 です。

JMS 存続時間

メッセージがキュー内に残るミリ秒単位の時間。指定された時間が経過すると、メッセージの有効期限が切れます。

値は 0 から n までの整数でなければなりません。

- 0: メッセージはタイムアウトにならない
- n : メッセージは n ミリ秒後にタイムアウトになる

デフォルト値は 0 です。

JMS メッセージ ID 使用不可

システムが JMS メッセージ ID を生成するかどうかを指定します。値は、以下のいずれかでなければなりません。

- 選択済み: システムは JMS メッセージ ID を生成しない
- クリア: システムは JMS メッセージ ID を自動的に生成する

デフォルトでは、JMS メッセージ ID は生成されます。

JMS メッセージ・タイム・スタンプ使用不可

システムが JMS メッセージ・タイム・スタンプを生成するかどうかを指定します。値は、以下のいずれかでなければなりません。

- 選択済み: メッセージ・タイム・スタンプを送信済みメッセージに追加しない
- クリア: メッセージ・タイム・スタンプを送信済みメッセージに自動的に追加する

デフォルトでは、メッセージ・タイム・スタンプは追加されます。

応答 JMS 接続ファクトリー JNDI 名

出力ポートによって処理される応答で使用する JMS 接続ファクトリーの JNDI 名 (例えば、jms/connFactory1)。このフィールドの値はストリングにする必要があります。

応答 JMS 宛先 JNDI 名

出力ポートによって処理される応答で使用する JMS 宛先の JNDI 名 (例えば、jms/destn1)。このフィールドの値はストリングにする必要があります。

WebSphere ESB の管理

WebSphere ESB の管理には、サービス・バス環境の管理、メディエーション・モジュールのデプロイと管理、サービス統合およびメディエーション・モジュールで使用されるリソースの管理、およびメディエーション・モジュールと相互作用するクライアントおよびアダプターの管理が関係しています。管理者は、基盤となる WebSphere Application Server の全機能を管理することもできます。

管理者は、WebSphere ESB により、サービス・アプリケーションとしてのメディエーション・モジュールのデプロイメントをサポートする、ESB サーバーおよびサービス統合バスの環境を作成します。WebSphere ESB をインストールすると、ESB にデプロイされたサービス・アプリケーションで使用する 2 つのサービス統合バスが得られます。1 つのバスでの 1 台のサーバーから始め、オプションとして、複数のサーバーまたはサーバー・クラスターを構築することによって容量や拡張可能性を追加することもできます。必要であればバスを追加し、別の ESB を作成し、ESB に接続するアプリケーションをデプロイし、WebSphere MQ との統合を可能にすることができます。

管理者は、サーバーおよびバス環境にメディエーション・モジュールを (SCA モジュールとして) デプロイできます。それらのモジュールにより、ユーザーがデプロイしたモジュールを表示し、要求が正常に処理されているかどうかをモニターできます。管理者は、メディエーション・モジュールを開始または停止できます。他の方法でモジュールを管理することもでき、例えば、モジュールの構成を変更したり、モジュールを停止または更新したり、あるいはその活動を管理したりすること

ができます。また管理者は、メディエーション・モジュールの SCA インポートに変更を加え、例えば、1 つのモジュールを別のモジュールに宛先変更できます。これによりモジュールは、異なるサービス・プロバイダーを呼び出すことができ、モジュールの再ビルドと再デプロイを行わなくても、サービス要求と応答をさまざまな方法で処理できます。

管理者は、WebSphere 管理コンソール、WebSphere 管理 (wsadmin) スクリプト・プログラム、コマンド行ツール、および管理プログラムなど、さまざまなツールを使用して WebSphere ESB を管理できます。

管理用タスクの主な記述は、管理コンソールの使用に基づいています。管理コンソール内の各タスクは、1 つ以上のパネルでサポートされます。タスク・フィルターを使用すれば、完了させたいタスクに最適なパネルのセットを表示し、それらのパネルでのみ集中して作業することができます。

すべて これにより、すべての管理コンソール・パネルが表示されます。これは、WebSphere ESB のすべてのパーツ、および基盤となる WebSphere Application Server の管理に関心のある管理者に最適です。

エンタープライズ・サービス・バス機能の管理について詳しくは、以下にリストする関連タスクを参照してください。

アプリケーションの統合

これにより、以下のタスク領域に適したパネルが表示されます。

- メディエーション・モジュールとサービス・アプリケーションのデプロイメントをサポートするために必要とされるサービス統合バス、サーバー、サーバー・クラスター、メッセージング・エンジン、およびネットワーク・トポロジーの構成の調整
- デプロイされたサービス・アプリケーションおよびメディエーション・モジュールに必要なリソース (例えば、JMS 接続ファクトリーや Common Event Infrastructure プロファイル) の作成
- メディエーション・モジュールとサービス・アプリケーションの動作制御

これは、227 ページの『サービス・アプリケーションの管理』の説明にあるように、メディエーション・モジュールをサービス・アプリケーションとしてデプロイして管理することに関心のある管理者に最適です。

サーバーおよびバス

これにより、以下のタスク領域に適したパネルが表示されます。

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

これは、サービス・アプリケーションおよびメディアーション・モジュールのデプロイメントをサポートするために必要なサーバーおよびバス環境の管理に関心のある管理者に最適です。管理には、ネットワークおよびバス・トポロジーの定義、適切なリソースの定義、ランタイム・システムのモニター、およびランタイム・エラーが発生した場合はそのトラブルシューティングが含まれます。バスおよびサービス環境の管理について詳しくは、『バス環境の管理』を参照してください。

バス環境の管理

Service Integrator では、サーバーおよび、エンタープライズ・サービス・バスを稼働するバス環境を考慮することなく SCA モジュールをデプロイできます。ただし、管理者はサーバーおよびバス環境を管理する必要がある場合があります。例えば、サーバーの始動および停止、SCA モジュールに提供されるサービス品質の変更、サーバー容量の追加、さらに分散されたバス環境の導入などです。

WebSphere ESB の完全 (デフォルト) インストールを選択すると、デフォルトのバス環境が作成されます。このバス環境は、SCA モジュールのデプロイに使用する 2 つのサービス統合バスに割り当てられるシングル・サーバーを構成します。このサーバーを使用して、サブレット、Enterprise Java Bean、JMS 宛先など、J2EE のコンポーネントおよびリソースをデプロイすることもできます。

デフォルトのバス環境が、ご使用の SCA モジュールに適切な場合もあります。ただし、バス環境のトポロジーを変更する必要がある場合があります。例えば、サーバー容量を追加したり、別の部門用にさらに分散されたバス環境を導入したり、テスト機能と実動機能を分離する場合です。

バス環境の管理には、WebSphere ESB のいくつかの設定が必要になります。場合によってはより大規模なシステムの一部として、一般的には実稼働環境または現実的なテスト環境として設定します。これには、いくつかのインストールおよびカスタマイズ・アクティビティ、バス・トポロジー計画、および製品構成の作成が含まれます。これは、サービス統合バス、サーバー、そのリソースの管理、セルおよびノードの管理可能論理ドメインの設定および管理、クラスタリングと高可用性の構成によるワークロードのバランス方法に重点を置いています。

WebSphere ESB でのバス環境は、WebSphere Application Server により提供されるサービス統合テクノロジーを使用します。ここでの各トピックでは、バス環境の管理に必要なサービス統合テクノロジーに関連するタスクについて説明します。

- サーバーおよびバス環境入門
- SCA.SYSTEM バス・トポロジーの管理
- SCA ランタイム用サーバーまたはクラスタの構成
- サーバー管理
- サーバー・クラスタを使用したワークロード・バランシング
- バス・トポロジーのその他の用途

バス環境入門

Service Integrator では、サーバーおよび、エンタープライズ・サービス・バスを稼働するバス環境を考慮することなく SCA モジュールをデプロイできます。ただし、サーバーおよびバス環境の管理が必要となる場合があるため、管理者はその管理について理解する必要があります。

ここでのトピックでは、バス環境を管理する背景について説明します。初めてバス環境の管理を始める場合に役立ちます。バス環境の管理について十分な知識がある場合は、182 ページの『バス環境の管理』のサブトピックとして提供している他の各タスクの説明に直接進んでください。

バス環境の概要:

バス環境は、1 つ以上のサービス統合バス、ESB サーバー、およびそのリソースで構成されており、セルとノードからなる論理的な管理可能ドメインに編成されます。

WebSphere ESB の完全 (デフォルト) インストールを作成した場合、サーバー構成を行うことなく Service Component Architecture (SCA) モジュールをデプロイできるスタンドアロン・サーバーがインストールされます。

ただし管理者は、さらにバス環境での作業を行う場合があるので、環境についていくらか詳細を知っていれば役に立ちます。

- SCA ランタイム (メディエーション・モジュールによって利用される) は、コンポーネントとモジュールの間の非同期相互作用をサポートするための堅固なインフラストラクチャーとして、SCA.SYSTEM サービス統合バス上のキューを使用します。このキューは、SCA.SYSTEM バスのメンバーとして、サーバーによりホストされます。
- ESB サーバーは、WebSphere ESB でメディエーション・モジュールおよびサービス・アプリケーションを実行するのに必要な統合テクノロジー、インフラストラクチャー・サービス、構成、およびランタイム管理を提供します。バス・メンバーとして、サーバーには、SCA.SYSTEM バスのコア・メッセージング機能を提供するメッセージング・エンジンがあります。

サーバーおよび SCA.SYSTEM は両方ともデフォルト・プロパティで構成され、これはご使用の SCA モジュールをデプロイおよび実行するのに適している場合があります。

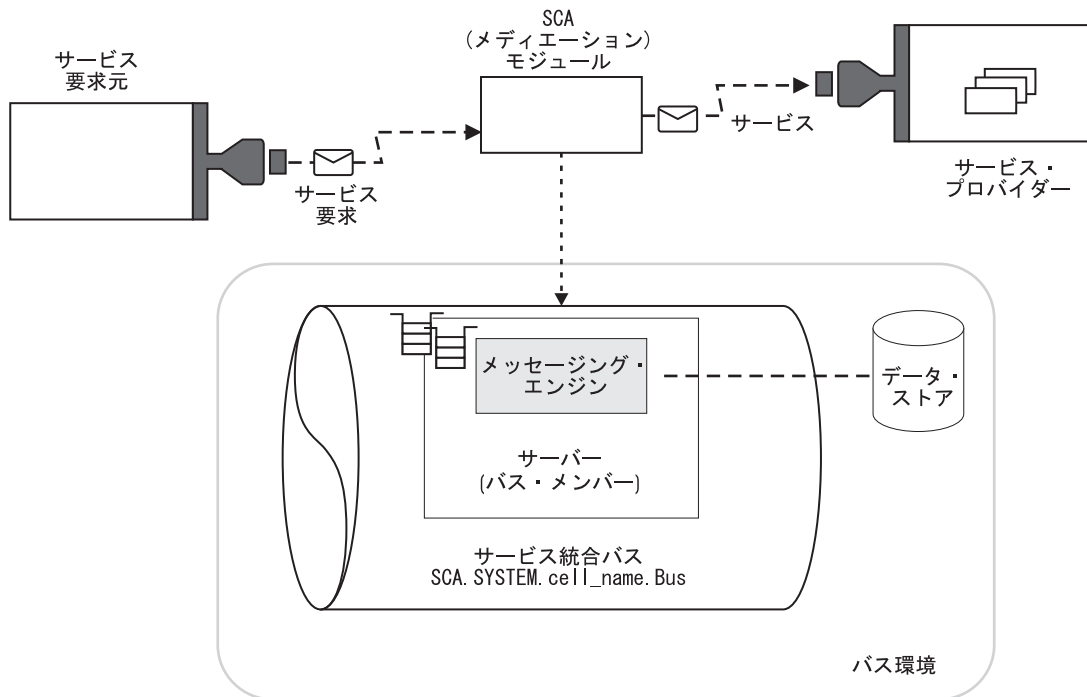


図7. 1つのサーバーが SCA.SYSTEM サービス統合バスに割り当てられたバス環境：バス・メンバーとして、サーバーには1つのメッセージング・エンジンが割り当てられており、このエンジンには状態データおよびメッセージを保管するためのデータ・ストアがあります。この図は、メディエーション・モジュールがバス環境にデプロイされ、バス・メンバーに割り当てられていることも示しています。

より高度な使用法として、スタンドアロン・サーバーのバス環境の構成を変更したり、デプロイメント・マネージャー・セルのバス環境を作成したりできます。以下に例を挙げます。

- セキュアで確実な配信（メッセージが失われないことが保証され、安全に移送される）からベスト・エフォート（システム障害が発生した場合はメッセージが失われる可能性がある）まで、さまざまなサービス品質を構成できます。
- メディエーション・モジュールをホストするサーバーをいくつか提供するようにデプロイメント・マネージャー・セルをセットアップすることもできます。このようにすると、スケーラビリティの確保、処理可能なクライアント接続数の拡大、およびメッセージ・スループットの向上、という利点が得られます。サーバー・クラスターも作成できます。サーバー・クラスターを使用すると、1グループのサーバーをまとめて管理でき、これらのサーバーをワークロード管理に加えることができます。
- 完全なバス環境は、各部門ごとに別個の管理可能ドメインを提供したり、テスト用設備と実動用設備を区別するために、いくつかのスタンドアロンおよびデプロイメント・マネージャーのプロファイルで構成されている場合があります。各プロファイルには独自の SCA.SYSTEM サービス統合バスがあります。

SCA モジュールに使用される SCA.SYSTEM バスのほかに、別のサービス統合バスを作成し、それを使用して、モジュールが提供するサービス統合ロジックをサポートすることもできます。例えば SCA.APPLICATION.cell_name.Bus は、JMS バインディングでデプロイされたモジュールの JMS キュー宛先やその他の JMS リソースを定義するために提供および使用されます。

他のバスを作成して、WebSphere Application Server の場合と同様に使用できます。例えば、WebSphere ESB 内でサービス要求元やサービス・プロバイダーとして動作するアプリケーション用、あるいは WebSphere MQ へのリンク用などです。WebSphere ESB デプロイメント・マネージャーを使用して、WebSphere Application Server 上にデプロイされたアプリケーションおよびモジュール用の個々のアプリケーション・サーバーを管理することもできます。

アプリケーション・サーバー:

WebSphere ESB は、WebSphere Application Server を組み込んでいます。これにより、エンタープライズ Bean、Web サービス、サーブレット、JCA コネクタ、および非同期 Bean などの J2EE アプリケーション・コンポーネントをサポートする、高性能で、セキュアかつ管理可能なミドルウェア・コンテナを提供します。

WebSphere ESB は、サービス・アプリケーションのデプロイメントおよび管理で使用する機能を提供します。例えば、ブラウザ・ベースの管理コンソール、コマンド・スクリプト、(サービス・バス、サーバー、およびクラスターの) トポロジー管理、JMX と呼ばれる標準管理プログラミング・インターフェース・セット、およびホストされるアプリケーション・コンポーネントとインフラストラクチャーすべてを包含する管理モデルなどがあります。

サーバー・インフラストラクチャーでは、ほとんどのミドルウェアに共通のネットワーク I/O、キューイング、スケジューリング、スレッド化、およびディスパッチング機能を提供します。また、さまざまなトランスポートとプロトコルをサポートすることにより、プロトコル・セマンティクスおよびメッセージ・エンコードのコーディングを担当する開発者の負担を軽減します。

WebSphere ESB は、通常はインプリメント困難な高品質のサービスをランタイム機能で提供することにより、アプリケーションを拡張および強化できます。例えば、アトミック・トランザクションや呼び出し元の特定のセキュリティー・ロールなどのホスティング・コンテナの要件は、サービスのメタデータ中で宣言として表現されます。パフォーマンスのモニターと測定、ワークロード・ルーティング、およびワークロード管理などの運用機能は、管理上使用可能にすることができます。

WebSphere ESB は、J2EE プログラミング・モデルの標準プログラミング・インターフェースをインプリメントする、アプリケーション・ランタイム・サービスを提供します。例えば、アプリケーションのトレースおよびロギングのインターフェースは、サーバー・ランタイムのインターフェースと統合され、問題診断についての一貫性のあるビューを提供します。セキュリティー・インターフェースは WebSphere ESB セキュリティーを拡張してアプリケーション固有のビジネス要件に対処します。ネーム・スペース・サービスは、サービスのディスカバリーおよびバインディングを可能にし、管理ドメイン全体で持続できます。

WebSphere ESB でデプロイされるサービス・インプリメンテーションでは、明示的にコード化したりサーバー・インフラストラクチャーを認識したりしなくても、これらの利点をすべて享受できます。

サービス統合バス:

サービス統合バスは、WebSphere ESB でデプロイされるメディエーション・モジュールと相互作用エンドポイントのリソースを構成する際の有効範囲を規定します。

要確認: エンタープライズ・サービス・バスは、単なるサービス統合バスではありません。エンタープライズ・サービス・バスはむしろ、SCA で記述された相互作用エンドポイントの統合ロジックおよび接続に使用される、アーキテクチャーです。サービス統合バスは、WebSphere ESB を構築するときの基盤である WebSphere Application Server によって提供されるテクノロジーの一部です。ご使用のエンタープライズ・サービス・バスには、複数のサービス統合バス、WebSphere MQ、および他のテクノロジーが組み込まれている場合があります。

エンタープライズ・サービス・バスの略称としては、*ESB* という用語を使用します。サービス統合バスの略称としては、*バス* という用語を使用しません。

WebSphere ESB では、バスにより、特定品質の相互作用サービスを提供するエンドポイント間でのメッセージ・ルーティングが可能になり、必要であれば一時的にメッセージを持続させることができます。セキュアで確実な配信 (メッセージが失われないことが保証され、安全に移送される) からベスト・エフォート (システム障害が発生した場合はメッセージが失われる可能性がある) まで、さまざまなサービス品質を構成できます。エンドポイントをインプリメントする人は、SCA モジュールのエクスポートとインポートで注釈を宣言することにより、望ましいサービス品質を選択します。サービス品質が指定されていない場合、WebSphere ESB はデフォルト設定を適用します。

WebSphere ESB をインストールすると、SCA モジュールをデプロイするための SCA.SYSTEM バスが作成されます。このバスの構成が SCA モジュールにとって不適切な場合、または他の用途のためにサービス統合バスが必要な場合は、マルチ・サーバー・バス、別々のサーバーを使用するいくつかのシングル・サーバー・バス、および WebSphere MQ にリンクされたバスなどのさまざまなバス環境から選択することができます。

多くのシナリオでは、単純なバス・トポロジーが必要とされます。例えば、シングル・サーバーが考えられます。複数のサーバーを単一のバスに追加することにより、接続ポイントの数を増やすことができます。サーバー・クラスターをバスのメンバーとして追加すれば、スケーラビリティを向上させ、高可用性を実現できます。ただしサーバーは、バスに接続するのにバス・メンバーである必要はありません。より複雑なバス・トポロジーでは、複数のバスが構成されており、相互接続で複雑なネットワークを構成している場合があります。企業は、組織上の理由で複数のバスを配置することもあります。例えば、独立部門をいくつか持つ企業では、部署ごとに別々に管理されるバスを持つ場合などです。

サーバー・クラスター:

サーバー・クラスターは、一緒に管理され、ワークロード管理に参加する一群のサーバーです。クラスターのメンバーになっているサーバーは、別々のホスト・マシンに配置することが可能です。

クラスターに属するサーバーはクラスター・セットのメンバーであり、それらのサーバーすべてには同一のアプリケーション・コンポーネントがデプロイされている必要があります。これらのサーバーで稼働するよう構成されたアプリケーション以外に、クラスター・メンバー間で構成データを共有する必要はありません。あるク

クラスター・メンバーは大規模なマルチプロセッサ・エンタープライズ・サーバー・システムで稼働し、同一クラスターの別メンバーはより小規模なシステムで稼働する場合があります。この 2 つのクラスター・メンバーのサーバー構成の設定はまったく異なりますが、これらに割り当てられたアプリケーション・コンポーネントの領域だけは例外です。この構成領域に関しては、これらのサーバーは同じです。これにより、すべてのワークロードを単一アプリケーション・サーバーで処理するのではなく、クライアント作業をクラスターのすべてのメンバー間で分散させることができます。

WebSphere ESB では、基盤となる WebSphere Application Server Network Deployment によってサーバー・クラスターをサポートします。唯一の違いは、既存のアプリケーション・サーバーをテンプレートにしてクラスターを作成できないことです。

サーバー・クラスターについて詳しくは、関連するトピックを参照してください。

バス環境の選択

WebSphere ESB の完全 (デフォルト) インストールを選択すると、シングル・サーバーのバス環境が設定されます。これがご使用のサービス・アプリケーションに適合しない場合は、デプロイメント・マネージャー・セル内の複数サーバー・バス、異なるサーバーを使用する複数のシングル・サーバー・バス、アプリケーション用のバス、WebSphere MQ にリンクするバスなど、バス・トポロジーの選択項目をインクルードするバス環境を作成できます。

WebSphere ESB の完全 (デフォルト) インストールを選択すると、固有の管理可能ドメイン内にセルとして知られるスタンドアロン・ノードが作成されます。このノードは、SCA モジュールをデプロイするためにセル用の SCA.SYSTEM バスに割り当てられた 1 つのサーバーをホストします。

SCA モジュールに必要なサーバーが 1 つだけの場合は、完全 (デフォルト) インストールの SCA.SYSTEM バスを使用できます。サービス・アプリケーションに複数のサーバーが必要な場合は、さまざまなバス・トポロジーからいずれかを選択する必要があります。

SCA モジュールをデプロイする前に、エンタープライズ・サービス・バスに必要なバス環境を選択してください。これは、SCA モジュールのデプロイによって、どの WebSphere ESB プロファイルを作成する必要があるか、メッセージング・エンジンでどのデータベースを使用するかなどといった、インストール関連のアクションが影響を受けるためです。

- デフォルトのスタンドアロン・サーバー・プロファイルで WebSphere ESB をインストールし、提供された SCA.SYSTEM バスで開始した後、必要に応じてデプロイメント・マネージャー・プロファイルおよび管理対象ノードのプロファイルを作成し、より高度なバス環境を実現できます。
- バス環境において複数のサーバーを使用するには、デプロイメント・マネージャー・セル内の管理対象ノードのプロファイルを使用する必要があります。
- WebSphere ESB をインストールする前にバス環境を選択し、選択したバス環境への最適なサポートを得るために必要なプロファイルをインストールできます。

SCA モジュールに提供されている SCA.SYSTEM バスを使用する以外にも、別のアプリケーション・サーバーおよびサービス統合バスを作成して別のアプリケーションやモジュールをサポートしたり、WebSphere MQ ネットワークに接続したりできます。この一連のトピックは主に、SCA モジュールをサポートするための SCA.SYSTEM バスの使用に重点を置いています。

別のサービス統合バスを WebSphere Application Server Network Deployment などを使用する場合についての説明は、WebSphere Application Server トピックへのリンクにより参照できます。

バス環境を選択するには、以下の点および以下に示すサブトピックで説明されているバス・トポロジーの記述を考慮します。

- バスに配置されたモジュールに必要なクライアント接続数およびスループットを考慮します。

その目的は、クライアントで認識される、モジュールのパフォーマンスが低下し始めるポイントを識別することです。

- それ以上の新たなクライアント接続が行われるとパフォーマンスが低下し始める、バスへのクライアントの同時接続数。
- それ以上の新たな要求がメッセージング・エンジンを介して送信されるとパフォーマンスが低下し始める、メッセージング・エンジンを通過する要求および応答の数。

サーバーが実行しているホストの特性や配置するモジュールの特質、その他の要因によって方式は異なるため、すべての環境に適用できる 1 つの特定の方式はありません。

シングル・サーバー・バスを使用している場合に、クライアント接続数に起因してパフォーマンスが低下したり、またはスループットが低下し始めたときは、いくつかの方法によりバス環境の容量を増大できます。

- スタンドアロン・プロファイルで、同じサーバーを使用する複数のシングル・サーバー・バスを作成します。こうすることで、クライアント接続は複数のバスに分散されますが、要求のスループットは 1 つのサーバーに依存したままです。
- さらにクライアント接続および要求のスループットの容量を増大させるには、複数のバスに分散された複数のサーバーを使用します。(1 つ以上のバスに分散された複数のサーバーを使用するには、デプロイメント・マネージャー・セル内の管理対象ノードに関するサーバー・プロファイルが必要です。)
- メッセージング・エンジンを通過する要求のサイズを考慮します。

各メッセージング・エンジンは、要求および要求関連データを含む 2 つのメモリー・バッファーを管理します。十分なスペースがない場合、メッセージング・エンジンはバッファーにデータを追加するときに、バッファーに含まれる既存のデータを廃棄してスペースを確保することがあります。

実行中のメッセージング・エンジンが許容されている以上にバッファーからデータを廃棄する場合があります。その場合は、別のサーバーをバスに追加して、別のメッセージング・エンジンを提供します。または、異なるサーバーをそれぞれのバス・メンバーとして使用する複数のシングル・サーバー・バスを作成するこ

ともできます。各サーバー内のメッセージング・エンジンは個別のメモリー・バッファ・セットおよび個別のデータ・ストアを使用します。(1 つ以上のバスに分散された複数のサーバーを使用するには、デプロイメント・マネージャー・セル内のノードに関するサーバー・プロファイルが必要です。)

- ご使用の各サービス・アプリケーションに異なるサービス品質を適用するかどうかを考慮します。

各バスは、サービス品質およびその他のプロパティの固有の構成を持っています。複数のバスを作成して、それぞれ異なるサービス品質で構成し、各モジュールを適切な構成のバスにデプロイできます。

- バス内で複数のサーバーを使用するその他の理由を考慮します。

アプリケーションによっては、1 つのサーバーのみで構成されるサービス統合バスが適切な場合もあります。しかし、バス内で複数のサーバーを使用する(各サーバーがメッセージング・エンジンを提供する)ことで、次のような利点が得られます。

- メッセージング・ワークロードを複数のサーバーに分散します。
 - 要求処理をリクエスター・アプリケーションに近い場所に配置して、ネットワーク・トラフィックを削減します。例えば、送信アプリケーションと受信アプリケーションが同じサーバー・プロセス内で実行している場合は、それらのアプリケーション間を移動するすべての要求を、リモート・サーバー内で実行しているメッセージング・エンジンに発送する方法は非効率です。
 - システム障害やリンク障害時の可用性が向上します。これにより、**Single Point of Failure** を除去し、必要な場合には、2 つのサーバー間のストア・アンド・フォワードが可能になります。
 - スケーラビリティが向上します。
 - すべて単一のメッセージング・エンジンに接続するネットワーク・ホストの能力を制限するファイアウォールまたはその他のネットワーク制限に対応できます。
- 複数の **SCA.SYSTEM** バス環境を使用するその他の理由を考慮します。

各サービス統合バスは個別の構成を持つため、それぞれのバスを個別のモジュールに適切な異なる構成にできます。例えば、いくつかのバスをセキュリティーのある実稼働環境用に構成し、それ以外のバスをセキュリティーなしのテスト用に構成できます。

また、複数のバスを作成してモジュール管理を分離できます。例えば、個別の管理セルとその **SCA.SYSTEM** バスを組織内の異なる部門で使用したり、テスト機能と実動機能に分けて使用したりすることもできます。

SCA.SYSTEM バスに加えて、別のバスを他のアプリケーションで使用するように作成したり、バスを接続してバス間のメッセージングを可能にしたりできます。異なる組織内のバスも接続できます。バスを相互に接続すると、アプリケーションから別のバスに存在するアプリケーションへのメッセージ送信や、他のバスに提供されているリソースの使用が可能になります。バス間リンクの構成により許可されていれば、公開されたメッセージを複数のバスに送信することもできます。

- **SCA** 以外のサービス統合バスを使用する理由を考慮します。

SCA モジュールに使用される SCA.SYSTEM バスのほかに、別のサービス統合バスを作成し、それを使用して、モジュールが提供するサービス統合ロジックをサポートすることもできます。例えば SCA.APPLICATION.cell_name.Bus は、JMS バインディングでデプロイされたモジュールの JMS キュー宛先やその他の JMS リソースを定義するために提供および使用されます。

他のバスを作成して、WebSphere Application Server の場合と同様に、WebSphere ESB 内でサービス要求元やプロバイダーとして動作するアプリケーション用に使用したり、WebSphere MQ にリンクしたりできます。WebSphere ESB デプロイメント・マネージャーを使用して、WebSphere Application Server 上にデプロイされたアプリケーションおよびモジュール用の個々のアプリケーション・サーバーを管理することもできます。

- SCA モジュールをサポートしないアプリケーション・サーバーを使用するかどうかを考慮します。

WebSphere ESB のデプロイメント・マネージャー・セルは、WebSphere Application Server サーバーを実行するアプリケーション・サーバー・ノードをインクルードできます。これらのアプリケーション・サーバーを、WebSphere Application Server によりサポートされるアプリケーションおよびモジュールに使用できます。WebSphere Application Server のサービス統合テクノロジーを活用しない場合は、アプリケーション・サーバーをサービス統合バスに追加する必要はありません。

提供される SCA* バス:

WebSphere ESB のスタンドアロン・プロファイルまたはデプロイメント・マネージャー・プロファイルを作成すると、2 つのサービス統合バスを基にしたエンタープライズ・サービス・バスを使用できるようになります。これらのバスは、名前が SCA で始まり、SCA モジュールをデプロイする際に使用できます。

SCA.SYSTEM.cell_name.Bus

エンタープライズ・サービス・バスでサービス統合ロジックを提供するメディエーション・モジュールなどの SCA モジュールのキュー宛先をホストするために使用されるバスです。SCA ランタイム (メディエーション・モジュールによって利用される) は、SCA.SYSTEM バスのキュー宛先を堅固なインフラストラクチャーとして使用して、コンポーネントとモジュールの間の非同期相互作用をサポートします。

SCA.APPLICATION.cell_name.Bus

このバスを使用して、JMS バインディングを使用してデプロイされるモジュールのリソースを作成します。このバスは、SCA モジュールのデプロイ以外の目的でサービス統合バスを作成する場合の一例です。

SCA.SYSTEM バスは、キュー宛先などのリソースがメディエーション・モジュールと相互作用のエンドポイント用に構成されるとき有効範囲を提供します。バスにより、特定品質の相互作用サービスを提供するエンドポイント間でのメッセージ・ルーティングが可能になり、必要であれば一時的にメッセージを持続させることができます。セキュアで確実な配信 (メッセージが失われないことが保証され、安全に移送される) からベスト・エフォート (システム障害が発生した場合はメッセージが失われる可能性がある) まで、さまざまなサービス品質を構成できます。エンドポイントをインプリメントする人は、SCA モジュールのエクスポートとインポート

で注釈を宣言することにより、望ましいサービス品質を選択します。サービス品質が未指定の場合、WebSphere ESB はデフォルト設定を適用します。

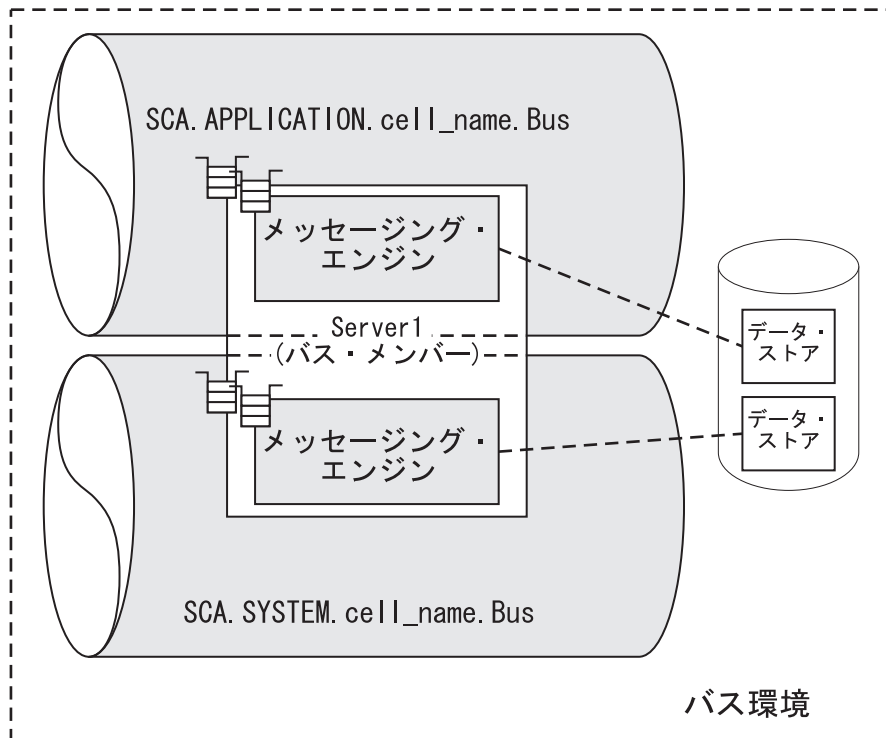


図8. スタンドアロン・サーバー・プロファイルの SCA.* バス： WebSphere ESB のインストール時に作成されるサービス統合バス。

スタンドアロン・プロファイルには、バスで 1 つのメッセージング・エンジンとなるバス・メンバーが 1 つあり、このトポロジーは、一部のアプリケーションにとっては適切です。

単一のメッセージング・エンジンは、クライアント接続の数が超過する場合、1 つのメッセージング・エンジンでメッセージのスループット率を維持できない場合、またはメッセージのサイズがメッセージング・エンジンによって使用されるメッセージ・バッファに有害な影響を与える場合には適切ではありません。

複数のメッセージング・エンジンをサービス統合バスに追加するには、デプロイメント・マネージャー・セルにある管理対象ノードのプロファイルを使用する必要があります。

シングル・サーバー・エンタープライズ・サービス・バスの作成:

最もシンプルなトポロジーは、スタンドアロンのサーバー・プロファイルに WebSphere ESB をインストールして作成される場合のような、シングル・サーバーから成るエンタープライズ・サービス・バスです。

シングル・サーバーのエンタープライズ・サービス・バスでは、SCA.SYSTEM バスには、自身の唯一のバス・メンバーである 1 台のサーバーが配置されます。WebSphere ESB にメディアエーション・モジュールをインストールすると、モジュール

ルで使用されるキュー宛先がバス・メンバーに定義されます。これらのキュー宛先は SCA ランタイムによって使用され、このランタイムは、コンポーネントとモジュールの間の非同期対話をサポートする堅固なインフラストラクチャーとして、メディエーション・モジュールによって活用されます。

サーバーは、キュー宛先や要求の処理で使用するサービス統合テクノロジーを提供するメッセージング・エンジンをホストします。

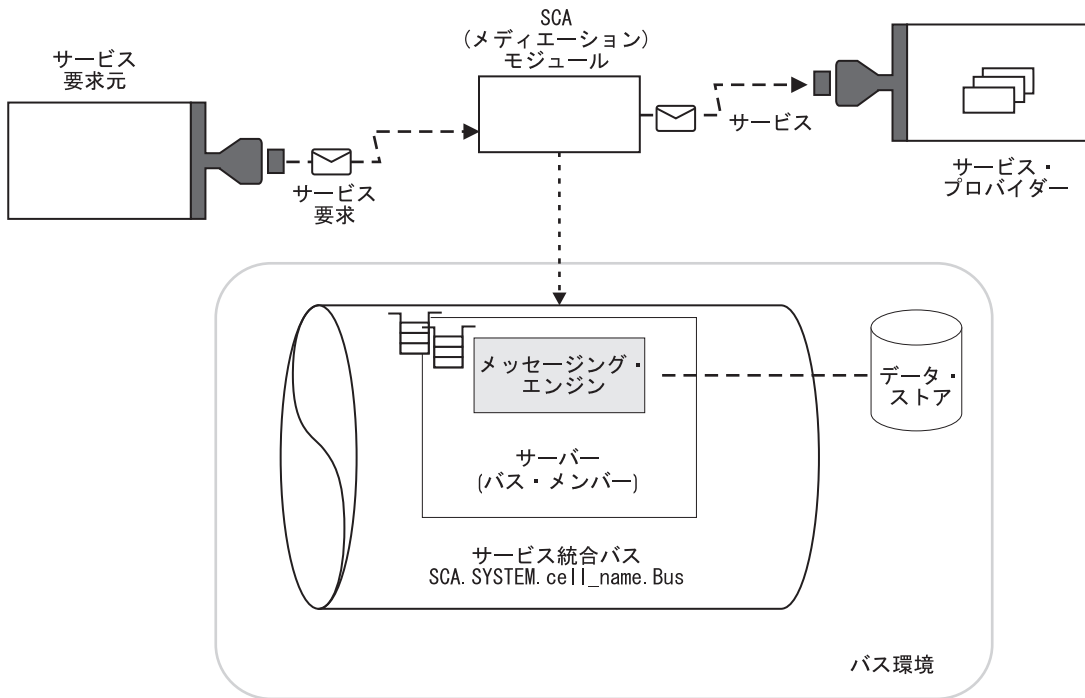


図9. シングル・サーバー・バス： バス・メンバーとして 1 台の ESB サーバーを持つ、1 つのサービス統合バスからなるバス環境を表しています。この図は、SCA モジュールがバス環境にデプロイされ、必要な宛先がバス・メンバーに割り当てられていることも示しています。

スタンドアロンのプロファイルでは、バスに配置可能なメッセージング・エンジンは 1 台のみで、いくつかのシナリオにはこのトポロジーが適しています。

単一のメッセージング・エンジンが適さないシナリオとしては、クライアント接続数が極端に多くなる場合、メッセージのスループット速度が単一のメッセージング・エンジンでは維持できない場合、または、メッセージのサイズが、単一のメッセージング・エンジンで提供できる以上のスペースをデータ・ストアに必要とする場合などがあります。

複数のメッセージング・エンジンをサービス統合バスに追加するには、デプロイメント・マネージャー・セルにある管理対象ノードのプロファイルを使用する必要があります。

スタンドアロン・サーバーを作成する最も容易な方法は、完全インストールを行うことです。完全インストールを実行すると、server1 というサーバー名をもつ default という名前のスタンドアロン・サーバー・プロファイルが作成されます。

1. WebSphere ESB の完全インストールを実行します。 コア・プロダクト・ファイルがインストールされ、最初のスタンドアロン・サーバー・プロファイルが作成されます。
2. 最初のステップのコンソールを使用するか、 `startServer server1` コマンドを実行して `server1` を始動します。

これで、WebSphere ESB サンプルを実行し、エンタープライズ・サービス・バスにサービス・アプリケーションをデプロイできます。

クラスタリングが存在しない複数サーバーのエンタープライズ・サービス・バスの作成:

複数のサーバーで構成されるエンタープライズ・サービス・バスでは、スケーラビリティの確保、処理可能なクライアント接続数の拡大、およびメッセージ・スループットの向上、という利点が得られます。また、SCA モジュールを異なるサーバーに配置して、サービスのさまざまなリソースや品質を提供したり、組織内の異なる部門を何らかの形で分離したり、テスト機能と実動機能を分離したりすることもできます。

バス環境に複数のサーバーを作成するには、デプロイメント・マネージャー・セル内に管理対象ノードが存在している必要があります。

メディエーション・モジュールに必要な SCA ランタイムに対して、各サーバーを構成します。この拡張構成によって、SCA ランタイムで使用されるキューの宛先をこのサーバー上でローカルにホストするか、リモート・サーバー上でホストするかが定義されます。このサーバー上でキューの宛先をホストするように指定した場合、サーバーは SCA.SYSTEM バスのメンバーとなり、キューの宛先が割り当てられるメッセージング・エンジンを取得します。このサーバー上でキューの宛先をホストしないように指定した場合、サーバーは SCA.SYSTEM バスのメンバーとなる必要がなく、メッセージング・エンジンも必要ありません。

194 ページの図 10 に示すシナリオを考慮してください。

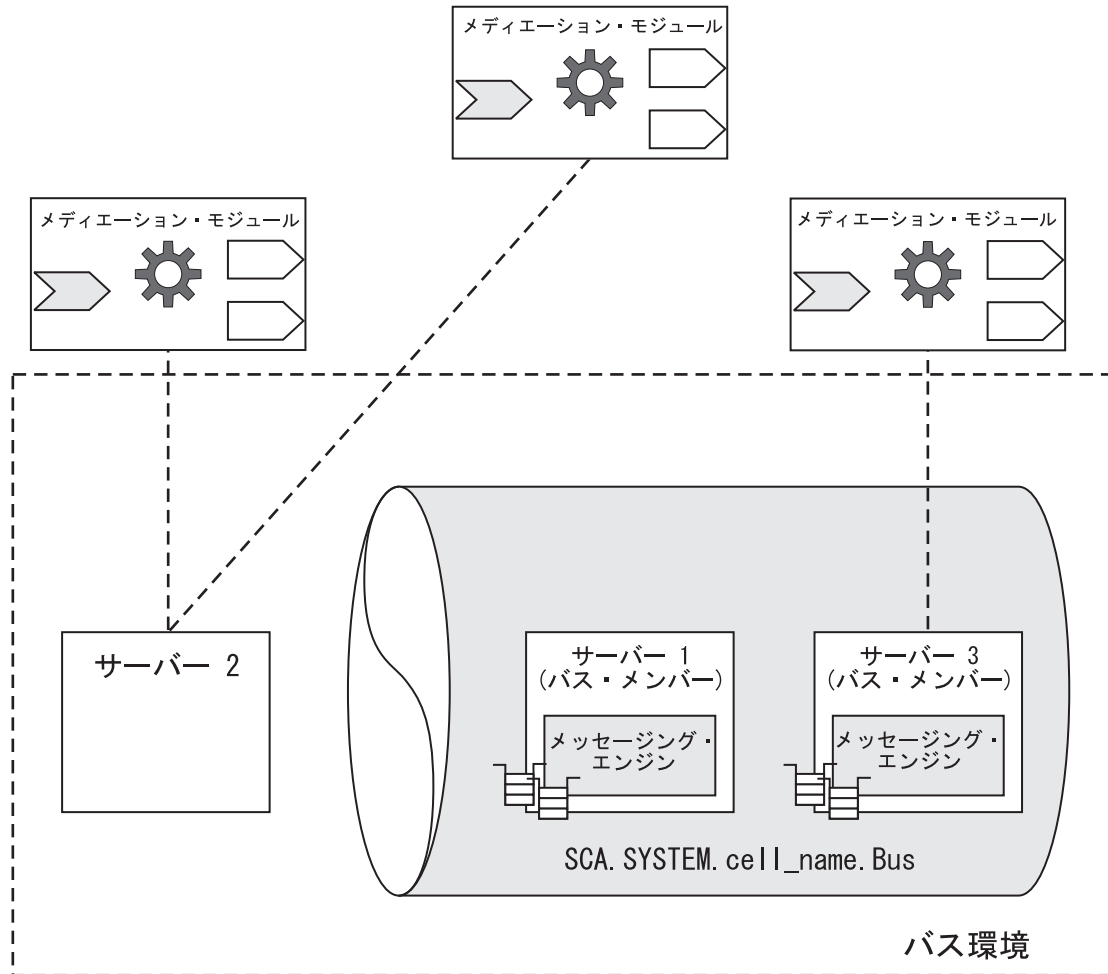


図 10. クラスタリングが存在しない複数サーバーのバス：サーバー 1 は、任意のサーバーに配置されるメディエーション・モジュールのキューの宛先をホストし、メディエーション・モジュールやその他のサービス・アプリケーションをホストしないように構成されます。サーバー 2 は、メディエーション・モジュールやその他のサービス・アプリケーションをホストし、別のサーバー（この図のサーバー 2 またはサーバー 3）上でホストされるキューの宛先を使用するように構成されます。サーバー 3 は、メディエーション・モジュールやその他のサービス・アプリケーションをホストし、また任意のサーバーに配置されるメディエーション・モジュールのキューの宛先もホストするように構成されます。

SCA.SYSTEM バスに存在するすべてのメッセージング・エンジンは暗黙的に接続されているため、要求はバス内のいずれかのメッセージング・エンジンによって処理されます。バス内のそれぞれのメッセージング・エンジンに割り当てられたリソースの知識は、バス内のすべてのメッセージング・エンジン間で共有されます。

バス内のすべてのメッセージング・エンジンが同時に実行している必要はありません。1 つのメッセージング・エンジンが停止した場合でも、残りのメッセージング・エンジンが稼働し続けます。ただし、メッセージング・エンジンが所有するリソース、特にメディエーション・モジュールのキュー・ポイントは、エンジンが停止すると使用不可になります。また、あるサーバーで使用するために作成されたメッセージング・エンジンはそのサーバーでのみ実行できます。したがって、そのサーバーは Single Point of Failure であり、サーバーが稼働できない場合、メッセージング・エンジンは使用できません。サーバー・クラスターをバスのメンバーとし

て構成することにより、メッセージング・エンジンはクラスター内の 1 つのサーバーで実行可能となるため、サーバーに障害が起こった場合でも、メッセージング・エンジンはクラスター内の代替サーバー上で実行できます。

複数サーバーのエンタープライズ・サービス・バスを作成するには、次のようにいくつかの異なる方法があります。

- 1 つのマシンに管理対象サーバー・ノードのセルをインストールします。
- 複数のマシンに管理対象サーバー・ノードのセルをインストールします。

これで、WebSphere ESB サンプルを実行し、エンタープライズ・サービス・バスにサービス・アプリケーションをデプロイできます。

クラスタリングが存在する複数サーバーのエンタープライズ・サービス・バス:

複数のサーバー (その一部またはすべてがサーバー・クラスターのメンバーである) で構成されるエンタープライズ・サービス・バスにデプロイメント・マネージャー・セルを使用できます。

SCA モジュールのキューの宛先をホストするサーバーは、SCA.SYSTEM バス内に 1 つのメッセージング・エンジンを保有します。多くの用途においてこれで十分ですが、このメッセージング・エンジンを実行できるのは、そのエンジンを作成した目的のサーバー上だけです。したがって、そのサーバーは **Single Point of Failure** であり、サーバーが稼働できない場合、メッセージング・エンジンは使用できません。サーバー・クラスターをバスのメンバーとして構成することにより、メッセージング・エンジンはクラスター内の 1 つのサーバーで実行可能となるため、実行サーバーに障害が起こった場合でも、メッセージング・エンジンはクラスター内の代替サーバー上で実行できます。これについては、196 ページの図 11 で説明します。

クラスター・バス・メンバーを構成するもう 1 つの利点は、SCA モジュールに関連したワークロードを複数のサーバー間で共有できることです。あるクラスター・バス・メンバーに配置された SCA モジュールの場合、使用するキューの宛先はクラスター・サーバーにより実行される一連のメッセージング・エンジン間で分割されます。SCA モジュールを介して渡されるメッセージの共有については、クラスター内のメッセージング・エンジンがそれぞれ処理します。

要約すると、クラスター・バス・メンバーを使用することにより、構成可能なポリシーに応じてフェイルオーバーまたはワークロードの共有、あるいはその両方を実現できます。

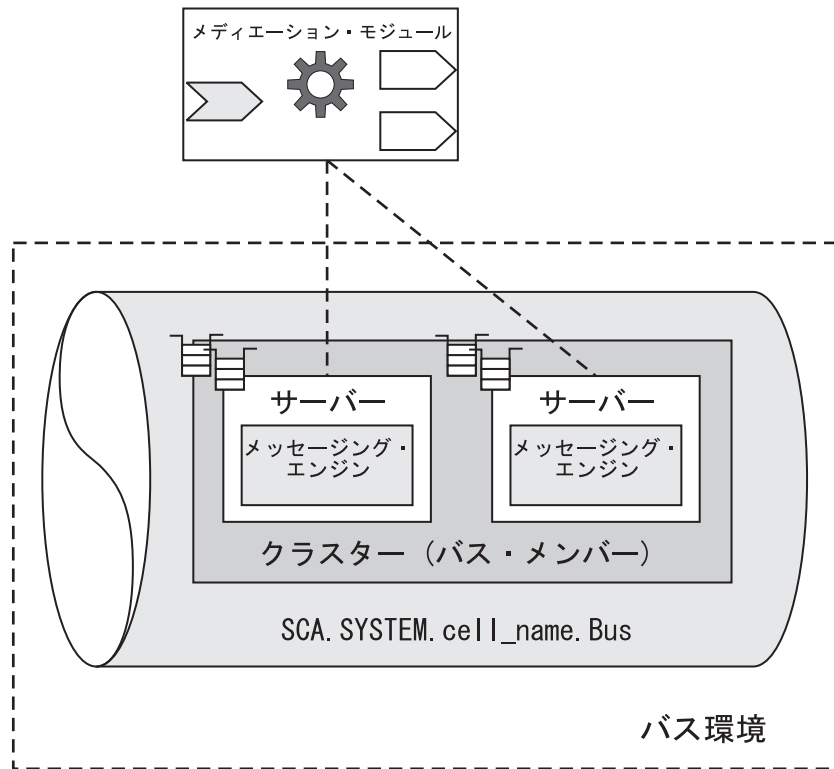


図 11. フェイルオーバー用クラスター・サーバーが存在する複数サーバーのバス

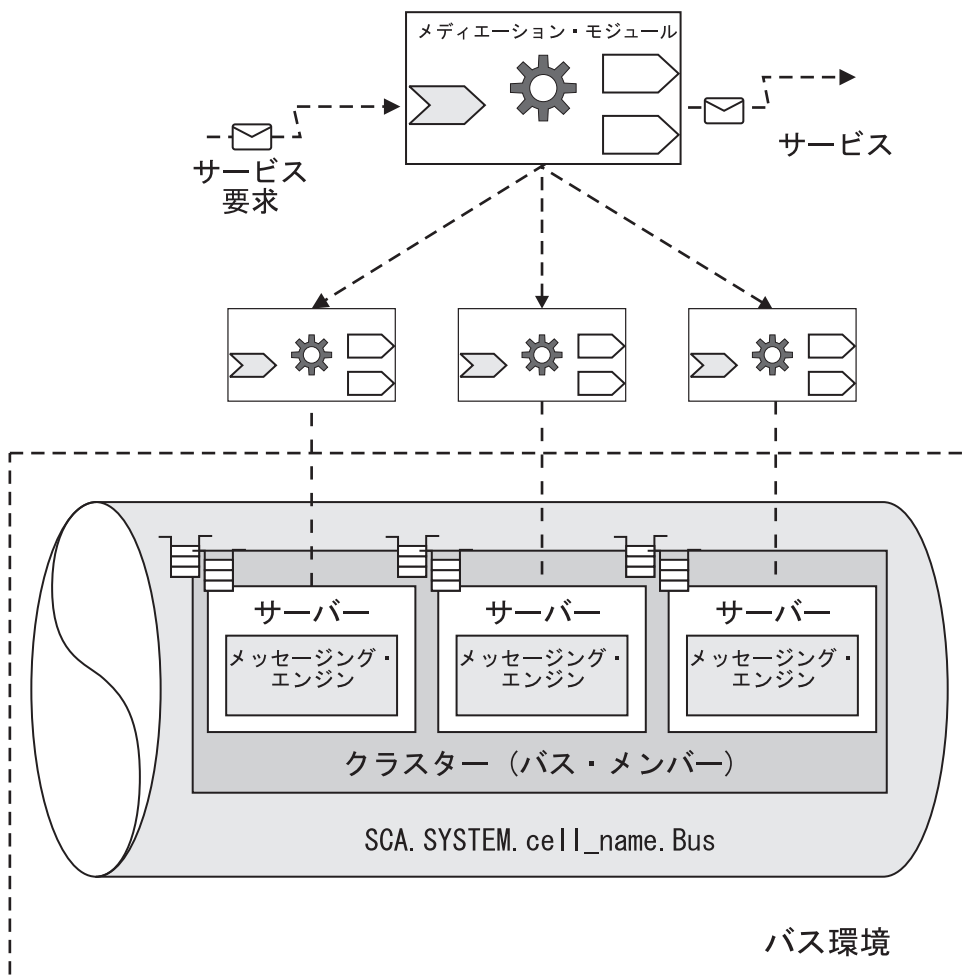


図 12. ワークロード共有用のクラスター・サーバーが存在する複数サーバーのバス

複数サーバーのエンタープライズ・サービス・バスを作成するには、次のようにいくつかの異なる方法があります。

- 1つのマシンに管理対象サーバー・ノードのセルをインストールします。
- 複数のマシンに管理対象サーバー・ノードのセルをインストールします。

これで、WebSphere ESB サンプルを実行し、エンタープライズ・サービス・バスにサービス・アプリケーションをデプロイできます。

複数のエンタープライズ・サービス・バス・トポロジーの作成:

分散バス環境に SCA モジュールをデプロイおよび管理することが必要になる場合があります。例えば、異なる部門間で個別のエンタープライズ・サービス・バスを使用したり、テスト用設備と実動用設備を区別したりする場合があります。

作成する各スタンドアロンのプロファイルまたはデプロイメント・マネージャーのプロファイルには、自身の管理セルがあり、これらは個別のエンタープライズ・サービス・バスの管理可能ドメインとみなせます。

完成したバス環境は、いくつかのスタンドアロン・セルおよびデプロイメント・マネージャー・セルで構成されている場合があります、それぞれのセルは SCA モジュール用に使用する独自の SCA.SYSTEM バスを持つ個別のエンタープライズ・サービス・バスを表しています。

SCA モジュールに使用される SCA.SYSTEM バスのほかに、別のサービス統合バスを作成し、それを使用して、モジュールが提供するサービス統合ロジックをサポートすることもできます。例えば SCA.APPLICATION.cell_name.Bus は、JMS バインディングでデプロイされたモジュールの JMS キュー宛先やその他の JMS リソースを定義するために提供および使用されます。

他のバスを作成して、WebSphere Application Server の場合と同様に使用したり (例えば、WebSphere ESB 内でサービス要求元やプロバイダーとして動作するアプリケーション用)、あるいは WebSphere MQ にリンクしたりできます。

WebSphere ESB デプロイメント・マネージャーを使用して、WebSphere Application Server 上にデプロイされたアプリケーションおよびモジュール用の個々のアプリケーション・サーバーを管理することもできます。

これらの他のバスは別々に使用できる一方で、バス同士を接続してバス間でメッセージの受け渡しができるようにすることも可能です。また、別の組織内のバスと相互に接続することもできます。バスを相互に接続すると、アプリケーションから別のバスに存在するアプリケーションへのメッセージ送信や、他のバスに提供されているリソースの使用が可能になります。公開されたメッセージは、バス間のリンクがこれを許可するように構成されている場合は、複数のバス間をスパンできます。

サービス統合バスは、単一セルに含まれている必要があります。つまり、1 つのバスは複数のセルをスパンできません。ただし、1 つのセルに複数のバスを含めることができます。この場合、セルの各バスは、セルの内部で互いに「外部」となります。1 つのセル内または異なるセル間でバスを相互接続することができます。

あるバスを別のバスにリンクするプロセスは、バス同士が同じセル内にあっても、異なるセルにあっても同じです。

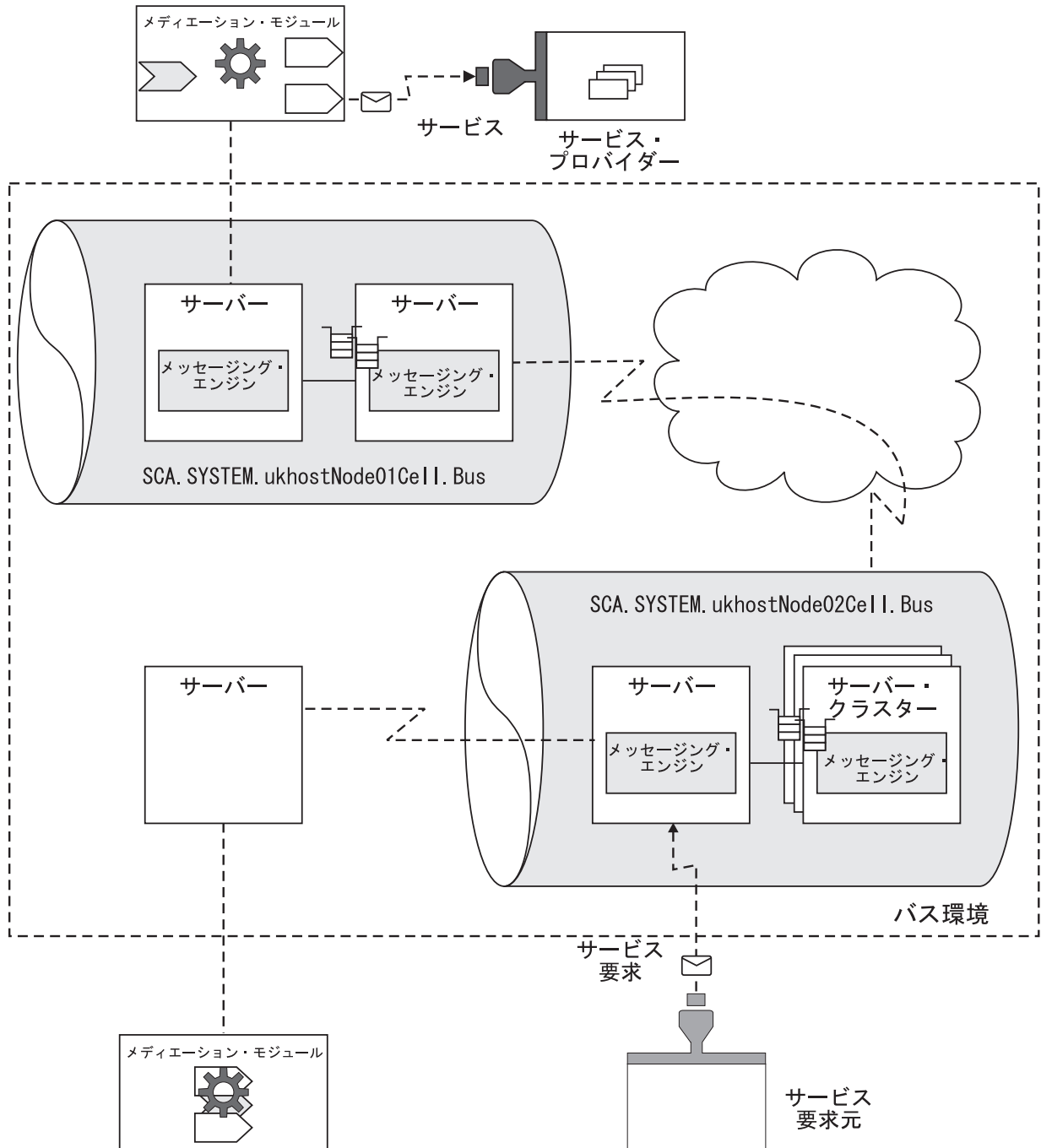


図 13. 複数のエンタープライズ・サービス・バス・トポロジー

複数のエンタープライズ・サービス・バス トポロジーを作成するには、次のようにいくつかの異なる方法があります。

1. 1 つのマシンにシングル・サーバーのエンタープライズ・サービス・バスをインストールします。この方法では、独自の SCA.SYSTEM バスを持つ 1 つのエンタープライズ・サービス・バスを効率よく作成します。
2. 複数のマシンに管理対象サーバー・ノードのセルをインストールします。

同じマシンに 1 つ以上のノードを配置する場合は、WebSphere ESB と同様のインストール方法と、同じマシン上に複数の WebSphere ESB のコピーを別々にインストールする方法のどちらでも実行できます。

これで、WebSphere ESB サンプルを実行し、エンタープライズ・サービス・バスにサービス・アプリケーションをデプロイできます。

WebSphere MQ ネットワークへのリンクを備えたエンタープライズ・サービス・バス:

サービス統合バスには、WebSphere MQ ネットワークへのリンクを含めることができます。これにより、WebSphere MQ キュー・マネージャーに接続されたアプリケーションは、サービス統合バスに接続されているアプリケーションにメッセージを送信することができ、そのようなアプリケーションは WebSphere MQ にメッセージを送信できます。

サービス統合バスが WebSphere MQ ネットワークに接続されている場合は、以下のサポートを追加することによってネットワークを拡張します。

- サービス統合バスにデプロイされたサービス・アプリケーション
- サービス統合バスに接続された JMS アプリケーション
- Web サービス要求元またはプロバイダー

WebSphere MQ ネットワークは、メッセージング・エンジン上で構成される外部バスによって代表されます。*WebSphere MQ* リンクと呼ばれる構成体は、送信側と受信側のチャンネルを使用して MQ キュー・マネージャーにメッセージング・エンジンを接続し、これにより、バスと WebSphere MQ ネットワークのいわゆるゲートウェイ・キュー・マネージャーとの間の橋渡しを行うことができます。

WebSphere MQ リンクは、リンクをホストするメッセージング・エンジンだけでなく、バスの他のメッセージング・エンジンとの接続も提供します。バスのメッセージング・エンジン全体は、WebSphere MQ ネットワークにとって、単一のキュー・マネージャーのように見えます (メッセージング・エンジンは WebSphere MQ リンクからキュー・マネージャー名を継承します)。

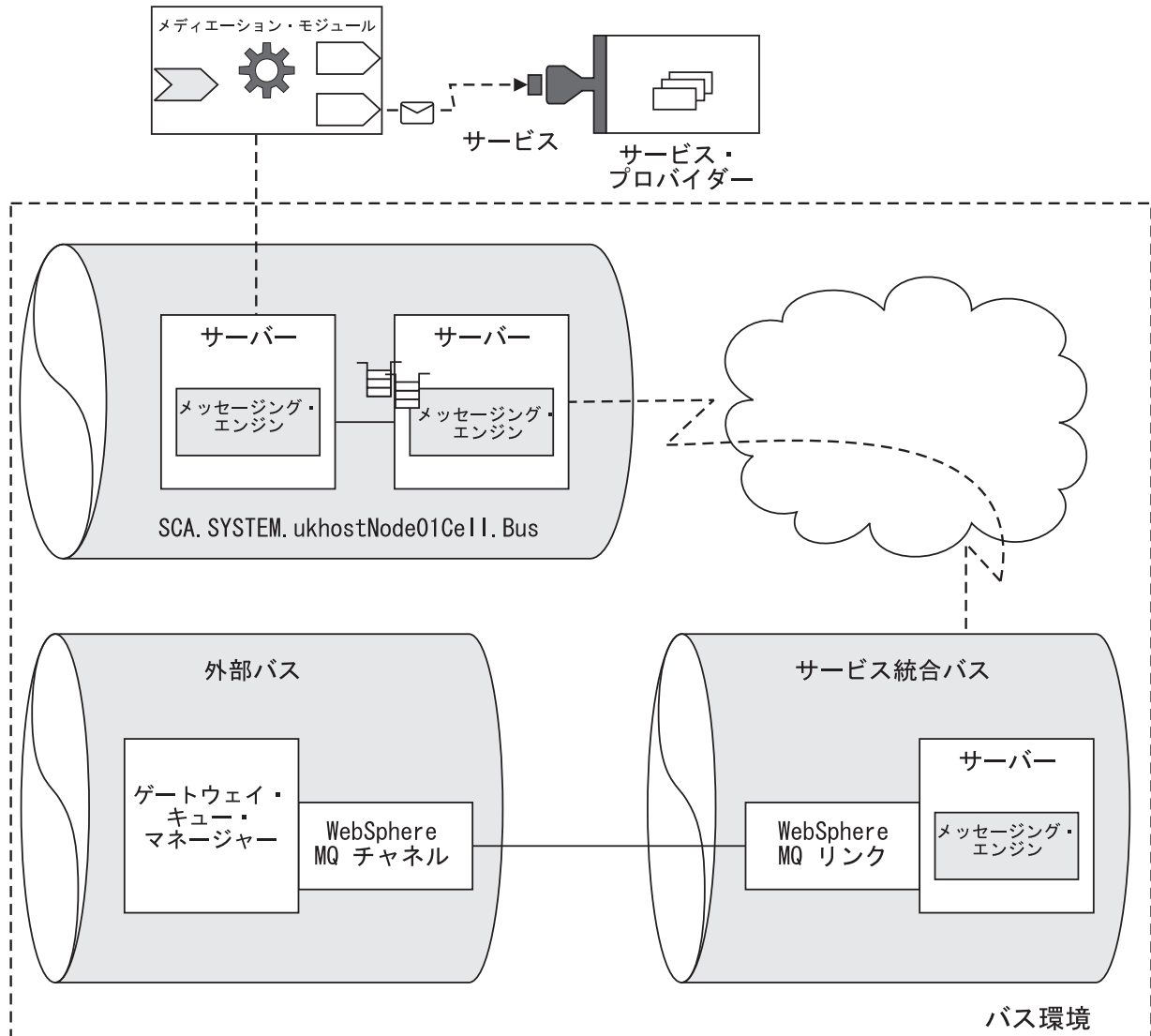


図 14. WebSphere MQ ネットワークへのリンクを備えたサービス統合バス

WebSphere MQ リンクは、さまざまな構成で使用できます。メッセージング・エンジンは、異なるゲートウェイ・キュー・マネージャーへの複数の WebSphere MQ リンクを含むことができます。

WebSphere MQ ネットワークへのリンクは、WebSphere Application Server のサービス統合テクノロジーによってインプリメントされます。

管理インターフェースのサーバーおよびバス機能

WebSphere ESB により、管理コンソールおよびコマンド行から、サーバーおよびバス環境の局面を表示および変更することができます。

「サーバー」 → 「アプリケーション・サーバー」

このページ・セットにより、エンタープライズ・サービス・バスの特別なテンプレートから作成されるアプリケーション・サーバーである、ESB サーバーにアクセスできます。

ESB サーバーは、WebSphere ESB でメディエーション・モジュールおよびサービス・アプリケーションを実行するのに必要な統合テクノロジー、インフラストラクチャー・サービス、構成、およびランタイム管理を提供します。バス・メンバーの一員として、サーバーには、サービス統合バスのコア・メッセージング機能を提供するメッセージング・エンジンがあります。メッセージング・エンジンは、バス・リソースを管理するとともに、サービス要求元の接続ポイントを提供します。

サーバーの詳細設定ページの「**サーバー・メッセージング (Server messaging)**」カテゴリ・ラベルの下からは、メッセージング・エンジン、メッセージング・エンジンのインバウンド・トランスポート、および WebSphere MQ リンクのインバウンド・トランスポートなどのサービス統合プロパティにアクセスできます。

「サービス統合」 → 「バス」

このページ・セットにより、メディエーション・モジュール (および他の SCA モジュール) のデプロイメントで使用する、WebSphere Application Server のサービス統合テクノロジーにアクセスできます。

各サービス統合バスには有効範囲があり、その範囲内で、WebSphere ESB でデプロイされるメディエーション・モジュールと相互作用エンドポイントのリソースを構成できます。バスにより、特定品質の相互作用サービスを提供するエンドポイント間でのメッセージ・ルーティングが可能になり、必要であれば一時的にメッセージを持続させることができます。

WebSphere ESB をインストールすると、2 つのサービス統合バスが作成されます。これらのサービス統合バスは、名前が SCA で始まり、SCA モジュールをデプロイする際に使用できます。メディエーション・モジュールを WebSphere ESB にインストールすると、そのモジュールは SCA.SYSTEM.cell_name.Bus と呼ばれるサービス統合バスのメンバーにデプロイされ、そのバスのメンバーにバス宛先セットが作成されます。

バスの詳細設定ページでは、「**[宛先リソース] 宛先 ([Destination resources] Destinations)**」からアクセスするバス宛先など、多くのカテゴリ・ラベルの下からサービス統合リソースにアクセスできます。

SCA.SYSTEM バス・トポロジーの管理

サービス統合バスである SCA.SYSTEM.cell_name.Bus は、SCA モジュールのデプロイに使用します。バスのローカル・トポロジーの表示、バスのプロパティの変更、バス・メンバーとしてのサーバーおよびクラスターの追加、トポロジー全体を管理するその他の管理タスクの実行を行うことができます。

WebSphere ESB の完全 (デフォルト) インストールを行うと、固有の管理セル内にスタンドアロン・ノードが作成されます。このノードは、セル用の SCA.SYSTEM サービス統合バスに割り当てられた 1 つのサーバーをホストします。

このバス・トポロジーは、いくつかのアプリケーション統合シナリオで SCA モジュールをデプロイする場合に適していますが、このバス・トポロジーを変更して、バス内で複数のメンバーを使用したり、複数のサービス統合バスや場合によっては WebSphere MQ へのリンクを含むより高度な環境を使用して、その利点を得ることができます。

デプロイメント・マネージャー・プロファイルを作成すれば、デプロイメント・マネージャー・セルに管理対象ノードを追加し、それらのノードに 1 つ以上のサーバーあるいはサーバー・クラスターを SCA.SYSTEM バスのメンバーとして作成して、SCA モジュールをデプロイできます。

SCA.SYSTEM バスとサーバーを作成すると、いくつかの重要なプロパティにのみ値が割り当てられ、その他のプロパティはデフォルト値のままです。

以下のサブトピックの情報をもとに、SCA.SYSTEM バスおよびサーバーのトポロジーとプロパティを表示し、必要に応じて構成し、それらのランタイム状態を管理します。これらのサブトピックでは、メディエーション・モジュールの SCA.SYSTEM バスへのデプロイメントをサポートするルーチン・タスクについて説明しています。サーバーおよびバス環境の一般管理に関するこれらのタスクおよびその他のタスクは、関連タスクのリストおよび、227 ページの『バス・トポロジーのその他の用途』に説明されています。

SCA.SYSTEM バスおよびサーバーを管理する場合、以下の項目を含むさまざまな点を考慮してください。

- サーバー間での SCA モジュールの分散方法
- サーバーをまとめて管理する場合、またはワークロード管理に組み込む場合に必要なサーバー・クラスター
- バス、サーバー、その他のリソースの命名規則
- 提供するメッセージのパーシスタンス (永続性) およびその他のサービス品質
- メッセージング・エンジンでメッセージの保持に使用するデータベース

要確認: 複数のバス・トポロジーを組み合わせたバス環境にすることができます。例えば、WebSphere ESB の複数のスタンドアロン・プロファイルまたはデプロイメント・マネージャー・プロファイルを作成すると、それぞれのプロファイルは個別のセル (固有の SCA.SYSTEM バスを持つ) を保有します。いくつかのバスはシングル・サーバーを保有し、それ以外のバスは複数のサーバーおよびサーバー・クラスターを保有することも可能です。

初期 SCA.SYSTEM バス・トポロジーの作成:

スタンドアロン・サーバー・プロファイルまたはデプロイメント・マネージャー・プロファイルを作成すると、個別の管理セルが得られます。自動的に サービス統合バス SCA.SYSTEM.cell_name.Bus が作成され、SCA モジュールをこのセルにデプロイできます。

WebSphere ESB の完全 (デフォルト) インストールを選択すると、固有の管理セル内にスタンドアロン・サーバーが作成されます。このサーバーは、セルの SCA.SYSTEM サービス統合バスに割り当てられたサーバーです。

デプロイメント・マネージャー・プロファイルを作成する場合は、セル内に管理対象ノードを作成し、次に SCA.SYSTEM バスのメンバーとしてサーバーまたはサーバー・クラスターを作成する必要があります。

このタスクを続行する前に、バス環境において複数のサーバーまたはサーバー・クラスターが必要かどうかを考慮します。スタンドアロン・サーバー・プロファイルで存在できるサーバーは 1 つだけです。同じバス環境で複数のサーバーまたはサー

バー・クラスターが必要となる可能性がある場合は、管理対象ノードを持つデプロイメント・マネージャー・セルの作成を考慮します。1 つまたは複数のマシン上で複数のスタンドアロン・サーバーを作成することもできます。

それぞれのスタンドアロン・ノードまたはデプロイメント・マネージャー・セルは個別の管理セルを構成します。管理セルには `SCA.SYSTEM.cell_name.Bus` と呼ばれる固有のサービス統合バスが存在し、これはセル内で `SCA` モジュールをデプロイするのに使用されます。

このタスクを実行して、管理セル用に `SCA.SYSTEM` バスを作成し、1 つの `ESB` サーバーをバス・メンバーとします。`SCA.SYSTEM` バスとサーバーは、いくつかの重要なプロパティの値のみを入力し、その他のプロパティはデフォルト値のままで作成します。

バスの構成は変更できます。例えば、バス・プロパティを変更する場合、またはデプロイメント・マネージャー・セルの場合は、サーバーまたはサーバー・クラスターをバス・メンバーとして追加します。

1 つのサーバーをバス・メンバーとして `SCA.SYSTEM` バスを作成するには、次のいずれかの手順を実行します。

- スタンドアロン・サーバーを作成します。

`WebSphere ESB` の完全 (デフォルト) インストールを選択すると、固有の管理セル内にスタンドアロン・サーバー・ノードが作成されます。このノードは、`SCA` モジュールをデプロイするために `SCA.SYSTEM` バスに割り当てられた 1 つのサーバーをホストします。

このサーバーはセル用に自動的に作成され、`SCA.SYSTEM` バスに追加されます。このトポロジーは 1 つのサーバーに限定されます。

- 管理対象ノード上に 1 つのサーバーを持つデプロイメント・マネージャー・セルを作成します。これにより、集中管理が行え、サーバー・クラスター間のワークロード・balancing が可能になるという利点が得られます。カスタム・プロファイル用の 1 つの管理対象ノードを含むセルから開始して、その後、必要に応じてそのノードにサーバーを追加したり、プロファイルを新たに作成したりできます。

セル用の管理コンソールにログインすると、「サービス統合」→「バス」ページにリストされた `SCA.SYSTEM` バスを参照できます。新規サーバーも「サーバー」→「アプリケーション・サーバー」ページにリストされています。

新規サーバーの開始後、`SCA.SYSTEM` バスを使用して `SCA` モジュールをデプロイできます。

バスの構成は変更できます。例えば、デプロイメント・マネージャー・セルにサーバーまたはサーバー・クラスターを追加できます。

サービス統合バスのトポロジーの表示:

管理コンソールのバス・トポロジー・ビューは、選択したバスのバス・メンバーおよびメッセージング・エンジンを表示します。各メッセージング・エンジンに関して、ランタイム状況が表示されます。

このタスクを使用して、バス・メンバーおよびサービス統合バス内のメッセージング・エンジンのツリー表示を実行します。このローカル・トポロジー・ビューを使用して、バスのメンバーとしてサーバーおよびサーバー・クラスターを追加したり、クラスター・バス・メンバーにメッセージング・エンジンを追加したりできます。

サービス統合バスのローカル・トポロジー・ビューを表示するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サービス統合」 → 「バス」をクリックします。コンテンツ・ペインにバスのリストが表示されます。
2. コンテンツ・ペインで、バスの名前をクリックします。例えば `SCA.SYSTEM.localhostCell01.Bus` をクリックします。
3. 「ローカル・トポロジー」タブをクリックします。

拡張可能なツリーとしてバスのトポロジーが表示されます。

ツリーのノードを展開すると、バス・メンバーとそのメッセージング・エンジンが表示されます。

バスのメンバーとしてサーバーおよびサーバー・クラスターを追加したり、クラスター・バス・メンバーにメッセージング・エンジンを追加したりできます。

バス・プロパティーの変更:

各プロパティーをデフォルト値のまま、サービス統合バスを作成できます。必要があれば、後で変更に応じてバスを使用するようプロパティーの値を変更することができます。

サービス統合バスの用途を変更する必要がある場合は、このタスクを使用すればどのプロパティーを変更するかを判断するのに役立ちます。このタスクは、バス・プロパティーを変更するための考慮事項の概要を提供し、基盤となる WebSphere Application Server のサービス統合テクノロジーが備える一連のタスク記述にリンクしています。

• SCA.SYSTEM バスのメッセージング・セキュリティの構成

メッセージング・セキュリティを作動させると、バスおよびバスに存在するすべての宛先へのアクセスには許可が必要になります。つまり、バスに接続しようとするすべてのユーザーは、直接またはユーザー・グループの一部としてのバス・リソースの使用許可が必要です。

SCA.SYSTEM バスを作成すると、デフォルトでバスのメッセージング・セキュリティが作動します。許可されたメッセージング・エンジンのみが、保護されたバスへの接続を作成できます。

SCA.SYSTEM バスは、`SCA_Auth_Alias` という認証別名、デフォルト権限許可の初期セットで構成されます。このデフォルト権限許可により、SCA は保護されたバスに接続でき、バス上のすべてのローカル宛先への全アクセス権限が与えられます。デフォルト権限許可を変更して、バスへのアクセスを特定のユーザーに限定したり、異なる認証別名を作成して使用したりできます。

デフォルトで、同じ `SCA_Auth_Alias` がエンジン間認証別名として使用され、バス上のメッセージング・エンジン間の通信が許可されます。異なるエンジン間認証別名を作成して使用することもできます。

さらにセキュリティを構成する場合は、セキュア・トランスポート接続 (SSL または HTTPS) を使用して、アプリケーション・クライアントとメッセージング・エンジン間の伝送、およびメッセージング・エンジン間の伝送におけるメッセージの機密性と保全性を確保できます。WebSphere ESB が備えるトランスポート・チェーン `InboundSecureMessaging (JFAP over SSL over TCP/IP)` を指定できます。または、別のトランスポート・チェーンを作成して指定することもできます。

- 削除されたメッセージ・ポイント上のメッセージの保存または廃棄の設定

SCA モジュール用のアプリケーションをアンインストールすると、SCA ランタイムで使用されるそのモジュールのキュー・ポイントが削除されます。デフォルトでは、「廃棄」メッセージ・プロパティがクリアされ、これらのキュー・ポイント上に存在するすべてのメッセージはシステムの例外宛先に保存されるため、後でメッセージを処理できます。「廃棄」メッセージ・プロパティを選択すると、削除されたキュー・ポイント上のすべてのメッセージを廃棄できます。

- 構成情報のすべての更新をメッセージング・エンジンに動的に渡すかどうかの設定

デフォルトでは、`SCA.SYSTEM` バスの `Configuration reload enabled` プロパティが選択されます。つまり、構成情報へのすべての更新が動的にサーバーに渡されるため、メッセージング・エンジンが開始済みかどうかにかかわらず、更新がメッセージング・エンジンで有効になります。(メッセージング・エンジンは開始されると、実行しているサーバー内の情報を使用します。)

- メッセージング・エンジンのメッセージ・ポイントに保存できるメッセージの最大総数の設定

デフォルトでは、`SCA.SYSTEM` バスの `High message threshold` プロパティが 50000 メッセージに設定されます。これにより、このプロパティの初期値がバス内の各メッセージング・エンジンで設定されます。キュー・ポイント上の現在のメッセージの深さがメッセージ数の上限しきい値に等しい場合、メッセージング・エンジンは、キューに入れられたメッセージが消費されるまでは新たなメッセージを受け入れません。メッセージのスループットの問題を解決する 1 つの方法として、(バス上またはメッセージング・エンジン上の) `High message threshold` プロパティを変更できます。

SCA ランタイム用サーバーまたはクラスタの構成

サーバーまたはサーバー・クラスタ上にメディアーション・モジュールをデプロイするには、事前にそのサーバーまたはクラスタを必要な SCA ランタイム用に構成しておく必要があります。この拡張構成で、SCA ランタイムで使用されるキュー宛先をローカルにホストするか、リモート・サーバーまたはクラスタ上でホストするかを定義します。

このタスクを開始する前に、必要な以下のサーバーまたはサーバー・クラスタを 1 つまたは複数作成しておく必要があります。

- メディアーション・モジュールをデプロイするサーバーまたはクラスタ。

- SCA ランタイムで使用するキュー宛先を作成するサーバーまたはクラスター。

両方に同じサーバーまたはクラスターを使用できます。

サーバーまたはサーバー・クラスターの作成に関する詳細は、以下のタスクの記述を参照してください。

- サーバーの作成
- サーバー・クラスターの作成

WebSphere ESB にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、SCA ランタイムによってメディエーション・モジュールの処理中のメッセージを保持するために使用されます。

キュー宛先を作成する場所を指定するには、サーバーまたはクラスターの以下の代替構成のいずれかを実行します。

- サーバーまたはクラスターの拡張構成を設定して、宛先をホストします。この構成では、ローカルにデプロイされたメディエーション・モジュールまたは別のサーバーまたはクラスターにデプロイされたメディエーション・モジュールのキュー宛先をこのサーバーまたはクラスターがホスト可能になるように定義します。また、この構成では、サーバーまたはクラスターを `SCA.SYSTEM` バスのメンバーとして追加し、サーバーまたはクラスターのメッセージング・エンジンを構成します。

この構成タスクを実行する場合は、208 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』を参照してください。

- リモート宛先を使用するように、サーバーまたはクラスターの拡張構成を設定します。この構成では、このサーバーまたはクラスターが宛先にリモート宛先ロケーションを使用するよう定義します。

このサーバーまたはクラスターは `SCA.SYSTEM` バスのメンバーである必要はありません。

このサーバーまたはクラスターにメディエーション・モジュールをデプロイすると、キュー宛先は `SCA.SYSTEM` バスのメンバーである別のサーバーまたはクラスター上に構成されます。

この構成タスクを実行する場合は、210 ページの『メディエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスターの構成』を参照してください。

サーバーまたはクラスターは、メディエーション・モジュールのデプロイメントを受け入れるように構成されます。

これで、サーバーまたはクラスター上にメディエーションをデプロイしたり、サーバーまたはクラスターを管理するその他のアクションを実行できるようになります。

メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスタの構成:

メディエーション・モジュールで、SCA ランタイムにより使用されるキュー宛先をホストするように、クラスタまたはサーバーを構成できます。サーバーまたはクラスタは、管理セル内の任意のサーバーまたはクラスタにデプロイされたメディエーション・モジュールのキュー宛先をホストできます。

このタスクを開始する前に、サーバーまたはクラスタを作成しておく必要があります。

WebSphere ESB にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、SCA ランタイムによってメディエーション・モジュールの処理中のメッセージを保持するために使用されます。

メディエーション・モジュールがデプロイされているサーバーまたはサーバー・クラスタ上、あるいは別のサーバーまたはクラスタ上にキュー宛先が作成されるように、サーバーまたはクラスタを構成できます。

注: サーバーまたはクラスタをメディエーション・モジュールのデプロイとそのキュー宛先のホストに使用するには、事前にこのタスクを完了する必要があります。または、210 ページの『メディエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスタの構成』に説明するように、宛先にリモート宛先ロケーションを使用するようサーバーまたはクラスタを構成できます。

メディエーション・モジュールのキュー宛先をホストするサーバーまたはクラスタはキュー処理用に最適化でき、メディエーション・モジュールがデプロイされた別の複数のサーバーおよびクラスタにこれらのキューを提供できます。

この構成タスクでは、ローカルにデプロイされたメディエーション・モジュールまたは別のサーバーまたはクラスタにデプロイされたメディエーション・モジュールのキュー宛先を、サーバーまたはクラスタがホストできるように定義します。また、この構成では、サーバーまたはクラスタを SCA.SYSTEM バスのメンバーとして追加し、サーバーまたはクラスタのメッセージング・エンジンを構成します。

- サーバーを構成すると、WebSphere はデフォルトのプロパティでメッセージング・エンジンを作成できます。デフォルトで、メッセージング・エンジンはデフォルトの JDBC データ・ソースおよびそのデータ・ストアに Cloudscape JDBC Provider を使用するように構成されます。この場合、追加の構成を行うことなくメッセージング・エンジンを実行できます。

デフォルトのデータ・ソース構成を使用しない場合は、異なるデータ・ソースの使用を選択するか、異なる JDBC プロバイダーを使用するようにデータ・ストアを構成できます。

- クラスタを構成する場合、データ・ストア、およびメッセージング・エンジンがデータ・ストアとの対話に使用する JDBC データ・ソースを明示的に作成する必要があります。また、接続プールの「ページ・ポリシー (Purge policy)」を「EntirePool」に設定する必要があります。

メディエーション・モジュールのキュー宛先をホストするようにサーバーまたはクラスターを構成するには、管理コンソールを使用して以下のステップを実行します。

1. サーバーまたはクラスターの「拡張構成」ページを表示します。
 - サーバーを構成する場合: 「サーバー」 → 「アプリケーション・サーバー」 → 「*server_name*」 → 「拡張構成」
 - クラスターを構成する場合: 「サーバー」 → 「クラスター」 → 「*cluster_name*」 → 「拡張構成」
2. 「エミッター・ファクトリー・プロファイル JNDI 名」フィールドの値を検証します。
 - CEI サーバーへの出力イベントのデフォルト構成を変更する場合は、このフィールドで該当するエミッター・ファクトリー・プロファイル JNDI 名を選択します。クラスター環境では、選択した値がクラスター内のすべてのサーバーに使用されます。
 - エミッター・ファクトリー・プロファイルを指定しない (既存の CEI 構成を保持する) 場合は、メニューから「なし」を選択します。
3. 「デフォルト宛先ロケーション」をクリックします。「SCA アプリケーションのホストを構成しない (Do not configure to host SCA applications)」オプションが選択されていないことを確認します。
4. サーバーまたはクラスターに対して、メッセージング・エンジンのオプションを構成します。

構成されるリソース	実行するステップ
サーバーの構成	<ul style="list-style-type: none"> • メッセージング・エンジンでデフォルトの JDBC データ・ソースおよびそのデータ・ストアに Cloudscape JDBC Provider を使用する場合は、「デフォルトのデータ・ソース値を使用 (Use default data source values)」チェック・ボックスを選択して、5 (210 ページ) ステップに進みます。 • メッセージング・エンジンでデフォルト以外のオプションを使用する場合は、クラスターに関してリストされた同じステップを、1 (210 ページ) ステップから実行します。

構成されるリソース	実行するステップ
<p>クラスタの構成</p>	<ol style="list-style-type: none"> 1. 「JDBC プロバイダー」メニューで適切な JDBC プロバイダー・テンプレートを選択します。 2. 「アプリケーション・バス・スキーマ名 (Application bus schema name)」フィールドに、SCA.APPLICATION バスのデータ・ソース用テーブルを含めるのに使用するデータベース・スキーマ名を入力します。 <p>各メッセージング・エンジンはテーブルなどのリソースをシングル・スキーマで保管します。各データベース・スキーマは、1 つのメッセージング・エンジンのみに使用されます。各メッセージング・エンジンが同じテーブル名を使用している場合、スキーマとのリレーションシップによって各メッセージング・エンジンでは固有のテーブルの排他使用が可能です。デフォルトのデータ・ソースに使用されるスキーマは IBMWSSIB です。</p> <ol style="list-style-type: none"> 3. 「システム・バス・スキーマ名 (System bus schema name)」フィールドに、SCA.SYSTEM バスのデータ・ソース用テーブルを含めるのに使用するデータベース・スキーマ名を入力します。 4. 「データ・ソースのユーザー名」フィールドに、データベースにアクセスするときに使用するユーザー名を入力します。 5. 「データ・ソースのパスワード」フィールドに、上記のユーザー名に関連付けられたパスワードを入力します。 6. 「アプリケーション・バス・データベース名」フィールドに、SCA.APPLICATION バス上に作成されたメッセージング・エンジンにより使用されるデータベースの名前を入力します。 7. 「システム・バス・データベース名」フィールドに、SCA.SYSTEM バス上に作成されたメッセージング・エンジンにより使用されるデータベースの名前を入力します。 8. メッセージング・エンジンにデータ・ソース用のデータベース表を作成させるには、「テーブルの作成」チェック・ボックスを選択します。(メッセージング・エンジンをホストするサーバーの始動時に、テーブルが作成されます。) それ以外の場合は、データベース管理者がデータベース表を作成する必要があります。

5. 「**OK**」をクリックします。
6. 変更をマスター構成に保管します。

リモート宛先ロケーションを使用するように構成されている別のサーバーまたはクラスタにメディエーション・モジュールをデプロイできます。その別のサーバーまたはクラスタの「リモート宛先ロケーション (Remote Destination Location)」プロパティにこのサーバーまたはクラスタが設定されている場合は、メディエーション・モジュールのキュー宛先がこのサーバーまたはクラスタ上に構成されません。

パフォーマンスに関してメッセージング・エンジンの JDBC データ・ソースを調整できます。

メディエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスタの構成:

メディエーション・モジュールでキュー宛先が別のクラスターまたはサーバーに作成されるように、クラスターまたはサーバーを構成できます。

このタスクを開始する前に、必要な以下のサーバーまたはサーバー・クラスターを1つまたは複数作成しておく必要があります。

- メディエーション・モジュールをデプロイするサーバーまたはクラスター。これは、このタスクで構成するサーバーまたはクラスターです。
- SCA ランタイムで使用するキュー宛先を作成するサーバーまたはクラスター。

注: 208 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』で説明するように、このサーバーまたはクラスターはメディエーション・モジュールのキュー宛先をホストするように構成しておく必要があります。

両方に同じサーバーまたはクラスターを使用できます。

WebSphere ESB にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、SCA ランタイムによってメディエーション・モジュールの処理中のメッセージを保持するために使用されます。

メディエーション・モジュールがデプロイされているサーバーまたはサーバー・クラスター上、あるいは別のサーバーまたはクラスター上にキュー宛先が作成されるように、サーバーまたはクラスターを構成できます。

注: 別のサーバーまたはクラスターでキュー宛先をホストする場合のメディエーション・モジュールのデプロイにこのサーバーまたはクラスターを使用するには、事前にこのタスクを完了する必要があります。または、208 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』で説明するように、キュー宛先をホストするようサーバーまたはクラスターを構成できます。

メディエーション・モジュールのキュー宛先をホストしないサーバーまたはクラスターは、他のコンポーネントを実行するように最適化でき、SCA.SYSTEM バスのメンバーである必要はありません。

この構成タスクでは、別のサーバーまたはクラスター上に作成されたキュー宛先を使用するようにサーバーまたはクラスターを定義します。この構成では、サーバーまたはクラスターを SCA.SYSTEM バスのメンバーとして追加しません。

別のサーバーまたはクラスター上に作成されたキュー宛先を使用するようにサーバーまたはクラスターを構成するには、管理コンソールを使用して以下のステップを実行します。

1. サーバーまたはクラスターの「拡張構成」ページを表示します。
 - サーバーを構成する場合: 「サーバー」 → 「アプリケーション・サーバー」 → 「*server_name*」 → 「拡張構成」
 - クラスターを構成する場合: 「サーバー」 → 「クラスター」 → 「*cluster_name*」 → 「拡張構成」

2. 「エミッター・ファクトリー・プロファイル JNDI 名」フィールドの値を検証します。
 - CEI サーバーへの出力イベントのデフォルト構成を変更する場合は、このフィールドで該当するエミッター・ファクトリー・プロファイル JNDI 名を選択します。クラスター環境では、選択した値がクラスター内のすべてのサーバーに使用されます。
 - エミッター・ファクトリー・プロファイルを指定しない (既存の CEI 構成を保持する) 場合は、メニューから「なし」を選択します。
3. 「Process Server アプリケーションのホストを構成しない (Do not configure to host Process Server applications)」オプションの選択を解除します。
4. サーバーまたはクラスターに対して、リモート宛先ロケーション (Remote Destination Location) オプションを設定します。
 - a. 「リモート宛先ロケーション (Remote Destination Location)」をクリックします。
 - b. 関連メニューから、メディアエーション・モジュールのキュー宛先をホストするリモート・サーバーまたはクラスターの名前を選択します。
5. 「OK」をクリックします。
6. 変更をマスター構成に保管します。

このサーバーまたはクラスターにメディアエーション・モジュールをデプロイすると、メディアエーション・モジュール用のキュー宛先は Remote Destination Location プロパティで指定されたサーバーまたはクラスター上に構成されます。

サーバーまたはクラスターの構成を変更して、サービス統合バスのメンバーとして追加したり、サーバーまたはクラスターを管理するその他のアクションを実行したりできます。

ESB サーバーの管理

サーバー構成の定義によって、サーバーでアプリケーションとそのコンポーネントを実行するためのサービスの提供方法の制御が設定されます。管理者は、既存のサーバーおよびバス環境で、サーバーを作成および構成できます。

- サーバーを新規作成するには、デプロイメント・マネージャー・セルに管理対象ノードが配置されている必要があります。

WebSphere ESB の完全 (デフォルト) インストールを選択すると、デフォルトのバス環境が作成されます。このバス環境は、SCA モジュールの配置に使用される 2 つのサービス統合バスに割り当てる単一サーバーで構成されます。このサーバーを使用して、サーブレット、Enterprise Java Bean、JMS 宛先など、J2EE のコンポーネントおよびリソースをデプロイすることもできます。

ご使用の SCA モジュールによっては、単一サーバーの構成で十分な場合もあります。しかし、サーバー容量の追加、ネットワーク・サービスを提供するトランスポート・チェーンの変更、または、ユーザー自身のカスタム・サービスの提供などを行う場合は、この構成を変更できます。

- サーバーを作成します。

管理コンソールの「新規アプリケーション・サーバーの作成」ウィザード、または `createApplicationServer wsadmin` コマンドのいずれかを使用して、サーバーを作成することができます。

- 使用可能なサーバーを管理します。

サーバーの作成後、ランタイム状態のモニター、構成の変更、サーバーの始動および停止、その他の管理用タスクの実行などを通じてサーバーを管理することができます。

- トランSPORT・チェーンを構成します。

サービス統合テクノロジー、WebSphere Secure Caching Proxy、および高可用性マネージャーによるコア・グループ・ブリッジ・サービスなどの機能を実施するネットワーク・サービスを提供するための、トランSPORT・チェーンを構成する必要があります。

- カスタム・サービスを作成します。

サーバーまたはノード・エージェントの始動時およびシャットダウン時にフック・ポイントが実行されるように定義するには、カスタム・サービス・クラスを作成してカスタム・サービス・インスタンスを構成します。アプリケーション・サーバーまたはノード・エージェントの始動時に、カスタム・サービスが開始され、初期化されます。

- サーバーのプロセスを定義します。

サーバーの操作を向上させるには、アプリケーション・サーバー・プロセスを開始または初期化するためのコマンド行情報を定義します。この設定では、実行するプログラム、プログラムを実行する引数、作業ディレクトリーなどのランタイム・プロパティを定義します。

- サーバーによる Java 仮想マシンの使用を管理します。

サーバー構成の一部として、システムによる Java 仮想マシン (JVM) の使用率を高める設定を定義できます。

サーバーの作成:

管理コンソールの「新規アプリケーション・サーバーの作成」ウィザード、または `createApplicationServer wsadmin` コマンドのいずれかを使用して、サーバーを作成することができます。

サーバーを作成するには、例えば新しい WebSphere ESB カスタム・プロファイルの作成の説明のように、事前にカスタム・プロファイルを作成し、その後ノードをデプロイメント・マネージャー・セルにフェデレートしておく必要があります。

SCA モジュールをデプロイする前に、SCA モジュールを実行するサーバーを作成しておく必要があります。

複数の SCA モジュールを同じサーバー上にデプロイすることが可能なため、新規サーバーを作成する場合に完了する必要があるのはこのタスクのみです。例えば、1 台のサーバーより多くのクライアント接続を処理したり、より大規模なメッセージ・スループットを提供したり、異なるサーバー上に SCA モジュールの管理を分離したりする場合などです。

サーバーを作成するには、管理コンソールを使用して以下のステップを実行します。

1. サーバーのリストを表示します。

ナビゲーション・ペインで、「サーバー」 → 「アプリケーション・サーバー」を展開します。

2. 「新規」をクリックします。
3. 「新規アプリケーション・サーバーの作成」 ページで、以下の指示に従ってサーバーを定義します。
 - a. サーバーを実行するノードを選択します。

このノードは、WebSphere ESB 管理対象ノードでなければなりません。

- b. サーバーの固有の名前を入力します。

この名前はノード内で固有でなければなりません。

- c. 「次へ」をクリックします。
- d. サーバーを作成するための **defaultESBServer** テンプレート (または既存サーバー) を選択します。

この **defaultESBServer** テンプレートの代わりに、既存の ESB サーバーをテンプレートとして使用することができます。新規サーバーは、テンプレート・サーバーのすべてのプロパティを継承します。

- e. 「次へ」をクリックします。
- f. 「固有 **Http** ポートの生成」 オプションで、新規サーバーに対し、HTTP トランスポートごとに固有のポートを生成するかどうかを選択します。

デフォルトでは、「固有 **Http** ポートの生成」 オプションが選択されています。このオプションを選択すると、上記の新しいポート値を入れるために、このサーバーとともに使用する予定の仮想ホストの別名リストを更新することが必要になる場合があります。このオプションを選択解除する場合は、同じ物理マシン上の他のサーバーとデフォルトのポート値が競合しないようにしてください。

固有の HTTP ポートの使用に関する詳細は、『HTTP トランスポートの構成』を参照してください。

- g. 「次へ」をクリックします。
- h. 既存のサーバーをテンプレートとして使用して新規サーバーを作成する場合、アプリケーションを既存のサーバーから新規サーバーにマップする選択は行わないでください。デフォルトでは、このオプションは使用不可になっています。
- i. 「次へ」をクリックします。
- j. 「新規サーバーの確認」 ページで、要約を確認します。サーバーの作成を完了するには、「終了」をクリックします。

変更する設定がある場合は、「前へ」をクリックしてサーバー設定を検討または変更してください。

4. **オプション:** 複数言語のエンコード・サポートを管理コンソールで使用するには、UTF-8 エンコードが使用可能になるようにアプリケーション・サーバーを構成してください。
5. 変更をマスター構成に保管します。

サーバーが作成され、セルの SCA* バスに自動的に追加されます。

新規サーバーが「アプリケーション・サーバー」ページのサーバー・リストに表示されます。

これで、サーバーの開始や SCA モジュールのデプロイなどの、サーバーの管理を実行できるようになります。

注: サーバーは、いくつかのプロパティでデフォルト値を使用して作成されています。サーバーには設定が可能な多くのプロパティがあり、「新規アプリケーション・サーバーの作成」ページでサーバーを作成すると、いくつかの重要なプロパティの値のみ指定されます。ご使用のサーバーのすべてのプロパティを表示し、さらにサーバーをカスタマイズするには、「アプリケーション・サーバー」ページでサーバーの名前をクリックします。サーバーの詳細設定のページが表示されるので、必要に応じてプロパティを参照し変更することができます。

SCA ランタイム用サーバーまたはクラスタの構成:

サーバーまたはサーバー・クラスタ上にメディエーション・モジュールをデプロイするには、事前にそのサーバーまたはクラスタを必要な SCA ランタイム用に構成しておく必要があります。この拡張構成で、SCA ランタイムで使用されるキュー宛先をローカルにホストするか、リモート・サーバーまたはクラスタ上でホストするかを定義します。

このタスクを開始する前に、必要な以下のサーバーまたはサーバー・クラスタを 1 つまたは複数作成しておく必要があります。

- メディエーション・モジュールをデプロイするサーバーまたはクラスタ。
- SCA ランタイムで使用するキュー宛先を作成するサーバーまたはクラスタ。

両方に同じサーバーまたはクラスタを使用できます。

サーバーまたはサーバー・クラスタの作成に関する詳細は、以下のタスクの記述を参照してください。

- サーバーの作成
- サーバー・クラスタの作成

WebSphere ESB にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、SCA ランタイムによってメディエーション・モジュールの処理中のメッセージを保持するために使用されます。

キュー宛先を作成する場所を指定するには、サーバーまたはクラスタの以下の代替構成のいずれかを実行します。

- サーバーまたはクラスターの拡張構成を設定して、宛先をホストします。この構成では、ローカルにデプロイされたメディアエーション・モジュールまたは別のサーバーまたはクラスターにデプロイされたメディアエーション・モジュールのキュー宛先をこのサーバーまたはクラスターがホスト可能になるように定義します。また、この構成では、サーバーまたはクラスターを `SCA.SYSTEM` バスのメンバーとして追加し、サーバーまたはクラスターのメッセージング・エンジンを構成します。

この構成タスクを実行する場合は、208 ページの『メディアエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』を参照してください。

- リモート宛先を使用するように、サーバーまたはクラスターの拡張構成を設定します。この構成では、このサーバーまたはクラスターが宛先にリモート宛先ロケーションを使用するよう定義します。

このサーバーまたはクラスターは `SCA.SYSTEM` バスのメンバーである必要はありません。

このサーバーまたはクラスターにメディアエーション・モジュールをデプロイすると、キュー宛先は `SCA.SYSTEM` バスのメンバーである別のサーバーまたはクラスター上に構成されます。

この構成タスクを実行する場合は、210 ページの『メディアエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスターの構成』を参照してください。

サーバーまたはクラスターは、メディアエーション・モジュールのデプロイメントを受け入れるように構成されます。

これで、サーバーまたはクラスター上にメディアエーションをデプロイしたり、サーバーまたはクラスターを管理するその他のアクションを実行できるようになります。

使用可能なサーバーの管理:

サーバーの作成後、ランタイム状態のモニター、構成の変更、サーバーの始動および停止、その他の管理用タスクの実行などを通じてサーバーを管理することができます。

これらのタスクは、作成済みのサーバーに対して実行します。

アプリケーションやサービスの実行は、ESB サーバーの構成および実行によって異なります。サーバーの作成後、他のさまざまな管理用タスクを実行してサーバーを管理することができます。

サーバーの管理には、管理コンソールまたは各種の `wsadmin` コマンドのいずれかを使用することができます。

このトピックでは、管理コンソールによって使用可能なサーバーを管理する代替的な方法について要約を説明し、関連タスクの説明へのリンクを提供します。

- サーバーのリストを表示します。

ナビゲーション・ペインで、「サーバー」 → 「アプリケーション・サーバー」をクリックします。

「アプリケーション・サーバー」コレクション・ページが表示され、サーバー、サーバーのセル、およびサーバーを保持するノードがリストされます。このページでは、それぞれのサーバーが開始済み、停止中、あるいは使用不可のいずれの状況にあるかも表示されます。このページを使用して、サーバーに対するいくつかのアクションを実行できます。例えば、新規サーバーの作成、サーバーの始動または停止、サーバーの詳細設定を行うページのオープンなどです。

- サーバーに関する詳細を表示します。

「アプリケーション・サーバー」ページでサーバーのリストを表示し、サーバーの名前をクリックします。

すると、このサーバーの設定ページが表示されます。このページを使用して、サーバーおよびそのリソースの詳細情報を表示したり、構成を変更したりできます。

- 『サーバーの始動』を行います。

アプリケーションおよびサービスを実行可能にするには、サーバーが開始済みでなければなりません。サーバーがデプロイメント・マネージャー・セルにある場合、ノード・エージェントが既に実行中である必要があります。

– 管理対象ノードのノード・エージェントを開始するには、

`wesb_install¥profiles¥profile_name¥bin¥startNode` コマンドを実行します。ここで、*profile_name* には、管理対象ノードのカスタム・プロファイルの名前が入ります。

– サーバーを開始するには、「アプリケーション・サーバー」コレクション・ページでサーバーのリストを表示し、サーバーのチェック・ボックスを選択してから、「開始」をクリックします。

- サーバーの稼働をモニターします。

ランタイム・コンポーネントの状況をモニターして、開始後に必要な操作が可能な状態であることを確認する必要があります。

サーバー・ログに表示されたメッセージを参照できます。管理コンソールの「トラブルシューティング」 → 「ログとトレース (Logs and Trace)」ページを使用して、ランタイム・コンポーネントの状況をモニターすることもできます。

- サーバーを停止します。

サーバーを停止すると、アプリケーションおよびサービスを実行するためにこのサーバーを使用することができなくなります。

「アプリケーション・サーバー」コレクション・ページでサーバーのリストを表示し、サーバーのチェック・ボックスを選択してから「停止」をクリックします。

- 218 ページの『サーバーの削除』

サーバーを削除すると、サーバーの ID および構成が削除され、そのサーバーへのアプリケーションやサービスのデプロイメントも同時に削除されます。そのため、このサーバーはアプリケーションやサービスのデプロイに使用できなくなります。

「アプリケーション・サーバー」ページでサーバーのリストを表示し、サーバーのチェック・ボックスを選択してから「**削除**」をクリックします。

サーバーの削除:

サーバーを削除すると、サーバーの ID および構成が削除され、そのサーバーへのアプリケーションやサービスのデプロイメントも同時に削除されます。そのため、このサーバーはアプリケーションやサービスのデプロイに使用できなくなります。

アプリケーションやサービスを実行するための特定のサーバーが不要になった場合は、サーバーを削除することができます。

サーバーを削除するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サーバー」 → 「アプリケーション・サーバー」をクリックします。「アプリケーション・サーバー」ページにサーバーのリストが表示され、サーバーが開始済み、停止中、使用不可のいずれの状況にあるか表示されます。
2. サーバーが停止中であることを確認してください。

サーバーを停止するには、サーバー名の横のチェック・ボックスを選択してから、「**停止**」をクリックします。

3. サーバーを削除します。

サーバーを削除するには、サーバー名の横のチェック・ボックスを選択してから、「**削除**」をクリックします。

4. 変更をマスター構成に保管します。

これで、サーバーは「アプリケーション・サーバー」ページのリストから削除され、アプリケーションやサービスのデプロイに使用できなくなります。

サーバー・クラスターによるワークロードの平衡化

サーバー・クラスターは、一緒に管理され、ワークロード管理に参加する一群のサーバーです。管理者は、既存のサーバーおよびバス環境にサーバー・クラスターを作成および構成できます。

クラスター環境を作成する前に、以下のアクションを行ってください。

1. サーバー・クラスターを使用する環境を計画して準備します。例えば、次の項目について考慮します。
 - クラスタリングを正常にインプリメントするための十分なリソースがあることを確認します。
 - サーバーをクラスター化することはアプリケーションにとって有益かどうかを判断します。アプリケーションに大量の要求がある場合は、クラスターにインストールするとスループット処理の改善に役立つ場合があります。それ以外に

クラスターへのインストールを考慮する理由としては、定期保守および装置の障害時におけるアプリケーションの可用性の向上などがあります。

- いずれかのアプリケーションにサービスが含まれるかどうか判別します。
 - WebSphere Application Server Network Deployment によって提供される Network Deployment、およびクラスタリング・サポートについて理解を深めます。クラスタリング・サポートについて詳しくは、関連するトピックを参照してください。
 - まずこのタスクについて検討してから、ステップを実行してください。
2. デプロイメント・マネージャー・プロファイルと 1 つ以上のカスタム・プロファイルを作成し、カスタム・プロファイルをデプロイメント・マネージャー・セルにフェデレートすることにより、ネットワーク・デプロイメント・セルを作成します。

一連のサーバーによりサービス要求などのワークロードの平衡を取る場合は、サーバー・クラスターを作成してから、クラスターのメンバーとしてサーバーを追加することができます。また、バックアップ・クラスターを作成し、割り当て先のサーバー・クラスターのフェイルオーバー機能をサポートすることもできます。

ワークロード管理用のクラスターを構成および使用する方法について理解するため、次のシナリオについて考えてみます。クライアント要求は、単一マシンにある複数のクラスター・メンバー間で分散されます。(クライアントとは、エンド・ユーザーとサーバーを接続する、任意のサービス要求元、サーブレット、Java アプリケーション、あるいは他のプログラムまたはコンポーネントを指します。)さらに複雑なワークロード管理のシナリオでは、クラスター・メンバーを同じシスプレックス内に分散させることができます。

サーバー・クラスターの管理方法について詳しくは、以下のサブトピックを参照してください。サブトピックでは、WebSphere ESBでのサーバー・クラスターの使用に関連した情報を提供し、適切な WebSphere Application Server のトピックへのリンクを掲載しています。

- サーバー・クラスターを作成します。

サーバー・クラスターの作成について詳しくは、『サーバー・クラスターの作成』を参照してください。

- クラスターのサーバー・メンバーをいくつか作成します。

サーバー・メンバーの作成について詳しくは、222 ページの『クラスター・メンバーの作成』を参照してください。

- 使用可能なクラスターを管理します。

サーバー・クラスターを作成した後は、そのランタイム状態のモニター、構成の変更、クラスターの開始と停止を含め、広範な管理タスクを実行してクラスターを管理できます。

サーバー・クラスターの作成:

このタスクによりサーバー・クラスターを作成します。サーバー・クラスターは、一緒に管理され、ワークロード管理に参加する一群のサーバーです。

サーバー・クラスターを作成するには、カスタム・プロファイルを作成し、ノードをデプロイメント・マネージャー・セルにフェデレートしておく必要があります。。

一連のサーバーによりサービス要求などのワークロードの平衡を取る場合は、サーバー・クラスターを作成してから、クラスターのメンバーとしてサーバーを追加することができます。また、バックアップ・クラスターを作成し、割り当て先のサーバー・クラスターのフェイルオーバー機能をサポートすることもできます。

注: z/OS プラットフォーム上で、シस्पレックス内の複数システムに及び、Transaction の活動化ポリシーがデプロイされたステートフル・セッション Bean を備えたサーバーのクラスターを作成する計画の場合は、クラスタリングされたサーバーが稼働するシस्पレックス内の複数システム全体で共用する HFS (階層ファイル・システム) にパッシベーション・ディレクトリーを置いてください。

サーバー・クラスターを作成するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サーバー」 → 「クラスター」をクリックします。
2. 「新規」をクリックします。
3. 「基本クラスター情報の入力」ページで、以下の詳細を指定します。
 - a. 「クラスター名」には、クラスターの名前を入力します。
 - b. **オプション:** ノード有効範囲のルーティング最適化を使用不可にするには、「ローカルを優先」をクリアします。

デフォルトでは、「ローカルを優先」オプションが選択されています。これは、可能な場合には Enterprise JavaBean (EJB) 要求がクライアント・ノードに送付されることを示します。この機能を使用可能にすると、クライアント要求がローカル・エンタープライズ Bean に送信されるので、パフォーマンスが向上します。

- c. **オプション:** このクラスターの複製ドメインを作成するには、「このクラスターの複製ドメインを作成する」を選択します。

複製ドメインを使用し、セッション・マネージャー、動的キャッシュ、またはステートフル・セッション Bean のデータ、オブジェクト、またはイベントを、クラスター内のアプリケーション・サーバー間で転送します。分離された複製ドメインを作成し、複製のコンシューマーとして動作する各コンポーネントで使用します。例えば、セッション・マネージャーで使用するために 1 つの複製ドメインを、動的キャッシュで使用するために別の複製ドメインを構成できます。作成される複製ドメイン名は、クラスター名と同じになります。

- d. 「既存のアプリケーション・サーバー」の下で、「このクラスターに既存のサーバーを組み込まない」オプションが選択されたままにします (デフォルト)。

このオプションによって指定されるとおり、空のクラスターを作成する必要があります。

- e. 「次へ」をクリックします。
- 4. 「クラスター・メンバーの作成 (Create cluster members)」 ページで、クラスターで作成するクラスター・メンバーの詳細を指定します。

新規のクラスター・メンバーごとに、以下の操作を実行します。

- a. 「メンバー名」では、クラスターのメンバーとして追加する新規サーバーの名前を入力します。
 - b. 「ノードの選択」では、サーバーを稼働させるノードを選択します。
 - c. 「固有 HTTP ポートの生成」が選択されていることを確認します。
 - d. 「テンプレートの選択:」の下で、defaultESBServer テンプレートを選択します。
 - 1) 「デフォルトのアプリケーション・サーバー・テンプレート」を選択します。
 - 2) ドロップダウン・リストから、defaultESBServer を選択します。
 - e. 「適用」をクリックしてクラスター・メンバーを終了します。クラスター・メンバーはさらに追加できます。追加するすべてのクラスター・メンバーは、同じサーバー・テンプレートを基にして作成されるので、少ないステップで追加できます。
- 5. 「次へ」をクリックします。
 - 6. 「要約の表示。(View the summary.)」 ページでクラスターを作成するには、「終了」をクリックします。

詳細情報に誤りがある場合は、「戻る」をクリックして前のページに戻り、詳細の内容を変更できます。

- 7. 固有のポート番号を持つ仮想ホストを定義します。
 - a. ナビゲーション・ペインで、「環境」 → 「仮想ホスト」をクリックします。
 - b. コンテンツ・ペインで、「default_host」 → 「[追加プロパティ] ホスト別名」をクリックします。
 - c. 「新規」をクリックします。
 - d. 仮想ホストの設定ページで、ホスト名とポート番号を指定します。

注: z/OS では、仮想ホストのセットアップはオプションです。

- 8. 変更をマスター構成に保管します。構成の変更を保管する作業の一部として、「保管」ページで「保管」をクリックする前に、「ノードと変更を同期化」を選択できます。
- 9. クラスターを始動する前に、ノードと構成を同期させる必要があります。前のステップで構成を保管するときに「ノードと変更を同期化」を選択した場合は、このステップを無視できます。

このオプションを選択していない場合は、以下の操作のいずれかを行ってください。

- 自動同期を使用している場合は、同期処理が実行されるまで待ちます。
- 手動の同期処理を実行して、ノードに移された構成ファイルを取得します。
 - a. 「システム管理」 → 「ノード」をクリックします。
 - b. 「ノード」 ページで、ノード名を選択します。

- c. 「同期」または「完全再同期」をクリックします。

「ノード」ページに、ノードが同期されているかどうかを示す状況が表示されます。

クラスタのプロパティおよびローカル・トポロジーを表示できます。「サーバー」 → 「クラスタ」 → 「[コンテンツ・ペイン] *cluster_name*」をクリックします。

クラスタの構成は、ユーザーが変更できます。「保管」をクリックせず、管理構成を保管しなかった場合は、「構成」タブと「ローカル・トポロジー」タブのみが表示されます。「ランタイム」タブを表示するには、管理構成を保管する必要があります。また上述のとおり、変更がノードと同期されていることも確認してください。

この他に、さらにクラスタ・メンバーを作成したり、クラスタを開始したりすることができます。クラスタ構成オプションについて詳しくは、『クラスタによるワークロードの平衡化』を参照してください。

スクリプトを使用してクラスタを作成するタスクを自動化できます。クラスタでスクリプトを使用する方法については、「スクリプトによるクラスタの作成」を参照してください。

サーバー・クラスタを管理するには、「Server cluster collection」を使用できます。このページを使用して、クラスタのプロパティおよびトポロジーを表示し、必要に応じて変更を加えることができます。

クラスタ・メンバーの作成:

このタスクを使用して、サーバー・クラスタのメンバーとしてサーバーを作成します。

クラスタ・メンバーを作成できるようにするには、219 ページの『サーバー・クラスタの作成』の説明に従ってクラスタを作成しておく必要があります。

サーバー・クラスタでワークロードの分散にサーバーを参加させる場合は、そのサーバーのクラスタ・メンバーを作成します。

クラスタ・メンバーを作成するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、「サーバー」 → 「クラスタ」をクリックします。「サーバー・クラスタ」ページが表示され、デプロイメント・マネージャー・セルのクラスタがリストされます。
2. コンテンツ・ペインで、クラスタの名前をクリックします。
3. 「クラスタ・メンバー」をクリックします。「クラスタ・メンバー」ページが表示され、クラスタのメンバーのリスト、メンバーが置かれているノードの説明、およびメンバーが開始、停止、または問題が発生しているかどうかの説明が表示されます。
4. 「新規」をクリックし、「新規クラスタ・メンバーの作成 (Create new cluster members)」ページのステップに従います。

- a. 「メンバー名」では、クラスターのメンバーとして追加する新規サーバーの名前を入力します。
 - b. 「ノードの選択」では、サーバーを稼働させるノードを選択します。
 - c. 「固有 HTTP ポートの生成」が選択されていることを確認します。
 - d. 「テンプレートの選択:」の下で、defaultESBServer テンプレートを選択します。
 - 1) 「デフォルトのアプリケーション・サーバー・テンプレート」を選択します。
 - 2) ドロップダウン・リストから、defaultESBServer を選択します。
 - e. 「適用」をクリックしてクラスター・メンバーを終了します。クラスター・メンバーはさらに追加できます。追加するすべてのクラスター・メンバーは、同じサーバー・テンプレートを基にして作成されるので、少ないステップで追加できます。
 - f. 「次へ」をクリックします。
 - g. 新規クラスター・メンバーの要約情報を確認し、「終了」をクリックします。
5. 変更をマスター構成に保管します。構成の変更を保管する作業の一部として、「保管」ページで「保管」をクリックする前に、「ノードと変更を同期化」を選択できます。
6. クラスター・メンバーのプロパティを表示するには、クラスター・メンバー・ページの「メンバー名」の下でメンバーの名前をクリックします。これにより、クラスター・メンバー・インスタンスの設定ページが表示されます。

これで、既存のサーバー・クラスターのメンバーになるアプリケーション・サーバーが作成されました。

クラスターのプロパティおよびローカル・トポロジを表示できます。「サーバー」→「クラスター」→「[コンテンツ・ペイン] *cluster_name*」をクリックします。

クラスターの構成は、ユーザーが変更できます。「保管」をクリックせず、管理構成を保管しなかった場合は、「構成」タブと「ローカル・トポロジ」タブのみが表示されます。「ランタイム」タブを表示するには、管理構成を保管する必要があります。また上述のとおり、変更がノードと同期されていることも確認してください。

この他に、さらにクラスター・メンバーを作成したり、クラスターを開始したりすることができます。クラスター構成オプションについて詳しくは、『クラスターによるワークロードの平衡化』を参照してください。

スクリプトを使用してクラスター・メンバーを作成するタスクを自動化できます。

クラスター・メンバーを管理するには、クラスター・メンバー・ページを使用できます。このページを使用して、クラスター・メンバーをリストし、詳細を表示したり操作したりするメンバーを選択できます。

スクリプトを使用してクラスター・メンバーを作成するタスクを自動化できます。クラスター・メンバーでスクリプトを使用する方法については、「スクリプトによるクラスターの作成」を参照してください。

SCA ランタイム用サーバーまたはクラスターの構成:

サーバーまたはサーバー・クラスター上にメディエーション・モジュールをデプロイするには、事前にそのサーバーまたはクラスターを必要な SCA ランタイム用に構成しておく必要があります。この拡張構成で、SCA ランタイムで使用されるキュー宛先をローカルにホストするか、リモート・サーバーまたはクラスター上でホストするかを定義します。

このタスクを開始する前に、必要な以下のサーバーまたはサーバー・クラスターを 1 つまたは複数作成しておく必要があります。

- メディエーション・モジュールをデプロイするサーバーまたはクラスター。
- SCA ランタイムで使用するキュー宛先を作成するサーバーまたはクラスター。

両方に同じサーバーまたはクラスターを使用できます。

サーバーまたはサーバー・クラスターの作成に関する詳細は、以下のタスクの記述を参照してください。

- サーバーの作成
- サーバー・クラスターの作成

WebSphere ESB にメディエーション・モジュールをインストールすると、非同期対話を使用するメディエーション・モジュールのコンポーネントに多数のキュー宛先が作成されます。キュー宛先は、SCA ランタイムによってメディエーション・モジュールの処理中のメッセージを保持するために使用されます。

キュー宛先を作成する場所を指定するには、サーバーまたはクラスターの以下の代替構成のいずれかを実行します。

- サーバーまたはクラスターの拡張構成を設定して、宛先をホストします。この構成では、ローカルにデプロイされたメディエーション・モジュールまたは別のサーバーまたはクラスターにデプロイされたメディエーション・モジュールのキュー宛先をこのサーバーまたはクラスターがホスト可能になるように定義します。また、この構成では、サーバーまたはクラスターを `SCA.SYSTEM` バスのメンバーとして追加し、サーバーまたはクラスターのメッセージング・エンジンを構成します。

この構成タスクを実行する場合は、208 ページの『メディエーション・モジュールのキュー宛先をホストするためのサーバーまたはクラスターの構成』を参照してください。

- リモート宛先を使用するように、サーバーまたはクラスターの拡張構成を設定します。この構成では、このサーバーまたはクラスターが宛先にリモート宛先ロケーションを使用するよう定義します。

このサーバーまたはクラスターは `SCA.SYSTEM` バスのメンバーである必要はありません。

このサーバーまたはクラスターにメディアエーション・モジュールをデプロイすると、キュー宛先は SCA.SYSTEM バスのメンバーである別のサーバーまたはクラスター上に構成されます。

この構成タスクを実行する場合は、210 ページの『メディアエーション・モジュールのリモート宛先を使用するためのサーバーまたはクラスターの構成』を参照してください。

サーバーまたはクラスターは、メディアエーション・モジュールのデプロイメントを受け入れるように構成されます。

これで、サーバーまたはクラスター上にメディアエーションをデプロイしたり、サーバーまたはクラスターを管理するその他のアクションを実行できるようになります。

使用可能なサーバー・クラスターの管理:

サーバー・クラスターの作成後に、そのランタイム状態のモニタリング、構成の変更、クラスターまたはクラスター・メンバーの開始と停止、およびクラスター管理のためのその他のさまざまな管理用タスクの実行を行うことができます。

アプリケーションおよびサービスの実行は、クラスターとそのクラスターのメンバーであるサーバーの構成および実行に依存します。サーバー・クラスターの作成後に、クラスター管理のためのさまざまな管理用タスクを実行できます。

クラスターの管理は、管理コンソールまたは一連の `wsadmin` コマンドを使用して行えます。

このトピックでは、使用可能なクラスターを管理コンソールを使用して管理する別の方法についての概要と、関連するタスクの説明へのリンクを提供します。

- クラスターのリストを表示します。

ナビゲーション・ペインで、「サーバー」 → 「クラスター」をクリックします。

クラスターがリストされた「サーバー・クラスター (Server Cluster)」コレクション・ページが表示されます。このページには、クラスターが開始または停止しているか、あるいは使用不可かを示す各クラスターの状況も表示されます。このページを使用して、クラスターに関する一連のアクションを実行できます。例えば、新規クラスターを作成したり、クラスターを開始および停止したり、クラスターの詳細設定ページを開いたりできます。

- クラスターに関する詳細を表示します。

クラスターのリストを表示し、「サーバー・クラスター (Server Cluster)」コレクション・ページでクラスター名をクリックします。

そのクラスターの設定ページが表示されます。このページを使用して、サーバー・クラスターの構成およびローカル・トポロジを表示したり、変更したりできます。サーバー・クラスターの作成後に管理構成を保存すれば、サーバー・クラスターのランタイム情報を表示することもできます。

- クラスターのメンバーであるサーバーのリストを表示します。

クラスターに関する詳細を表示した後、「サーバー・クラスター (Server Cluster)」設定ページで「**クラスター・メンバー**」をクリックします。

「クラスター・メンバー」コレクション・ページが表示されます。このページにクラスターのメンバーがリストされ、メンバーが存在するノード、メンバーが開始しているか、停止しているか、または問題が発生していないかが示されます。このページを使用して、クラスター・メンバーに関する一連のアクションを実行できます。例えば、新規のクラスター・メンバーを作成したり、クラスター・メンバーの詳細設定ページを開いたりできます。

- クラスターの新規メンバーを作成します。

「クラスター・メンバー」コレクション・ページを表示し、「**クラスター・メンバー**」をクリックします。「クラスター・メンバー」ページにクラスターのメンバーがリストされ、メンバーが存在するノード、メンバーが開始しているか、停止しているか、または問題が発生していないかが示されます。

- クラスターを開始します。

アプリケーションおよびサービスを実行するには、クラスターが開始している必要があります。

クラスターのリストを表示し、「サーバー・クラスター (Server Cluster)」コレクション・ページでクラスターに関するチェック・ボックスを選択し、「**開始**」または「**リップルスタート (Ripplestart)**」をクリックします。

- クラスターおよびそのメンバーの実行をモニターします。

ランタイム・コンポーネントの状況をモニターして、開始後に必要な操作が可能な状態であることを確認する必要があります。

クラスター・ログに表示されたメッセージを参照できます。管理コンソールの「**トラブルシューティング**」 → 「**ログとトレース (Logs and Trace)**」ページを使用して、ランタイム・コンポーネントの状況をモニターすることもできます。

- ワークロード管理ランタイムの振る舞いを調整します。

アプリケーションにタイムアウトの問題が発生した場合や、ネットワークの待ち時間が極端な場合は、`com.ibm.CORBA.RequestTimeout` プロパティのタイムアウト間隔を変更します。または、クライアントのワークロード管理状態の最新表示が早すぎたり遅すぎたりする場合は、`com.ibm.websphere.wlm.unusable.interval` プロパティの間隔を変更します。

- クラスターを停止します。

サーバー・クラスターを停止すると、アプリケーションおよびサービスの実行に使用できなくなります。

クラスターのリストを表示し、「サーバー・クラスター (Server Cluster)」コレクション・ページでクラスターに関するチェック・ボックスを選択し、「**停止**」または「**即時停止 (Immediate Stop)**」をクリックします。

- クラスター・メンバーを削除します。

クラスター・メンバーを削除すると、クラスターからサーバーが除去され、そのサーバーは削除されます。サーバーを削除せずにクラスター・メンバーを削除することはできません。

クラスター・メンバーのリストを表示し、「クラスター・メンバー」コレクション・ページでクラスター・メンバーに関するチェック・ボックスを選択し、「削除」をクリックします。

- クラスターを削除します。

サーバー・クラスターを削除すると、クラスターおよびすべてのクラスター・メンバーが削除されます。クラスターを削除する場合、特定のクラスター・メンバーまたはそのクラスターの任意の箇所にインストールしたアプリケーションを残すことはできません。

クラスターのリストを表示し、「サーバー・クラスター (Server Cluster)」コレクション・ページでクラスターに関するチェック・ボックスを選択し、「削除」をクリックします。

バス・トポロジーのその他の用途

SCA モジュールに使用される SCA.SYSTEM バスは、WebSphere Application Server で使用される場合と同様、標準のサービス統合バスです。SCA.SYSTEM バスのほかにも、サービス統合バスを作成してサービス統合ロジックやその他のアプリケーションをサポートできます。

例えば SCA.APPLICATION.cell_name.Bus は、JMS バインディングでデプロイされたモジュールの JMS キュー宛先やその他の JMS リソースを定義するために提供および使用されます。

他のバスを作成して、WebSphere Application Server の場合と同様に使用したり (例えば、WebSphere ESB 内でサービス要求元やプロバイダーとして動作するアプリケーション用)、あるいは WebSphere MQ にリンクしたりできます。WebSphere ESB デプロイメント・マネージャーを使用して、WebSphere Application Server 上にデプロイされたアプリケーションおよびモジュール用の個々のアプリケーション・サーバーを管理することもできます。

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 ESB 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、関連した Service Component Architecture (SCA) モジュールを備えています。

WebSphere ESB 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、簡単な方法でサービス要求および応答のフォーマット、内容、またはターゲットを変更できます。

サービス・アプリケーション入門

サービス・アプリケーションのデプロイ後に、そのサービス・アプリケーションのコンポーネントを表示し、管理できます。

アプリケーションおよび関連する Service Component Architecture (SCA) モジュールを表示し、管理できます。WebSphere ESB がサポートする SCA モジュールの種類は、メディエーション・モジュールと呼ばれます。

デプロイしたすべての SCA モジュールをリストでき、SCA モジュールのサービス要求元およびサービス・プロバイダーへの接続方法の詳細を表示できます。

アプリケーション統合環境の概要:

サービス・アプリケーションには、関連付けられた Service Component Architecture (SCA) モジュールがあります。サービス・アプリケーションは、WebSphere ESB の EAR (Enterprise ARchive) ファイル内にデプロイします。

サービス・アプリケーションのデプロイ

サービス・アプリケーションを含む EAR ファイルをデプロイするプロセスは、他の EAR ファイルをデプロイするプロセスと同じです。ただし、SCA モジュールを含む EAR ファイルをデプロイした後は、以下に関する詳細を表示できます。

- サービス・アプリケーション。
- サービス・アプリケーションに関連付けられた SCA モジュール。
 - SCA モジュールがサービス要求元およびサービス・プロバイダーに接続する方法を確認できます。SCA モジュールは、エクスポートによってサービス要求元に、インポートによってサービス・プロバイダーに接続されます。

SCA モジュールの詳細の表示

表示可能な SCA モジュールの詳細には、以下に挙げるいくつかが含まれます。表示できる詳細の正確な内容は、SCA モジュールによって異なります。

- SCA モジュール名。
 - SCA モジュールの説明。
 - 関連アプリケーション名。
 - SCA モジュールのインポート。
 - インターフェース。
 - インポート・インターフェースは、SCA モジュールがサービスにアクセスするときの方法を記述した抽象定義です。
- 注:** WebSphere ESB は、WSDL インターフェースをサポートしますが、Java インターフェースはサポートしません。
- バインディング。
 - インポート・バインディングは具象定義です。SCA モジュールがサービスにアクセスするときの物理メカニズムを指定します。例えば、SOAP/HTTP を使用します。
 - SCA モジュールのエクスポート。
 - インターフェース。
 - エクスポート・インターフェースは、サービス要求元が SCA モジュールにアクセスするときの方法を記述した抽象定義です。

注: WebSphere ESB は、WSDL インターフェースをサポートしますが、Java インターフェースはサポートしません。

- バインディング。
 - エクスポート・バインディングは具象定義です。サービス要求元が SCA モジュールにアクセスするときと、間接的にサービスにアクセスするときの物理メカニズムを指定します。

デフォルトのインポート・バインディングが 2 つの SCA モジュールに接続する場合は、WebSphere ESB を使用してバインディングのターゲットを変更できます。SCA インポート・バインディングを変更すると、別のサービス・プロバイダーを呼び出すことができます。

サービス・アプリケーションの学習:

サービス・アプリケーションはサービスを提供し、関連した Service Component Architecture (SCA) モジュールを備えています。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 ESB がサポートする SCA モジュールの種類は、メディエーション・モジュールと呼ばれます。

メディエーション・モジュール:

メディエーション・モジュールは、サービス要求のフォーマット、内容、またはターゲットの変更が可能な 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・プロバイダーの間の送信途中にあるメッセージを操作します。これにより、メッセージをさまざまなサービス・プロバイダーに送付できます。メッセージを変換することもできます。つまり、メッセージの内容または形式を変更できます。さらに、メディエーション・モジュールは、メッセージ・ロギングやユーザーの要件に応じたエラー処理などの機能を提供します。

メディエーション・モジュールのコンポーネント

メディエーション・モジュールに含まれる項目には、以下のものがあります。

- インポート
 - インポートは、Service Component Architecture (SCA) モジュールとサービス・プロバイダー間の対話を定義します。
 - インポートによって、SCA モジュールは外部サービスをローカルのように呼び出すことができます。
 - メディエーション・モジュールのインポートは WebSphere ESB から表示できます。インポート・バインディングが SCA バインディングの場合は、別の SCA モジュールを指すように変更できます。
- エクスポート
 - エクスポートは、Service Component Architecture (SCA) モジュールとサービス要求元間の対話を定義します。
 - エクスポートによって、SCA モジュールはサービスを提供することができます。エクスポートは、SCA モジュールの外部インターフェース (アクセス・ポイント) を定義します。

- メディエーション・モジュールのエクスポートは WebSphere ESB から表示できます。
- Service Component Architecture (SCA) コンポーネント
 - SCA コンポーネント、つまりサービス・コンポーネントは SCA ビルディング・ブロックです。SCA コンポーネントを使用して、メディエーション・モジュールなどの SCA モジュールを作成します。WebSphere Integration Developer を使用して、SCA モジュールおよびコンポーネントをグラフィカルに作成しカスタマイズすることができます。
 - 一般に、メディエーション・モジュールには、メディエーション・フロー・コンポーネントと呼ばれる特定のタイプの SCA コンポーネントが含まれます。メディエーション・モジュールに含めることができるのは、最大 1 つのメディエーション・フロー・コンポーネントです。
 - メディエーション・フロー・コンポーネントは、1 つまたは複数のメディエーション・プリミティブを含むことも、メディエーション・プリミティブをまったく含まないようにすることもできます。WebSphere ESB は、メッセージ・ルーティングおよび変換の機能を備えたメディエーション・プリミティブの提供セットをサポートします。WebSphere ESB がサポートするメディエーション・プリミティブの 1 つを使用すれば、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・モジュールには、メディエーション・フロー・コンポーネントが含まれている必要はありません。メディエーション・フロー・コンポーネントが含まれていないメディエーション・モジュールの目的は、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ことです。例えば、サービス要求は SOAP/JMS を使用して作成されたが、送信前に SOAP/HTTP に変換する必要がある場合があります。

注: メディエーション・モジュールは WebSphere ESB から表示できます。

WebSphere ESB から、メディエーション・モジュールに対して、制限付きで変更を行うこともできます。ただし、WebSphere ESB から SCA コンポーネントまたはメディエーション・プリミティブを表示あるいは変更することはできません。SCA コンポーネントおよびメディエーション・プリミティブをカスタマイズするには、WebSphere Integration Developer を使用してください。

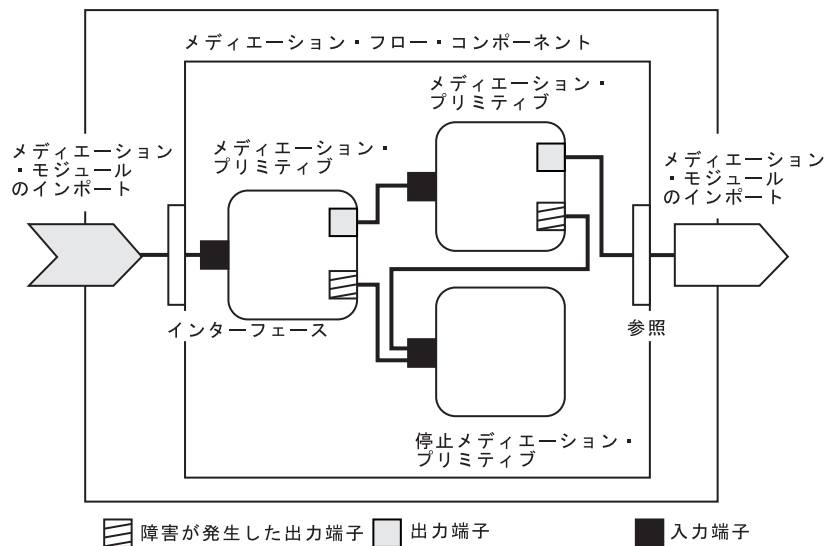


図 15. メディエーション・モジュールの簡単な例： メディエーション・モジュールには 1 つのメディエーション・フロー・コンポーネントが含まれます。メディエーション・フロー・コンポーネントにはメディエーション・プリミティブが含まれます。

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、エンタープライズ・アーカイブ (EAR) ファイル内で WebSphere ESB にデプロイされます。したがってメディエーション・モジュールは、エンタープライズ・アプリケーションのデプロイと同じ方法で WebSphere ESB にデプロイされます。

WebSphere Integration Developer はメディエーション・モジュールを Java アーカイブ (JAR) ファイル内にパッケージし、次にこの JAR ファイルが EAR ファイルに保管されます。

論理的には、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。しかし実際には、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。

- EAR ファイル
 - JAR ファイルを格納します。
 - メディエーション・モジュールを格納します。

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

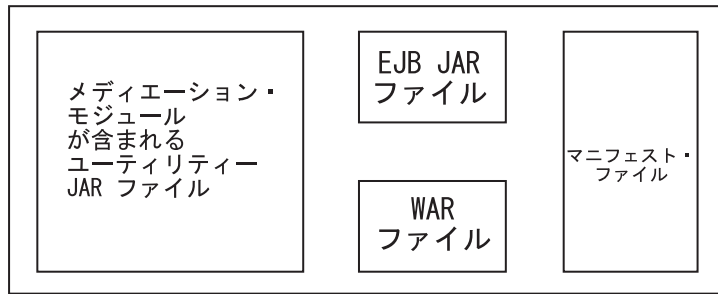


図 16. メディエーション・モジュールが含まれる EAR ファイルの簡単な例：EAR ファイルには JAR ファイルが含まれます。ユーティリティー JAR ファイルにはメディエーション・モジュールが含まれます。

インポートとインポート・バインディング:

インポートにより、Service Component Architecture (SCA) モジュールは、ローカルのサービスに対するように外部サービス (SCA モジュールの外部にあるサービス) にアクセスできます。インポートは、SCA モジュールとサービス・プロバイダーの間の相互作用を定義します。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ない場合、インポートは不要です。メディエーション・モジュールは SCA モジュールのタイプです。

インターフェースとバインディング

1 つの SCA モジュールのインポートでは、少なくとも 1 つのインターフェースを必要とし、1 つの SCA モジュールのインポートにはバインディングが 1 つだけあります。

- インターフェース。
 - インポート・インターフェースは抽象定義です。アクセス・ポイントを定義します。
 - インポート・インターフェースは、Web サービスを記述する際に使用する XML 言語である、Web Services Description Language (WSDL) によって定義されます。
 - 1 つの SCA モジュールは、多くのインポート・インターフェースを持つことができます。
- バインディング。
 - インポート・バインディングは具象定義です。SCA モジュールが外部サービスにアクセスするために使用する物理的なメカニズムを指定します。

サポートされるインポート・バインディング

WebSphere ESBは、次のインポート・バインディングをサポートします。

- Web サービス・バインディング
 - SOAP/HTTP
 - SOAP/JMS

- SCA バインディング
 - SCA モジュールには、SCA バインディングを指定できます。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。
- Java Message Service (JMS) 1.1 のバインディング
 - JMS は、WebSphere ファミリーとのインターオペラビリティを許可します。
 - JMS は、TCP/IP や HTTP(S) などのさまざまなトランスポート・タイプを利用します。
 - ビジネス・オブジェクト (BO) XML を含んだ JMS テキスト・メッセージをサポートする、事前定義 JMS バインディングがあります。事前定義 JMS バインディングは、シリアル化 Java ビジネス・オブジェクトを含んだ JMS オブジェクト・メッセージもサポートします。
 - JMS カスタム・バインディングを使用して、他のタイプの JMS メッセージをサポートできます。ただし、カスタム・バインディングでは、メッセージを変換するためのコーディングが必要です。
- WebSphere アダプターのバインディング
 - WebSphere アダプターにより、Enterprise Information Systems (EIS) との相互作用を可能にします。

エクスポートとエクスポート・バインディング:

エクスポートにより、Service Component Architecture (SCA) モジュールは他のモジュールにサービスを提供できます。エクスポートでは、SCA モジュールとサービス要求元の間の相互作用を定義します。エクスポート・バインディングでは、他のモジュールが SCA モジュールにアクセスする特定の方法を定義します。

メディエーション・モジュールは SCA モジュールのタイプです。

インターフェースとバインディング

SCA モジュールのエクスポートには、少なくとも 1 つのインターフェースが必要です。

- インターフェース。
 - エクスポート・インターフェースは抽象定義です。アクセス・ポイントを定義します。
 - エクスポート・インターフェースは、Web サービスを記述する際に使用する XML 言語である、Web Services Description Language (WSDL) を使用して定義されます。
 - 1 つの SCA モジュールは、多くのエクスポート・インターフェースを持つことができます。
- バインディング。
 - エクスポート・バインディングは具象定義です。サービス要求元がサービスにアクセスするために使用する物理的なメカニズムを指定します。
 - 通常、1 つの SCA モジュールには、1 つのバインディングが指定されています。バインディングが指定されていないエクスポートは、ランタイムにより SCA バインディングを持つエクスポートと解釈されます。

サポートされるエクスポート・バインディング

WebSphere ESBは、次のエクスポート・バインディングをサポートします。

- Web サービス・バインディング
 - SOAP/HTTP
 - SOAP/JMS
- SCA バインディング
 - SCA モジュールには、SCA バインディングを指定できます。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。
- Java Message Service (JMS) 1.1 のバインディング
 - JMS は、WebSphere ファミリーとのインターオペラビリティを許可します。
 - JMS は、TCP/IP や HTTP(S) などのさまざまなトランスポート・タイプを利用します。
 - ビジネス・オブジェクト (BO) XML を含んだ JMS テキスト・メッセージをサポートする、事前定義 JMS バインディングがあります。事前定義 JMS バインディングは、シリアル化 Java ビジネス・オブジェクトを含んだ JMS オブジェクト・メッセージもサポートします。
 - JMS カスタム・バインディングを使用して、他のタイプの JMS メッセージをサポートできます。ただし、カスタム・バインディングでは、メッセージを交換するためのコーディングが必要です。
- WebSphere アダプターのバインディング
 - WebSphere アダプターにより、Enterprise Information Systems (EIS) との相互作用を可能にします。

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 ESB により、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局面を表示および変更することができます。

サービス・アプリケーションはサービスを提供し、関連した Service Component Architecture (SCA) モジュールを備えています。WebSphere ESBでサポートされる SCA モジュールのタイプはメディエーション・モジュールです。

表示可能な SCA モジュールの詳細

Service Component Architecture (SCA) モジュールを含む EAR (Enterprise ARchive) ファイルをデプロイした後は、SCA モジュールの詳細を表示できます。ご使用の SCA モジュールすべてとそれらの関連アプリケーションをリストし、特定の SCA モジュールについて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、以下の情報の一部が含まれています。

- 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。

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した後は、管理コンソールを使用して次の SCA モジュールの詳細を変更できます。

- タイプ SCA のインポート・バインディング。
 - インポート・バインディングを変更することにより、サービス相互作用を変更できます。
 - 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、3 番目の 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。

SCA モジュールでは、タイプの異なるバインディングを行うことができます。WebSphere ESB は、異なるタイプの SCA モジュール バインディングを数多くサポートします。例えば、Web サービス・バインディングと SCA バインディングがあります。ただし、WebSphere ESB からは、SCA インポート・バインディングのみを変更できます。

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。

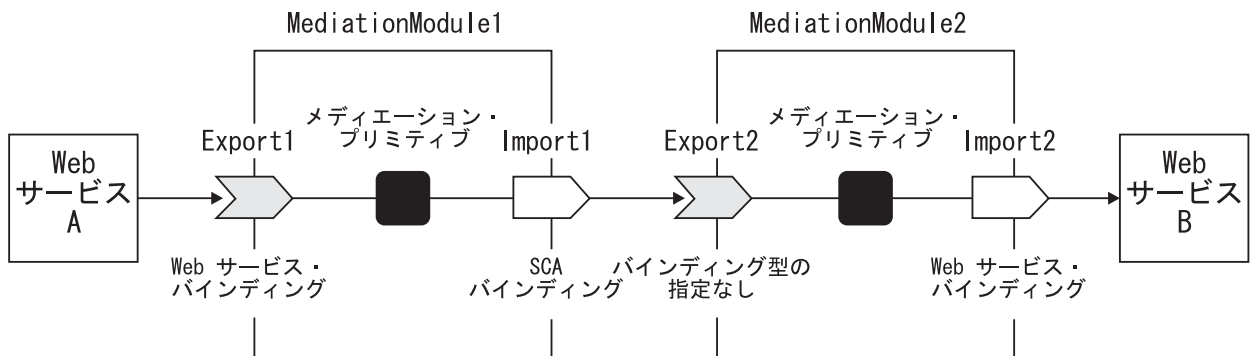


図 17. 別のメディエーション・モジュールと相互作用するメディエーション・モジュールの例： MediationModule1 は、MediationModule2 に接続

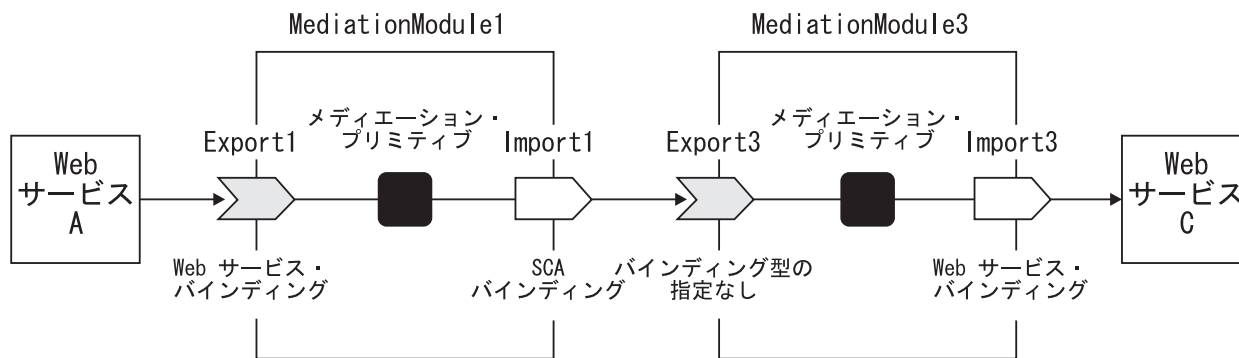


図 18. 別のメディエーション・モジュールと相互作用するメディエーション・モジュールの例： MediationModule1 は MediationModule3 に接続

実動サーバーへのモジュールのインストール

このトピックでは、テスト・サーバーからアプリケーションを取り出して実稼働環境にデプロイするステップについて説明します。

実動サーバーにサービス・アプリケーションをデプロイする前に、テスト・サーバーでアプリケーションをアセンブルおよびテストします。テストが終了したら、『サーバーへのデプロイの準備』の説明のとおりに関連するファイルをエクスポートし、このファイルをデプロイする実動システムに移します。詳しくは、インフォメーション・センターの WebSphere Integration Developer および WebSphere Application Server Network Deployment バージョン 6 を参照してください。

1. モジュールおよび他のファイルを実動サーバーにコピーします。

アプリケーションに必要なモジュールおよびリソース (EAR、JAR、RAR、および WAR の各ファイル) が実稼働環境に移動します。

2. serviceDeploy コマンドを実行して、インストール可能な EAR ファイルを作成します。

このステップでは、アプリケーションを実動サーバーにインストールする準備として、モジュールをサーバーに定義します。

- a. デプロイするモジュールを含む JAR ファイルを見つけ出します。
 - 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: アプリケーションをインストールするには、管理コンソールとスクリプトのどちらでも使用できます。詳しくは、WebSphere Application Server インフォメーション・センターを参照してください。

4. 構成を保管します。これでモジュールがアプリケーションとしてインストールされます。
5. アプリケーションを開始します。

これで、アプリケーションはアクティブになり、処理はモジュールを介して流れるようになります。

アプリケーションをモニターして、サーバーが要求を正しく処理しているか確認してください。

コンソールによるメディエーション・モジュール EAR ファイルのインストール:

メディエーション・モジュールのインストールは、そのメディエーション・モジュールのインストール可能 EAR ファイルを、メディエーション・モジュールをホストするサーバーまたはクラスターに移動する操作で構成されます。インストールされたメディエーション・モジュールが正常に開始および実行されると、デプロイされたと見なすことができます。

メディエーション・モジュールを JAR ファイルにエクスポートしている場合は、`serviceDeploy` コマンドを使用して、メディエーション・モジュールの JAR ファイルからインストール可能 EAR ファイルを作成します。メディエーション・モジュールのインストール可能 EAR ファイルの作成方法については、236 ページの『実動サーバーへのモジュールのインストール』を参照してください。

WebSphere ESB がメディエーション・モジュールによって提供される機能を使用してアプリケーションとサービスを統合できるようにするには、そのモジュールの EAR ファイルをサーバーまたはクラスターにインストールしてから、デプロイされたモジュールを開始する必要があります。

このトピックでは、管理コンソールを使用してメディエーション・モジュール EAR ファイルをインストールする方法について説明します。それとは別に、エンタープライズ・アプリケーション・ファイルを WebSphere Application Server にインストールする場合のように、`wsadmin` ツールの `install` コマンドや `installinteractive` コマンドのような他の方法を使用してインストールすることも可能です。

重要: 以下のステップを実行し始めてからアプリケーションのインストールを中止する場合は、「キャンセル」をクリックして終了します。管理コンソールの別のページに移動する場合は、その前に必ず、アプリケーションのインストール・ページで「キャンセル」をクリックしてください。

管理コンソールを使用してメディエーション・モジュール EAR ファイルをインストールするには、以下のステップを実行します。

1. コンソールのナビゲーション・ツリーで、「アプリケーション」 → 「新規アプリケーションのインストール」をクリックします。2 ページある「アプリケーション・インストールの準備」ページの最初のページが表示されます。
2. 最初の「アプリケーション・インストールの準備」ページで、以下のサブステップを実行します。
 - a. メディエーション・モジュール EAR ファイル (.ear ファイル) の絶対パス名を指定します。インストールする EAR ファイルは、クライアント・マシン (管理コンソール Web ブラウザーを実行するマシン) またはサーバー・マシン (クライアントの接続先マシン) にあります。クライアント・マシンの EAR ファイルを指定すると、管理コンソールは管理コンソールが動作しているマシンにその EAR ファイルをアップロードし、アプリケーションのインストールを続行します。

- b. 「次へ」をクリックします。
3. 2 番目の「アプリケーション・インストールの準備」ページで、以下のサブステップを実行します。
 - a. デフォルト・バインディングを生成するかどうかを選択します。

デフォルト・バインディングを使用すると、アプリケーション内の不完全なバインディングすべてにデフォルト値が入力されます。既存のバインディングは変更されません。

デフォルト・バインディングの生成で使用されるデフォルト値は、カスタマイズすることができます。

- b. 「次へ」をクリックします。 セキュリティ警告が表示されたら、「**継続**」をクリックします。「新規アプリケーションのインストール」ページが表示されます。デフォルト・バインディングを生成することを選択し、アプリケーションでリソース・アダプターを使用しない場合は、要約ステップ(以下の最後のステップ)に進むことができます。
4. 「ステップ: インストール・オプションの選択」パネルで、以下の設定の値を入力します。このページの設定値について詳しくは、「インストール・オプションの選択」設定を参照してください。デフォルト値は、値が指定されない場合に使用されます。
 - a. 「アプリケーションをインストールするディレクトリー」では、アプリケーション EAR ファイルをインストールするディレクトリーを指定します。デフォルト値は `APP_INSTALL_ROOT/cell_name` の値です。`APP_INSTALL_ROOT` 変数は、`install_root/installedApps` です。例えば、WebSphere ESB が Windows の `C:\Program Files\WESB\` にインストールされている場合、デフォルト・ロケーションは `C:\Program Files\WESB\profile_name\installedApps\cell_name` になります。

注: アプリケーションがスタンドアロン・プロファイルにインストールされるときにインストール・ディレクトリーが指定されていないと、アプリケーションは `APP_INSTALL_ROOT/base_cell_name` にインストールされます。スタンドアロン・サーバーをデプロイメント・マネージャー・セルに追加すると、新規サーバー構成のセル名は、デプロイメント・マネージャー・ノードのセル名になります。addNode ユーティリティーで `-includeapps` オプションを使用すると、addNode 操作より先にインストールされていたアプリケーションは、引き続き `APP_INSTALL_ROOT/base_cell_name` をインストール・ディレクトリーとして使用します。ただし、スタンドアロン・サーバーがネットワーク構成に追加された後にインストールされるアプリケーションは、デフォルトのインストール・ディレクトリー `APP_INSTALL_ROOT/network_cell_name` を使用します。addNode オペレーションの実行時にアプリケーションを `APP_INSTALL_ROOT/network_cell_name` ロケーションに移動するには、インストール時に、`${APP_INSTALL_ROOT}/${CELL}` としてインストール・ディレクトリーを明示的に指定します。そのような場合、アプリケーション・ファイルは必ず `APP_INSTALL_ROOT/current_cell_name` 下に置かれます。

- b. 「アプリケーションの配布」では、WebSphere ESB によってインストール先のアプリケーション・バイナリーを展開/削除するかどうかを指定します。デフォルトでは、アプリケーションの配布が使用可能になっています。したがって、コンソールで変更を保管すると、新規にインストールされたアプリケーションのアプリケーション・バイナリーは指定されたディレクトリーに展開されます。またアプリケーション・バイナリーは、アンインストールして構成の変更を保管すると削除されます。このオプションを使用不可に設定する場合は、アプリケーションを実行するすべてのノードの宛先ディレクトリーで、必ずアプリケーション・バイナリーを適切に展開する必要があります。

重要: このオプションを使用不可に設定し、アプリケーション・バイナリーをノードにコピーおよび展開しない場合は、後で構成を保管したり、手動で同期をとったりしても、アプリケーション・バイナリーはノードに移動しません。

- c. 「バイナリー構成の使用」では、サーバーまたはクラスターが、バインディング、拡張子、またはデプロイメント記述子に関して、アプリケーション・デプロイメント文書 `deployment.xml` ファイルにあるものを使用する (デフォルト) か、EAR ファイルにあるものを使用するかを指定します。
- d. 「アプリケーション名」では、アプリケーションの名前を入力します。アプリケーション名はセル内で固有のものにする必要があります。オブジェクト名での使用が許可されていない文字は使用できません。オブジェクト名での使用が許可されていない文字のリストについては、「オブジェクト名」を参照してください。
- e. 「リソース用の MBean の作成」では、アプリケーションの開始時に、アプリケーション内のさまざまなリソース (サーブレットや JSP ファイルなど) 用に MBean を作成するかどうかを指定します。デフォルトでは、MBean インスタンスが作成されます。
- f. 「クラスの再ロードを使用可能にする」では、アプリケーション・ファイルの更新時にクラスの再ロードを使用可能にするかどうかを指定します。デフォルトでは、クラスの再ロードは使用可能になりません。クラスの再ロードを可能にすると、メディエーション・モジュールの `deployment.xml` ファイルで `reloadEnabled` が `true` に設定されます。メディエーション・モジュールのクラス定義が変更されると、サーバー・ランタイムはアプリケーションを停止して開始し、アプリケーション・クラスを再ロードします。
- g. 「再ロード間隔 (秒)」では、更新済みファイルを探してアプリケーションのファイル・システムをスキャンするときの秒数を指定します。デフォルトは、EAR ファイルの IBM 拡張 (META-INF/ibm-application-ext.xmi) ファイルで指定されている再ロード間隔属性の値です。再ロードを使用可能にするには、ゼロより大きい値 (例えば、1 から 2147483647 の範囲) を指定します。再ロードを使用不可にするには、ゼロ (0) を指定します。

ここで指定する再ロード間隔は、クラスの再ロードが使用可能になっている場合にのみ有効になります。

- h. 「Web サービスのデプロイ」では、アプリケーションのインストール時に Web サービスのデプロイ・ツール `wsdeploy` を実行するかどうかを指定します。このツールは、Web サービスを使用してアプリケーションを実行す

るために必要なコードを生成します。デフォルトでは、wsdeploy ツールは実行されません。EAR ファイルに Web サービスを使用するモジュールが含まれており、前にこのファイルで wsdeploy ツールが実行されたことがない場合は、アセンブリー・ツールの「デプロイ」メニュー選択またはコマンド行から、この設定を使用可能にする必要があります。

- i. 「入力のオフ/警告/失敗の妥当性検査」では、WebSphere ESB により、アプリケーションのインストール時または更新時に指定したアプリケーション参照を検査するかどうか、さらに妥当性検査を使用可能にする場合は、参照に誤りがあることを警告するか、またはオペレーションを失敗させるかどうかを指定します。アプリケーションは通常、コンテナ管理パーシスタンス (CMP) Bean のデータ・ソースを使用して、またはデプロイメント記述子で定義されるリソース参照またはリソース環境参照を使用して、リソースを参照します。妥当性検査では、アプリケーションによって参照されるリソースが、そのアプリケーションのデプロイメント・ターゲットの有効範囲内で定義されているかどうかを検査します。リソースの妥当性検査を行わない場合は「オフ」、誤りのあるリソース参照について警告メッセージを出す場合は「警告」、誤りのあるリソース参照の結果、失敗するオペレーションを停止する場合は「失敗」を選択します。
 - j. 「プロセス組み込み構成」では、組み込み構成を処理するかどうかを指定します。組み込み構成は、resource.xml や variables.xml などのファイルで構成されています。選択されたか、または true の場合、組み込み構成は .ear ファイルからアプリケーション有効範囲にロードされます。.ear ファイルに組み込み構成が含まれていない場合のデフォルトは false です。.ear ファイルに組み込み構成が含まれている場合のデフォルトは true です。
5. 「ステップ: サーバーにモジュールをマップ」パネルで、すべてのモジュールに対し、「クラスターおよびサーバー」リストからターゲットのサーバーまたはクラスターを選択します。「モジュール」の横にあるチェック・ボックスを選択して、メディエーション・モジュールを選択します。

アプリケーションが WebSphere Adapter を使用する場合は、RAR ファイルごとにターゲットのサーバーまたはクラスターを指定します。また、RAR モジュールに定義されているリソース・アダプターを使用するその他のモジュールは、必ず、すべて同じターゲットにマップしてください。

注: サーバーに RAR ファイルをインストールする場合、WebSphere ESB はコネクタ・モジュールのマニフェスト (MANIFEST.MF) を探します。最初、RAR ファイルの connectorModule.jar ファイルで探し、_connectorModule.jar ファイルからマニフェストをロードします。クラスパス・エントリが connectorModule.jar ファイルからのマニフェストにある場合は、RAR はそのクラスパスを使用します。インストール済みコネクタ・モジュールが必要なクラスおよびリソースを見付けることを確認するには、コンソールを使用して、RAR の「クラスパス」設定を確認します。クラスパス設定について詳しくは、管理コンソールの「リソース・アダプター」設定および WebSphere 関連のリソース・アダプターの設定を参照してください。

Web サーバーは、アプリケーションへの要求を送付するターゲットとして指定できます。各 Web サーバーのプラグイン構成ファイル (plugin-cfg.xml) は、

Web サーバーを経由して送付されるアプリケーションに基づいて生成されます。Web サーバーによってアプリケーションにサービスを提供する場合は、**Ctrl** キーを使用して、アプリケーション・サーバーまたはクラスターを Web サーバーと一緒に選択します。これは、その Web サーバーのプラグイン構成ファイル `plugin-cfg.xml` を、Web サーバーを経由して送付されるアプリケーションに基づいて生成するためです。

6. アプリケーションがリソース参照を定義する場合は、「ステップ: リソース参照をリソースにマップ」で、リソース参照で定義される論理名を表すリソースの JNDI 名を指定します。アプリケーションで定義される各リソース参照は、「要約」パネルで「終了」をクリックする前に、WebSphere ESB 構成で定義されているリソースにバインドする必要があります。
 - a. **オプション:** ログイン構成名とリソースの認証プロパティを指定します。
 - b. 「**OK**」をクリックして値を保管し、マッピング・ステップに戻ります。
7. アプリケーションで Web モジュールを使用する場合は、「ステップ: Web モジュールの仮想ホストをマップ」で、アプリケーションで定義されている Web モジュールにマップする仮想ホストをリストから選択します。

仮想ホスト定義で指定されたポート番号は、Web モジュールのサーブレットや JSP ファイルなどの成果物にアクセスするために使用する URL で使用されます。各 Web モジュールには、マップ先となる仮想ホストが必要です。必要とされる仮想ホストすべてを指定しないと、「要約」パネルで「終了」をクリックした後に、検証エラーが表示されます。

8. アプリケーションのデプロイメント記述子でセキュリティー・ロールが定義されている場合は、「ステップ: セキュリティー・ロールをユーザー/グループにマップ」で、セキュリティー・ロールのそれぞれにマップするユーザーとグループを指定します。

「**ロール**」を選択して、ロールのすべてまたは個々のロールを選択します。以下の選択項目から、ロールごとに 1 つセキュリティーの適用方法を選択します。

オプション	説明
全員	これは、セキュリティーなしと同等です。
全認証者	正当なユーザー名とパスワードで認証するユーザーは、だれでもこのロールのメンバーです。
マップされたユーザー	個々のユーザーがこのロールのメンバーとしてリストされます。
マップされたグループ	グループは、ユーザーを追加するために最も便利な方法です。指定されたグループのメンバーすべてがこのロールのメンバーになります。

「マップされたユーザー」または「マップされたグループ」では、ユーザー・レジストリーから特定のユーザーまたはグループを選択し、以下のサブステップを実行します。

- a. ロールを選択し、「ユーザーのルックアップ」または「グループのルックアップ」をクリックします。
 - b. 表示される「ユーザー/グループのルックアップ」パネルで検索条件を入力し、ユーザー・レジストリーからユーザーまたはグループのリストを抽出します。
 - c. 表示される結果から、個々のユーザーまたはグループを選択します。
 - d. 「OK」をクリックして、選択したユーザーまたはグループを、「ステップ: セキュリティー・ロールをユーザー/グループにマップ」パネルで選択したロールにマップします。
9. アプリケーションのデプロイメント記述子で Run As ロールが定義されている場合は、「ステップ: RunAs ロールをユーザーにマップ」で、すべての Run As ロールに対して Run As ユーザー名とパスワードを指定します。Run As ロールは、別のエンタープライズ Bean との相互作用中に、特定のロールとして実行する必要のあるエンタープライズ Bean によって使用されます。「ロール」を選択して、ロールのすべてまたは個々のロールを選択します。ロールを選択した後、ユーザー名、パスワード、および確認パスワードの値を入力し、「適用」をクリックします。
 10. アプリケーションにリソース環境参照が含まれている場合は、「ステップ: リソース環境参照をリソースにマップ」で、リソース環境参照で定義される論理名にマップするリソースの JNDI 名を指定します。各リソース環境参照にリソースが関連付けられていない場合は、「終了」をクリックした後に検証エラーが表示されます。
 11. アプリケーションが Run-As ID をシステム ID として定義する場合は、「ステップ: RunAs システムを RunAs ロールに置換」において、オプションでその ID を Run-As ロールに変更し、指定された Run As ロールのユーザー名とパスワードを指定できます。システム ID を選択することは、WebSphere Application Server のセキュリティー・サーバー ID を使用して呼び出しが行われることを意味します。この ID には多くの特権が付与されるので、慎重に使用してください。
 12. アプリケーションに、Oracle データベースがバックエンド処理を実行しているリソースにマップするリソース参照がある場合は、「ステップ: Oracle タイプ・プロバイダーの分離レベルの指定 (Step: Specify the isolation level for Oracle type provider)」で、アプリケーションによって使用されるときにそれらのリソースで使用される分離レベルを指定または訂正します。Oracle データベースは、ReadCommitted と Serializable の分離レベルのみをサポートします。
 13. 「要約」パネルで、アプリケーション・モジュールをインストールするセル、ノード、およびサーバーを検証します。
 - a. 「セル/ノード/サーバー」の横で、「ここをクリックする」をクリックします。
 - b. 設定を検証します。
 - c. 「終了」をクリックします。

いくつかのメッセージが表示され、アプリケーション・ファイルが正常にインストールされたかが示されます。

OutOfMemory 例外を受け取り、ソース・アプリケーション・ファイルがインストールされていない場合は、システムのメモリーが不足しているか、アプリケーションに含まれるモジュールが多すぎてサーバーに正常にインストールできなくなっている可能性があります。例外の原因がシステム・メモリーの不足ではない場合は、.ear ファイルに含まれるモジュールが少なくなるように再度アプリケーションをパッケージします。システム・メモリーの不足とモジュールの数が例外の原因ではない場合は、管理コンソールを実行しているアプリケーション・サーバーの Java 仮想マシン・ページで指定したオプションを確認します。その後、アプリケーション・ファイルを再度インストールしてください。

Windows インストール時には、特定のアプリケーション・ファイルが、構成セッションによって表されるディレクトリーに展開され、構成が保管されると、これらのファイルは WebSphere Application Server の構成リポジトリーに保管されます。Windows マシンでは、ファイル・パスの長さが 256 文字に制限されています。よって、構成セッションまたは構成リポジトリーのアプリケーション・ファイルのパスが 256 文字より長くなると、アプリケーションのインストールが失敗します。FileNotFound 例外が発生し、「*path name too long*」という文字列を含むメッセージが表示される場合もあります。この問題から回復するには、アプリケーション名とモジュール URI 名の長さを短くし、ファイル・パスの長さが短くなるようにします。その後、アプリケーション・ファイルを再度インストールしてください。

アプリケーション・ファイルを正常にインストールしたら、以下のアクションを実行します。

1. アプリケーションが必要とする共有ライブラリーすべてをアプリケーションに関連付けます。
2. 変更を構成に保管します。アプリケーションは管理構成に登録され、アプリケーション・ファイルは、デフォルトでターゲット・ディレクトリー *install_root/installedApps/cell_name* に、またはユーザーが指定するディレクトリーにコピーされます。Network Deployment プロファイルにインストールされると、デプロイメント・マネージャーの構成と個々のノードの構成が同期されるときに、ファイルがリモート・ノードにコピーされます。
3. モジュールをサーバー・クラスターでデプロイする場合は、「エンタープライズ・アプリケーション」ページの「更新のロールアウト」をクリックして、変更された構成をクラスターのメンバーすべてに伝搬します。ロールアウト更新では、クラスター・メンバーを含むノードの構成を順次更新します。

WebSphere ESB がメディエーション・モジュールによって提供される機能を使用してアプリケーションとサービスを統合できるようにするには、デプロイされたモジュールを開始する必要があります。モジュールは、手動で開始するか、自動的に開始するように構成できます。また例えば、モジュールの構成を変更したり、モジュールを停止または更新したり、あるいはそのアクティビティーを管理したりするなど、他の方法でモジュールを管理することもできます。

ANT タスクを使用したアプリケーションのデプロイ:

このトピックでは、ANT タスクを使用して、WebSphere Process Server に対するアプリケーションのデプロイメントを自動化する方法について説明します。ANT タスクを使用すると、複数のアプリケーションのデプロイメントを定義したり、サーバーで不在実行したりできるようになります。

このタスクで前提となるのは以下の点です。

- デプロイされるアプリケーションは開発およびテストが完了済みである。
- アプリケーションは、同一サーバーまたは複数のサーバーにインストールする予定である。
- ユーザーには、ANT タスクに関する一通りの知識がある。
- ユーザーは、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の情報は、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。

インフォメーション・センターの WebSphere Application Server Network Deployment バージョン 6 の参照部分に、アプリケーション・プログラミング・インターフェースに関するセクションがあります。ANT タスクについては、パッケージ `com.ibm.websphere.ant.tasks` に説明があります。このトピックの目的に合う関連タスクは、`ServiceDeploy` と `InstallApplication` です。

複数のアプリケーションを並行してインストールする必要がある場合は、デプロイメントの前に ANT タスクを作成します。そうすれば、プロセスにユーザーが介在しなくても、ANT タスクによってアプリケーションをサーバーにデプロイおよびインストールすることができます。

1. デプロイするアプリケーションを識別します。
2. 各アプリケーションに 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. 各サーバー用の EAR ファイルを作成するために `ServiceDeploy` コマンドを実行する ANT タスクを作成します。
5. ステップ 4 で作成した各 EAR ファイルに対して、`InstallApplication` コマンドをアプリケーション・サーバーで実行するための ANT タスクを作成します。
6. `ServiceDeploy` ANT タスクを実行してアプリケーションに EAR ファイルを作成します。
7. `InstallApplication` ANT タスクを実行して、ステップ 6 で作成した EAR ファイルをインストールします。

アプリケーションがターゲット・サーバーに正しくデプロイされます。

アプリケーションの不在デプロイの例

次に、`myBuildScript.xml` ファイルに含まれる ANT タスクの例を示します。

```
<?xml version="1.0">
<project name="OwnTaskExample" default="main" basedir=".">
  <taskdef name="servicedeploy"
    classname="com.ibm.websphere.ant.tasks.ServiceDeployTask" />
  <target name="main" depends="main2">
    <servicedeploy scaModule="c:/synctest/SyncTargetJAR"
      ignoreErrors="true"
      outputApplication="c:/synctest/SyncTargetEAREAR"
      workingDirectory="c:/synctest"
      noJ2eeDeploy="true"
      cleanStagingModules="true"/>
  </target>
</project>
```


このステートメントは、ANT タスクの呼び出し方を指定します。

```
${WAS}/bin/ws_ant -f myBuildScript.xml
```

ヒント: ファイルにプロジェクト・ステートメントを追加すると、複数のアプリケーションを不在デプロイすることができます。

管理コンソールを使用して、新しくインストールし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 ESB 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。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もできます。

サービス・アプリケーションのデプロイ後に、関連する Service Component Architecture (SCA) モジュールを参照および管理できます。メディエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

メディエーション・モジュールの使用:

WebSphere ESB にデプロイされたメディエーション・モジュールをリストできます。また、メディエーション・モジュールの開始および停止、メディエーション・モジュールまたはそのアプリケーションの詳細表示も行えます。

デプロイしたメディエーション・モジュールを表示するには、管理コンソールを使用して以下のステップを実行します。

注: メディエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

SCA モジュールをリストします。ナビゲーション・ペインで「**アプリケーション**」 → 「**SCA モジュール**」を展開します。

コンテンツ・ペインに、WebSphere ESB 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およびそのアプリケーションが実行中かどうかも表示されます。

メディエーション・モジュール詳細の表示:

WebSphere ESB にデプロイされたメディエーション・モジュールに関する情報を表示できます。

デプロイされているメディエーション・モジュールに関する詳細を表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。ナビゲーション・ペインで「**アプリケーション**」 → 「**SCA モジュール**」を展開します。
2. 参照したい特定の SCA モジュールを選択します。

コンテンツ・ペインに SCA モジュール名および記述、ならびに関連するエンタープライズ・アプリケーションの名前が表示されます。また、インポートおよびエクスポートの拡張可能リストも表示されます。

メディエーション・モジュール用アプリケーション詳細の表示:

メディエーション・モジュールを WebSphere ESB にデプロイするために使用するアプリケーションに関する詳細を表示できます。

メディエーション・モジュールのデプロイに使用するアプリケーションでは、メディエーション・モジュールと関連コンポーネントの使用に影響を及ぼす構成プロパティを定義します。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またはほとんどを指定しています。

アプリケーションのインストール後にプロパティを検討して、必要があればいくつかの値を変更します。

メディエーション・モジュールのデプロイに使用するアプリケーションの詳細を表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。ナビゲーション・ペインで「アプリケーション」→「SCA モジュール」を展開します。
2. 参照したい SCA モジュールのアプリケーション名をクリックします。

アプリケーション名は「アプリケーション」というラベルの列にリストされています。

コンテンツ・ペインに「アプリケーション詳細」ページが開き、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・トポロジーを表示します。このページからプロパティ値を変更して、アプリケーションの構成を検討し、変更するための追加コンソール・ページにリンクできます。

『アプリケーションの構成』で説明されているように、アプリケーションの構成プロパティを検討し、必要に応じて変更できます。

メディエーション・モジュールの始動と停止:

実行していない (停止状態の) メディエーション・モジュールを始動したり、実行中の (始動済み状態の) メディエーション・モジュールを停止したりできます。メディエーション・モジュールの状態を変更するには、モジュールのデプロイに使用するアプリケーションを始動または停止します。

メディエーション・モジュールに使用するアプリケーションを開始または停止するには、236 ページの『実動サーバーへのモジュールのインストール』で説明するように、事前にそのメディエーション・モジュールを WebSphere ESB にデプロイしておく必要があります。デプロイすることにより、アプリケーションがアプリケーション・サーバー (またはサーバー・クラスター) にインストールされます。

メディエーション・モジュールおよび関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、サーバーの始動時にアプリケーションが自動的に開始します。

以下の機能を使用して、アプリケーションを手動で開始または停止できます。

- 管理コンソール
- `wsadmin startApplication` コマンドおよび `stopApplication` コマンド
- ApplicationManager または AppManagement MBeans を使用する Java プログラム

メディエーション・モジュールを開始または停止するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで「アプリケーション」 → 「SCA モジュール」を展開します。
2. 開始または停止する SCA モジュールのチェック・ボックスを選択します。
3. 「開始」ボタンまたは「停止」ボタンをクリックします。

オプション	説明
開始	アプリケーションを実行して、アプリケーションの状態を「開始済み」に変更します。実行しているサーバーが、アプリケーションがデプロイされているサーバーのすべてでない場合は、状態が「一部開始済み (partially started)」に変更されます。
停止	アプリケーションの処理を停止して、アプリケーションの状態を「停止済み」に変更します。

アプリケーションを再始動するには、再始動するアプリケーションを選択して「停止」をクリックした後、「開始」をクリックします。

アプリケーションの状態が変更され、アプリケーションが開始済みまたは停止済みであることを示すメッセージがページの上部に表示されます。

アプリケーションが存在するサーバーの始動時に自動的にアプリケーションを開始するかどうかを変更できます。WebSphere アプリケーションの開始および停止の詳細については、『アプリケーションの始動と停止』を参照してください。

インポートの使用:

WebSphere ESB 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・インターフェースも表示でき、SCA インポート・バインディングを変更できます。

デプロイしたメディエーション・モジュールのインポートをリストするには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、アプリケーション → SCA モジュール を展開します。
2. 参照したい特定の SCA モジュールを選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「インポート」を展開します。

コンテンツ・ペインにインポートのリストが表示されます。インポートが存在しない場合は、情報メッセージが表示されます。

インポート・インターフェースの表示:

WebSphere ESB にデプロイされたメディエーション・モジュールのインポート・インターフェースを表示できます。

デプロイしたメディエーション・モジュールのインポート・インターフェースを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、**アプリケーション** → **SCA モジュール** を展開します。
2. SCA モジュール を選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「**インポート**」を展開します。
4. インポート詳細を表示します。 コンテンツ・ペインでモジュール・コンポーネントの下にある目的のインポートを展開します。
5. インポート・インターフェースを表示します。「**インターフェース**」を展開します。
6. インターフェースを選択します。

コンテンツ・ペインに WSDL (Web サービス記述言語) インターフェースが表示されます。

インポート・バインディングの表示:

WebSphere ESB 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・バインディングを表示できます。

メディエーション・モジュールのインポート・バインディングを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、**アプリケーション** → **SCA モジュール** を展開します。
2. SCA モジュール を選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「**インポート**」を展開します。
4. インポート詳細を表示します。 コンテンツ・ペインでモジュール・コンポーネントの下にある目的のインポートを展開します。
5. インポート・バインディングを表示します。「**バインディング**」を展開します。 バインディングが **JMS** バインディングまたは **アダプター** バインディングの場合は、そのバインディング・タイプが表示されます。バインディング・タイプは選択できません。
6. **オプション:** バインディングを選択します。 バインディングが **Web サービス**・バインディング または **SCA** インポート・バインディングの場合は、そのバインディングを選択してインポート・バインディングの詳細を取得できます。

コンテンツ・ペインにインポート・バインディングの詳細が表示されます。表示される詳細は、バインディングのタイプによって異なります。

バインディングが **Web サービス**・バインディングの場合は、そのサービス名とポート名が表示されます。バインディングが **SCA** インポート・バインディングの場合は、表示される詳細に現在のターゲット・モジュールが含まれます。現在のターゲット・エクスポートと現在のエクスポートのインターフェースも表示されます。表示されるターゲット・モジュールは **WebSphere ESB** にデプロイされた **SCA** モジュールで、ドロップダウン・メニューに表示されます。別のターゲット・モジュールを選択すると、ターゲット・エクスポートおよびエクスポート・インターフェ

ースのリストが変わります。別のターゲット・エクスポートを選択すると、エクスポート・インターフェースのリストが変わります。

インポート・バインディングの変更:

場合によって、WebSphere ESB からのメディエーション・モジュールのインポート・バインディングを変更できます。インポート・バインディングを変更できるのは、それが SCA インポート・バインディングであり、他のタイプのバインディングでない場合です。

メディエーション・モジュールの SCA インポート・バインディングを変更するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、**アプリケーション** → **SCA モジュール** を展開します。
2. SCA モジュール を選択します。
3. SCA モジュールのインポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「**インポート**」を展開します。
4. インポート詳細を表示します。 コンテンツ・ペインでモジュール・コンポーネントの下にある目的のインポートを展開します。
5. インポート・バインディングを表示します。「**バインディング**」を展開します。
6. SCA インポート・バインディングを選択します。 SCA インポート・バインディングは、ID [SCA] を使用して示されます。
7. 新規ターゲットの SCA モジュールを選択します。「**ターゲット**」ドロップダウン・メニューからモジュールを選択します。異なる SCA モジュールを選択すると、表示されるエクスポートおよびエクスポート・インターフェースが変更されます。
8. エクスポートを選択します。「**エクスポート**」ドロップダウン・メニューからエクスポートを選択します。
9. 変更を保管します。「**OK**」をクリックします。次に、変更をマスター構成に保管します。

特定の SCA モジュール・インポートに対する SCA インポート・バインディングを変更します。

WebSphere ESB では、エクスポート・インターフェースが適合しない各インポート・インターフェースに関する警告が出されます。WebSphere ESB 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は、WebSphere ESB は行われた操作を等価とみなし、警告は出されません。

エクスポートの使用:

WebSphere ESB にデプロイされたメディエーション・モジュールのエクスポートをリストできます。エクスポート・インターフェースおよびエクスポート・バインディングも表示できます。

デプロイしたメディエーション・モジュールのエクスポートをリストするには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、**アプリケーション** → **SCA モジュール** を展開します。
2. 参照したい特定の SCA モジュールを選択します。
3. SCA モジュールのエクスポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「**エクスポート**」を展開します。

コンテンツ・ペインにエクスポートのリストが表示されます。エクスポートが存在しない場合は、情報メッセージが表示されます。

エクスポート・インターフェースの表示:

WebSphere ESB にデプロイされたメディエーション・モジュールのエクスポート・インターフェースを表示できます。

デプロイされているメディエーション・モジュールのエクスポート・インターフェースを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで「**アプリケーション**」 → 「**SCA モジュール**」を展開します。
2. SCA モジュール を選択します。
3. SCA モジュールのエクスポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「**エクスポート**」を展開します。
4. エクスポートの詳細を表示します。 コンテンツ・ペインで、モジュール・コンポーネントの下にある目的のエクスポートを展開します。
5. エクスポート・インターフェースを表示します。「**インターフェース**」を展開します。
6. インターフェースを選択します。

コンテンツ・ペインに WSDL (Web サービス記述言語) インターフェースが表示されます。

エクスポート・バインディングの表示:

WebSphere ESB 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・バインディングを表示できます。

メディエーション・モジュールのエクスポート・バインディングを表示するには、管理コンソールを使用して以下のステップを実行します。

1. SCA モジュールをリストします。 ナビゲーション・ペインで、**アプリケーション** → **SCA モジュール** を展開します。
2. SCA モジュール を選択します。
3. SCA モジュールのエクスポートをリストします。 コンテンツ・ペインで、モジュール・コンポーネントの下にある「**エクスポート**」を展開します。
4. エクスポートの詳細を表示します。 コンテンツ・ペインで、モジュール・コンポーネントの下にある目的のエクスポートを展開します。

5. エクスポート・バインディングを表示します。「バインディング」を展開します。バインディングが JMS バインディングまたは アダプター バインディングの場合は、そのバインディング・タイプが表示されます。バインディング・タイプは選択できません。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、ランタイムは SCA タイプのバインディングとみなします。
6. バインディングを選択します。バインディングが Web サービス・バインディングの場合は、そのバインディングを選択してエクスポート・バインディングの詳細を取得できます。

コンテンツ・ペインにエクスポート・バインディングの詳細が表示されます。Web サービス・バインディングはサービスおよびポート名を表示します。

SCA 要求のスループット管理

WebSphere ESB にデプロイされた各 SCA モジュールでは、処理中の要求はキュー・ポイントおよびメッセージング・エンジンのデータ・ストアに保持されます。SCA 要求のデータを表示し、必要があれば SCA 要求のスループット管理のアクションを実行できます。

SCA モジュールが WebSphere ESB で実行している場合は、要求は通常そのエンタープライズ・サービス・バスを通過するため、管理の必要はありません。ときどき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必要があります。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といったその他のアクションの実行が必要な場合もあります。

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、使用する可能性のあるメインタスクの概要、および詳細が記述された WebSphere Application Server タスクへのリンクを提供します。

- 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.cell_name.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。要求がロックされているかどうかを示す、メッセージの「State」プロパティを表示できます。必要があればアクションを実行して問題を解決します。

- メッセージ・ポイントにあるメッセージの削除 (Deleting messages on a message point)

例外的な事情により、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・ポイントに存在する 1 つまたは複数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。

- データ・ストア内のメッセージング・エンジンに関するデータの表示 (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 (永続的) 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・エンジンに関するデータ・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)

要求メッセージは、メッセージ配信の信頼性を指定するサービス属性品質を備えています。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、個々の製作者により (通常、API 呼び出しによるアプリケーション制御のもとで) 信頼性を指定することができます。

データ・ストア内のデータの表示:

メッセージング・エンジンはメッセージ、トランザクション状態、通信チャネル状態など、揮発性データと永続的データの両方をデータ・ストアに保持します。データベース・ツールを使用して、メッセージング・エンジンに関するデータ・ストア内のデータを表示できます。

CloudView ツールを使用して、メッセージング・エンジンに関する Cloudscape データ・ストア内のデータを表示するには、その前にメッセージング・エンジンを停止する必要があります。

制御された方法、または制御されない方法でメッセージング・エンジンが停止すると、揮発性データは失われます。永続的データは、サーバーの再始動後に使用可能になります。

場合によっては、メッセージング・エンジンにより処理中のメッセージを検査するために、データ・ストア内のデータを表示する必要があることがあります。

データ・ストア用のデータベース・ツールを使用して、メッセージング・エンジンに関するデータ・ストア内のデータを表示できます。例えば、メッセージング・エンジンがデフォルトの Cloudscape データベースを使用している場合は、CloudView ツールを使用して要求メッセージを表示できます。

1. CloudView ツールを開始します。例えば、Windows では次のサブステップを実行します。
 - a. コマンド・ウィンドウを開きます。
 - b. ディレクトリーを `install_root/cloudscape/bin/embedded` に変更します。
 - c. `cview` と入力します。
2. メッセージング・エンジンのデータ・ストアを開きます。CloudView ツールを使用して、次のサブステップを実行します。
 - a. 「ファイル (File)」 → 「開く (Open)」をクリックします。
 - b. データベース・ファイルを参照して選択します。

メッセージング・エンジンの場合、データベースはディレクトリー `install_root/profiles/profile_name/databases/com.ibm.ws.sib` にメッセージング・エンジンの名前でご格納されます。例えば、Windows でのデフォルトのスタンドアロン・プロファイルの場合、メッセージング・エンジン `localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus` (SCA.SYSTEM バス上のサーバー 1) のデータベース・ファイルのディレクトリーは次のとおりです。

`install_root/profiles/default/databases/com.ibm.ws.sib/localhostNode01.server1-SCA.SYSTEM.localhostNode01Cell.Bus`

- c. 「開く (Open)」をクリックします。
3. CloudView 制御を使用してデータを表示します。
 - a. ナビゲーション・ペインのテーブルを拡張します。
 - b. テーブル名をクリックします。
 - c. コンテンツ・ペインで「データ」タブをクリックします。
 4. オプション: CloudView ツールの使用方法に関するヘルプを表示します。

CloudView の使用に関するヘルプを表示するには、「ヘルプ」ボタンをクリックするか、メニュー・バーのオプション 「ヘルプ」 → 「Cview ヘルプ... (Cview help...)」をクリックします。

バス宛先のメッセージ信頼性の変更:

メッセージは、メッセージ配信の信頼性を指定するサービス属性品質を備えています。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、個々の製作者により (通常、API 呼び出しによるアプリケーション制御のもとで) 信頼性を指定することができます。管理者は、製作者が宛先に設定されたデフォルトの信頼性をオーバーライド可能かどうか、および接続している製作者が要求できる最大の信頼性を指定できません。

宛先のメッセージの信頼性設定を参照または変更するには、管理コンソールを使用して以下のステップを実行します。

1. ナビゲーション・ペインで、以下をクリックします。「サービス統合」 → 「バス」
2. コンテンツ・ペインで、宛先が存在するバスの名前をクリックします。
3. 「宛先」をクリックします。

- 宛先名をクリックします。宛先の詳細ページが表示されます。
- 信頼性プロパティを調べます。次のプロパティによって、宛先のメッセージの信頼性が制御されます。

デフォルトの信頼性

製作者が信頼性を明示的に設定していないときに、この宛先に生成されるメッセージに割り当てる信頼性。

最大の信頼性

この宛先が受け入れるメッセージの最大の信頼性。

これらのプロパティは、以下にリストされた値をとることができます。

ベスト・エフォートの非永続

メッセージング・エンジンが停止または障害が発生すると、メッセージが破棄されます。また、メッセージ送信用の接続が使用不可になった場合、およびシステム・リソースが制約された場合も、メッセージが破棄されることがあります。

即時の非永続

メッセージング・エンジンが停止または障害が発生すると、メッセージが破棄されます。また、メッセージ送信用の接続が使用不可になった場合も、メッセージが破棄されることがあります。

高信頼性の非永続

メッセージング・エンジンが停止または障害が発生すると、メッセージが破棄されます。

高信頼性の永続

メッセージング・エンジンに障害が発生すると、メッセージが破棄される場合があります。

保証付きの永続

メッセージは破棄されません。

メッセージの信頼性を制御するこれらのプロパティの使用については、『メッセージ信頼性レベル』を参照してください。

- 製作者が設定されたデフォルトの信頼性をオーバーライド可能にするかどうかを検討します。

デフォルトの信頼性を製作者がオーバーライドできるようにする

このオプションを選択すると、宛先で設定されたデフォルトの信頼性を製作者がオーバーライドできるようになります。

- オプション:** 必要に応じて、宛先プロパティを変更します。

『バス宛先の構成』の説明に従って、必要に応じてその他のプロパティを設定することにより、宛先をより詳細に構成できます。

- 「OK」をクリックします。
- 変更をマスター構成に保管します

メディエーション・モジュールのその他の用途

WebSphere 管理コンソールを使用して、メディエーション・モジュールを管理する以外にも、モジュールで使用されるリソースの管理、モジュールのデプロイに使用するアプリケーションの管理、コマンドを使用したメディエーション・モジュール管理などのタスクも実行できます。

メディエーション・モジュール管理のためのその他のルーチン・タスクについては、245 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、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に関連するタスクについての WebSphere Application Server のトピックへのリンクが提供されています。

メディエーション・モジュールのその他の用途については、以下のサブトピックを参照してください。

メディエーション・モジュールのリソースの管理:

メディエーション・モジュールは、WebSphere Application Server のサービス統合テクノロジーによって提供されるリソースを利用します。またメディエーション・モジュールは、Java Message Service (JMS) および Common Event Infrastructure によって提供されるリソースを含む、広範なリソースも利用します。メディエーション・モジュールのリソースを管理するため、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の間の非同期相互作用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 ESB 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.cell_name.Bus と呼ばれるサービス統合バスのメンバーで定義されます。これらのバス宛先は、非同期相互作用を使用するメディエーション・モジュールのコンポーネントに合わせて処理されるメッセージを、保持するために使用します。

Queue sca/module_name

モジュール *module_name* に送信される非同期要求をバッファーに入れるために使用する宛先です。

Queue sca/module_name/export/export_name

モジュール・エクスポート *export_name* に送付される非同期要求をバッファーに入れるために使用する宛先です。

Queue sca/module_name/exportlink/export_name

非同期要求をモジュールに送信するエクスポートによって使用される宛先です。要求は、そのエクスポートにリンクされたコンポーネント・ターゲットに送付されます。

Queue `sca/module_name/component/component_name`

コンポーネント `component_name` に送信される非同期要求をバッファーに入れるために使用する宛先です。

Queue `sca/module_name/component/component_name/source/source_name`

コンポーネントのソース・インポート `source_name` に送付される非同期要求をバッファーに入れるために使用する宛先です。

Queue `sca/module_name/component/component_name/target/target_name`

コンポーネントのターゲット・エクスポート `target_name` に送付される非同期要求をバッファーに入れるために使用する宛先です。

Queue `sca/module_name/import/import_name`

インポート `import_name` に送信される非同期要求をバッファーに入れるために使用する宛先です。

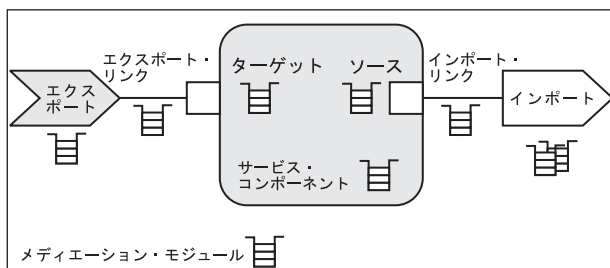
Queue `sca/module_name/importlink/import_name`

モジュールからの非同期要求を送信するインポートによって使用される宛先です。要求は、そのインポートにリンクされたモジュール・エクスポートに送付されます。

Queue `sca/module_name/import/sca/dynamic/import/scaimport` [for SCA binding]

Queue `sca/module_name/import/sca/dynamic/import/wsimport` [for Web service binding]

Queue `sca/contextStore/module_name`



宛先ごとに、キュー・ポイントも作成され、バス・メンバーのメッセージング・エンジンで定義されます。

メディエーション・モジュールは、これらのリソースを管理しなくても、デプロイおよび使用できます。ただし、リソースの構成を調整 (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) したり、トラブルシューティングでメッセージを見つける際に使用したり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の手法として、非同期メッセージングを使用できます。メディエーション・モジュールの場合は、WebSphere Application Server のデフォルトのメッセージング・プロバイダーによって JMS がサポートされます。Java Message Service の使用状態を管理するため、次のリソースを管理することができます。

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 Publish/Subscribe メッセージングの両方について、JMS 宛先の関連する JMS プロバイダーへの接続を作成する場合に使用します。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。

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。デフォルト・メッセージング・プロバイダーの JMS キューを管理するには、JMS キュー宛先の管理オブジェクトを使用します。

JMS トピック

JMS トピックは、Publish/Subscribe メッセージングの宛先として使用されます。デフォルト・メッセージング・プロバイダーの JMS トピックを管理するには、トピック宛先の管理オブジェクトを使用します。

JMS 活動化仕様

JMS 活動化仕様は、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。

Common Event Infrastructure (CEI)

CEI リソースにより、メディアエーション・モジュールでは、イベント・データ管理用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、選択子式で判別されるイベントのリストを定義します。JMS キューおよび JMS トピックを各イベント・グループに関連付けることができます。イベント・サーバー配布サービスが使用可能で、イベントがイベント・グループに一致する場合、イベントはそのイベント・グループ用に構成されたトピックまたはキューに配布されます。

イベント・サーバー・プロファイル

このプロファイルは、イベント・サーバーのプロパティを定義します。

フィルター・ファクトリー・プロファイル

このプロファイルは、フィルターのプロパティを定義します。フィルターはフィルター構成ストリングを使用して、イベントがバスに渡されるかどうかを判別します。

JMS 伝送プロファイル

イベント・テーブルを含むデータベース・スキーマ。

非同期メッセージングの使用:

これらのトピックでは、JMS リソースおよびメッセージ・ドリブン Bean をエンタープライズ・アプリケーションで使用可能にするため、WebSphere ESB で非同期メッセージングを使用する方法を説明します。

WebSphere ESB 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の手法として、非同期メッセージングをサポートしています。この JMS サポートは、エンタープライズ・アプリケーションによって使用するために構成した、1 つ以上の JMS プロバイダーおよび関連したサービスとリソースによって提供されます。JMS 1.1 インターフェースを使用する EJB 2.1 アプリケーションと、JMS 1.0.2 インターフェースを使用する EJB 2.0 アプリケーションをデプロイできます。このサポートは、WebSphere ESB を構築するときの基盤である WebSphere Application Server によって提供されます。

WebSphere 管理コンソールを使用して、非同期メッセージングのための WebSphere ESB サポートを管理できます。例えば、メッセージング・プロバイダーおよびそのリソースを構成して、メッセージング・サービスのアクティビティを制御できます。

非同期メッセージングを使用する WebSphere エンタープライズ・アプリケーションをインプリメントすることについて詳しくは、以下のサブトピックを参照してください。

非同期メッセージングを使用する WebSphere エンタープライズ・アプリケーションをインプリメントすることについて詳しくは、インフォメーション・センターの以下のサブトピックを参照してください。または、

<http://www-306.ibm.com/software/webservers/appserv/was/library/> の WebSphere Application Server Network Deployment ライブラリー Web ページから入手できる「アプリケーションとその環境の管理」およびその他の PDF 資料を参照することもできます。

- WebSphere Application Server によるメッセージングの学習

このトピックを使用して、WebSphere ESB を用いたエンタープライズ・アプリケーションの非同期メッセージングの使用について学習します。

- JMS プロバイダーのインストールおよび構成

このトピックでは、WebSphere ESB で JMS プロバイダーを使用できるさまざまな方法について説明します。JMS プロバイダーにより、Java Message Service (JMS) および WebSphere ESB で Java Message Service (JMS) および他のメッセージ・リソースを使用することができます。

- デフォルトのメッセージング・プロバイダーの使用

このトピックは、デフォルトのメッセージング・プロバイダーによって提供されるメッセージング・リソースを WebSphere アプリケーションが使用できるようにするための方法に関する一連のトピックへの入り口です。デフォルトのメッセー

ジング・プロバイダーは、WebSphere Application Server の一部としてインストールされ実行されており、サービス統合テクノロジーを基礎としています。

- バージョン 5 デフォルト・メッセージング・リソースのメンテナンス

このトピックは、デフォルト・メッセージング・プロバイダーによって、WebSphere Application Server バージョン 5 アプリケーション用に提供されるメッセージング・リソースの保守に関する一連のトピックへの入り口です。

- WebSphere MQ の JMS リソースの使用

このトピックは、WebSphere MQ によって提供される JMS リソースを WebSphere アプリケーションが使用できるようにするための方法に関する一連のトピックへの入り口です。

- 一般プロバイダーの JMS リソースの使用

このトピックは、WebSphere のデフォルトのメッセージング・プロバイダーや WebSphere MQ 以外の一般メッセージング・プロバイダーによって提供される JMS リソースを WebSphere アプリケーションが使用できるようにするための方法に関する一連のトピックへの入り口です。

- メッセージ・ドリブン Bean の管理サポート

これらのタスクを使用して、メッセージ・ドリブン Bean のサポートに使用するリソースを管理します。これらのタスクは、リソース・アダプター、JMS プロバイダー、およびそれらが提供するリソースを管理するタスクへの追加です。

- WebSphere メッセージングのトラブルシューティング

この概説タスクを使用して、WebSphere Messaging に関連すると思われる問題を解決してください。WebSphere Messaging に関連すると思われる問題を識別して解決するには、標準の WebSphere ESB トラブルシューティング機能を使用できます。

コマンドを使用したサービス・アプリケーション管理:

コマンドを使用してサービス・アプリケーションを管理できます。コマンドはスクリプト内で使用できます。

サービス・アプリケーションのコマンドを実行するには、wsadmin ツールを使用する必要があります。

wsadmin ツールはいくつかの方法で使用できます。このツールを個々のコマンドとして、またはスクリプト内で対話式で使用できます。複数のマシンを管理する場合には、スクリプト内で複数のコマンドを実行すると便利です。

WebSphere ESB には、SCA モジュールおよびそのインポートとエクスポートを表示できるコマンドがあります。また、インポート・バインディングが SCA バインディングであれば、そのバインディングを変更できますが、それ以外のタイプのバインディングは変更できません。

1. SCA 管理コマンドをリストします。 `$AdminTask help SCAAdminCommands`
2. 特定のコマンドの詳細ヘルプを表示します。 `$AdminTask help command_name`

`$AdminTask help listSCAModules`

メディエーション・モジュールの管理:

コマンド行により WebSphere ESB 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。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もできます。

wsadmin ツールを使用して、WebSphere ESB コマンドを実行する必要があります。

コマンドは個別に、またはスクリプトで実行できます。複数のマシンを管理する場合や、定期的にレポートを作成する場合には、スクリプトで複数のコマンドを実行すると便利です。

メディエーション・モジュールのリスト表示:

コマンドを使用して、WebSphere ESB 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。

wsadmin ツールを使用して、次のコマンドを実行する必要があります。

デプロイしたメディエーション・モジュールをリストするには、wsadmin ツールを使用して以下のステップを実行します。

メディエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。したがって、デプロイしたメディエーション・モジュールをリストするには、SCA モジュールをリストします。

デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`

WebSphere ESB 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。これにより、後続のコマンドで使用するための、スクリプトによる出力の構文解析と名前の抽出が容易になります。

メディエーション・モジュール詳細の表示:

コマンドを使用して、メディエーション・モジュールの属性を表示できます。

wsadmin ツールを使用して、次のコマンドを実行する必要があります。

メディエーション・モジュールの記述を表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュールの記述を表示するには、メディエーション・モジュール名がわかっている必要があります。

注: メディエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

`listSCAModules` コマンドにより、WebSphere ESB にデプロイされている SCA モジュールをリストします。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`

2. 特定の SCA モジュールの詳細を表示します。 `$AdminTask showSCAModule {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュールの名前および記述を表示します。

```
$AdminTask showSCAModule {-moduleName myModule -applicationName myApplication}
```

インポートのリスト表示:

コマンドを使用して、WebSphere ESB にデプロイされている任意のメディエーション・モジュールのインポートをリストできます。

wsadmin ツールを使用して、次のコマンドを実行する必要があります。

メディエーション・モジュールのインポートをリストするには、wsadmin ツールを使用して以下のステップを実行します。

メディエーション・モジュールのインポートをリストするには、メディエーション・モジュール名がわかっている必要があります。

注: メディエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

listSCAModules コマンドにより、WebSphere ESB にデプロイされているすべての SCA モジュールをリストし、listSCAImports コマンドにより、特定の SCA モジュールのインポートをリストします。SCA モジュールにインポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのインポートをリストします。 `$AdminTask listSCAImports {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュールのインポートをリストします。

```
$AdminTask listSCAImports {-moduleName myModule -applicationName myApplication}
```

インポート詳細の表示:

コマンドを使用して、WebSphere ESB 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の詳細を表示できます。

wsadmin ツールを使用して、次のコマンドを実行する必要があります。

メディエーション・モジュールのインポートの詳細を表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュール・インポートの詳細を表示するには、メディエーション・モジュール名およびインポート名がわかっている必要があります。

注: メディエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

`listSCAModules` コマンドにより、WebSphere ESB にデプロイされているすべての SCA モジュールをリストし、`listSCAImports` コマンドにより、特定の SCA モジュールのインポートをリストします。SCA モジュールにインポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのインポートをリストします。 `$AdminTask listSCAImports {-moduleName moduleName}`

注: `moduleName` の指定に加えて、オプションで `applicationName` も指定できます。`applicationName` を指定することにより、パフォーマンスが向上します。

3. 特定の SCA モジュール・インポートの詳細を表示します。 `$AdminTask showSCAImport {-moduleName moduleName -import importName}`

注: `moduleName` および `importName` の指定に加えて、オプションで `applicationName` も指定できます。`applicationName` を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・インポートのインポート詳細を表示します。

```
$AdminTask showSCAImport {-moduleName myModule -applicationName myApplication -import myImport}
```

インポート・バインディングの表示:

コマンドを使用して、WebSphere ESB 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。

`wsadmin` ツールを使用して、次のコマンドを実行する必要があります。

特定のメディエーション・モジュールのインポート・バインディングを表示するには、`wsadmin` ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュール・インポートのインポート・バインディングを表示するには、メディエーション・モジュール名およびインポート名がわかっている必要があります。

注: メディエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

`listSCAModules` コマンドにより、WebSphere ESB にデプロイされているすべての SCA モジュールをリストし、`listSCAImports` コマンドにより、特定の SCA モジュールのインポートをリストします。SCA モジュールにインポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのインポートをリストします。 `$AdminTask listSCAImports {-moduleName moduleName}`

注: `moduleName` の指定に加えて、オプションで `applicationName` も指定できます。`applicationName` を指定することにより、パフォーマンスが向上します。

3. 特定のインポートのインポート・バインディングを表示します。 `$AdminTask showSCAImportBinding {-moduleName moduleName -import importName}`

注: `moduleName` および `importName` の指定に加えて、オプションで `applicationName` も指定できます。`applicationName` を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・インポートのインポート・バインディングを表示します。

```
$AdminTask showSCAImportBinding {-moduleName myModule -applicationName myApplication -import myImport}
```

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 ESB 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。

`wsadmin` ツールを使用して、次のコマンドを実行する必要があります。

特定のメディエーション・モジュールによって別のメディエーション・モジュールが起動されるようにしたい場合、SCA インポート・バインディングを変更します。インポート・バインディングを変更する場合は、インポートとエクスポートが一致するよう、つまり、行う操作が等価となる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。

注: Service Component Architecture (SCA) モジュール インポート・バインディング を変更できるのは、それが SCA バインディングである場合だけで、その他のタイプのバインディングの場合は変更できません。SCA バインディングによって、ある SCA モジュールが別の SCA モジュールに接続されます。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。

特定のメディエーション・モジュール・インポートの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分かっている必要があります。

`listSCAModules` コマンドにより、WebSphere ESB にデプロイされているすべての SCA モジュールがリストされます。`listSCAImports` コマンドにより、特定の SCA モジュールのすべてのインポートが、`listSCAExports` コマンドにより、特定の SCA モジュールのすべてのエクスポートがリストされます。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`

2. 特定の SCA モジュールのインポートをリストします。 `$AdminTask listSCAImports {-moduleName moduleName}`

注: `moduleName` の指定に加えて、オプションで `applicationName` も指定できます。`applicationName` を指定することにより、パフォーマンスが向上します。

3. SCA タイプの SCA モジュール・インポート・バインディングを変更します。
`$AdminTask modifySCAImportSCABinding -moduleName moduleName -import importName -targetModule targetModuleName -targetExport targetExportName`

注: オプションで、`applicationName` および `targetApplicationName` も指定できます。`applicationName` および `targetApplicationName` を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・インポートに対する SCA インポート・バインディングを変更します。

WebSphere ESB では、エクスポート・インターフェースが適合しない各インポート・インターフェースに関する警告が出されます。WebSphere ESB 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は、WebSphere ESB 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}
```

エクスポートのリスト表示:

コマンドを使用して、WebSphere ESB にデプロイされている任意のメディアエーション・モジュールのエクスポートをリストできます。

`wsadmin` ツールを使用して、次のコマンドを実行する必要があります。

メディアエーション・モジュールのエクスポートをリストするには、`wsadmin` ツールを使用して以下のステップを実行します。

特定のメディアエーション・モジュールのエクスポートをリストするには、メディアエーション・モジュール名がわかっている必要があります。

注: メディアエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

`listSCAModules` コマンドにより、WebSphere ESB にデプロイされているすべての SCA モジュールをリストし、`listSCAExports` コマンドにより、特定の SCA モジュールのエクスポートをリストします。SCA モジュールにエクスポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのエクスポートをリストします。 `$AdminTask listSCAExports {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュールのエクスポートをリストします。

```
$AdminTask listSCAExports {-moduleName myModule -applicationName myApplication}
```

エクスポート詳細の表示:

コマンドを使用して、WebSphere ESB 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。

wsadmin ツールを使用して、次のコマンドを実行する必要があります。

メディエーション・モジュールのエクスポートの詳細を表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディエーション・モジュールのエクスポート詳細を表示するには、メディエーション・モジュール名およびエクスポート名が分かっている必要があります。

注: メディエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

listSCAModules コマンドにより、WebSphere ESB にデプロイされているすべての SCA モジュールをリストし、listSCAExports コマンドにより、特定の SCA モジュールのエクスポートをリストします。SCA モジュールにエクスポートが存在しない場合もあります。

1. デプロイされている SCA モジュールをリストします。 `$AdminTask listSCAModules`
2. 特定の SCA モジュールのエクスポートをリストします。 `$AdminTask listSCAExports {-moduleName moduleName}`

注: *moduleName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

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`

注: *moduleName* および *exportName* の指定に加えて、オプションで *applicationName* も指定できます。*applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・エクスポートのエクスポート詳細を表示します。

```
$AdminTask showSCAExport {-moduleName myModule -applicationName myApplication -export myExport}
```

エクスポート・バインディングの表示:

コマンドを使用して、WebSphere ESB 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。

wsadmin ツールを使用して、次のコマンドを実行する必要があります。

特定のメディアエーション・モジュールのエクスポート・バインディングを表示するには、wsadmin ツールを使用して以下のステップを実行します。

特定のメディアエーション・モジュール・エクスポートのエクスポート・バインディングを表示するには、そのメディアエーション・モジュール名およびエクスポート名が分かっている必要があります。

注: メディアエーション・モジュールは Service Component Architecture (SCA) モジュールのタイプです。

listSCAModules コマンドにより、WebSphere ESB 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。

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. 特定のエクスポートのエクスポート・バインディングを表示します。
`$AdminTask showSCAExportBinding {-moduleName moduleName -export exportName}`

注: *moduleName* および *export* の指定に加えて、オプションで *applicationName* も指定できます。 *applicationName* を指定することにより、パフォーマンスが向上します。

特定の SCA モジュール・エクスポートのエクスポート・バインディングを表示します。表示される情報は、バインディングのタイプによって異なります。指定されたバインディングがエクスポートに存在しない場合は、ランタイムは SCA タイプのバインディングとみなします。

```
$AdminTask showSCAExportBinding {-moduleName myModule -applicationName myApplication -export myExport}
```

エンタープライズ・アプリケーションの管理:

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

アプリケーションの構成で指定した値を表示するには、リストからアプリケーション名をクリックします。「アプリケーション詳細」ページが開き、アプリケーションの構成プロパティ、および該当する場合はローカル・トポロジーを表示します。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、次に以下のボタンのいずれかを使用します。

表 11. 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始動すると、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"> • 開始済み: アプリケーションは、すべてのデプロイメント・ターゲットで始動しました。 • 部分開始: アプリケーションは、1 つ以上のデプロイメント・ターゲットでまだ開始中です。
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"> • 停止済み: アプリケーションは、すべてのデプロイメント・ターゲットで停止しました。 • 部分停止: アプリケーションは、1 つ以上のデプロイメント・ターゲットでまだ停止中です。
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) をサーバーに配置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、構成が保管された後にアプリケーション・モジュールがインストールされる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、サーバー上に配置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、その新規ファイルまたはモジュールが既存のファイルまたはモジュールと置き換えられます。同じ名前のファイルまたはモジュールがない場合は、配置されたアプリケーションに追加されます。</p>
ファイルの除去	<p>配置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。配置されたアプリケーションをバックアップしたり、そのバインディング情報を保存したりする場合にも、「エクスポート」アクションを使用します。</p>

表 11. 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 for z/OS インフォメーション・センターを参照してください。

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、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提供されています。

Relationship Manager によって、参加プログラムおよびそれらのデータを含むリレーションシップ・ランタイム・データにおける操作の構成、照会、表示、および実行を行うことができます。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は、別のアプリケーションからの情報を関連付けるデータを取り込みます。このリレーションシップ・インスタンス・データは、マップまたはその他 WebSphere Process Server コンポーネントの実行時、およびそれらにリレーションシップ・インスタンスが必要になった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 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・インターフェースを提供します。

リレーションシップ・インスタンスごとに、Relationship Manager は、その参加プログラムの階層リストを表示できます。それぞれの参加プログラムは、リレーションシップにロールを記入し、インスタンス・データ、プロパティ、およびキー属性を持っています。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、最終変更日時などの、リレーションシップ・インスタンス内の各参加プログラム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。

ユーザーは、Relationship Manager を使用して、リレーションシップ・インスタンス、参加プログラム・インスタンス、および属性データのすべてのレベル、およびプロパティ・データ・レベルにおけるエンティティを管理できます。例えば、Relationship Manager を使用して以下を行うことができます。

- リレーションシップ・インスタンスを作成、および削除します。

- 参加プログラムの追加および削除など、リレーションシップ・インスタンスの内容を変更します。
- 参加プログラムのデータを追加して、別のリレーションシップの参加プログラムのデータをリレーションシップ・インスタンスにコピー・アンド・ペーストし、新規参加プログラムを作成します (参加プログラムのタイプが同一の場合)。
- 参加プログラムの活動化および非活動化を行います。
- インスタンス ID またはデータに基づいて、参加プログラムを検索します。
- 問題発生時に状態を修復します。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の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の詳細な情報は、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・サービスおよび管理コンソールを参照してください。

リレーションシップ・タイプの表示

リレーションシップ名、表示名、静的属性または識別の属性、およびロールなどの、リレーションシップ・タイプに関連した情報を表示するには、このタスクを実行します。

必要なセキュリティー・ロール: このタスクを実行するには、モニター、オペレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。

リレーションシップ詳細の表示

リレーションシップ名、表示名、プロパティ値、ロール・タイプ属性、および静的属性と識別の属性などの、リレーションシップ・タイプに関する詳細情報を表示するには、このタスクを実行します。

必要なセキュリティー・ロール: このタスクを実行するには、モニター、オペレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 次の 2 つの方法で、リレーションシップの詳細を表示することができます。
 - a. リレーションシップ名をクリックします。

- b. 「**選択**」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「**詳細**」をクリックします。
5. リレーションシップ詳細のリストに戻るには、ページの上部のパスから「**リレーションシップ**」をクリックします。

リレーションシップ・インスタンスの照会

リレーションシップ・ベースのインスタンス照会を実行するには、このタスクを使用します。

必要なセキュリティ・ロール: このタスクを実行するには、モニター、オペレーター、または管理者としてログインしている必要があります。

照会パラメーター (すべて、ID ごと、プロパティごと) を選択して、リレーションシップ用のすべてまたはサブセットのインスタンス・データを検索します。この戻りはその照会の結果セットであり、リレーションシップ・インスタンス・ページの「**設定**」フィールドでの設定に基づいて表示されます。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「**統合アプリケーション**」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「**リレーションシップ**」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「**選択**」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「**照会**」をクリックします。
5. 適切なタブを選択して照会オプションの 1 つを選択します。

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検索します。すべての活動化された、すべての非活動化された、またはすべての活動化されたおよび非活動化されたリレーションシップ・インスタンス・データを表示することを選択することができます。
「 ID ごと 」タブ	開始および終了インスタンス ID の範囲のリレーションシップ・インスタンスを検索します。1 つのフィールドが空白になっている場合は、単一インスタンスのみが戻ります。照会は見つけるインスタンス用のロールをすべて戻します。
「 プロパティごと 」タブ	特定のプロパティ値でリレーションシップ・インスタンスを検索します。

6. 照会パラメーターを選択したら、以下のオプションを行えます。
 - 「**適用**」をクリックして、照会から結果データを表示します。
 - 「**OK**」をクリックして、照会から結果データを表示します。
 - 「**リセット**」をクリックして選択をクリアするか、エントリーを最も最近に行った変更のセットに戻します。

- 「キャンセル」をクリックして行ったすべての変更を破棄し、リレーションシップ・タイプのリストに戻ります。

リレーションシップ・インスタンス詳細の表示

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、ロール・ベースの情報、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。

必要なセキュリティ・ロール: このタスクを実行するには、モニター、オペレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「照会」をクリックします。
5. 3 つの照会オプションから 1 つを選択します。
 - 「すべて」タブを選択し、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検索します。すべての活動化された、すべての非活動化された、またはすべての活動化されたおよび非活動化されたリレーションシップ・インスタンス・データを表示することを選択することができます。
 - 「ID ごと」タブを選択して、開始および終了インスタンス ID の範囲のリレーションシップ・インスタンスを検索します。1 つのフィールドが空白になっている場合は、単一インスタンスのみが戻ります。照会は見つけるインスタンス用のロールをすべて戻します。
 - 「プロパティごと」タブを選択して、特定のプロパティ値でリレーションシップ・インスタンスを検索します。
6. 照会オプションを選択し必要な情報を入力したら、「適用」または「OK」のいずれかをクリックします。
7. 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することができます。
 - リレーションシップ・インスタンス ID をクリックします。
 - 「選択」列で、リレーションシップ・インスタンス ID の横にあるラジオ・ボタンをクリックして、「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情報を編集するには、このタスクを実行します。

必要なセキュリティ・ロール: このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

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

3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「照会」をクリックします。
5. 3 つの照会オプションの 1 つを選択し、照会情報を記入し、「適用」または「OK」のいずれかをクリックします。
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することができます。
 - リレーションシップ・インスタンス ID をクリックします。
 - 「選択」列で、リレーションシップ・インスタンス ID の横にあるラジオ・ボタンをクリックして、「詳細」をクリックします。
7. 必要に応じて、プロパティ値を変更します。
8. 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。
 - 「適用」をクリックして変更をローカルに保管します。
 - 「OK」をクリックして変更をローカルに保管します。
 - 「リセット」をクリックして変更をクリアするか、記入項目を最も最近に行った変更のセットに戻します。
 - 「キャンセル」をクリックして行ったすべての変更を破棄し、リレーションシップ・インスタンスのリストに戻ります。
9. 「変更を適用」をクリックしてこの時点までに行った変更をデータベースに保管します。

リレーションシップ・インスタンスの新規作成

新しいリレーションシップ・インスタンスを作成するには、このタスクを実行します。

必要なセキュリティ・ロール: このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「照会」をクリックします。
5. 3 つの照会オプションから 1 つを選択します。
6. 照会オプションを選択し必要な情報を入力したら、「適用」または「OK」のいずれかをクリックします。
7. 「作成」をクリックします。
8. デフォルト値以外の値が必要な場合は、プロパティ値用の値情報を追加し、「適用」または「OK」のいずれかをクリックして、新しいリレーションシップ・インスタンスをローカルに保管します。

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。

9. リレーションシップ・インスタンスおよびロール・インスタンスの両方を作成したら、「**変更を適用**」をクリックしてこの時点までに行った変更をデータベースに保管します。

リレーションシップ・インスタンスの削除

選択したリレーションシップ・インスタンスを削除するには、このタスクを実行します。

必要なセキュリティ・ロール: このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「**統合アプリケーション**」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「**リレーションシップ**」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「**選択**」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「**照会**」をクリックします。
5. 3 つの照会オプションのうちの 1 つを選択し、必要な情報を入力して、「**適用**」または「**OK**」のいずれかをクリックします。
6. 削除するリレーションシップ・インスタンス ID の横にあるラジオ・ボタンをクリックします。
7. 「**削除**」をクリックしてリレーションシップ・インスタンスをローカルに削除します。
8. 「**変更を適用**」をクリックしてこの時点までに行った変更をデータベースに保管します。

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

リレーションシップのインスタンス・データを指定した日時にロールバックするには、このタスクを実行します。

必要なセキュリティ・ロール: このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「**統合アプリケーション**」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「**リレーションシップ**」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「**選択**」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「**ロールバック**」をクリックします。
5. 「**開始日**」および「**終了日**」フィールドに記入してロールバックの時間を選択します。
6. 「**適用**」または「**OK**」をクリックします。

リレーションシップに関連したロール・タイプの表示

ロール名、表示名、キー (ビジネス・オブジェクト用)、ロール・オブジェクト・タイプ (ビジネス・オブジェクト名)、および管理対象の属性設定などの、リレーションシップのロールの説明を表示するには、このタスクを実行します。

必要なセキュリティ・ロール: このタスクを実行するには、モニター、オペレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. リレーションシップの「ロール」列でロール名をクリックします。

注: このコレクション・ページの表示設定を設定するには、「設定」をクリックします。フィールド値が必要な値に変更し、「適用」をクリックします。

ロールを基にしたリレーションシップ・インスタンスの照会

さまざまなロール・ベースのインスタンス照会を実行するには、このタスクを使用します。

必要なセキュリティ・ロール: このタスクを実行するには、モニター、オペレーター、または管理者としてログインしている必要があります。

照会パラメーター (キーごと、日付ごと、プロパティごと、または 拡張) を選択して、選択したロールに基づいたリレーションシップ用のすべてまたはサブセットのインスタンス・データを検索します。この戻りはその照会の結果セットであり、リレーションシップ・インスタンス・ページの「設定」フィールドでの設定に基づいて表示されます。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. リレーションシップの「ロール」列でロール名をクリックします。
5. 「選択」列で、ロール名の横にあるラジオ・ボタンをクリックして、「照会」をクリックします。
6. 適切なタブを選択して照会オプションの 1 つを選択します。

オプション	説明
「キーごと」タブ	選択された参加者のキー属性用の値に基づいてリレーションシップ・インスタンスを検索します。 この情報を検索するには、ID の値を記入し、「適用」または「OK」をクリックします。

オプション	説明
「日付ごと」タブ	<p>指定した日付間に作成または修正されたロールを持つリレーションシップ・インスタンスを検索します。</p> <p>この情報を検索するには、希望する日付を「開始日」および「終了日」フィールドに記入し、「OK」または「適用」をクリックします。</p>
「プロパティごと」タブ	<p>特定のロール・プロパティ値でリレーションシップ・インスタンスを検索します。</p> <p>この情報を検索するには、以下のステップを実行してください。</p> <ol style="list-style-type: none"> 1. ロール・プロパティ名およびタイプ・パラメーターをドロップダウン・リストから選択します。 2. 指定したロール・プロパティ用の値パラメーターを「プロパティ値」フィールドに入力し、「OK」または「適用」をクリックします。
「拡張」タブ	<p>他の照会パネルのフィールドを結合して、さらに洗練された検索を要求します。</p> <p>他のオプションで説明したように情報を記入し、「OK」または「適用」をクリックします。</p>

注: また、各タブには以下のオプションもあります。

- 「リセット」をクリックして選択をクリアするか、エントリーを最も最近に行った変更のセットに戻します。
- 「キャンセル」をクリックして行ったすべての変更を破棄し、ロールのリストに戻ります。

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、および管理対象の属性設定などの、ロールに関する詳細情報を表示するには、このタスクを実行します。

必要なセキュリティー・ロール: このタスクを実行するには、モニター、オペレーター、または管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「**Relationship Manager**」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. リレーションシップの「ロール」列でロール名をクリックします。

- 次の 2 つの方法で、ロールの詳細を表示することができます。
 - ロール名をクリックします。
 - 「選択」列で、ロール名の横にあるラジオ・ボタンをクリックして、「詳細」をクリックします。

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、プロパティ値、状況、および論理状態などの、選択されたロール・インスタンスに関する詳細情報を表示するには、このタスクを実行します。

必要なセキュリティー・ロール: このタスクを実行するには、モニター、オペレーター、または管理者としてログインしている必要があります。

- 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。
- 「選択」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「照会」をクリックします。
- 照会オプションのうちの 1 つを選択して、「適用」または「OK」をクリックします。
- リレーションシップの「ロール・インスタンス」列でロール名をクリックします。
- 次の 2 つの方法で、ロール・インスタンスを表示することができます。
 - ロール名をクリックします。
 - 「選択」列で、ロール名の横にあるラジオ・ボタンをクリックし、次に「インスタンス」をクリックします。
- 次の 2 つの方法で、インスタンスの詳細を表示することができます。
 - ロール・インスタンス名をクリックします。
 - 「選択」列で、ロール・インスタンスの横にあるラジオ・ボタンをクリックして、「詳細」をクリックします。

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ロール・インスタンスを新規に作成するには、このタスクを実行します。

必要なセキュリティー・ロール: このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

注: 識別リレーションシップ・インスタンスに追加のロール・インスタンスを追加することはできません。

- 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。

3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「照会」をクリックします。
5. 照会オプションのうちの 1 つを選択し、必要な情報を入力して、「適用」または「OK」をクリックします。
6. リレーションシップの「ロール・インスタンス」列でロール名をクリックします。
7. ロール名の横にあるラジオ・ボタンをクリックして、「作成」をクリックします。
8. ID 用の値およびプロパティ値用の値情報を追加し、「適用」または「OK」のいずれかをクリックして、新規ロール・インスタンスをローカルに保管します。

ロール・インスタンスの作成時には、キー値情報のみを設定することができます。変更内容をデータベースに再度適用した後は、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。

注: また、以下のオプションもあります。

- 「リセット」をクリックして選択をクリアするか、エントリーを最も最近に行った変更のセットに戻します。
 - 「キャンセル」をクリックして行ったすべての変更を破棄し、ロール・インスタンスのリストに戻ります。
9. 「変更を適用」をクリックしてこの時点までに行った変更をデータベースに保管します。

ロール・インスタンスの削除

リレーションシップの選択したロール・インスタンスを削除するには、このタスクを実行します。

必要なセキュリティー・ロール: このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」>「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「照会」をクリックします。
5. 照会オプションのうちの 1 つを選択し、必要な情報を入力して、「適用」または「OK」をクリックします。
6. リレーションシップの「ロール・インスタンス」列でロール名をクリックします。
7. 「選択」列で、ロール名の横にあるラジオ・ボタンをクリックし、次に「インスタンス」をクリックします。

8. 「選択」列で、ロール・インスタンスの横にあるラジオ・ボタンをクリックし、「削除」をクリックして、このロール・インスタンスをローカルに削除します。
9. 「変更を適用」をクリックしてこの時点までに行った変更をデータベースに保管します。

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情報を編集するには、このタスクを実行します。

必要なセキュリティ・ロール: このタスクを実行するには、オペレーターまたは管理者としてログインしている必要があります。

1. 管理コンソールが稼働していることを確認します。
2. ナビゲーション・ペインで、「統合アプリケーション」 > 「Relationship Manager」をクリックします。
3. そのリレーションシップ・サービス MBean の横にある「リレーションシップ」をクリックして、管理するサーバーのリレーションシップ・ページを開きます。
4. 「選択」列で、リレーションシップ名の横にあるラジオ・ボタンをクリックして、「照会」をクリックします。
5. 照会オプションのうちの 1 つを選択し、必要な情報を入力して、「適用」または「OK」をクリックします。
6. リレーションシップの「ロール・インスタンス」列でロール名をクリックします。
7. 「選択」列で、ロール名の横にあるラジオ・ボタンをクリックし、次に「インスタンス」をクリックします。
8. 「選択」列で、ロール・インスタンスの横にあるラジオ・ボタンをクリックして、「詳細」をクリックします。
9. 必要に応じて情報を編集し、「適用」または「OK」をクリックして、この変更をローカルに保管します。

注: また、以下のオプションもあります。

- 「リセット」をクリックして変更をクリアするか、記入項目を最も最近に行った変更のセットに戻します。
 - 「キャンセル」をクリックして行ったすべての変更を破棄し、ロール・インスタンスのリストに戻ります。
10. 「変更を適用」をクリックしてこの時点までに行った変更をデータベースに保管します。

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、データの同じ項目用の異なる環境から ID を相関するために使用されます。例えば、1 つの環境では、状態は 2 文字略語 (AZ、TX) で識別されます。別の環境では、異なる略語が使用されます (Ariz、Tex.)。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」に相関させるように作成されます。

ここで参照されるサンプル・リレーションシップは、カスタマー ID を相関させます。多くのビジネス・アプリケーションは、カスタマーのデータベースを維持し、このアプリケーションのほとんどはその独自の ID を各カスタマーに割り当てます。企業環境では、同じカスタマーが各ビジネス・アプリケーションで異なる ID を持つ可能性があります。このチュートリアルでは、リレーションシップはカスタマー ID を相関させるために定義されます。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、他のロールは 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、ロールと少ないサンプル・データと共にリレーションシップ・サービス・サンプルによって作成されました。

Relationship Manager は、リレーションシップ・インスタンスの追加、変更、除去と同様に、リレーションシップ・インスタンスのロール・インスタンスを追加、変更、除去するために設計されています。WebSphere Integration Developer は、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ために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した後で、ユーザーはリレーションシップ・インスタンスの値を変更することができます。

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は、完了するのに約 10 分かかります。

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・サンプルによっ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の手順に従う前に、サンプル・ギャラリーに進み、リレーションシップ・サービス・サンプルに説明されている手順を実行し、必要なリレーションシップおよびロールを作成します。

リレーションシップ・インスタンスの値の変更

カスタマーの 1 人が A004 というカスタマー ID を CIS アプリケーションに持っています。同じカスタマーが 801 というカスタマー ID を GL アプリケーションに持っています。ただし、データ入力エラーのため、このカスタマーのカスタマー ID を相関するリレーションシップ・インスタンスは、現在、GL カスタマー ID に対して 801 の代わりに 901 という値を持っています。このチュートリアルでは、リレーションシップのこの入力を訂正する手順を順を追って説明します。

1. WebSphere Process Server の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、管理者権限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. 「ロール」列を探し、SampleCustID リレーションシップ用のこの列内のリンクをクリックします。

MyCISCustomer_0、MyGLCustomer_0、および GenericCustomer_0 という 3 つのロールが使用可能になっている必要があります。

6. MyGLCustomer_0 ロール上の特定の値を検索し、変更する必要があるリレーションシップを探します。 MyGLCustomer_0 の横にあるラジオ・ボタンを選択して、「照会」をクリックします。
7. 「キーごと」タブの下で、「キー」属性の下の「値」フィールドに値 901 を入力し、「適用」をクリックします。

これによって、必要なカスタマーのリレーションシップ・インスタンスが見つかります。

8. リレーションシップ・インスタンス ID をクリックします。

このページには、GL アプリケーションでのカスタマー ID 901 用の実際のリレーションシップ・データが表示されます。特定のロールに対して照会が実行されたので、そのロール用のデータのみが戻されました。

9. 「名前」列の MyGLCustomer_0 リンクをクリックします。

これによって、このリレーションシップ・インスタンス用の照会に一致するロールがすべて表示されます。

注: このロールは、customerNumber=901 を持ち、プロパティを持っていない必要があります。他のデータが表示されたら、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。

10. 「選択」列で、ロール・インスタンスの横にあるラジオ・ボタンをクリックして、「削除」をクリックします。
11. 「作成」をクリックして、このリレーションシップ・インスタンス用の新規ロール・インスタンスを作成します。
12. customerNumber 用の値として 801 を入力し、「OK」をクリックします。

テーブルに新規の未保管インスタンスが表示されるはずですが。

13. 「変更を適用」をクリックして、リレーションシップ・データベース表にすべての更新を保管し、変更済みインスタンスを戻します。

ユーザーは、GL アプリケーション用のリレーションシップ・インスタンスに正しいカスタマー ID 値を持つようになりました。

WebSphere Process Server の失敗したイベントの管理

管理者は管理コンソールで使用可能な Failed Event Manager を使用して、WebSphere Process Server の失敗したイベントを管理できます。

失敗したイベントとは

WebSphere Process Server のコンテキストでは、イベントとは WebSphere Process Server アプリケーションによって受け取った要求のことです。これは外部ソース (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に由来します。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。イベントを受信すると、しかるべき WebSphere Process Server アプリケーションのビジネス・ロジックで処理されます。

実行の単一スレッドは複数の枝 (またはスレッド) に分岐できます。個々の枝は、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枝のビジネス・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、イベントは障害のある状態に移動します。複数の枝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 サブシステムは、以下のタイプの失敗したイベントを取り扱います。

- 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 (言い換えると、ビジネス・ロジックで使用されたメソッドで宣言されなかった例外すべて)

Recovery サブシステムはこれらのタイプの失敗したイベントを収集して、Failed Event Manager インターフェースを通じてこれらを管理目的で使用可能にします。

失敗したイベントは一般的に、これに関連したソースおよび宛先情報を持ちます。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) を基にしており、対話のタイプには無関係です。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。

- 初期の要求中に例外が発生した場合、Failed Event Manager の目的としては、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 の目的としては、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。

これは、すべての非同期呼び出しについて該当します。(Failed Event Manager は、同期呼び出しからの障害を取り扱いません。)

失敗したイベントの管理方法

管理者は管理コンソールで使用可能な Failed Event Manager を使用して、すべての WebSphere Process Server の失敗したイベントをブラウズおよび管理します。失敗したイベントは再サブミットするか、またはシステムから削除できます。

失敗したイベントを管理する共通のタスクは、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ためのデータの編集
- 失敗したイベントの再サブミット
- 失敗したイベントの削除

Failed Event Manager にアクセスするには、「統合アプリケーション」>「Failed Event Manager」をクリックします。

関連概念

288 ページの『失敗したイベントのデータの操作』

292 ページの『失敗したイベントの再サブミット』

関連タスク

『失敗したイベントの検索』

293 ページの『失敗したイベントの削除』

Failed Event Manager のロール・ベースのアクセス

Failed Event Manager は、失敗したイベント・データおよびタスクに対して、ロール・ベースのアクセス制御を使用します。管理者およびオペレーターのロールのみが、Failed Event Manager 内のタスクの実行を許可されています。管理者またはオペレーターのいずれかとしてログインしたユーザーは、失敗したイベントに関連したすべてのデータを表示でき、すべてのタスクを実行できます。

注: このセキュリティー・インフラストラクチャーは、ベースの WebSphere Application Server 製品から継承されたものです。セキュリティー情報について詳しくは、WebSphere Application Server および WebSphere Process Server のインフォメーション・センターを参照してください。

失敗したイベントの検索

失敗したイベントを編集、再サブミット、または削除する前に、これらを識別する必要があります。Failed Event Manager の検索機能を使用して、サーバーのすべての失敗したイベントを検索するか、または失敗したイベントの特定のサブセットを検索します。

このトピックでは、サーバー上のすべての失敗したイベントを検索するための説明があります。また、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を基にした別の検索を実行するためのトピックへの参照もあります。

必要なセキュリティー・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. 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 を実行します。
3. 「このサーバー上の失敗したイベント」ボックスで、「失敗したすべてのイベントを取得」をクリックします。

「検索結果」ページが開き、サーバー上にある WebSphere Process Server の失敗したイベントすべてのリストが表示されます。

関連概念

288 ページの『失敗したイベントのデータの操作』

292 ページの『失敗したイベントの再サブミット』

関連タスク

284 ページの『ソースによる失敗したイベントの検索』

『宛先による失敗したイベントの検索』

285 ページの『日付による失敗したイベントの検索』

286 ページの『ビジネス・オブジェクト・タイプによる失敗したイベントの検索』

286 ページの『例外による失敗したイベントの検索』

287 ページの『失敗したイベントの拡張検索の実行』

293 ページの『失敗したイベントの削除』

宛先による失敗したイベントの検索

「検索」ページの「宛先による検索」タブを使用すれば、特定の宛先のモジュール、コンポーネント、またはメソッドに関連した、失敗したイベントのみを検索できます。Failed Event Manager は、対話のタイプに関係なく、失敗したポイントを基にして宛先を判断します。

検索を実行するには、以下に注意します。

- フィールドの値は、大/小文字の区別があります。
- 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す。
- このタブのいずれかのフィールドを空白にすると、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、モジュール、またはメソッドを検索します。
- 検索は、単一の宛先基準または複数の基準で実行できます。複数の宛先基準で検索すると、失敗したイベントのリストがより詳細化されて提供されます。

必要なセキュリティー・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. 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 を実行します。
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、「宛先による検索」タブが選択された状態になります。

3. 使用する検索基準を指定します。以下のフィールドのあらゆる組み合わせを使用して、検索をカスタマイズできます。
 - 「宛先モジュール」フィールド: このフィールドを使用して、失敗したイベントの宛先モジュールを指定します。
 - 「宛先コンポーネント」フィールド: このフィールドを使用して、失敗したイベントの宛先コンポーネントを指定します。

- 「宛先メソッド」フィールド: このフィールドを使用して、失敗したイベントの宛先メソッドを指定します。
4. 「OK」をクリックして検索を開始します。

「検索結果」ページが開き、指定したモジュール、コンポーネント、またはメソッドを宛先として失敗したイベントすべてのリストが表示されます。

関連概念

281 ページの『WebSphere Process Server の失敗したイベントの管理』
管理者は管理コンソールで使用可能な Failed Event Manager を使用して、
WebSphere Process Server の失敗したイベントを管理できます。

ソースによる失敗したイベントの検索

「検索」ページの「ソースによる検索」タブを使用すれば、特定のソースのモジュール、コンポーネント、またはその両方から発信した、失敗したイベントのみを検索できます。Failed Event Manager は、対話のタイプに関係なく、失敗したポイントを基にしてソースを判断します。

検索を実行するには、以下に注意します。

- フィールドの値は、大/小文字の区別があります。
- 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す。
- このタブのいずれかのフィールドをブランクにすると、ブランク・フィールドはワイルドカードとして扱われます。Failed Event Manager はすべてのコンポーネントまたはモジュールを検索します。
- 失敗したイベントについて最も詳細化した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。

必要なセキュリティー・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. 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 を実行します。
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、「ソースによる検索」タブが選択された状態になります。

3. 検索基準を指定します。以下のフィールドのうちの 1 つまたは両方を使用できます。
 - 「ソース・モジュール」フィールド: このフィールドを使用して、失敗したイベントの発信元モジュールを指定します。
 - 「ソース・コンポーネント」フィールド: このフィールドを使用して、失敗したイベントの発信元コンポーネントを指定します。
4. 「OK」をクリックして検索を開始します。

「検索結果」ページが開き、指定したモジュール、コンポーネント、またはその両方から発信した、失敗したイベントすべてのリストが表示されます。

関連概念

281 ページの『WebSphere Process Server の失敗したイベントの管理』
管理者は管理コンソールで使用可能な Failed Event Manager を使用して、
WebSphere Process Server の失敗したイベントを管理できます。

日付による失敗したイベントの検索

「検索」ページの「日付ごと」タブを使用して、特定の期間で失敗したイベントのみを検索できます。

検索を実行するには、以下に注意します。

- 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は、各フィールドで提供されています。

注: 入力する値は、必須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で置換します。

- 時刻は常時、サーバーのローカル時間です。時刻は、管理コンソールを実行中の個々のマシンのローカル時刻を反映するようには更新されません。
- このタブの両方のフィールドについて、値を指定する必要があります。

必要なセキュリティ・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. 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 を実行します。
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。値に必要なフォーマットはロケールによって異なるため、Failed Event Manager はこのフィールドの上にロケールに適した例を提供します。入力した値が、提供された例と同じ方法でフォーマット設定されているか確認します。(例えば、en_US ロケールに必要なフォーマットは *MM/DD/YY HH:MM Meridiem* です。したがって、このフィールドについて正しくフォーマット設定された値は 11/10/05 4:30 PM となります。)
4. 「**終了日**」フィールドを使用して、終了日時を指定します。値に必要なフォーマットはロケールによって異なるため、Failed Event Manager はこのフィールドの上にロケールに適した例を提供します。入力した値が、提供された例と同じ方法でフォーマット設定されているか確認します。(例えば、en_US ロケールに必要なフォーマットは *MM/DD/YY HH:MM Meridiem* です。したがって、このフィールドについて正しくフォーマット設定された値は 11/17/05 4:30 PM となります。)
5. 「**OK**」をクリックして検索を開始します。

「検索結果」ページが開き、指定した期間に発生して失敗したイベントすべてのリストが表示されます。

ビジネス・オブジェクト・タイプによる失敗したイベントの検索

「検索」ページの「タイプによる検索」タブを使用すれば、特定のビジネス・オブジェクトに関連した、失敗したイベントのみを検索できます。

必要なセキュリティ・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. 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 を実行します。
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、「タイプによる検索」タブが選択された状態になります。

3. 以下のいずれか 1 つを使用して、検索対象とするビジネス・オブジェクト・タイプを指定します。
 - 「**ビジネス・オブジェクト・タイプの選択 (Select the business object type)**」メニュー: このドロップダウン・メニューを使用して、失敗したイベントに関連付けられたビジネス・オブジェクトのタイプを選択します。このメニューには、サーバー上の失敗したイベントから検出した、すべてのビジネス・オブジェクト・タイプのリストがあります。
 - 「**別のビジネス・オブジェクト・タイプ**」フィールド: このフィールドを使用して、失敗したイベントに関連付けられたビジネス・オブジェクトのタイプを指定します。このフィールドはアスタリスク (*) ワイルドカード文字を受け入れます。すべての値は、大/小文字の区別があります。
4. 「**OK**」をクリックして検索を開始します。

「検索結果」ページが開き、指定したビジネス・オブジェクト・タイプに関連した、失敗したイベントすべてのリストが表示されます。

例外による失敗したイベントの検索

「検索」ページの「例外による検索」タブを使用すれば、特定の例外に関連した、失敗したイベントのみを検索できます。例外テキストは一部または全部を指定できます。

必要なセキュリティ・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. 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 を実行します。
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、「例外による検索」タブが選択された状態になります。

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し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は、大/小文字の区別があります。

注: 「例外テキスト」フィールドを空白にすると、ワイルドカードとして扱われます。失敗したイベントすべてが戻されます。

4. 「OK」をクリックして検索を開始します。

「検索結果」ページが開き、指定した例外テキストに関連した、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拡張検索の実行

「検索」ページの「拡張」タブを使用すれば、他の検索タブにあるソース、宛先、日付、ビジネス・オブジェクト・タイプ、および例外テキストなどの基準の組み合わせを使用して、失敗したイベントのより詳細な検索が実行できます。

次のことに注意してください。

- 以下は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。
- フィールドを空白のままにすると、ワイルドカードとして取り扱われます。

拡張検索は最適化されていません。失敗したイベントの大規模な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。

必要なセキュリティ・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. 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 を実行します。
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、「拡張」タブが選択された状態になります。

3. 使用する検索基準を指定します。以下のフィールドのあらゆる組み合わせを使用して、検索をカスタマイズできます。
 - 「宛先モジュール」フィールド: このフィールドを使用して、失敗したイベントの宛先モジュールを指定します。
 - 「宛先コンポーネント」フィールド: このフィールドを使用して、失敗したイベントの宛先コンポーネントを指定します。
 - 「宛先メソッド」フィールド: このフィールドを使用して、失敗したイベントの宛先メソッドを指定します。
 - 「ソース・モジュール」フィールド: このフィールドを使用して、失敗したイベントの発信元モジュールを指定します。
 - 「ソース・コンポーネント」フィールド: このフィールドを使用して、失敗したイベントの発信元コンポーネントを指定します。
 - 「開始日」フィールド: 特定の期間について検索をしたい場合は、このフィールドを使用して、開始日時を指定します。このフィールドはアスタリスク (*) ワイルドカード文字を受け入れません。

- 「終了日」フィールド: 特定の期間について検索をしたい場合は、このフィールドを使用して、終了日時を指定します。このフィールドはアスタリスク (*) ワイルドカード文字を受け入れません。
 - 「ビジネス・オブジェクト・タイプ」フィールド: このフィールドを使用して、失敗したイベントに関連付けられたビジネス・オブジェクトのタイプを指定します。
 - 「例外テキスト」フィールド: このフィールドを使用して、検索の条件とする例外に関連するテキストを指定します。
4. 「OK」をクリックして検索を開始します。

「検索結果」ページが開き、指定した基準に合致する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データの操作

失敗したイベントには、それぞれに関連し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。失敗したイベントに対して 2 つのデータの基本型があります。これらは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、それぞれに関連する以下のデータがあります。

- 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起動型
- イベントが発生したモジュールおよびコンポーネントの名前 (ソース)。Failed Event Manager は、起動が失敗したロケーションを基にして、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、起動が失敗したロケーションを基にして、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、失敗したイベントは関連したトレースおよび有効期限データを持つことがあり、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か、またはビジネス・オブジェクトの一部でない単純データとすることが可能です。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。

関連タスク

289 ページの『失敗したイベントのデータのブラウズ』

289 ページの『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し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。

失敗したイベントのデータのブラウズ

失敗したイベントには、それぞれに関連する以下の 2 タイプのデータがあります。

- 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するか、またはビジネス・オブジェクトの一部でない単純データとすることが可能です。

必要なセキュリティー・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. Failed Event Manager が開いていて、失敗したイベントのリストをシステムに取得したか確認します。
2. Failed Event Manager の「検索結果」ページで、ブラウズしたい失敗したイベントの ID (メッセージ ID 列で検索) 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すべてが表示されます。

3. 失敗したイベントに関連したビジネス・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したビジネス・データを表示します。階層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。

関連タスク

282 ページの『失敗したイベントの検索』

『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し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。

290 ページの『失敗したイベントのビジネス・データの編集』

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し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。

重要: トレースまたは有効期限データに行った編集は、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、イベントの処理をモニターするのを支援するために、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) サーバーに送信できます。「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、デフォルトのトレース値 `SCA.LOG.INFO;COMP.LOG.INFO` がイベントに対して表示されます。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、トレースは起こりません。

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。この失敗を回避するには、イベントを再サブミットするときに有効期限が切れていないようにするために、イベントの有効期限を編集できます。

必要なセキュリティ・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. Failed Event Manager が開いていて、失敗したイベントのリストをシステムに取得したか確認します。
2. Failed Event Manager の「検索結果」ページで、編集したい失敗したイベントの ID (メッセージ ID 列で検索) 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、フィールドの上に提供されています。

4. 失敗したイベントのために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の詳細な情報は、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。
 - 行っ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を再サブミットされ、Failed Event Manager から除去されます。

関連タスク

282 ページの『失敗したイベントの検索』

失敗したイベントのビジネス・データの編集

Failed Event Manager にはビジネス・データ・エディターが提供されており、失敗したイベントを再サブミットする前に、それに関連したビジネス・データを編集できます。それぞれの失敗したイベントについて、エディターでは関連したビジネ

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、階層でどこにいるかが明確に理解できます。

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か、またはビジネス・オブジェクトの一部でない単純データとすることが可能です。失敗したイベントは、単純データとそれに関連したビジネス・オブジェクトの両方を持つことができます。

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到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」列で省略符号 (...) で示されます。

重要: ビジネス・データに行ったすべての編集は、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。

必要なセキュリティー・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. Failed Event Manager が開いていて、失敗したイベントのリストをシステムに取得したか確認します。
2. Failed Event Manager の「検索結果」ページで、編集したい失敗したイベントの ID (メッセージ ID 列で検索) 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し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で保管され、「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. 行っ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・データが正しい場合、「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。

編集済みの失敗したイベントは処理を再サブミットされ、Failed Event Manager から除去されます。

失敗したイベントの再サブミット

イベントの実行を再度試行したい場合、Failed Event Manager から再サブミットする必要があります。イベントは変更せずに再サブミットできます。または再サブミットする前にビジネス・データ・パラメーターを編集できます。

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、失敗した枝に対してのみ処理が再開します。

再サブミットしたイベントについて、イベントの処理のモニターを支援するために、トレースが使用可能です。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、イベントは元のメッセージ ID と、更新された失敗時刻とともに 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。処理の再開は、イベント全体ではなく、失敗した枝に対してのみ行われます。

必要なセキュリティ・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. Failed Event Manager が開いていて、失敗したイベントのリストをシステムに取得したか確認します。
2. 「検索結果」ページで、再サブミットするそれぞれの失敗したイベントの横にあるチェック・ボックスを選択します。
3. 「再サブミット」をクリックします。

選択した各イベントは処理を再サブミットされ、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

失敗したイベントが正常に実行されるかどうかを判断するため、失敗したイベントの再サブミットをモニターできます。Failed Event Manager は、すべての失敗したイベントのために、トレースのオプションを提供しています。

トレースはサービスまたはコンポーネントに対して設定でき、ログまたは Common Event Infrastructure (CEI) サーバーに出力できます。トレースの設定および表示についての詳細な情報は、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。

必要なセキュリティー・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. Failed Event Manager が開いていて、失敗したイベントのリストをシステムに取得したか確認します。
2. 「検索結果」ページで、再サブミットするそれぞれの失敗したイベントの横にあるチェック・ボックスを選択します。
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、トレースは起こりません。

5. 「OK」をクリックして失敗したイベントを再サブミットし、「検索結果」ページに戻ります。

再サブミットしたイベントのトレース・ログを表示するには、対応するコンポーネントのロガーを開くか、または CEI ログ・ビューアーを使用します。

失敗したイベントの削除

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。

必要なセキュリティー・ロール: このタスクを実行するには、管理者またはオペレーターとしてログインしている必要があります。

1. Failed Event Manager が開いていて、失敗したイベントのリストをシステムに取得したか確認します。
2. Failed Event Manager の「検索結果」ページから、以下のいずれか 1 つを行います。
 - 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、「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。これは現在の検索結果のセットから、有効期限が切れたイベントのみを削除することに注意してください。
 - 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。

Common Event Infrastructure の使用

Common Event Infrastructure は、WebSphere Process Server のコア・コンポーネントです。

Common Event Infrastructure は、ランタイム環境で、さまざまなプログラミング環境からのイベントを永続的および保管するための機能を提供します。イベントは、Common Base Event モデル、規格、イベントの構造を定義する XML ベースのフォーマットを使用して表現されます。

Common Event Infrastructure は、イベント・データの管理用の標準の形式および機構を備えた WebSphere Process Server を提供します。以下の機能が提供されています。

- イベント・オブジェクトを作成し、保存し、送信して、後でリトリブする、WebSphere アプリケーションの標準インターフェースおよびサービス。
- (ソース) アプリケーションの作成に関連して、イベント・オブジェクトを登録アプリケーションに直接または Java Message Service (JMS) を通して間接的に渡す機能。Business Process Execution Language (BPEL) ベースのプロセス用、およびデプロイメント記述子拡張子に基づいた JavaBeans (EJB) 呼び出し用のイベント・エミッターがあります。
- 保管されたイベントをブラウズするための Common Base Event ブラウザー。

Common Event Infrastructure の管理

以下の管理用タスクを実行すると、実行時に Common Event Infrastructure コンポーネントの操作を制御することができます。

Common Event Infrastructure 内のロギングおよびトレース

アプリケーションの問題をデバッグするために、Common Event Infrastructure を使用してロギングとトレースを使用可能にすることができます。

Common Event Infrastructure のコンポーネントは、WebSphere Process Server 環境とクライアント環境で使用可能な JSR47 Java ロギング・フレームワークを使用します。ロギング・フレームワークの使用方法について詳しくは、WebSphere Application Server のトラブルシューティングの資料を参照してください。

以下の表に、Common Event Infrastructure のコンポーネントによって使用されるロガー名を示します。

表 12. ロガー名

コンポーネント	ロガー名
ルート・ロガー名	com.ibm.events
イベント・カタログ	com.ibm.events.catalog
イベント・サーバーのサブコンポーネント	com.ibm.events.access com.ibm.events.bus com.ibm.events.distribution com.ibm.events.server
デフォルト・データ・ストア・プラグイン	com.ibm.events.datastore
イベント・エミッター	com.ibm.events.emitter
通知ヘルパー	com.ibm.events.notification
構成	com.ibm.events.configuration
インストール	com.ibm.events.install

表 12. ロガー名 (続き)

コンポーネント	ロガー名
各種ユーティリティ	com.ibm.events.util

DB2 データベース保守

DB2 イベント・データベースを使用する場合は、提供されたスクリプトを実行してデータベース保守を定期的に行ってください。

これらの保守スクリプトの実行対象として現在サポートされる DB2 バージョンは以下のとおりです。

- DB2 Universal バージョン 8.1
- DB2 Universal バージョン 8.2.1

データベース統計の更新:

DB2 データベースを使用可能にして、照会を最適化し、フリー・スペースを検出するために、`runstats` スクリプトを使用してデータベース統計を更新します。

データベース統計は、特に以下のいずれかの条件に当てはまる場合は、定期的に更新することをお勧めします。

- イベントがデータベースからパージされた場合
- 多数のイベントがデータベースに挿入された場合
- テーブルが、`reorg` スクリプトを使用して再編成された場合
- 索引がテーブルに追加されたか、テーブルから除去された場合

`runstats` スクリプトは、`install_root/event/dbscripts/db2` ディレクトリーにあります。

データベース統計を更新するには、以下のコマンドを実行します。

```
runstats.sh db_alias db_user [db_password]
```

パラメーターは、以下のとおりです。

db_alias

データベース別名。DB2 クライアントではイベント・データベースがカタログされている必要があります。DB2 サーバーでスクリプトを実行している場合、既にデータベースはカタログされています。

db_user

使用するためのデータベース・ユーザー ID。このパラメーターは必須です。

db_password

データベース・パスワード。このパラメーターはオプションです。コマンド行にパスワードを指定しないと、DB2 データベースはプロンプトを表示し、パスワードを指定するよう要求します。

例えば、以下のコマンドは、DB2 データベース統計を更新します。ここでは、データベース名は `event` で、データベース・ユーザー ID は `dbadmin`、パスワードは `mypassword` です。

```
runstats.sh event dbadmin mypassword
```

データベース・テーブルの再編成:

イベントが DB2 イベント・データベースからパージされたあとで、reorg スクリプトを使用してデータベース・テーブルを再編成します。

reorg スクリプトは、*profile_root/event/dbscripts/db2* ディレクトリーにあります。

イベント・データベース・テーブルを再編成するには、以下のコマンドを実行します。

```
reorg.sh db2_alias db_user [db_password]
```

パラメーターは、以下のとおりです。

db2_alias

データベース別名。DB2 クライアントではイベント・データベースがカタログされている必要があります。DB2 サーバーでスクリプトを実行している場合、既にデータベースはカタログされています。

db_user

使用するためのデータベース・ユーザー ID。このパラメーターは必須です。

db_password

データベース・パスワード。このパラメーターはオプションです。コマンド行でパスワードを指定しない場合、DB2 データベースからプロンプトが出ます。

例えば、以下のコマンドは、イベント・データベース・テーブルを再編成し、データベース・ユーザー ID dbadmin、パスワード mypassword を使用し、データベース名は event です。

```
reorg.sh event dbadmin mypassword
```

reorg スクリプトを実行したら、runstats スクリプトを使用して、データベース統計を更新する必要があります。詳しくは、『データベース統計の更新』を参照してください。

イベント・データベースからのイベントのパージ:

提供されているスクリプトを使用すると、イベント・データベースから多数のイベントを迅速にパージできます。

デフォルトのデータ・ストア・プラグインは、ユーティリティーのセットを提供しています。これらを使用すると、多数の古いイベントをイベント・データベースから定期的および迅速にパージできます。これらのユーティリティーは、指定した基準にマッチするイベントを削除する **eventpurge.jacl** イベント・サーバー・コマンドとは異なります。

この高速パージ機能では、**バケット**という概念が使用されます。バケットはテーブルのセットで、イベント・データベースのイベントの保管に使用されます。デフォルトのデータ・ストア・プラグインは 2 つのバケットを使用します。

- **アクティブ・バケット**は、最新のイベントを収容するバケットです。つまり、新規のイベントはアクティブ・バケットに格納されます。このアクティブ・バケットは、高速パージ・ユーティリティを使用してパージすることはできません。
- **非アクティブ・バケット**は、古いイベントを収容します。非アクティブ・バケットに格納されるイベントは、照会、削除、または変更できますが、通常、新規のイベントは非アクティブ・バケットには格納されません。非アクティブ・バケットは、高速パージ・ユーティリティでパージできます。

各イベントは 1 つのバケットにのみ格納されます。イベント・コンシューマーからは、アクティブ・バケットと非アクティブ・バケットの違いは見えません。コンシューマーは、特定のイベントを照会、変更、または削除できますが、この際イベントがどのバケットに格納されているか認識する必要はありません。この方法の利点は、データベース固有のインターフェースを使用して、アクティブ・バケットに影響を与えることなく、非アクティブ・バケットを迅速にパージできる点です。通常のイベント・トラフィックは、パージ操作の実行中も継続して実行できます。

非アクティブ・バケットのパージ後、バケットをスワップして、アクティブ・バケットを非アクティブ・バケットに、非アクティブ・バケットをアクティブ・バケットにすることができます。バケットのスワッピングが行えるのは、非アクティブ・バケットが空の場合に限ります。

注: 新規のイベントは、通常はアクティブ・バケットにのみ格納されますが、特定の条件下では、バケットのスワップ直後に新規イベントが非アクティブ・バケットに格納される場合もあります。データ・ストア・プラグインは、定期的にチェックを実行し、どのバケットが現在アクティブとマークされているか判別します。ただし、次のチェックが実行されるまで、一部のイベントは非アクティブ・バケット内に継続して格納されます。また、バッチの一部として送信されるイベントは、すべて同じバケット内に格納されます。これは、バッチの処理中にバケットが非アクティブになった場合も同様です。

この高速パージ機能を使用する場合、バケットをスワップする頻度、または非アクティブ・バケットをパージする頻度を決定するのはユーザーの責任になります。イベント・トラフィック、ストレージ・スペース、アーカイブ要件、およびその他の考慮事項に基づき頻度を決定してください。

アクティブなバケット状況の表示または変更:

アクティブ・バケット状況では、現在どのバケットがアクティブか、および現在のバケットが非アクティブかが示されます。

アクティブ・バケット状況を表示または変更するには、**eventbucket.jacl** スクリプト (`profile_root/event/bin` フォルダ内にある) を次の例のように使用します。

```
wsadmin -f eventbucket.jacl [-status] [-change]
```

このコマンドには、以下のオプションがあります。

-status

このオプションを使用すると、現在のバケット構成についての情報を表示できます。例えば、アクティブ・バケットの設定、およびバケットのチェック間隔 (データ・ストア・プラグインが、どのバケットがアクティブか判定する頻度) を表示できます。

-change

このオプションを使用すると、アクティブ・バケットと非アクティブ・バケットをスワップできます。非アクティブ・バケットは空でないと、このオプションは使用できません。

非アクティブなバケットの消去:

非アクティブ・バケットのパージ方法は、データベース・ソフトウェアにより異なります。

注: Cloudscape データベースは、高速パージ・ユーティリティーをサポートしていません。

DB2 データベース (z/OS システム) の非アクティブなバケットの消去:

z/OS システムでは、DB2 イベント・データベース用の高速パージ・ユーティリティーは、DB2 ロード・ユーティリティーを使用して実装されています。

非アクティブ・バケットをパージする方法

1. **eventbucket.jacl** コマンドを使用して非アクティブ・バケットを識別します (バケット 0 またはバケット 1)。
2. 適切なユーティリティー制御ファイルをアップロードします。これらのファイルは、データベースの構成中に生成され、*profile_path/event/dbscripts/db2zos* ディレクトリに配置されます (*profile_path* は、WebSphere Process Server ランタイム環境のプロファイルを格納するディレクトリのパスです)。以下ファイルのいずれかをアップロードします。
 - バケット 0 が非アクティブな場合、fastpurge00.ctl をアップロードします。
 - バケット 1 が非アクティブな場合、fastpurge01.ctl をアップロードします。

注: 制御ファイルは、論理レコード長 80 の固定レコード・フォーマットを使用してアップロードする必要があります。

3. z/OS ホストで、ISPF DB2I 基本オプション・メニューに移動し、「ユーティリティー」オプションを選択します。
4. 以下の情報を指定します。

フィールド	値
機能	EDITJCL
ユーティリティー	LOAD
ステートメント・データ・セット	アップロードした制御ファイルを収容するデータ・セットの名前
LISTDEF	NO
テンプレート	NO

5. Enter キーを押して、次のパネルに移動します。
6. recdsn 入力フィールドで、アップロードした制御ファイルを収容するデータ・セットの名前を指定します。
7. Enter キーを押します。非アクティブ・バケットをパージする JCL スクリプトが生成されます。

8. Enter キーを押して、出力メッセージをクリアします。
9. 必要に応じて生成された JCL スクリプトを編集します。
10. JCL スクリプトを実行依頼します。

バケット確認間隔の変更:

バケットの確認間隔は、DataStoreEjb.jar ファイルで指定します。

この値では、データ・ストア・プラグインがどのバケットがアクティブか判定する頻度を指定します。デフォルト値は 5 分 (300 秒) です。これより短い間隔を設定すると、スワッピング後にイベントが非アクティブ・バケットに格納される可能性は低下しますが、パフォーマンスも低下します。

バケット確認間隔の変更方法。

1. WebSphere Process Server 管理コンソールで、「エンタープライズ・アプリケーション」>「EventServer」>「EJB モジュール」>「DataStoreEjb.jar」>「デプロイメント記述子の表示」にナビゲートします。
 - a. DefaultDataStoreEJB パラメーターを探します。
 - b. <session id="DataStoreHelperEJB"> メニューを展開し、<env-entry-value> を探します。

これは、バケット・チェック間隔値で、秒数で設定します。
2. BucketCheckInterval 環境変数の値を変更し、秒単位でバケットの確認間隔を指定します。

イベントの使用

Java オブジェクトは、Common Base Event 仕様の表記として使用します。

Common Event Infrastructure は、イベントを Java オブジェクトとして表します。特に、各イベントは、Common Base Event 仕様の Java 表記である `org.eclipse.hyades.logging.events.cbe.CommonBaseEvent` インターフェースをインプリメントするクラスのインスタンスとなります。 `org.eclipse.hyades.logging.events.cbe` package は、Eclipse-based Hyades 環境の一部です。これは、標準と、テスト、トレースおよびモニター用の標準オープン・ソース・ツールのセットです。詳しくは、<http://www.eclipse.org/hyades/> を参照してください。

イベントの一般的なライフ・サイクルは以下のとおりです。

1. イベントを送信する場合、イベント・ソースは `CommonBaseEvent` の新規インスタンスを作成し、このインスタンスをプロパティ・データとともに取り込み、エミッターに対してこのインスタンスをサブミットします
2. エミッターはオプションでコンテンツ完了メカニズム (インプリメントされている場合) を使用し、必要なプロパティ・データを持つイベントを取り込みます。次にエミッターはイベントを検証し、現在の構成済みフィルター基準に対してチェックします。イベントが有効でフィルター基準を満たしている場合、エミッターはこのイベントをイベント・サーバーに送信します。エミッターによるイベント処理について詳しくは、309 ページの『イベントの送信』を参照してください。

3. パーシスタンスが使用可能な場合、イベント・サーバーはイベントをパーシスタント・データ・ストアに格納します。
4. パブリッシングが使用可能な場合、イベント・サーバーはイベントを 1 つ以上の Java Message Service (JMS) 宛先にパブリッシュします。次に、これらの宛先にサブスクライブするイベント・コンシューマーが、新規イベントの通知を受信します。イベント・コンシューマーは次に通知ヘルパーを使用して、受信した JMS メッセージを元の `CommonBaseEvent` のインスタンスに変換します。

また、イベント・コンシューマーは、照会をサブミットして、データ・ストアからイベントを検索する場合があります。通常、コンシューマーは照会インターフェースを使用して、特に開始処理中にヒストリカル・イベントを検索します。

イベントを受信した後、イベント・コンシューマーはイベント・プロパティ・データを読み込み、イベントを処理します。

5. 必要なくなったイベントは、データ・ストアからパーズできます。

XML スキーマ定義言語に基づく `Common Base Event` 仕様は、2 種類のイベント・プロパティ・データを定義します。

- 単純データ型で表されるプロパティ。 `CommonBaseEvent` エレメントの属性として XML でエンコードされます。このプロパティには、 `globalInstanceId`、 `severity`、 `msg` などがあります。 `CommonBaseEvent` Java クラスでは、これらの値は適切なストリングまたは整数で表されます。
- 複素数データ型で表されるプロパティ。 `CommonBaseEvent` エレメントのサブエレメントとして XML でエンコードされます。このプロパティには、 `situation`、 `sourceComponentId`、 `extendedDataElements` などがあり、各プロパティにはネストされたプロパティがあります。これらの複合タイプは、 `org.eclipse.hyades.logging.events.cbe` パッケージで定義された、特別な Java クラスで表されます。例えば、 `sourceComponentId` プロパティは、 `ComponentIdentifier` のインスタンスで表されます。

`CommonBaseEvent` インターフェースは、各プロパティの `getter` および `setter` メソッドと複素数プロパティの作成を単純化する `helper` メソッドを定義します。イベント・ソースは、 `setter` メソッド (または `helper` メソッド) を使用して、エミッターにイベントをサブミットする前にプロパティ・データとともにこのイベントを取り込みます。イベント・コンシューマーは、 `getter` メソッドを使用して、受信イベントからプロパティ・データをリトリブします。

XML スキーマ仕様について詳しくは、 <http://www.w3.org/XML/Schema> を参照してください。

イベント・オブジェクトの作成:

新規イベントは、イベント・ファクトリー を使用して作成します。

イベント・ソース内に新規イベントを作成するには、イベント・ファクトリー を使用します。イベント・ファクトリーとは、 `CommonBaseEvent` の新規インスタンスまたは複素数プロパティ・データ型を表す特別なクラスの新規インスタンスを戻すオブジェクトのことです。

イベント・ファクトリーにアクセスするには、以下の 2 つの方法があります。

- イベント・ファクトリー を使用して、新規イベント・ファクトリーを作成します。この方法は、使用可能な適切なイベント・ファクトリーがない場合に使用します。新規イベント・ファクトリーを作成する場合は、オプションでコンテンツ・ハンドラーを指定して、コンテンツが自動的に完了するようにすることができます。
- Java Naming and Directory Interface (JNDI) ネーム・スペース内にバインドされている既存のイベント・ファクトリーを使用できます。この方法は、使用するイベント・ファクトリーが管理者によって提供されている場合に使用します。これにより、作成するイベントは適切なビジネス・ルールに準拠します。それは、イベント・ファクトリーがコンテンツ・ハンドラーで構成されるためです。

新規のイベント・ファクトリーの作成:

イベント・ファクトリー は、新規イベントを作成するために使用されます。

新規イベント・ファクトリーを作成するには、EventFactoryFactory クラスとして実装される、イベント・ファクトリーを使用します。このクラスにはインスタンスがありません。代わりに、イベント・ファクトリーを作成するのに使用される 2 つの静的メソッドが用意されています。どちらのメソッドを使用するかは、自動コンテンツ完了をインプリメントするためにコンテンツ・ハンドラーを使用するかどうかによって異なります。詳しくは、303 ページの『イベント・コンテンツの自動終了』を参照してください。

コンテンツ・ハンドラーを含まない汎用イベント・ファクトリーを作成する場合は、EventFactoryFactory の createEventFactory 静的メソッドを使用します。

```
EventFactory eventFactory =
    (EventFactory) EventFactoryFactory.createEventFactory();
```

コンテンツ・ハンドラーを備えたイベント・ファクトリーを作成する場合は、createEventFactory(ContentHandler) メソッドを使用して、使用するコンテンツ・ハンドラーを指定します。

```
EventFactory eventFactory =
    (EventFactory) EventFactoryFactory.createEventFactory(contentHandler);
```

いずれの場合も、戻されるオブジェクトは、新規イベントを作成する場合に使用可能なイベント・ファクトリーです。

JNDI ルックアップによるイベント・ファクトリーの取得:

JNDI ルックアップを使用して、イベント・ソースのために JNDI を使用する、イベント・ファクトリーを検索することができます。

イベント・ソースが使用するように、既存のイベント・ファクトリーが管理者によって JNDI にバインドされている場合は、以下のように標準 JNDI ルックアップを実行してイベント・ファクトリーをリトリブします。

```
import javax.naming.*
import org.eclipse.hyades.logging.events.cbe.*

Context context = new InitialContext();
EventFactory eventFactory =
    (EventFactory) context.lookup("com/ibm/events/EventFactory");
```

戻されたオブジェクトは提供されたイベント・ファクトリーです。イベント・ファクトリーがコンテンツ・ハンドラーを使用して構成されている場合は、コンテンツ・ハンドラーのインスタンスもローカルに作成されます。コンテンツ・ハンドラーおよび JNDI については、303 ページの『イベント・コンテンツの自動終了』を参照してください。

イベントの作成と取り込み:

イベント・ファクトリーの取得すると、イベント・オブジェクトを作成して、それらのオブジェクトをプロパティ・データとともに取り込むことができます。

ほとんどのイベント・プロパティは、Common Base Event 仕様によってオプションとして定義されています。ただし、以下のプロパティは必須です。

- *version* (string 属性)
- *creationTime* (XML スキーマ dateTime 属性。
<http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#dateTime> を参照)
- *sourceComponentId* (複雑な ComponentIdentification エレメント)
- *situation* (複雑な Situation エレメント)

注: *version* 属性は、Common Base Event 仕様によってオプションとして定義されています。ただし、この属性を指定しない場合は、デフォルト値 1.0 が想定されます。Common Event Infrastructure ではこの仕様のバージョン 1.0.1 のみがサポートされているため、この値を指定する必要があります。

これらのプロパティのいずれかが欠落しているイベントを送信すると、エミッターはイベントをリジェクトし、EventsException 例外をスローします。

以下のコード・フラグメントは、イベントを作成し、必要な最小限のプロパティ・データでそのイベントを取り込みます。

```
CommonBaseEvent event = eventFactory.createCommonBaseEvent();

event.setVersion("1.0.1"); // set version

long currentTime = System.currentTimeMillis(); // get current time
event.setCreationTimeAsLong(currentTime); // and set creationTime

// set sourceComponentId (a complex type)
event.setSourceComponentId("Windows", // application
                            "svchost.exe", // component
                            "tlntsvr.exe", // subcomponent
                            "http://www.ibm.com/namespaces/autonomic/Windows",
                            // componentType
                            "win386_svc", // componentIdType
                            "9.45.72.138", // location
                            "IPV4" // locationType
                            );

// create situation object
Situation situation = eventFactory.createSituation();

// set situationType to AvailableSituation (a complex type)
situation.setAvailableSituation("EXTERNAL", // reasoningScope
                                "NOT_AVAILABLE", // availabilityDisposition
                                "STARTABLE", // operationDisposition
```

```
"FUNCTION_PROCESS"); // processingDisposition

// set situation
event.setSituation(situation);
```

この例は最初にイベント・ファクトリーを使用して新規のイベント・インスタンス、*event* を作成します。まず、*version* プロパティを設定して、次に現在のシステム時間を検索し、*setCreationTimeAsLong(long)* メソッドを使用して *creationTime* プロパティの値を設定します。代替として、XML *dateTime* フォーマット (例えば、"2004-07-29T13:12:00-05:00") を使用して作成時刻を設定する *setCreationTime(String)* メソッドを使用することもできます。

次の必須プロパティ *sourceComponentId* は、独自のプロパティを持つ *ComponentIdentification* のインスタンスによって表される複合プロパティです。ただし、直接インスタンスを生成したり、このオブジェクトと相互作用する必要はありません (これを行うことは可能ですが)。代わりに、例の次のステートメントはヘルパー・メソッド *setSourceComponentId()* を使用してネストされたプロパティを指定します。ヘルパー・メソッドはこれらの値を使用して、*ComponentIdentification* のインスタンスを作成してから、作成したインスタンスを使用して、イベントの *sourceComponentId* プロパティの値を設定します。

その他の複合プロパティを設定する場合は、同様のヘルパー・メソッドを使用できます (例えば、*setMsgDataElement()*、*addAssociatedEvent*、および *addExtendedDataElement()* があります)。これらのメソッドの多くは異なるシグニチャーを持つ複数のバージョンに存在しているため、プロパティ値をさまざまな方法で指定することができます。これらのメソッドについて詳しくは、Javadoc API の資料を参照してください。

例の中の最後の必須プロパティ *Situation* も複合プロパティです。この場合は、イベント・ファクトリーを使用して状態オブジェクトを直接インスタンス化する必要があります。例では、ヘルパー・メソッドを使用して、複合サブエレメントである *situationType* プロパティを設定します。

実際のアプリケーションでは、有益なイベントには、この例に示すよりもさらに多くの情報を含める必要があります。しかし、これは、Common Base Event 仕様および Common Event Infrastructure で必要となる最低限です。イベントが有効になり、エミッターにサブミットできます。

イベント・コンテンツの自動終了:

プロパティおよびポリシーを設定して、イベント・コンテンツを自動的に完了することができます。

状況に応じて、作成したすべてのイベントのプロパティ・データが自動的に設定されるようにすることができます。これは、変化しない特定の標準値 (アプリケーション名など) を入力する場合や、ランタイム環境から使用可能な情報に基づくプロパティの一部 (作成時間やスレッド情報など) を設定する場合に役立ちます。ビジネス・ルールに基づいてイベント・コンテンツを管理するポリシーを設定することもできます。例えば、特定の拡張名を持つイベントに、特定の値に設定された重大度を指定する必要がある場合があります。

これは、コンテンツ・ハンドラー を作成して実行できます。コンテンツ・ハンドラーは、使用する任意のポリシーに基づいた各イベントのプロパティ値を自動的に設定するオブジェクトです。 Common Event Infrastructure では、イベントが Common Base Event 仕様に準拠している限り、コンテンツ・ハンドラーがイベント・データを変更する方法に対して制限が設定されていません。

すべてのイベント・ソースが同じポリシーに準拠するようにするには、(EventFactoryFactory を使用して) コンテンツ・ハンドラーに関連付けられているイベント・ファクトリーを作成してから、作成したイベント・ファクトリーを JNDI ネーム・スペースにバインドします。それ自身のイベント・ファクトリーを作成する代わりに、イベント・ソースは JNDI ルックアップを実行して、コンテンツ・ハンドラーを使用せずに既に存在するイベント・ファクトリーにアクセスできます。ビジネス・ルールが後で変更された場合は、1 箇所でコンテンツ・ハンドラーを変更できます。

イベント・ソースは、コンテンツ完了を使用可能にするために何もする必要はありません。イベント・ファクトリーがコンテンツ・ハンドラーに関連付けられている場合、イベント・ファクトリーが作成する各イベントは、そのコンテンツ・ハンドラーへの参照を保持します。イベントをエミッターにサブミットすると、イベントはコンテンツ・ハンドラーの completeEvent() メソッドを呼び出し、自身への参照を渡します。これにより、イベント・ソースがイベント固有のプロパティを設定した後、イベントがエミッターによって検証および処理されるより前に、イベントに正しいポリシーが適用されます。

注: イベントがプロセスからプロセスに伝送される場合、コンテンツ・ハンドラーへの参照は伝送されません。これは、コンテンツ完了が、イベントが生成された環境に依存しており、必要な情報が他の場所では利用できないためです。この制限は、相互にローカルであるアプリケーション間の呼び出し (例えば、ローカル・インターフェースを使用した Enterprise Bean への呼び出し) には影響を与えません。

コンテンツ・ハンドラーを作成するには、以下のステップに従います。

1. org.eclipse.hyades.logging.events.cbe.ContentHandler インターフェースをインプリメントする新規 Java クラスを作成します。このインターフェースは completeEvent(CommonBaseEvent) と呼ばれる単一メソッドを定義します。パラメーターは、完成する必要があるコンテンツを持つイベントです。このメソッドをインプリメントする場合は、CommonBaseEvent の getter メソッドおよび setter メソッドを使用し、適用されるポリシーに従ってイベント・プロパティ・データを処理します。

注: イベント・ソースが JNDI を使用してイベント・ファクトリーを検索すると、イベント・ファクトリーと共にコンテンツ・ハンドラーとが戻されます。このため、コンテンツ・ハンドラーは順次実行される必要があります。

以下の例は、各イベントの拡張名を自動的に設定する簡単なコンテンツ・ハンドラーです。

```
import java.io.Serializable;
import org.eclipse.hyades.logging.events.cbe.*;

public class BusinessContentHandler
    implements ContentHandler, Serializable {
```



```

        public void completeEvent(CommonBaseEvent event)
            throws CompletionException {
            event.setExtensionName("business");
        }
    }
}

```

2. コンテンツ・ハンドラーとイベント・ファクトリーを関連付けます。 関連付けを行うには、イベント・ファクトリーを作成する際にコンテンツ・ハンドラーを指定します。

```

EventFactory eventFactory =
    (EventFactory) EventFactoryFactory.createEventFactory(contentHandler);

```

戻されるイベント・ファクトリーは、指定したコンテンツ・ハンドラーと永続的に関連付けられます。

受信されたイベントからのデータの検索:

イベント・ソースは `CommonBaseEvent` のメソッドを使用してイベント・プロパティ・データを検索します。

イベント・ソースはイベントを受信すると、`CommonBaseEvent` の `getter` メソッドを使用して、イベント・プロパティ・データをリトリブします。例えば、以下のコード・フラグメントでは、単一イベントをリトリブしてから、`msg` プロパティの内容を読み取ります。

```

CommonBaseEvent event = eventAccess.queryEventByGlobalInstanceId(eventId);
String eventMessage = event.getMsg();

```

リトリブするプロパティが複素数プロパティ (`Common Base Event` 仕様の `CommonBaseEvent` のサブエレメント) の場合、戻り値は、複素数データ型を表す特別なクラスのインスタンスとなります。戻されたオブジェクトの `getter` メソッドを使用して、そのオブジェクトからプロパティ・データをリトリブします。例えば、以下のコード・フラグメントは、複合プロパティである `componentId` の値を検索します。続いて、ストリングであるネストされた `component` プロパティのコンテンツを検索して、ソース・コンポーネントの名前を読み取ります。

```

CommonBaseEvent event = eventAccess.queryEventByGlobalInstanceId(eventId);
ComponentIdentification componentId = event.getSourceComponentId();
String componentName = componentId.getComponent();

```

XML イベントの変換:

イベント・ソースは、他のアプリケーションから XML 形式のイベントに変換できます。

イベント・ソースは、新規イベントを作成するだけでなく、他のアプリケーションから受け取った XML フォーマットのイベントを変換します。同様に、イベント・コンシューマーは、別のアプリケーションに転送するイベントを XML フォーマットに変換することができます。 `org.eclipse.hyades.logging.events.cbe.EventFormatter` クラスは、`CommonBaseEvent` インスタンスと XML の間の変換に使用できるメソッドを提供します。

EventFormatter を使用して、CommonBaseEvent のインスタンスを XML 文書または XML フラグメントのいずれかを含むストリングに変換できます。同様に、XML 文書またはフラグメントから CommonBaseEvent のインスタンスに変換することができます。

EventFormatter について詳しくは、org.eclipse.hyades.logging.events.cbe パッケージにある Javadoc の資料を参照してください。

イベント・インスタンス・メタデータへのアクセス:

イベント・インスタンス・メタデータへのアクセスは、Java ベースの Eclipse Modeling Framework によって実行されます。

イベント・オブジェクトを使用する場合に必要なクラスおよびインターフェースを提供する org.eclipse.hyades.logging.events.cbe パッケージは、Eclipse Modeling Framework (EMF) に基づいています。EMF とは、構造化されたデータ・モデルに基づいてアプリケーション・コードを生成するときに使用される Java フレームワークのことです。また、EMF は、データ・モデルを記述するメタデータにアクセスするために使用するインターフェースも生成済みコード内に提供します (EMF について詳しくは、<http://www.eclipse.org/emf> にある Eclipse Modeling Framework 資料を参照)。

EMF と互換性のあるツールは、このインターフェースを使用して、データ・モデルに関する予備知識やインプリメンテーションへのアクセスなしで CommonBaseEvent イベント・データと対話できます。これにより、開発ツールで他のデータ・モデルのデータを CommonBaseEvent モデルに転送するコードを生成できます。その後、アプリケーション開発者は、データを構築するコードではなくデータを使用するコードの作成に焦点を合わせることができます。

例えば、ネットワーク・イベントをモニターし、EMF で独自のデータ・モデルを記述するイベント・ソースがあると仮定します。両方のデータ・モデルへのアクセスによって、開発ツールには、イベント・ソース・データ・モデルのフィールドとともに CommonBaseEvent データ・モデルのフィールドが表示されます。次に開発者は、グラフィカル・インターフェースを使用して、イベント・ソース・モデルのフィールドから CommonBaseEvent モデルのフィールドへのマップ方法を示すことができます。例えば、イベント・ソース・データ・モデルの Workstation.name というフィールドは、CommonBaseEvent データ・モデルの

CommonBaseEvent.sourceComponentId.location フィールドに対応します。どちらのデータ・モデルも標準の EMF インターフェースを使用して記述されているので、ツールにより、2 つのモデル間でのデータ転送を処理するコードを生成することができます。

以下のコード・フラグメントは、開発ツールがどのように EMF インターフェースを使用して CommonBaseEvent データ・モデルに関する情報を照会し、この情報を使用してイベント・インスタンスと対話するかを示す例です。この例は、単純なイベント・コンシューマーの一部と見なすことができます。イベント・コンシューマーは、イベント・インスタンスのすべてのフィールドを繰り返し処理し、フィールドごとに、すべてのフィールドの名前と値を出力します。

```
// event is a valid CommonBaseEvent instance

// Get list of event instance structural features (fields)
```

```

List features = event.eClass().getEAllStructuralFeatures();

// iterate through list; print names and values
for (int i = 0 ; i < features.size() ; i++)
{
    EStructuralFeature feature = (EStructuralFeature)features.get(i);
    Object value = eObj.eGet(feature);
    System.out.println(feature.getName() + ":" + value);
}

```

CommonBaseEvent データ・モデルは、EMF ファイル cbe.ecore および cbe.genmodel に記述されています。これらのファイルは、Common Event Infrastructure SDK に組み込まれています。ユーザーは、Eclipse ベースの開発環境にこれらのファイルをインポートし、EMF を使用して CommonBaseEvent オブジェクトと対話するコードを生成できます。

イベント・ソースの作成

イベント・ソースは、エミッター・オブジェクトを介して、イベント・サーバーと対話します。

イベント・ソース は、エミッターを使用してイベントをイベント・サーバーに送信する任意のアプリケーションです。イベント・ソースには、以下のアプリケーションがあります。

- モニター対象リソースに関連するイベントを生成するアダプターまたはモニター
- 通知イベントを生成するアプリケーション
- 他のソースからイベントを転送するアプリケーション

イベント・ソースは、Java 2 Platform, Standard Edition (J2SE) または Java 2 Platform, Enterprise Edition (J2EE) のいずれかを使用して Java プログラム言語にインプリメントされます。イベント・ソースは、Common Base Event モデルに準拠した有効なイベントをサブミットする必要があります。各イベントは、Java オブジェクトとして表されます。

エミッターとエミッター・ファクトリー

イベント・ソースは、イベント・サーバーとは直接対話をしません。イベント・ソースは、エミッターと呼ばれるオブジェクトと対話します (com.ibm.events.emitter.Emitter インターフェースのインプリメンテーション)。エミッターとは、イベントを送信するメソッドを提供するローカル・オブジェクトのことです。

通常、エミッターはイベント送信の詳細を処理します。このため、イベント・ソースの開発者は、イベント・サーバーのロケーション、フィルター設定、および基礎となる送信メカニズムを考慮する必要がありません。これらの詳細は、エミッター・ファクトリー で管理されます。これは、管理者が構成したオブジェクトで、Java Naming and Directory Interface (JNDI) ネーム・スペース内でバインドされます。エミッター・ファクトリーは、com.ibm.events.emitter.EmitterFactory のインスタンスで、エミッター・オブジェクトを作成するために使用されます。また、エミッター・ファクトリーが作成するエミッターの動作も定義します。以下の設定が含まれています。

- 優先トランザクション・モード。この設定は、エミッターが新規トランザクション内の各イベントを送信するか、または現行トランザクション内の各イベントを送信するかを指定します。イベント・ソースは特定のエミッターまたはイベントのサブミットに関してこの設定を変更できますが、プロファイルはデフォルト値を指定します。(この設定は、J2EE コンテナでのみ有効です。J2SE プラットフォームには、トランザクション制御がありません)。
- 優先同期モード。この設定は、同期伝送または非同期伝送のいずれかを使用してイベントを送信するかどうかを指定します。同期伝送では、sendEvent() メソッドは、イベントが処理されるまで呼び出し元に制御を戻しません。非同期伝送では、sendEvent() メソッドはイベントがサブミットされた直後に戻り、呼び出し元にはイベント処理に関する詳細情報が送信されません。イベント・ソースはエミッターまたはイベント・サブミットに関してこの設定を変更できますが、デフォルト値はプロファイルにより指定されます。
- 使用する伝送プロファイル。伝送プロファイルとは、イベントをイベント・サーバーに送信する特定の伝送メカニズムを定義する構成オブジェクトのことです。エミッター・ファクトリー・プロファイルでは、同期伝送のプロファイルと非同期伝送のプロファイルを指定できます。イベント・ソースは、エミッターが使用している伝送プロファイルを変更することはできません。
- エミッターに使用するフィルター構成。このフィルター構成は、エミッターにサブミットされるイベントをフィルター操作するためにどのフィルター・プラグインを使用するかを定義します。Common Event Infrastructure にはデフォルトのフィルター・プラグインが含まれていますが、別のフィルタリング・エンジンを使用したい場合は、独自のフィルター・プラグインをインプリメントすることもできます。

管理者は、各プロファイルが異なるエミッター構成を定義する複数のエミッター・ファクトリー・プロファイルを作成できます。イベント・ソースは、既存のエミッター・ファクトリー・プロファイルに関連付けられているエミッター・ファクトリーを使用して、エミッターを取得します。このため、同じエミッター・ファクトリーで作成されたすべてのエミッターのデフォルトの振る舞いは同じです。詳しくは、『エミッターの取得』を参照してください。

注: イベント・ソースが Java 2 セキュリティーを使用可能にして実行されている状態で、ユーザー独自のグローバル固有 ID (GUIDs) を生成する必要がある場合は、正しく処理できるようにポリシー・ファイルを変更する必要があります。以下のエントリーを追加します。

```
permission java.io.FilePermission "${java.io.tmpdir}${/}guid.lock",
    "read,write,delete";
permission java.net.SocketPermission "*", "resolve";
```

エミッターの取得:

エミッターを取得する場合は、少なくとも 1 つのエミッター・ファクトリー・プロファイルが構成されている必要があります。

エミッター・ファクトリー・プロファイルごとにエミッター・ファクトリーが自動的に作成されます。エミッター・ファクトリーには、エミッター・ファクトリー・プロファイルの JNDI 名を使用してアクセスできます。

エミッターを取得するには、以下のステップに従います。

1. エミッターに使用したいエミッター・ファクトリーの名前を指定して JNDI ルックアップを実行します。これは、管理者がエミッター・ファクトリー・プロファイルを定義する際に指定する JNDI 名です。
2. エミッター・ファクトリーの `getEmitter()` メソッドを呼び出します。戻されるオブジェクトは、ユーザーが指定したエミッター・ファクトリー・プロファイルで定義されたオプションに従って構成されたエミッターです。エミッター・ファクトリーがエミッターを取得できない場合は、`EmitterException` 例外がスローされます。

注: イベント・ソースが、機密保護機能のある環境で稼働している J2EE クライアント・アプリケーションであり、使用しているエミッター・プロファイルが非同期伝送プロファイルを指定している場合は、エミッターを取得するために JMS のユーザー名とパスワードを指定する必要があります。そのためには、`getEmitter(String, String)` メソッドを使用して、使用する JMS のユーザー名とパスワードを渡します。詳しくは、`com.ibm.events.emitter` クラスの Javadoc 文書を参照してください。

以下のコード・フラグメントでは、プロファイル・デフォルトを使用して構成されたエミッターを取得します。

```
import javax.naming.*
import com.ibm.events.emitter.*

Context context = new InitialContext();
EmitterFactory emitterFactory =
    (EmitterFactory) context.lookup("com/ibm/events/configuration/emitter/Default");
Emitter emitter = emitterFactory.getEmitter();
```

イベントの送信:

イベント・ソースは、イベントを Java オブジェクト形式で送信します。

特に、各イベントは、Common Base Event 仕様の Java 表記である `org.eclipse.hyades.logging.events.cbe.CommonBaseEvent` インターフェースをインプリメントするクラスのインスタンスとなります。詳しくは、『*Common Base Event モデル*』を参照してください。

イベントを送信するには、`Emitter` インターフェースの `sendEvent()` メソッドを使用します。エミッターにイベントを送信すると、以下が発生します。

1. エミッターは、イベントの `complete()` メソッドを呼び出して、オプションのコンテンツ完了をトリガーします。詳しくは、303 ページの『*イベント・コンテンツの自動終了*』を参照してください。
2. エミッターは、まだシーケンス番号およびグローバル・インスタンス ID のないイベントにこれらを割り当てます。
3. エミッターは、イベントを検証して、Common Base Event 仕様に準拠しているかどうかを確認します。

注: 現行の Common Base Event 仕様は、イベント包含関係の階層の各レベルで指定された名前を持つ 1 つの拡張データ・エレメントのみを許可します。ただしこの制限は、本仕様の将来のバージョンには含まれず、Common Event Infrastructure では施行されていません。

4. フィルタリングがアクティブの場合、エミッターは、現行のフィルター基準に対してイベントを確認して、イベントが送信すべきか廃棄すべきかを判別します。
5. 最後に、イベントが有効でフィルター基準を満たしている場合、パーシスタンスを保ちイベント・コンシューマーへの配布を行うために、エミッターはこのイベントをイベント・サーバーに送信します。

イベントが無効な場合や、イベント・サーバーへイベントの送信を試みた際にエミッターで問題が発生した場合には、例外がスローされます。

注: WebSphere セキュリティーが使用可能になっている場合に同期 EJB 送信を使用してイベントを送信するには、アプリケーション・ユーザー ID が `eventCreator` または `eventAdministrator` のロールにマップされている必要があります。

現行のエミッター設定でのイベントの送信:

イベントは、伝送モードを指定する必要がない場合、現在のエミッター設定を使用して送信することができます。

特定の伝送モードまたはトランザクション・モードを指定する必要がない場合は、現行のエミッター設定を使用してイベントを送信することができます。これらの設定は、最初はエミッター・ファクトリー・プロファイル内の管理者によって定義されますが、後にイベント・コンシューマーによって変更することが可能です。

現在のエミッター設定を使用してイベントを送信するには、`sendEvent(CommonBaseEvent)` メソッドを使用します。

```
String eventId = emitter.sendEvent(event);
```

この例では、`emitter` がエミッター・インスタンスで、`event` が `CommonBaseEvent` インスタンスです。

戻り値 `eventId` は、イベントのグローバル固有 ID (`CommonBaseEvent` の `globalInstanceId` フィールドの値) です。イベントをサブミットするときにイベントに `globalInstanceId` がない場合、エミッターによって自動的に割り当てられます。

注: イベントがエミッターにサブミットされても、イベントがイベント・サーバーに送信されたことを保証するものではありません。フィルター設定によってイベントが廃棄される可能性があるためです。 `sendEvent()` の呼び出しが正常に終了するのは、イベントがエミッターで正常に処理された場合のみです。

現在のエミッター設定のオーバーライド:

エミッター・ファクトリー・プロファイル設定をイベント・コンシューマーで変更することができます。

イベントを送信する際に、エミッターに対して現在構成されている現行のトランザクション・モード、同期モード、または両方をオーバーライドするオプションを指定できます。これらの設定は、最初はエミッター・ファクトリー・プロファイル内の管理者によって定義されますが、後にイベント・コンシューマーによって変更することが可能です。

エミッターは、すべての同期およびトランザクション・モードをサポートしない場合があります。使用可能なモードは、以下の制限に基づきます。

- エミッターでサポートされる同期モードは、エミッター・ファクトリー・プロファイルで定義されます。 `isSynchronizationModeSupported()` メソッドを呼び出すことによって、特定のエミッターがサポートしているモードを確認できます。詳しくは、`com.ibm.events.emitter.Emitter` の Javadoc API の資料を参照してください。
- トランザクションは、J2EE コンテナ内でのみサポートされています。

サポートされていないモードを使用しようとする、エミッターは、`TransactionModeNotSupportedException` または `SynchronizationModeNotSupportedException` 例外をスローします。

エミッター設定をオーバーライドするには、`sendEvent(CommonBaseEvent, int, int)` メソッドを使用します。

```
String eventId = emitter.sendEvent(event,
                                synchronizationMode,
                                transactionMode);
```

パラメーターは、以下のとおりです。

event

送信するイベント・オブジェクト (`CommonBaseEvent` のインスタンス)

synchronizationMode

インターフェース `SynchronizationMode` で定義された整数定数。これは、以下の定数のいずれかです。

- `SynchronizationMode.ASYNCHRONOUS` (イベントを非同期に送信します。)
- `SynchronizationMode.SYNCHRONOUS` (イベントを同時に送信します。)
- `SynchronizationMode.DEFAULT` (現在のエミッター設定を使用します。)

transactionMode

インターフェース `TransactionMode` で定義された整数定数。

- `TransactionMode.NEW` (新規トランザクションのイベントを送信します。)
- `TransactionMode.SAME` (現在のトランザクションのイベントを送信します。)
- `TransactionMode.DEFAULT` (現在のエミッター設定を使用します。)

イベントは、指定されたオプションで送信されます。これらのオプションは、送信される単一イベントにのみ適用されます。エミッター設定は変更されず、後続のイベント・サブミットは影響を受けません。

戻り値 `eventId` は、イベントのグローバル固有 ID (`CommonBaseEvent` の `globalInstanceId` フィールドの値) です。イベントをサブミットするときイベントに `globalInstanceId` がない場合、エミッターによって自動的に割り当てられます。

注: イベントがエミッターにサブミットされても、イベントがイベント・サーバーに送信されたことを保証するものではありません。フィルター設定によってイベントが廃棄される可能性があるためです。 `sendEvent()` の呼び出しが正常に終了するのは、イベントがエミッターで正常に処理された場合のみです。

以下の例では、新規トランザクションでイベントを送信するためにエミッター設定をオーバーライドしますが、同期モードはオーバーライドしません。

```
String eventId = sendEvent(event,
                          SynchronizationMode.DEFAULT,
                          TransactionMode.NEW);
```


複数イベントの送信:

イベント・ソースがバッチで複数のイベントを送信する必要がある場合は、`sendEvents()` メソッドへの単一呼び出しで送信し、パフォーマンスを向上することができます。

基礎となるトランザクションが正常に完了した場合のみ送信する必要がある論理グループのイベントに対して、このようにイベントをバッチにすることは便利です。サブミットされたイベントのすべては単一トランザクションの一部として送信されます。

- 現在のエミッター設定を使用して複数のイベントを送信するには、`sendEvents(CommonBaseEvent[])` メソッドを使用します。

```
String[] eventIds = emitter.sendEvents(events);
```

この例では、`emitter` がエミッターのインスタンスで、`events` が `CommonBaseEvent` インスタンスの配列です。

- 複数のイベントを送信し、現在のエミッター設定をオーバーライドするには、使用する同期モードおよびトランザクション・モードを指定して、`sendEvents(CommonBaseEvent, int, int)` メソッドを使用します。

```
String eventId = emitter.sendEvent(event,  
                                  synchronizationMode,  
                                  transactionMode);
```

戻り値 `eventIds` は送信イベントのグローバル固有 ID を含む配列です。

各イベントは現在のフィルター基準に対して検証され、確認されます。その後、フィルター基準を通過した有効なイベントはすべて適切なメカニズムを使用して送信されます。

- 同期イベント・トランスポートを使用している場合、イベントは単一 EJB 呼び出しを使用して送信されます。EJB 呼び出し時にエラーが発生した場合、例外がスローされ、イベントは送信されません。
- 同期イベント・トランスポートを使用している場合、すべてのイベントは単一 JMS メッセージを使用して送信されます。JMS 処理時にエラーが発生した場合、例外がスローされ、イベントは送信されません。

エミッター設定の変更:

エミッター・ファクトリー・プロファイルでの設定は、イベント・ソースで変更することができます。

イベント・ソースは、エミッター用に構成されたトランザクション・モードおよび同期モードを変更できます。これらの設定は、初めエミッター・ファクトリー・プロファイルで定義されています。さらに、イベント・ソースは、現行のトランザクション・モードを照会して、現在エミッターで有効な設定を判別することができます。

同期モードの変更:

イベント・ソースは、エミッターが使用する同期モードを変更できます。

この変更は、後続のイベント実行依頼に有効です。しかし、エミッター・ファクトリー・プロファイルで定義された同期モード指定は変更しません。

エミッターでサポートされる同期モードは、エミッター・ファクトリー・プロファイルで定義されます。 `isSynchronizationModeSupported()` メソッドを呼び出すことによって、特定のエミッターがサポートしているモードを確認できます。詳しくは、`com.ibm.events.emitter.Emitter` の Javadoc API の資料を参照してください。サポートされていないモードを使用しようとする、エミッターは、`SynchronizationModeNotSupportedException` 例外をスローします。

同期モードを変更するには、`setSynchronizationMode(int)` メソッドを使用します。
`emitter.setSynchronizationMode(synchronizationMode);`

synchronizationMode は、インターフェース `SynchronizationMode` によって定義された整数定数です。

- `SynchronizationMode.ASYNCHRONOUS` (イベントを非同期に送信します。)
- `SynchronizationMode.SYNCHRONOUS` (イベントを同時に送信します。)
- `SynchronizationMode.DEFAULT` (現行のエミッター設定を使用してイベントを送信します。)

トランザクション・モードの変更:

イベント・ソースは、エミッターが使用するトランザクション・モードを変更できます。

このトランザクション・モードの変更は、後続のイベント実行依頼にも有効ですが、エミッター・ファクトリー・プロファイル内で定義されたトランザクション・モードは変更できません。

注: トランザクションは、J2EE コンテナ内でのみサポートされています。

トランザクション・モードを変更するには、`setTransactionMode(int)` メソッドを使用します。

`emitter.setTransactionMode(transactionMode);`

transactionMode は、以下のインターフェース `TransactionMode` で定義された整数定数です。

- `TransactionMode.NEW` (新規トランザクションのイベントを送信します。)
- `TransactionMode.SAME` (現在のトランザクションのイベントを送信します。)
- `TransactionMode.DEFAULT` (現在のエミッター設定を使用してイベントを送信します。)

トランザクション・モードの照会:

イベント・ソースは、エミッターが使用するトランザクション・モードを照会できます。

注: トランザクションは、J2EE コンテナ内でのみサポートされています。

現行のトランザクション・モードを照会するには、`getTransactionMode()` メソッドを使用します。

```
int transactionMode = emitter.getTransactionMode();
```

戻される値は、以下のいずれかのトランザクション・モード定数に対応する整数です。

- TransactionMode.NEW
- TransactionMode.SAME

エミッター・リソースの解放:

イベント・ソースが特定のエミッターを使用してイベントの送信を終了した場合は、そのエミッターが使用しているリソースを解放する必要があります。

エミッター・リソースを解放するには、close() メソッドを使用します。

```
emitter.close();
```

このメソッドは、エミッターが使用するすべてのリソースを解放します。

イベントのフィルター処理:

エミッターは、オプションでソースでイベントをフィルターに掛けるように構成することができます。

イベントのフィルター処理は、重要でないイベントをスクリーニングすることにより、イベントのトラフィックを削減するためのメカニズムを提供します。イベント・ソースがエミッターにイベントをサブミットするたびに、エミッターは現行のフィルター基準に対してイベントをチェックします。イベントがフィルター基準を満たしている場合、エミッターはこのイベントをイベント・サーバーに送信します。基準が満たされていない場合、エミッターはそのイベントを廃棄します。いずれの場合も、イベント・ソースは、管理者が構成したフィルター設定を変更することはできません。

エミッター・フィルターは、フィルター・プラグイン と呼ばれる別個のコンポーネントとしてインプリメントされます。Common Event Infrastructure には、デフォルト・フィルター・プラグインが含まれています。これにより、サブミットされたイベントが XPath イベント・セレクターに基づいてフィルター処理されます。異なるフィルター・メカニズムを使用する場合は、独自のフィルター・プラグインをインプリメントできます。

Common Event Infrastructure 構成では、各エミッター・ファクトリーはフィルター・ファクトリーに関連付けられています。フィルター・ファクトリーは、フィルター・プラグインのインスタンスの作成に使用されるオブジェクトです。エミッター・ファクトリーを使用してエミッターを作成すると、エミッターは、自動的に指定のフィルター・プラグインのインスタンスに関連付けられます。これにより、イベントのフィルター処理がそのエミッターにサブミットされます。

デフォルト・フィルター・プラグインによるイベントのフィルター処理:

デフォルト・エミッター・フィルター・プラグインが Common Event Infrastructure に含まれています。

Common Event Infrastructure には、デフォルト・エミッター・フィルターが含まれ、このプラグインは、どのイベントをイベント・サーバーに送信し、どのイベントを廃棄するかを定義する XPath イベント・セレクターを使用して、構成することができます。例えば、フィルター設定で、重大度が 20 (重大でない) より大きいイベントのみを送信するように指定します。

デフォルト・フィルター・プラグインを使用してイベントをフィルターに掛けるには、以下のステップに従います。

1. WebSphere 管理コンソールで、「**Common Event Infrastructure プロバイダー**」 > 「**フィルター・ファクトリー・プロファイル**」 ページに移動します。
2. 新規フィルター・ファクトリー・プロファイルを作成します。詳しくは、管理コンソールのオンライン・ヘルプを参照してください。
3. 「**フィルター構成ストリング**」 フィールドに、イベントをフィルター処理するために使用するイベントを記述する XPath イベント・セレクターを入力します。このイベント・セレクターに一致するイベントが、イベント・サーバーに送信されます。一致しないイベントは、エミッターによって廃棄されます。
4. 「**Common Event Infrastructure プロバイダー**」 > 「**エミッター・ファクトリー・プロファイル**」 ページに移動します。
5. 新規エミッター・ファクトリー・プロファイルを作成するか、既存のエミッター・ファクトリー・プロファイルに進みます。詳しくは、管理コンソールのオンライン・ヘルプを参照してください。
6. 「**フィルター・ファクトリー JNDI 名**」 フィールドで、作成した新規フィルター・ファクトリー・プロファイルの JNDI 名を入力します。

これで、イベント・ソースは、新規エミッター・ファクトリーを使用して、新規フィルター構成を使用するエミッターのインスタンスを作成することができます。エミッター・ファクトリーを使用してイベント・ソースのフィルター設定を後で調整する場合は、フィルター・ファクトリーに指定したイベント・セレクターを変更できます。

どのイベントがデフォルト・フィルター・プラグインで廃棄されるかを検出するために、トレースを使用できます。詳細は、294 ページの『*Common Event Infrastructure* 内のロギングおよびトレース』を参照してください。

注: デフォルト・フィルター・プラグインは、Apache JXPath コンポーネントを使用して、XPath イベント・セレクターを処理します。Java 2 セキュリティーが使用可能に設定されている場合は、必要なアクセスを許可するエントリーを含むようにポリシー・ファイルを変更する必要があります。

```
permission java.util.PropertyPermission "*" , "read";
permission java.io.FilePermission
    "${was.install.root}${/}java${/}jre${/}lib${/}jxpath.properties",
    "read";
```

フィルター・プラグインのインプリメント:

カスタム・フィルター・プラグインをインプリメントできます。

ユーザー独自のフィルター・エンジンをエミッター・フィルターとして使用する場合は、以下のステップに従ってカスタム・フィルター・プラグインをインプリメントできます。

1. インターフェース `com.ibm.events.filter.Filter` をインプリメントする Java クラスとしてフィルター・プラグインを作成します。このインターフェースは、以下のメソッドを定義します。

isEnabled(CommonBaseEvent)

指定されたイベントがフィルター基準を満たしているかどうかを示すブール値を返します。イベントがエミッターにサブミットされるたびに、エミッターはこのメソッドを呼び出し、サブミットされたイベントを渡します。戻り値が `true` である場合、エミッターは、パーシスタンスおよび配布用にイベントをイベント・サーバーに送信します。戻り値が `false` である場合、エミッターはイベントを廃棄します。

getMetaData()

プロバイダー名やバージョン番号などのフィルター・プラグインに関する情報を返します。

close() フィルター・プラグインによって使用されていたすべてのリソースを解放します。このメソッドは、エミッターの `close()` メソッドが呼び出されたときに呼び出されます。

2. インターフェース `com.ibm.events.filter.FilterFactory` をインプリメントするフィルター・ファクトリー・クラスを開発します。このインターフェースは、単一のメソッド `getFilter` を定義します。このメソッドは、フィルター・クラスのインスタンス (`Filter` インターフェースのインプリメンテーション) を返します。
3. フィルター・ファクトリーのインスタンスを JNDI ネーム・スペースにバインドします。初期化時に、エミッターは JNDI ルックアップを実行して、フィルター・ファクトリーにアクセスします。
4. WebSphere Process Server 管理コンソールで、エミッター・ファクトリー・プロファイルを変更するか、新規プロファイルを作成します。「**フィルター・ファクトリー JNDI 名**」フィールドで、`FilterFactory` インプリメンテーションの JNDI 名を指定します。エミッター・ファクトリー・プロファイルについての詳細は、管理コンソールのオンライン・ヘルプを参照してください。

フィルター・ファクトリーを指定するエミッター・ファクトリー・プロファイルを使用してエミッターを作成すると、新規エミッターは、フィルター・インプリメンテーションのインスタンスを使用します。これで、標準エミッター・インターフェースを使用してイベントを送信できるようになり、独自のフィルター・プラグインが使用されるようになりました。

ECSEmitter クラスを使用してのイベントの作成および取り込み:

Java Naming and Directory Interface を使用すると、イベント・ファクトリーに間接的にアクセスできます。

イベント・ファクトリーが Java Naming and Directory Interface ネーム・スペース内にバインドされている場合は、イベント・ファクトリーに間接的にアクセスできます。`com.ibm.websphere.cem.ECSEmitter` クラスを使用して、`Common Base Event` を作成し、そのイベントにデータを取り込むことができます。このクラスは、以下のメソッドを提供します。

- `createCommonBaseEvent` メソッド。このメソッドを使用する場合、指定する必要があるのは、`Common Base Event` の拡張名と状態プロパティのみです。他のすべてのプロパティは、自動的に設定されます。
- `addUserDataEvent` メソッド。このメソッドを使用すると、すべての必須プロパティが自動的に設定されます。拡張名は `ECS:UserDataEvent` に設定され、状態は `ReportSituation` に設定されます。一連のプロパティを渡すことによって、`Common Base Event` の拡張データ・エレメントを設定できます。

以下のいずれかの方法で `Common Base Event` を作成し、それにデータを取り込むことができます。

- `createCommonBaseEvent` メソッドを使用してイベントを作成し、データを取り込みます。

以下のコード・フラグメントは、新規のイベント相関範囲 `newECSID` を開始し、`createCommonBaseEvent` メソッドを使用してイベント・オブジェクトを作成します。

```
ECSEmitter myEmitter = new ECSEmitter("JNDI Emitter Factory Name", "newECSID");
CommonBaseEvent myEvent = myEmitter.createCommonBaseEvent("myEventType");
// get situation object
Situation mySituation = myEvent.getSituation();
// set situation properties
mySituation.setCategoryName("ReportSituation");
mySituation.setReportSituation("EXTERNAL", "STATUS");
// add other information to the the event
// send the event
myEmitter.sendEvent(myEvent);
```

この例では、`ECSEmitter` クラスのコンストラクター・メソッドを使用してエミッターを作成し、既存の `Common Event Infrastructure` エミッターの JNDI 名と新規イベント相関範囲の ID を渡します。

その後、新規エミッターを使用して、`getSituation` 呼び出しを用いてアクセスされる状態オブジェクトを含む `Common Base Event` を作成します。`setCategoryName` および `setReportSituation` メソッドを使用して、`ReportSituation` でイベントを送出するための状態オブジェクトに必須データを設定します。他の状態タイプを作成するには、`setCategoryName` 呼び出しで別のカテゴリ名を使用し、状態に対して別の `setter` メソッド呼び出しを使用します。

その他すべての必須情報は、ランタイム環境によって自動的に提供されます。`Common Base Event` に必須情報が明示的に設定されている場合、この情報はデフォルト情報で上書きされません。この段階で、イベントが有効になり、`sendEvent` メソッドを使用してエミッターにサブミットできます。

実際のアプリケーションでは、有益なイベントには、この例に示すよりもさらに多くの情報を含める必要があります。この例では、`Common Base Event` 仕様および `Common Event Infrastructure` で必要となる最低限のプロパティが含まれています。

- `addUserDataEvent` メソッドを使用してイベントを作成し、データを取り込みます。

以下のコード・フラグメントでは、`addUserDataEvent` メソッドを使用して現行のイベント相関範囲にイベント・オブジェクトを作成します。


```

ECSEmitter myEmitter = new ECSEmitter("JNDI Emitter Factory Name", null);
// prepare a set of user data properties
Properties myUserData = new Properties();
myUserData.setProperty("UserData1","UserDataValue1");
myUserData.setProperty("UserData2","UserDataValue2");
// create and send the event
myEmitter.addUserDataEvent(myUserData);

```

この例では、ECSEmitter クラスのコンストラクター・メソッドを使用してエミッターを作成し、既存の Common Event Infrastructure エミッターの JNDI 名を渡します。イベント相関範囲の ID は渡されていません (null)。そのため、新規イベント相関範囲は開始されません。イベント相関範囲が存在する場合は、ユーザー・データ・イベントがこの相関範囲に追加されます。

その後、ユーザー・データ・プロパティの集合が準備されます。名前と値の組がプロパティ・リストに追加されます。

例の最後のステップでは、新規エミッターの addUserDataEvent メソッドを使用して Common Base Event を作成し、送信します。新規 Common Base Event の extensionName プロパティは ECS:UserDataEvent に設定され、状態は ReportSituation に設定されます。その他すべての必須情報は、ランタイム環境によって自動的に提供されます。

イベント・コンシューマーの作成

イベント・コンシューマー とは、イベント・サーバーからイベントを受信するアプリケーションのことです。

イベント・コンシューマー は、非同期イベント通知を受け取るアプリケーションであったり、パーシスタント・データ・ストアからヒストリカル・イベント・データを照会および処理するアプリケーションであったりします。イベント・コンシューマーは、Java オブジェクトの形式でイベントを受信します。その後、CommonBaseEvent インターフェースを使用してイベント・プロパティ・データを取得するか、サポートされている別の形式 (XML など) にイベントを変換して別のアプリケーションに転送します。

イベント・コンシューマーは、以下の 2 つの方法のいずれかでイベントを受信できます。

- Java Message Service (JMS) インターフェースを使用してキューまたはトピックをサブスクライブすることにより、イベント通知を JMS メッセージとして非同期で受信できます。この方法は、新規または変更されたイベントがイベント・サーバーに到着したときに処理する必要があるイベント・コンシューマーの場合に最も効率的です。
- イベント・アクセス・インターフェースを使用してパーシスタント・データ・ストアでヒストリカル・イベントを照会することにより、要求されたイベントを同時に取得できます。この方法では、データ・ストアでヒストリカル・イベントを照会することにより、JMS によって新規イベントを受信し始める前にイベント・コンシューマーが現行状態の情報を判別することができるので、開始処理の場合に有効です。

イベント・コンシューマーは、イベントを受信する以外に、データ・ストアのイベントを変更、イベントを削除、および古いイベントをパージできます。

Java Message Service インターフェースの使用:

Java Message Service (JMS) インターフェースを使用して、JMS キューまたはトピックからイベント通知を非同期に受信するイベント・コンシューマーを開発できます。

イベント・コンシューマーは、標準 Java クラスまたはメッセージ駆動型 Bean (MDB) としてインプリメントできます。

JMS インターフェースを使用すると、標準 Java ツールおよびプログラミング・モデルを使用してイベント・コンシューマーをインプリメントできます。また、イベント・データ・ストアを直接照会する場合に起こるパフォーマンスの低下を回避することができます。イベント・コンシューマーは、Common Event Infrastructure と直接対話する代わりに、JMS 宛先 (キューおよびトピック) をサブスクライブして、イベント通知を JMS メッセージの形式で受信します。

Common Event Infrastructure は、イベントをイベント・グループに編成します。イベント・グループは、Common Event Infrastructure 構成で定義されたイベントの論理集合です。特定のイベント・コンシューマーは、通常、特定のイベント・グループのイベントのみを受信する必要があります。

各イベント・グループの構成プロファイルは、そのイベント・グループを 1 つ以上の JMS 宛先に関連付け、そのイベント・グループに関連した通知は、その JMS 宛先を介して配布されます。イベント・グループと JMS 宛先には、次のようなりレーションシップがあります。

- イベント・グループは、複数のキューに関連付けることができます。
- イベント・グループは、1 つのトピックにしか関連付けられません。(複数のイベント・コンシューマーは、同じトピックにサブスクライブすることができるため、同一のイベント・グループを複数のトピックに公開することは冗長になります)。
- JMS 宛先 (キューまたはトピック) は、通常、1 つのイベント・グループにしか関連付けられません。

イベント・グループからメッセージを受信するため、JMS コンシューマーは、適切な宛先にサブスクライブします。関連するイベント・グループと一致するイベントが、作成、変更、または削除されるごとに、通知がイベント通知を含む JMS メッセージ形式で配信されます。通知の内容は、以下のタイプによって異なります。

- 新規または変更されたイベントの場合、通知には、完全なイベント・データが含まれています。このデータは、CommonBaseEvent インスタンスに変換できます。
- 削除されたイベントの場合、通知には、削除されたイベントのグローバル・インスタンス ID が含まれています。

JMS イベント・コンシューマーは、標準 JMS インターフェースだけでなく、通知ヘルパーと呼ばれる機能とも対話します。通知ヘルパーは、Common Event Infrastructure エンティティ (イベントおよびイベント・グループ) と、それと同等の JMS エンティティ (メッセージおよび宛先) の間の変換を行います。通知ヘルパーには、次のような機能があります。

- 指定したイベント・グループに関連付けられている JMS トピックまたはキューを識別します。その後、イベント・コンシューマーで該当する宛先を使用して、サブスクリプションを作成することができます。
- 新規または変更されたイベントのイベント通知を `CommonBaseEvent` のインスタンスに変換します。
- コンシューマーにイベントのフィルタリングを提供します。各通知ヘルパーは、コンシューマーに戻す必要のあるイベントを指定するイベント・セレクターと関連付けることができます。コンシューマーが通知ヘルパーを使用してイベント通知をイベント・インスタンスに変換する場合、イベント・インスタンスが戻されるのは、それが、指定したイベント・セレクターと一致する場合に限られます。

注: 通知ヘルパーは、Apache XPath コンポーネントを使用して、XPath イベント・セレクターを処理します。Java 2 セキュリティーが使用可能になっている場合は、ポリシー・ファイルを以下のエントリーが含まれるように変更する必要があります。

```
permission java.util.PropertyPermission "*", "read";
permission java.io.FilePermission
    "${was.install.root}${/}java${/}jre${/}lib${/}jxpath.properties",
    "read";
```

メッセージ・ドリブン Bean (MDB) としてのイベント・コンシューマーの開発:

J2EE イベント・コンシューマーは、メッセージ・ドリブン Bean としてインプリメントされます。メッセージ・ドリブン Bean は、デプロイメント時に JMS 宛先および接続ファクトリーに関連付けられます。イベントを受信するには、以下のステップに従ってください。

1. 通知ヘルパーを入手します。JMS イベント・コンシューマーは、イベント・グループに関連付けられた JMS 宛先の識別、受信した JMS メッセージからイベント通知への変換、および受信したイベントのフィルター処理を実行するために通知ヘルパーを使用します。通知ヘルパーを取得するには、通知ヘルパー・ファクトリーを使用します。通知ヘルパー・ファクトリーは、JNDI ネーム・スペース内にバインドされた `NotificationHelperFactory` のインスタンスです。以下のコード・フラグメントは、通知ヘルパーを入手するために通知ヘルパー・ファクトリーを使用します。

```
// Get notification helper factory from JNDI
InitialContext context = new InitialContext();
Object notificationHelperFactoryObject =
    context.lookup("com/ibm/events/NotificationHelperFactory");
NotificationHelperFactory nhFactory = (NotificationHelperFactory)
    PortableRemoteObject.narrow(notificationHelperFactoryObject,
        NotificationHelperFactory.class);
```

```
// Create notification helper
NotificationHelper notificationHelper =
    nhFactory.getNotificationHelper();
```

2. **オプション:** イベント・セレクターを指定します。受信したイベントをフィルター処理する場合は、`setEventSelector()` メソッドを使用して通知ヘルパーにイベント・セレクターを設定できます。次に、イベント・コンシューマーは通知ヘルパーを使用して、イベント・セレクターに対して受信したイベントを確認することができます。以下のコード・フラグメントでは、重大度が 30 (警告) より大きいイベントを指定するイベント・セレクターを設定します。

```
notificationHelper.setEventSelector("CommonBaseEvent[@severity > 30]");
```

3. 受信したメッセージをイベント通知に変換します。

リスナーの `onMessage()` メソッドで、通知ヘルパーを使用して、受信した各 JMS メッセージを、イベント通知を含む配列に変換します。(イベントが通知ヘルパーで指定されたイベント・セレクターと一致しない場合、配列は空です。) イベント通知は、`EventNotification` インターフェースをインプリメントするクラスのインスタンスです。

```
public void onMessage(Message msg) {
    EventNotification[] notifications =
        notificationHelper.getEventNotifications(msg);
    // ...
}
```

4. 通知タイプを検査し、イベント・データを適切に検索します。各イベント通知には通知タイプを表すフィールドがあります(整数で、その値は、`NotificationHelper` インターフェースで定義される通知タイプ定数の 1 つです)。以下の 3 つの通知タイプが現在サポートされています。

通知タイプ	説明
<code>CREATE_EVENT_NOTIFICATION_TYPE</code>	宛先と関連するイベント・グループで新規イベントが作成されました。つまり、新規イベントが送信されたか、現在はイベント・グループ定義に一致するように既存のイベントが変更されたかのいずれかです。通知は完全なイベント・データも含んでいます。
<code>REMOVE_EVENT_NOTIFICATION_TYPE</code>	イベント・データベースに保管されたイベントが、宛先に関連したイベント・グループから除去されました。つまり、イベントがイベント・データベースから削除されたか、イベント・グループ定義に一致しないように既存のイベントが変更されたかのいずれかです。通知は、削除されたイベントのグローバル・インスタンス ID も含んでいます。
<code>UPDATE_EVENT_NOTIFICATION_TYPE</code>	イベント・データベースに保管されたイベントが、宛先に関連したイベント・グループのメンバーシップを変更しないような方法で更新されました。通知は完全なイベント・データも含んでいます。

`EventNotification` の `getNotificationType()` メソッドを使用して、各受信済み通知の通知タイプを検査します。通知タイプに基づいて、イベント・コンシューマーがさらに通知を処理する必要があるかどうか、および通知にどんな種類のイベント・データが含まれているかを判別できます。

- 通知タイプが `CREATE_EVENT_NOTIFICATION_TYPE` または `UPDATE_EVENT_NOTIFICATION_TYPE` の場合、コンシューマーは `EventNotification.getEvent()` を使用して、新規または更新済みイベントを検索できます。このメソッドは、新規または更新済みイベントの通知にのみ有効です。

- 通知が REMOVE_EVENT_NOTIFICATION_TYPE である場合、コンシューマーは EventNotification.getGlobalInstanceId() を使用して、削除済みイベントのグローバル・インスタンス ID を検索できます。このメソッドは、削除済みイベントの通知にのみ有効です。

```

for (int i = 0; i < notifications.length; i++)
{
    int notifType = notifications[i].getNotificationType();

    if(notifType == NotificationHelper.CREATE_EVENT_NOTIFICATION_TYPE)
    {
        CommonBaseEvent event = notifications[i].getEvent();
        if (event != null) {
            // process the new event
            // ...
        }
    }

    else if(notifType == NotificationHelper.UPDATE_EVENT_NOTIFICATION_TYPE)
    {
        CommonBaseEvent event = notifications[i].getEvent();
        if (event != null) {
            // process the updated event
            // ...
        }
    }

    else if(notifType == NotificationHelper.REMOVE_EVENT_NOTIFICATION_TYPE)
    {
        String eventId = notifications.[i].getGlobalInstanceId();
        // process the event deletion
        // ...
    }
}

```

JMS 宛先および接続ファクトリーを指定するデプロイメント記述子内で、メッセージ・ドリブン Bean をリスナー・ポートと関連付ける必要があります。 イベントを受信するイベント・グループに関連する宛先と接続ファクトリー（これはイベント・グループ・プロファイルに定義されている）を指定して、MDB をデプロイする前にイベント・コンシューマー向けのリスナー・ポートを作成する必要があります。

注: MDB をデプロイする際に CommonEventInfrastructure_ListenerPort リスナー・ポートを使用しないでください。このリスナー・ポートは、イベント・サーバーによって使用され、イベント・コンシューマーによる使用は意図されていません。

MDB 以外のイベント・コンシューマーの開発:

非メッセージ駆動 Bean を使用してイベント・コンシューマーを作成することもできます。

メッセージ・ドリブン Bean 以外のイベント・コンシューマーを書き込むためには、ステップに従ってください。

1. 通知ヘルパーを入手します。JMS イベント・コンシューマーは、イベント・グループに関連付けられた JMS 宛先の識別、受信した JMS メッセージからイベント通知への変換、および受信したイベントのフィルター処理を実行するために通知ヘルパーを使用します。通知ヘルパーを取得するには、通知ヘルパー・ファ

クトリーを使用します。通知ヘルパー・ファクトリーは、JNDI ネーム・スペース内にバインドされた `NotificationHelperFactory` のインスタンスです。以下のコード・フラグメントは、通知ヘルパーを入手するために通知ヘルパー・ファクトリーを使用します。

```
// Get notification helper factory from JNDI
InitialContext context = new InitialContext();
Object notificationHelperFactoryObject =
    context.lookup("com/ibm/events/NotificationHelperFactory");
NotificationHelperFactory nhFactory = (NotificationHelperFactory)
    PortableRemoteObject.narrow(notificationHelperFactoryObject,
        NotificationHelperFactory.class);
```

```
// Create notification helper
NotificationHelper notificationHelper =
    nhFactory.getNotificationHelper();
```

2. **オプション:** イベント・セレクターを指定します。受信したイベントをフィルター処理する場合は、`setEventSelector()` メソッドを使用して通知ヘルパーにイベント・セレクターを設定できます。次に、イベント・コンシューマーは通知ヘルパーを使用して、イベント・セレクターに対して受信したイベントを確認することができます。以下のコード・フラグメントでは、重大度が 30 (警告) より大きいイベントを指定するイベント・セレクターを設定します。

```
notificationHelper.setEventSelector("CommonBaseEvent[@severity > 30]");
```

3. 通知ヘルパーを使用して JMS 宛先を検索し、以下にサブスクライブします。

各イベント・グループは、単一の JMS トピックおよび任意の数の JMS キューに関連することが可能です。通知ヘルパーを照会して、特定のイベント・グループにどの宛先が関連しているかを検出することができます。イベント・グループに関連付けられているトピックを検索するには、イベント・グループの名前を指定した `NotificationHelper` の `getJmsTopic(String)` メソッドを使用します。

```
MessagePort msgPort = notificationHelper.getJmsTopic("critical_events");
```

イベント・グループに関連付けられているキューを検索するには、`getJmsQueues(String)` メソッドを使用します。

```
MessagePort[] msgPorts = notificationHelper.getJmsQueues("critical_events");
```

戻されるオブジェクトは、JMS トピックを表す単一 `MessagePort` オブジェクトか、JMS キューを表す `MessagePort` オブジェクトの配列です。`MessagePort` インスタンスは、宛先の JNDI 名とその接続ファクトリーを含むラッパー・オブジェクトです。

4. 宛先に接続します。`MessagePort` の `getter` メソッドを使用して、宛先および接続ファクトリーの JNDI 名をリトリブします。次に、標準 JMS インターフェースを使用して、宛先に接続することができます。以下のコード・フラグメントは、JMS トピックにサブスクライブします。

```
String connectionFactoryName = msgPort.getConnectionFactoryJndiName();
String destinationName = msgPort.getDestinationJndiName();
```

```
// create connection and session
ConnectionFactory connectionFactory =
    (ConnectionFactory) context.lookup(connectionFactoryName);
Connection connection = connectionFactory.createConnection();
Session session = connection.createSession(false,
    Session.CLIENT_ACKNOWLEDGE);
```



```
// Create consumer and register listener
Topic topic = (Topic) context.lookup(destinationName);
MessageConsumer consumer = session.createConsumer(topic);
consumer.setMessageListener(this);
connection.start();
```

- 受信したメッセージをイベント通知に変換します。

リスナーの `onMessage()` メソッドで、通知ヘルパーを使用して、受信した各 JMS メッセージを、イベント通知を含む配列に変換します。(イベントが通知ヘルパーで指定されたイベント・セレクターと一致しない場合、配列は空です。) イベント通知は、`EventNotification` インターフェースをインプリメントするクラスのインスタンスです。

```
public void onMessage(Message msg) {
    EventNotification[] notifications =
        notificationHelper.getEventNotifications(msg);
    // ...
}
```

- 通知タイプを検査し、イベント・データを適切に検索します。各イベント通知には通知タイプを表すフィールドがあります(整数で、その値は、`NotificationHelper` インターフェースで定義される通知タイプ定数の 1 つです)。以下の 3 つの通知タイプが現在サポートされています。

通知タイプ	説明
<code>CREATE_EVENT_NOTIFICATION_TYPE</code>	宛先と関連するイベント・グループで新規イベントが作成されました。つまり、新規イベントが送信されたか、現在はイベント・グループ定義に一致するように既存のイベントが変更されたかのいずれかです。通知は完全なイベント・データも含んでいます。
<code>REMOVE_EVENT_NOTIFICATION_TYPE</code>	イベント・データベースに保管されたイベントが、宛先に関連したイベント・グループから除去されました。つまり、イベントがイベント・データベースから削除されたか、イベント・グループ定義に一致しないように既存のイベントが変更されたかのいずれかです。通知は、削除されたイベントのグローバル・インスタンス ID も含んでいます。
<code>UPDATE_EVENT_NOTIFICATION_TYPE</code>	イベント・データベースに保管されたイベントが、宛先に関連したイベント・グループのメンバーシップを変更しないような方法で更新されました。通知は完全なイベント・データも含んでいます。

`EventNotification` の `getNotificationType()` メソッドを使用して、各受信済み通知の通知タイプを検査します。通知タイプに基づいて、イベント・コンシューマーがさらに通知を処理する必要があるかどうか、および通知にどんな種類のイベント・データが含まれているかを判別できます。

- 通知タイプが `CREATE_EVENT_NOTIFICATION_TYPE` または `UPDATE_EVENT_NOTIFICATION_TYPE` の場合、コンシューマーは

EventNotification.getEvent() を使用して、新規または更新済みイベントを検索できます。このメソッドは、新規または更新済みイベントの通知にのみ有効です。

- 通知が REMOVE_EVENT_NOTIFICATION_TYPE である場合、コンシューマーは EventNotification.getGlobalInstanceId() を使用して、削除済みイベントのグローバル・インスタンス ID を検索できます。このメソッドは、削除済みイベントの通知にのみ有効です。

```
for (int i = 0; i < notifications.length; i++)
{
    int notifType = notifications[i].getNotificationType();

    if(notifType == NotificationHelper.CREATE_EVENT_NOTIFICATION_TYPE)
    {
        CommonBaseEvent event = notifications[i].getEvent();
        if (event != null) {
            // process the new event
            // ...
        }
    }

    else if(notifType == NotificationHelper.UPDATE_EVENT_NOTIFICATION_TYPE)
    {
        CommonBaseEvent event = notifications[i].getEvent();
        if (event != null) {
            // process the updated event
            // ...
        }
    }

    else if(notifType == NotificationHelper.REMOVE_EVENT_NOTIFICATION_TYPE)
    {
        String eventId = notifications.[i].getGlobalInstanceId();
        // process the event deletion
        // ...
    }
}
```

イベント・サーバーからのイベントの照会:

イベント・コンシューマーは、イベント・サーバーの照会によって、パーシスタント・データ・ストアから同期的に履歴イベントをリトリブすることができます。

パーシスタント・データ・ストアは、データ・ストア・プラグイン と呼ばれる分離コンポーネントとして実装されています。Common Event Infrastructure は、XPath 構文のサブセットを基にしたイベント照会をサポートするデフォルトのデータ・ストア・プラグインを含んでいます。異なるデータ・ストアを使用したい場合は、ユーザー独自のデータ・ストア・プラグインを実装することができます。

イベント・サーバーを照会するには、イベント・アクセス・インターフェースを使用します。

1. イベント・アクセス Bean を作成します。

この Bean は、Java 2 Platform, Enterprise Edition (J2EE) の Stateless Session Bean です。この Bean インターフェースは、イベント・サーバーを照会するためのメソッドを提供します。イベント・コンシューマーは、すべての同期イベント照会のためのイベント・アクセス Bean のインスタンスを使用します。

2. イベントの照会

以下の方法でイベントの照会ができます。

- グローバル・インスタンス ID を指定して、特定の単一イベントをリトリブします。
- イベント・グループを指定し、そのイベント・グループに関連付けられているイベントをリトリブします。オプションで、追加のイベント・セレクターを指定することによって、照会を詳細化できます。これにより、イベント・グループとイベント・セレクターの両方に一致するイベントのみをリトリブします。
- 既知のイベントおよび関連タイプを指定し、既知のイベントに関連したイベントをリトリブします。
- イベント照会およびページ。

イベント・アクセス Bean の作成:

イベント・アクセスの取得は、イベント・アクセス Bean のインスタンスの作成から開始します。

イベント・アクセス・インターフェースは、Enterprise JavaBeans アーキテクチャーを使用する Stateless Session Bean としてインプリメントされます。イベント・アクセス・インターフェースを使用してイベント・サーバーを照会する場合、イベント・ソースは、最初にイベント・アクセス Session Bean のインスタンスを作成する必要があります。イベント・アクセス Bean は、ローカルまたはリモートのいずれにもすることができます。

イベント・アクセス Session Bean のインスタンスを作成するには、適切なホーム・インターフェース (com.ibm.events.access.EventAccessHome または com.ibm.events.access.EventAccessLocalHome) を使用します。

```
// use home interface to create remote event access bean
InitialContext context = new InitialContext();
Object eventAccessHomeObj = context.lookup("ejb/com/ibm/events/access/EventAccess");
EventAccessHome eventAccessHome = (EventAccessHome)
    PortableRemoteObject.narrow(eventAccessHomeObj,
        EventAccessHome.class);
eventAccess = (EventAccess) eventAccessHome.create();
```

グローバル・インスタンス ID を使用したイベントの照会:

基本キーによりイベントを照会できます。

Common Base Event 仕様は、イベントを識別するための 1 次キーとして使用できる globalInstanceId イベント・プロパティを定義します。このプロパティのコンテンツはグローバル固有 ID であり、アプリケーションまたはエミッターによって生成されます。Common Base Event 仕様はオプションとして globalInstanceId プロパティを定義しますが、イベント・エミッターは、まだ ID を持っていないイベントに ID を自動的に割り当てます。

リトリブするイベントの globalInstanceId プロパティを照会すると、イベント・サーバーから特定の単一イベントをリトリブできます。この照会は、(イベントがイベント・データベース内に保管されていることを確認するために) テストを行ったり、以前受け取ったイベントに関連付けられているイベントをリトリブする場合に役立ちます。

グローバル・インスタンス ID を使用してイベントを照会するには、イベント・アクセス Bean の `queryEventByGlobalInstanceId` メソッドを使用します。

1. **オプション:** イベント・アクセス Bean を作成します。
2. EventAccess Bean の `queryEventByGlobalInstanceId(String)` メソッドを呼び出します。ここでは、リトリートするイベントのグローバル・インスタンス ID を指定します。

```
CommonBaseEvent event = eventAccess.queryEventByGlobalInstanceId(eventId);
```

戻されたオブジェクトは、指定されたグローバル・インスタンス ID を持つイベントです。永続的データ・ストア内に一致するイベントがない場合、戻されたオブジェクトはヌルです。

イベント・グループを使用したイベントの照会:

指定したイベントにより簡単にアクセスするには、そのイベントをイベント・グループに関連付けます。

イベントは、1 つ以上のイベント・グループに関連付けることができます。イベント・グループは、特定のイベント・セレクターと一致するイベントの論理グループです。イベント・グループは、イベント・インフラストラクチャー構成で定義されます。イベント・グループについて詳しくは、『デフォルト構成』を参照してください。

イベント・アクセス・インターフェースを使用すると、指定したイベント・グループに属するイベントをリトリートすることができます。追加のイベント・セレクターを指定すると、照会結果を制限できます。また、リトリートせずにイベントを照会することもできます。

次の方法で、イベント・グループを照会できます。

イベント・グループからの限定数のイベント照会:

イベント・グループからの限定数のイベントを照会することができます。

イベント・グループから限定数のイベントを照会するには、EventAccess Bean の `queryEventsByEventGroup(String, String, Boolean, int)` メソッドを使用します。

1. イベント・アクセス Bean を作成する必要がある場合は、326 ページの『イベント・アクセス Bean の作成』を参照してください。
2. EventAccess.queryEventsByEventGroup(String, String, boolean, int) メソッドを呼び出します。

```
CommonBaseEvent[] events = eventAccess.queryEventsByEventGroup(eventGroup,
                                                                    eventSelector,
                                                                    ascendingOrder,
                                                                    maxEvents);
```

このメソッドのパラメーターは、以下のとおりです。

eventGroup

イベント・グループの名前を含む文字列。このグループ内のイベントが照会されます。これは、イベント・インフラストラクチャー構成で定義されている既存のイベント・グループの名前である必要があります。

eventSelector

照会を詳細化するイベント・セレクターを含む文字列 (オプション)。この照会を行うと、指定したイベント・グループと追加のイベント・セレクターの両方にマッチするイベントのみ返されます。 イベント・セレクターは、XPath 式の形式で指定されています (詳しくは、332 ページの『イベント・セレクターの書き込み』を参照してください)。追加のイベント・セレクターを指定しない場合は、このパラメーターをヌルにします。

ascendingOrder

返されるイベントを、creationTime プロパティの値に従い昇順にソートするか降順にソートするか指定するブール値。このパラメーターを *true* に設定すると、イベントは昇順 (発生順) にソートされます。また *false* に設定すると、降順 (新しい順) にソートされます。

maxEvents

返されるイベントの最大数を指定する整数。

返されるオブジェクトは、指定したイベント・グループ内のイベントを含む配列です。

注: マッチするイベントの数が、データ・ストア・プロファイルで定義されている照会しきい値を超える場合、QueryThresholdExceededException 例外がスローされます。デフォルトの照会しきい値は 100 000 です。

以下のコードの断片では、*critical_hosts* という名前のイベント・グループに属する、重大度が 30 (警告) より大きいイベントがすべて返されます。ただし、返されるマッチング・イベントの最大数は 5000 よりも少なくなります。

```
CommonBaseEvent[] events =
    eventAccess.queryByEventGroup("critical_hosts",
        "CommonBaseEvent[@severity > 30]",
        true,
        5000);
```

イベント・グループからのすべてのイベントの照会:

イベント・グループからすべてのイベントを照会するためには、EventAccess Bean の queryEventsByEventGroup(String, String, boolean) メソッドを使用します。

1. イベント・アクセス Bean を作成する必要がある場合は、326 ページの『イベント・アクセス Bean の作成』を参照してください。
2. EventAccess.queryEventsByEventGroup(String, String, boolean) メソッドを呼び出します。

```
CommonBaseEvent[] events = eventAccess.queryEventsByEventGroup(eventGroup,
                                                                eventSelector,
                                                                ascendingOrder);
```

このメソッドのパラメーターは、以下のとおりです。

eventGroup

イベント・グループの名前を含む文字列。このグループ内のイベントが照会されます。これは、イベント・インフラストラクチャー構成で定義されている既存のイベント・グループの名前である必要があります。

eventSelector

照会を詳細化するイベント・セレクターを含む文字列 (オプション)。この照会を行うと、指定したイベント・グループと追加のイベント・セレクターの両方にマッチするイベントのみ返されます。 イベント・セレクターは、XPath 式の形式で指定されています (詳しくは、332 ページの『イベント・セレクターの書き込み』を参照してください)。追加のイベント・セレクターを指定しない場合は、このパラメーターをヌルにします。

ascendingOrder

返されるイベントを、creationTime プロパティの値に従い昇順にソートするか降順にソートするか指定するブール値。このパラメーターを *true* に設定すると、イベントは昇順 (発生順) にソートされます。また *false* に設定すると、降順 (新しい順) にソートされます。

返されるオブジェクトは、指定したイベント・グループ内のイベントを含む配列です。

注: マッチするイベントの数が、データ・ストア・プロファイルで定義されている照会しきい値を超える場合、QueryThresholdExceededException 例外がスローされます。デフォルトの照会しきい値は 100 000 です。

以下のコードの断片では、*critical_hosts* という名前のイベント・グループに属する、重大度が 30 (警告) より大きいイベントがすべて返されます。

```
CommonBaseEvent[] events =
    eventAccess.queryByEventGroup("critical_hosts",
                                  "CommonBaseEvent[@severity > 30]",
                                  true);
```

イベント・グループ内のイベントの存在の照会:

イベントが存在しているかどうかを、そのイベントを取得することなく判別できません。

状況によっては、現実にイベントをリトリブすることなく、特定のイベント・グループ内にイベントが存在するかどうかを検索したい場合があります。この場合は、イベント・アクセス Bean の *eventExists()* メソッドを使用します。

1. イベント・アクセス Bean を作成する必要がある場合は、326 ページの『イベント・アクセス Bean の作成』を参照してください。
2. EventAccess Bean の *eventExists(String, String)* メソッドを呼び出します。

```
boolean hasEvents = eventAccess.eventExists(eventGroup,
                                             eventSelector);
```

このメソッドのパラメーターは、以下のとおりです。

eventGroup

イベント・グループの名前を含む文字列。このイベント・グループ内にイベントが存在しているかどうかチェックされます。これは、イベント・インフラストラクチャー構成で定義されている既存のイベント・グループの名前である必要があります。

eventSelector

照会を詳細化するイベント・セレクターを含む文字列 (オプション)。この照

会を行うと、指定したイベント・グループと追加のイベント・セレクターの両方にマッチするイベントのみチェックされます。 イベント・セレクターは、XPath 式の形式で指定されています (詳しくは、332 ページの『イベント・セレクターの書き込み』を参照してください)。追加のイベント・セレクターを指定しない場合は、このパラメーターをヌルにします。

戻された布尔・オブジェクトは、指定したイベント・グループおよびイベント・セレクターと一致するイベントが存在すれば、true に等しく、存在しなければ、false に等しくなります。

以下のコードの断片は、critical_hosts という名前のイベント・グループ内にイベントが存在しているかどうかチェックし、存在している場合はそのイベントを取得します。

```
if (eventAccess.eventExists("critical_hosts",null)) {
    CommonBaseEvent[] events =
        eventAccess.queryByEventGroup("critical_hosts",
                                     null,
                                     true);
}
```

関連タイプによるイベントの照会:

関連タイプによりイベントを照会できます。

Common Base Event 仕様は、イベント間のリレーションシップを確立するプロパティーを定義します。 associatedEvents プロパティーは、AssociatedEvent タイプの 1 つ以上のサブエレメント (個々が関連したイベントを表す) を含む複合エレメントです。各 AssociatedEvent エレメントには、関連のタイプおよび関連を確立したアプリケーションを識別するサブエレメントが含まれます。関連タイプの例として、CausedBy や Correlated があります。

既知のイベントのグローバル・インスタンス ID および関連タイプを指定することで、指定された関連を満たすイベントをリトリブすることができます。関連タイプでイベントを照会するために、EventAccess.queryEventsByAssociation(String, String) メソッドを使用します。

1. **オプション:** イベント・アクセス Bean を作成します。
2. EventAccess.queryEventsByAssociation(String, String) メソッドを呼び出します。

```
CommonBaseEvent[] events = eventAccess.queryEventsByAssociation(associationType,
                                                                eventId);
```

このメソッドのパラメーターは、以下のとおりです。

associationType

関連のタイプ。これは、associationEngineInfo プロパティーで指定された関連タイプの名前です。

eventId

既知のイベントのグローバル・インスタンス ID。

戻されるオブジェクトは、既知のイベントとの関連が指定したタイプであるイベントを含む配列です。照会時にイベント・データベースに存在しているイベントだけが戻されます (関連イベントはデータベースからページされます)。

以下のコード・フラグメントでは、イベント・データベースから、既知のイベントとの間に `CausedBy` 関連を持つすべてのイベントを戻します。

```
String eventId = causeEvent.getGlobalInstanceId();
CommonBaseEvent[] resultEvents = eventAccess.queryEventsByAssociation("CausedBy",
                                                                    eventId);
```

データ・ストアからのイベントの削除:

イベント・コンシューマーまたは管理ツールを使用すると、イベント・アクセス・インターフェースを使用してデータ・ストアからイベントを削除できます。

データ・ストアからすべてのイベントを削除するか、またはイベント・グループかイベント・セレクター (あるいはその両方) のみを削除することができます。

注: WebSphere セキュリティーが使用可能に設定されている場合、アプリケーション・ユーザーID を `eventAdministrator` ロールにマップして、イベントを削除する必要があります。

データ・ストアからイベントを削除するには、イベント・アクセス Bean の `purgeEvents()` メソッドを使用します。

```
int purged = eventAccess.purgeEvents(eventGroup,
                                    eventSelector,
                                    transactionSize);
```

パラメーターは、以下のとおりです。

eventGroup

削除するイベントを含むイベント・グループの名前を含むストリング。これは、イベント・インフラストラクチャー構成で定義されている既存のイベント・グループの名前である必要があります。イベント・グループを指定しない場合は、このパラメーターをヌルにします。

eventSelector

削除するイベントを識別するオプションのイベント・セレクターを含むストリング。イベント・セレクターは、XPath 式の形式で指定されています (詳しくは、332 ページの『イベント・セレクターの書き込み』を参照してください)。イベント・セレクターを指定しない場合は、このパラメーターをヌルにします。

transactionSize

単一データベース・トランザクションで削除するイベントの数を指定するゼロ以外の整数。ほとんどの場合、`EventAccess` インターフェースで定義される定数 `DEFAULT_PURGE_TRANSACTION_SIZE` を使用できます。

`purgeEvents()` メソッドは、指定したすべての基準と一致するすべてのイベントを削除します。*eventGroup* および *eventSelector* パラメーターの両方がヌルの場合は、データ・ストア内のすべてのイベントが削除されます。削除操作が開始された後に着信するイベントは、ページされません。戻される値は、削除されたイベントの数を示す整数です。

注: *transactionSize* パラメーターの値がデータ・ストア・プロファイルで定義されている最大ページ・トランザクション・サイズを超過する場合は、`PurgeThresholdExceededException` 例外がスローされ、イベントは削除されません。デフォルトの最大ページ・トランザクション・サイズは 100 000 です。

イベント・セレクターの書き込み:

イベント・セレクターは、イベントの集合を定義します。

イベント・セレクターは、プロパティ・データ (属性またはサブエレメント) に基づいて一連のイベントを定義する正規表現です。例えば、イベント・セレクターは、重大度が 30 より大きい (警告) 特定のホストのすべてのイベントを指定します。イベント・セレクターを使用して、イベント・グループの定義、フィルター基準の指定、およびイベント・サーバーの照会を行います。

Common Base Event 仕様は XML に基づいているため、イベント・セレクターは、XPath 構文のサブセットを使用して作成されます。イベント・セレクターに使用できる特定の構文は、次の表に要約を示すように、そのイベント・セレクターの使用方法によって異なります。

イベント・セレクターの目的	構文
イベント・グループの定義	デフォルトのデータ・ストア・プラグインによってサポートされる XPath サブセットに限定されます。
イベント・アクセス・インターフェースを介したイベントの照会およびページ	デフォルトのデータ・ストア・プラグインによってサポートされる XPath サブセットに限定されます。
エミッター・フィルター構成	任意の有効な XPath
通知ヘルパー・インターフェースを介したサブスクリプション	任意の有効な XPath

注: デフォルトのデータ・ストア・プラグインは XPath 構文のサブセットを使用します。ただし、異なるデータ・ストア・プラグインが使用されている場合は、異なる XPath のサブセットをサポートすることもあります。イベント・グループ定義とイベント・アクセス・インターフェース用に作成するイベント・セレクターは、データ・ストア・プラグインがサポートする構文を使用する必要があります。

XPath イベント・セレクターの作成

XPath は、XML 文書のパーツを識別するために使用する標準言語です。詳しくは、<http://www.w3.org/TR/xpath> の XPath 仕様を参照してください。

属性値を指定する単純な XPath イベント・セレクターの形式は、以下のとおりです。

```
CommonBaseEvent[@attribute = value]
```

value は、数値または単一引用符または二重引用符で囲まれたストリングのいずれかです。

次のように、サブエレメントの属性を指定することもできます。

```
CommonBaseEvent[/subelement/@attribute = value]
```

XPath 演算子を使用する場合は、以下の一般規則に従ってください。

- 比較演算子を使用して XML dateTime 値を比較すると、時差を認識する論理比較が実行されます。
- 論理演算子および関数名は、常に小文字を使用して指定する必要があります (例えば、AND ではなく and)。
- 演算子は、前後の属性名および値から空白文字を使用して区切る必要があります (@severity>30 ではなく @severity > 30)。
- 括弧を使用して演算子優先順位を変更することができます。

以下は、有効な XPath イベント・セレクターの例です。

XPath イベント・セレクター	セレクターの定義
CommonBaseEvent[@extensionName = 'ApplicationStarted']	extensionName 属性が ApplicationStarted であるすべてのイベント
CommonBaseEvent[sourceComponentId/@location = "server1"]	location 属性が server1 である sourceComponentId エレメントを含むすべてのイベント
CommonBaseEvent[@severity]	値にかかわらず severity 属性を持つすべてのイベント
CommonBaseEvent[@creationTime < '2003-12-10T12:00:00-05:00' and @severity > 30]	2003 年 12 月 10 日米東部標準時正午より前に作成され、重大度が 30 (警告) より大きいすべてのイベント
CommonBaseEvent[contains(@msg, 'disk full')]	msg 属性内に出現する、句 disk full を持つすべてのイベント
CommonBaseEvent[(@severity = 30 or @severity = 50) and @priority = 100]	severity 属性が 30 または 50 に等しく、priority 属性が 100 に等しいすべてのイベント

デフォルト・データ・ストア・プラグイン用のイベント・セレクターの作成

イベント・セレクターを使用してイベント・グループを定義したり、パーシスタント・データ・ストアを照会したりする場合は、デフォルト・データ・ストア・プラグインの制約事項に従います。これらの制約事項は、以下のとおりです。

- イベント・プロパティは、演算子または XPath 関数の左辺にのみ指定できません。演算子の右辺の値は、リテラル値でなければなりません。次の例は、無効なイベント・セレクターを示しています。

```
CommonBaseEvent[30 < @priority and
contains('this message', @msg)]
```

代わりに、この例は以下のように書き直すことができます。

```
CommonBaseEvent[@priority > 30 and
contains(@msg, 'this message')]
```

- サポートされているのは、以下の XPath 関数のみです。
 - contains
 - starts-with
 - false
 - true
 - not
- 共用体演算子 (|) はサポートされていません。

- イベント・セレクターは、以下の形式にする必要があります。

```
CommonBaseEvent[predicate_expression]
```

`CommonBaseEvent` エlementと関連付けられるのは、単一の述部式のみです。スタック述部はサポートされていません (例えば、

```
CommonBaseEvent[@extensionName = "server_down"][@severity = 10])。
```

- 述部と関連付けられるのは、ロケーション・パスの最後のステップのみです。次の例は、無効なイベント・セレクターを示しています。

```
CommonBaseEvent[contextDataElement[@contextValue = "myContextValue"]
/@contextId = "myContextId"]
```

代わりに、この例は以下のように書き直すことができます。

```
CommonBaseEvent[contextDataElement[@contextValue = "myContextValue"
and @contextId = "myContextId"]]
```

- イベント・セレクターが、異なるレベルの XML 包含関係の階層にある拡張データ・Elementのプロパティを参照する場合、これらのElementは、レベルごとにグループ化されなければなりません。以下の例は有効なイベント・セレクターではありません。理由は、`extendedDataElements` の `type` および `value` 属性 (両方ともトップレベル) への参照が分離されているからです。

```
CommonBaseEvent[extendedDataElements[@type = 'int' and
children/@type = 'intArray' and
children/@name = 'myName' and
@value = 10]]
```

代わりに、この例は、トップレベルと第 2 レベルの属性をまとめてグループ化し、以下のように書き直すことができます。

```
CommonBaseEvent[extendedDataElements[@type = 'int' and
@value = 10 and
children/@type = 'intArray' and
children/@type = 'myName']]
```

- ノード索引はサポートされていません (例えば、`CommonBaseEvent[extendedDataElements[1]]`)。
- ワイルドカード文字はサポートされていません (例えば、`CommonBaseEvent[extendedDataElements/*/children/values = "text"]`)。
- 拡張データ・Elementの `values` プロパティを参照する場合は、値だけでなくプロパティの型も指定する必要があります。

```
CommonBaseEvent[extendedDataElements[values = "myVal"
and @type = "string"]]
```

比較を括弧でグループ化することによって、1 つの複合式内で複数の比較に使用する型を指定できます。

```
CommonBaseEvent[extendedDataElements[(values = "myVal" or
values = "yourVal") and
@type = "string"]]
```

この例では、`type` 式は括弧で囲んだ複合式の両方の辺に適用されます。括弧内で異なる `type` 式を指定することによってこの式をオーバーライドすることはできません。

また、starts-with または contains 関数を使用して、複数の関連した型をグループ化することもできます。例えば、以下の式は string 型または stringArray 型のプロパティーに一致します。

```
CommonBaseEvent[extendedDataElements[values = "myVal" and  
starts-with(@type, 'string')]]
```

イベント・カタログ・アプリケーションの作成

イベント・カタログ は、イベント・メタデータのリポジトリです。

event catalog メタデータは、イベント定義から構成されています。イベント定義は、イベントのクラスとそれらに許可された内容を記述するものです。(これは、306 ページの『イベント・インスタンス・メタデータへのアクセス』で説明されている Eclipse Modeling Framework のインターフェースを使用してアクセスできるイベント・インスタンス・メタデータとは区別されます。) アプリケーションは、イベント・カタログを使用してエンタープライズ固有のイベント定義を管理できますが、独自の検証ロジックをインプリメントしてイベントを確実にこれらの定義に準拠させる必要があります。

Common Base Event 仕様に従って定義されたイベントは、拡張名 (*extensionName* 属性の値) に基づいてイベント・クラスに分類することができます。イベント・カタログを使用すれば、Common Base Event の他のプロパティーに許可される値の他に、そのクラスのイベントが含むことのできる拡張データ・エレメントを指定することにより、特定のイベント・クラスに許可されるコンテンツを定義できます。イベント定義は、Common Base Event 仕様で定める制約外のイベント・コンテンツに対する制約を定義します。

イベント定義は、階層形式で定義されており、親イベントの定義を継承します。単一のルート・イベント定義である *event* は、Common Base Event 仕様の基本要件を定義します。他のすべてのイベント定義はこのルート定義から継承します。デフォルトでは、このルート・イベント定義は、イベント・カタログ通知イベントのイベント定義と共に、イベント・カタログに自動的にインストールされます (詳しくは、340 ページの『変更の通知』を参照してください)。

注: 現在、イベント定義は、Common Base Event 仕様を完全に記述するために必要な制約の形式すべてをサポートしているわけではありません (例えば、*globalInstanceId* プロパティーの先頭は英字にする必要があるという要件)。このため、イベントが *event* イベント定義に準拠していても、イベント・エミッターによる妥当性検査にパスしない可能性があります。

イベント・カタログ・インターフェースを使用することによって、イベント定義を作成、削除および照会できます。(一度作成されると、イベント定義は変更できません。) また、イベント定義を XML フォーマットでインポートおよびエクスポートすることに加えて、既存のイベント定義を読み取り可能な形式でリストすることもできます。

イベント定義:

イベント定義は、いくつかの種類の情報で構成されています。

イベント定義 には、いくつかの種類情報が含まれます。

名前 イベント定義の名前で、定義によって記述されるイベントの拡張名と同じです。特定の拡張名を持つすべてのイベントは同じイベント定義を共有します。

親 親イベント定義の名前。任意のイベント定義 (ルート定義 `event` を除く) には、親イベント定義があり、そこからイベント定義はプロパティの記述および拡張データ・エレメントの記述を継承します (ただし、継承されたデータのいくつかの要素はオーバーライドできます)。親はイベント・カタログに存在している任意の有効なイベント定義であることができます。

プロパティの記述

イベント定義の、許可された `Common Base Event` プロパティの記述。プロパティの記述は、`Common Base Event` 仕様でシンプル・タイプ (複合サブエレメントのプロパティを含む) と定義された任意のプロパティを記述できます。

拡張データ・エレメントの記述

イベント定義についての、許可された拡張データ・エレメントの記述です。拡張データ・エレメントの説明は、拡張データ・エレメントの名前および型を定義します。また、この説明は、デフォルト値、許可される拡張データ・エレメントの数、および子拡張データ・エレメントの説明も定義することができます。

XML 文書で表されるイベント定義は、以下のような一般的な形式を取ります。

```
<eventDefinition name="eventDefinitionName"
                 parent="parentEventDefinitionName">
  <property name="propertyName" ... />
  <extendedDataElement name="extendedDataElementName"
                      type="type" ... />
</eventDefinition>
```

プロパティの記述:

プロパティ記述は、`Common Base Event` 仕様で定義されています。

プロパティ記述は、イベントに含めることのできるプロパティを記述します。これは、`Common Base Event` 仕様によって単純型として定義されている、任意のプロパティにすることができます。プロパティの説明は、`msgDataElement` などの複合プロパティを記述することができません。ただし、複合プロパティの子であるシンプル・プロパティを記述することができます。イベント定義には、任意の数 (ゼロを含む) のプロパティ記述を含めることができます。

プロパティ記述には、次のようなフィールドが含まれています。

name プロパティの名前。これは、`CommonBaseEvent` エレメントの属性、または `CommonBaseEvent` の複雑なサブエレメントの属性の名前である必要があります。例として、`severity`、`priority`、および `globalInstanceId` を挙げることができます。

path プロパティが `CommonBaseEvent` の属性ではない場合、プロパティへのパスを指定する XPath ロケーション・パス。path は、記述されているプロパティの親プロパティを識別します。以下に例を挙げます。

- `severity` などの `CommonBaseEvent` のプロパティを記述する場合、path を指定しない。ヌル・パスで、最上位のプロパティが指定されます。

- `CommonBaseEvent` の複合プロパティである `msgDataElement` のプロパティを記述するには、`path msgDataElement` を指定します。
- `msgDataElement` の複合プロパティである `msgHelp` のプロパティを記述するには、`path msgDataElement/msgHelp` を指定します。

パスでは、繰り返されるプロパティの特定インスタンスを記述することもできます。例えば、イベント定義がいくつかの `contextDataElements` プロパティについて記述している場合、`businessContext` という名前のプロパティを指定し、`path contextDataElements[@name='businessContext']` を指定することができます。

defaultValue

プロパティのデフォルト値。デフォルト値は、必要なプロパティが欠落しているイベントのコンテンツ完了中に使用される値です。(このため、プロパティの説明を要求し、それによってデフォルト値を定義することには意味があります。) このフィールドはオプションです。

required

プロパティが必須か、またはオプションかを指定するブール値。このフィールドが `true` であれば、そのプロパティは必須です。このフィールドはオプションです。このフィールドに値が指定されていない場合、プロパティはオプションであると想定されます。

permittedValue

プロパティの許可値。イベント定義で 1 つのプロパティに対して特定の値のみが許可されている場合、それぞれの値は、プロパティの説明の `permittedValue` フィールドによって表されます。プロパティ記述には、任意の数の許可値を含めることができます。このフィールドはオプションで、`minValue` または `maxValue` フィールドが指定されている場合は指定してはいけません。

minValue

maxValue

プロパティの最小許可値および最大許可値。イベント定義でプロパティの値の範囲が許可されている場合、これらのフィールドは、この範囲の上限と下限を定義します。`minValue` のみを指定した場合、許可される範囲に上限はありません。同様に、`maxValue` のみを指定した場合、許可される範囲に下限はありません。これらのフィールドはオプションで、`permittedValue` フィールドが指定されている場合は指定する必要はありません。

拡張データ・エレメントの記述:

拡張されたデータ・エレメントは、イベント定義に組み込まれる可能性がある情報の 1 つのタイプです。

拡張データ・エレメントの記述は、特定のイベント・クラスのイベントが含むことのできる、拡張データ・エレメントを記述します。イベント定義は、任意の数の拡張データ・エレメントの記述 (ゼロを含む) を含むことができます。

拡張データ・エレメントの記述には以下のフィールドが含まれています。

name 拡張データ・エレメントの名前。エレメントの `名前` 属性の値を定義します。

type 拡張データ・エレメントのデータ型。エレメントの型 属性の値を定義します。これは、サポートされている以下のデータ・タイプのいずれかである必要があります。

- noValue
- byte
- short
- int
- long
- float
- double
- string
- dateTime
- boolean
- byteArray
- shortArray
- intArray
- longArray
- floatArray
- doubleArray
- stringArray
- dateTimeArray
- booleanArray
- hexBinary

defaultValue

拡張データ・エレメントのデフォルト値、または、タイプが配列の場合は複数のデフォルトの値。デフォルト値は、必要な拡張データ・エレメントが欠落しているイベントのコンテンツ完了中に使用される値を表しています。このフィールドはオプションです。

minOccurs

表示されなければならない拡張データ・エレメントのインスタンスの最小数です。このフィールドはオプションです。デフォルト値は 1 です。

maxOccurs

表示可能な拡張データ・エレメントのインスタンスの最大数です。このフィールドはオプションです。デフォルト値は 1 です。

注: 現行の Common Base Event 仕様は、イベント包含関係の階層の各レベルで指定された名前を持つ 1 つの拡張データ・エレメントのみを許可します。ただしこの制限は、本仕様の将来のバージョンには含まれず、Common Event Infrastructure では施行されていません。

イベント・カタログの継承:

イベント定義は、そのイベントの親のプロパティを継承します。

イベント定義は、デフォルトで、その親からプロパティの説明および拡張データ・エレメントの説明を継承します。しかし子イベントの定義はこれらの継承された記述をオーバーライドします。ただし、これには一定の制限があります。イベント・カタログにイベント定義を追加すると、そのカタログは、継承を管理するルールに新規イベント定義が違反しないことを検証します。同様に、子孫を持つ既存のイベント定義を置き換える場合、イベント・カタログでは、既存の継承リレーションシップの妥当性を検証し、それらが無効になっている場合は `InheritanceNotValidException` 例外をスローします。いずれの場合も、すべての継承リレーションシップが有効にならない限り、新規イベント定義はカタログに追加されません。

イベント定義は、未解決 および解決済み のいずれかの形式で存在します。

- 未解決のイベント定義には、イベント定義内で定義されているプロパティの定義および拡張データ・エレメントの説明のみが含まれます。
- 解決済みのイベント定義には、未解決のイベント定義内のデータ、および継承されるプロパティの定義と拡張データ・エレメントの説明が含まれます。

継承したプロパティの記述のオーバーライド

子イベントの定義は、ローカルで定義された同名、同じパス (大/小文字の区別があることに注意) のプロパティの記述を既に持っていない限り、その親イベントから各プロパティの記述を継承します。子イベントに同名、同じパスのプロパティ記述がある場合、子の記述の各フィールドは以下のように親の記述の各フィールドを以下のようにオーバーライドできます。

デフォルト値

子は、親のプロパティ記述で指定したデフォルト値をオーバーライドできます。子にデフォルト値が指定されていない場合は、親から値を継承します。

必須またはオプション

子は常に親をオーバーライドします。ただし、親がプロパティを必須として定義した場合、子もこのプロパティが必須であることを指定する必要があります。継承された必須プロパティはオプションのプロパティとして再定義できません。

許可された値または最小値と最大値

親が許可された値または最小値および最大値を定義した場合、子は許可された値または最小値および最大値を指定することによって、これらをオーバーライドできます。イベント定義には、許可値、または最小値と最大値の両方ではなく、いずれか 1 つのみを組み込むことができます。

- 親が最小値および最大値を定義しているが、子が許可された値を定義している場合は、親が定義した最小値と最大値が無視されます。
- 親が許可された値を定義し、子が最小値と最大値を定義している場合は、親が定義する許可された値が無視されます。
- 親が最大値、子が最小値しか定義していない場合、子は親が定義した最大値を継承します。
- 子が許可された値または最小値と最大値を定義しない場合は、親が指定した値が継承されます。

継承された拡張データ・エレメントの記述のオーバーライド

子イベント定義は、同名の拡張データ・エレメントの記述を既にローカルで定義していない限り、親から各拡張データ・エレメントの記述を継承します。子が同名の拡張データ・エレメントの記述を持っている場合は、子の記述の各フィールドが、以下のように親の記述をオーバーライドできます。

型 子は親と同じ型を指定する必要があります。

最小発生

子は常に親をオーバーライドします。

最大発生

子は常に親をオーバーライドします。

デフォルト値

子は、親の拡張データ・エレメントの記述によって指定されたデフォルト値をオーバーライドできます。子にデフォルト値が指定されていない場合は、親から値を継承します。

デフォルトの 16 進値

子は、親の拡張データ・エレメント記述によって指定されたデフォルトの 16 進値をオーバーライドできます。子にデフォルトの 16 進値が指定されていない場合は、親から値を継承します。

ネストされた拡張データ・エレメントの記述

子は、同名のネストされた記述を定義することによって、ネストされた拡張データ・エレメントの記述をオーバーライドできます。子が、継承したネストされた記述をオーバーライドするときには、個々のフィールドのオーバーライドに同じ規則が適用されます。子に同名のネストされた拡張データ・エレメントの記述が指定されていない場合は、親からネストされた記述を継承します。

変更の通知:

イベント定義を追加、除去、または置換するたびに、イベント・カタログは、これらの処置が行われたことを示すイベントをイベント・サーバーに送信します。

イベント・コンシューマーはこれらのイベントをサブスクライブして、イベント・カタログ変更の通知を受信します。デフォルトでは、イベント・カタログは、デフォルトのエミッター・ファクトリーを使用して、これらのイベントを送信するためのエミッターを取得します。ただし、この動作はイベント・カタログ構成で変更できます。

イベント・カタログは、以下の拡張名を使用して 3 種類の通知クラスを送信することができます。

- `cei_event_definition_added`
- `cei_event_definition_replaced`
- `cei_event_definition_removed`

これら 3 つのイベント・クラスは、共通の親クラス `cei_event_definition` からプロパティ記述を継承します。4 つの全イベント・クラスのイベント定義は、インストール時に、デフォルトのルート・イベント定義と共にイベント・カタログに自動的にロードされます。

注: イベント定義がイベント・カタログから除去されると、そのイベント定義の子またはその子孫も除去されます。イベント・カタログは、除去された各イベント定義の変更通知イベントを別途送信します。

各変更通知イベントには以下のプロパティが含まれます。

プロパティ	値
<code>version</code>	1.0.1
<code>globalInstanceId</code>	イベントのグローバルに固有な ID
<code>creationTime</code>	イベントが生成された現在日時
<code>severity</code>	10 (情報)
<code>priority</code>	10 (低)
<code>sourceComponentId</code>	イベント・カタログ・コンポーネントおよびイベント・サーバー・ホスト・マシンの ID
<code>situation</code>	状態カテゴリーの以下の値のいずれかを含む状態データ <ul style="list-style-type: none"> • <code>CreateSituation</code> (追加されたイベント定義) • <code>ConfigureSituation</code> (置換されたイベント定義) • <code>DestroySituation</code> (除去されたイベント定義)
<code>extensionName</code>	以下のいずれかの値です。 <ul style="list-style-type: none"> • <code>cei_event_definition_added</code> • <code>cei_event_definition_replaced</code> • <code>cei_event_definition_removed</code>
<code>extendedDataElements</code>	1 つの属性 <code>eventDefinitionName</code> を持つ単一の拡張データ・エレメント。この属性は、追加、置換、または除去されたイベント定義の名前を指定する文字列です。

イベント定義の作成:

イベント定義は、`EventDefinition` クラスのインスタンスです。

イベント定義は、`EventDefinition` クラスのインスタンスです。イベント定義を作成するには、先にこのクラスの新しいインスタンスを作成してから、プロパティ記述および拡張データ・エレメント記述を取り込みます。イベント定義を作成した後、作成したイベント定義をイベント・カタログに追加できます。詳しくは、345 ページの『イベント定義のイベント・カタログへの追加』を参照してください。

空の新規イベント定義を作成するには、`EventDefinition` のインスタンスを作成します。

```
EventDefinition definition = new EventDefinition(name, parent);
```

このコンストラクターのパラメーターは、以下のとおりです。

name

イベント定義の名前。これは、記述しているイベントの *extensionName* 属性の値です。

parent

親イベント定義の名前。Common Base Event 仕様で必要となるもの以外のプロパティ記述または拡張データ・エレメント記述をイベント定義に継承したくない場合は、このパラメーターを *event* にする必要があります。このパラメーターがヌルの場合、新規イベント定義はルート・イベント定義として定義されます。ルート・イベント定義は、空である場合、または現行のルート・イベント定義を置き換える場合に限り、カタログに追加できます。

戻されるオブジェクトは、プロパティ記述または拡張データ・エレメント記述を含まない未解決の新規イベント定義です。

次のコード・フラグメントは、*insurance_claim_start_auto* と呼ばれる新規イベント定義を作成します。これは、イベント定義 *insurance_claim_start* の子です。

```
EventDefinition definition = new EventDefinition("insurance_claim_start_auto",  
                                                "insurance_claim_start");
```

これで、イベント定義にプロパティの記述および拡張データ・エレメントの記述を取り込むことができます。

プロパティ記述のイベント定義への追加:

拡張データ・エレメント記述は、クラス *PropertyDescription* のインスタンスです。

イベント定義にプロパティ記述を追加するには、先に新しいプロパティ記述を作成してから、そのプロパティ記述のフィールドの値を設定する必要があります。これで、プロパティ記述をイベント定義に追加できます。

1. 新規プロパティ記述を作成するには、プロパティの名前とパスを指定して *PropertyDescription* のインスタンスを作成します。

```
PropertyDescription propDesc = new PropertyDescription(name, path);
```

このコンストラクターのパラメーターは、以下のとおりです。

name

プロパティの名前。これは、*CommonBaseEvent* エレメントまたはその子エレメントのいずれかの、単純プロパティの名前でなければなりません。

path

プロパティへのパスを指定する XPath ロケーション・パス。

CommonBaseEvent (*severity* や *priority* など) のトップレベル・プロパティの場合、*path* はヌルにする必要があります。

戻されるオブジェクトは、新しい *PropertyDescription* オブジェクトです。

2. プロパティ記述のフィールドにデータを取り込みます。 *PropertyDescription* クラスは、プロパティ記述の各フィールドに *setter* メソッドを提供します。これらのメソッドについて詳しくは、Javadoc API の資料を参照してください。例えば、プロパティが必須であることを指定するには、*setRequired(boolean)* メソッドを使用して *required* プロパティを *true* に設定します。

```
propDesc.setRequired(true);
```

3. `EventDefinition.addPropertyDescription()` メソッドを使用して、イベント定義にプロパティ記述を追加します。

```
definition.addPropertyDescription(propDesc);
```

イベント定義に、同じ名前とパスを持つ別のプロパティ記述が既に含まれている場合は、`DescriptionExistsException` 例外がスローされます。

次のコード・フラグメントは、新しいプロパティ記述を作成し、作成した記述にデータを取り込み、その記述をイベント定義に追加します。

```
PropertyDescription propDesc = new PropertyDescription("severity",null);
propDesc.setRequired(true);
propDesc.setMinValue('30');
```

```
// definition is a valid event definition
definition.addPropertyDescription(propDesc);
```

拡張データ・エレメント記述のイベント定義への追加:

拡張データ・エレメント記述は、`ExtendedDataElementDescription` クラスのインスタンスです。

イベント定義に拡張データ・エレメント記述を追加するには、先に新しい拡張データ・エレメント記述を作成してから、そのフィールドの値を設定する必要があります。また、ネストされた (子の) 拡張データ・エレメント記述 (これは、ネストされた拡張データ・エレメントを記述します) を追加することもできます。これで、拡張データ・エレメント記述をイベント定義に追加できます。

1. 新しい拡張データ・エレメント記述を作成するには、拡張データ・エレメントの名前と型を指定した `ExtendedDataElementDescription` のインスタンスを作成します。

```
ExtendedDataElementDescription edeDesc =
    new ExtendedDataElementDescription(name, type);
```

このコンストラクターのパラメーターは、以下のとおりです。

name

拡張データ・エレメントの名前。これは、記述したい拡張データ・エレメントの *name* プロパティの値でなければなりません。

type

拡張データ・エレメントのデータ型。これは、`org.eclipse.hyades.logging.events.cbe.ExtendedDataElement` クラスで定義されている以下のいずれかの整数定数でなければなりません。

- `TYPE_BOOLEAN_ARRAY_VALUE`
- `TYPE_BOOLEAN_VALUE`
- `TYPE_BYTE_ARRAY_VALUE`
- `TYPE_BYTE_ARRAY`
- `TYPE_DATE_TIME_ARRAY_VALUE`
- `TYPE_DATE_TIME_VALUE`
- `TYPE_DOUBLE_ARRAY_VALUE`
- `TYPE_DOUBLE_VALUE`

- TYPE_FLOAT_ARRAY_VALUE
- TYPE_FLOAT_VALUE
- TYPE_HEX_BINARY_VALUE
- TYPE_INT_ARRAY_VALUE
- TYPE_INT_VALUE
- TYPE_LONG_ARRAY_VALUE
- TYPE_LONG_VALUE
- TYPE_NO_VALUE_VALUE
- TYPE_SHORT_ARRAY_VALUE
- TYPE_SHORT_VALUE
- TYPE_STRING_ARRAY_VALUE
- TYPE_STRING_VALUE

戻されるオブジェクトは、新しい `ExtendedDataElementDescription` オブジェクトです。

2. 拡張データ・エレメント記述のフィールドにデータを取り込みます。
`ExtendedDataElementDescription` クラスは、拡張データ・エレメント記述の各フィールドに `setter` メソッドを提供します。これらのメソッドについては、Javadoc API の資料を参照してください。例えば、拡張データ・エレメントが少なくとも一度生じるように指定するには、`setMaxOccurs(int)` メソッドを使用して `maxOccurs` プロパティを 4 に設定します。

```
edeDesc.setMaxOccurs(4);
```

3. オプション: 子の拡張データ・エレメント記述を追加するには、`ExtendedDataElementDescription.addChild()` メソッドを使用します。

```
edeDesc.addChild(childEdeDesc);
```

`childEdeDesc` パラメーターは、有効な拡張データ・エレメント記述でなければなりません。

4. `EventDefinition.addExtendedDataElementDescription()` メソッドを使用して、イベント定義に拡張データ・エレメント記述を追加します。

```
definition.addExtendedDataElementDescription(edeDesc);
```

イベント定義に、同じ名前とパスを持つ別の拡張データ・エレメント記述が既に含まれている場合は、`DescriptionExistsException` 例外がスローされます。

次のコード・フラグメントは、新しい拡張データ・エレメント記述を作成し、作成した記述にデータを取り込み、その記述をイベント定義に追加します。

```
ExtendedDataElementDescription edeDesc =
    new ExtendedDataElementDescription("age", TYPE_SHORT_VALUE);
edeDesc.setMinOccurs(1);
edeDesc.setMaxOccurs(1);
```

```
// definition is a valid event definition
definition.addExtendedDataElementDescription(edeDesc);
```

イベント・カタログ Bean の作成:

イベント・カタログ Bean がイベント・カタログをアクセスするために使用されま
す。

イベント・カタログは、Enterprise JavaBean アーキテクチャーを使用する Stateless
Session Bean としてインプリメントされます。イベント・カタログにアクセスする
には、最初にイベント・カタログ・アプリケーションでイベント・カタログ Session
Bean のインスタンスを作成する必要があります。

ホーム・インターフェースを使用して、イベント・カタログ Session Bean のイン
スタンスを作成します。

```
//use home interface to create event catalog bean
InitialContext context = new InitialContext();
Object eventCatalogHomeObj =
    context.lookup("ejb/com/ibm/events/catalog/EventCatalog");
EventCatalogHome eventCatalogHome = (EventCatalogHome)
    PortableRemoteObject.narrow(eventCatalogHomeObj,
        EventCatalogHome.class);
eventCatalog = (EventCatalog) eventCatalogHome.create();
```

イベント定義のイベント・カタログへの追加:

新規に作成したイベントをイベント・カタログに追加することができます。

新規イベント定義を作成し、プロパティ記述と拡張データ・エレメント記述を取
り込んだ後は、そのイベント定義をイベント・カタログに追加することができま
す。一度イベントがイベント・カタログに追加されると、イベント定義を変更する
ことはできませんが、置き換えることはできます。

注: WebSphere セキュリティーが使用可能に設定されている場合、イベント・カタ
ログにイベント定義を追加するために、アプリケーション・ユーザーID を
catalogAdministrator ロールにマップする必要があります。

イベント定義をイベント・カタログに追加するには、addEventDefinition メソッドを
使用します。

```
boolean result = eventCatalog.addEventDefinition(definition, replace)
```

このメソッドのパラメーターは、以下のとおりです。

definition

追加するイベント定義。これは、有効な EventDefinition のインスタンスでなけ
ればなりません。

replace

指定したイベント定義によって同じ名前を持つ既存の定義を置き換えるかどう
かを示すブール値。

replace パラメーターが false の場合、指定したイベント定義の名前は、カタログ
内にある既存のイベント定義の名前と一致しないようにする必要があります。一致
する場合は、EventDefinitionExistsException 例外がスローされます。

replace パラメーターが true である場合は、新規イベント定義で、カタログ内に既
に存在する同じ名前の既存のイベント定義が置き換えられます。ただし、継承階層
を保持するには、新規イベント定義で古いイベント定義と同じ親を指定する必要が
あります。 そうしないと、ParentNotValidException 例外がスローされます。

戻されたブール値で、既存のイベント定義が置換されたかどうか分かります。*replace* が *true* で、しかも同じ名前のイベント定義が新しい定義で置換された場合にのみ、この値が *true* になります。

イベント定義がイベント・カタログに追加されると、イベント・カタログはイベントをイベント・サーバーに送信し、イベント・コンシューマーにこの変更が行われたことを通知します。 340 ページの『変更の通知』を参照してください。

注: 継承規則に違反するイベント定義を追加しようとする、

InheritanceNotValidException 例外がスローされ、そのイベント定義はカタログに追加されません。これが発生するのは、新規イベント定義が、継承されたプロパティの記述または拡張データ・エレメントの記述を無効な方法でオーバーライドする場合、または既存のイベント定義を置き換えることによって、継承された記述を子孫が無効な方法でオーバーライドする場合です。詳しくは、338 ページの『イベント・カタログの継承』を参照してください。

カタログからのイベント定義の除去:

必要なくなったイベント定義をイベント・カタログから削除することができます。

イベント定義が不要であれば、これをイベント・カタログから除去することができます。

注: WebSphere セキュリティーが使用可能に設定されている場合、アプリケーション・ユーザー ID を *catalogAdministrator* ロールにマップして、イベント・カタログからイベント定義を除去する必要があります。

イベント定義をイベント・カタログから除去するには、*removeEventDefinition* メソッドを使用します。

```
eventCatalog.removeEventDefinition(name)
```

name パラメーターは、イベント・カタログから除去するイベント定義の名前です。一致するイベント定義がイベント・カタログにない場合、*EventDefinitionNotFoundException* 例外がスローされます。

イベント・カタログからイベント定義が除去されると、その子および他のすべての子孫も除去されます。除去されるイベント定義ごとに、イベント・カタログはイベント・サーバーにイベントを送り、イベント・コンシューマーに変更が行われた旨を通知します。詳しくは、340 ページの『変更の通知』を参照してください。

注: イベント定義を削除する際に、イベント・カタログはイベント・サーバーをチェックして、イベント・データ・ストアの既存のイベントがそのイベント定義により記述されているかを判別することはしません。このため、イベント定義が不要であることを確認してから、イベント・カタログから除去する必要があります。

コマンド解説

コマンド行のスクリプトは、Common Event Infrastructure のいくつかの機能にアクセスするために利用できます。

これらのスクリプトは、Jacl スクリプトとしてインプリメントされています。 Jacl スクリプトは、WebSphere wsadmin ツール (*profile_root/bin* ディレクトリーにある) を使用して実行する必要があります。 wsadmin ツールについての詳細は、WebSphere Application Server の資料を参照してください。

スクリプトを実行するには、以下の構文を使用します。

```
wsadmin -f scriptname.jacl
```

他のパラメーターとの区別が可能な範囲内でのみ、パラメーター名を短くすることができます。例えば、*eventquery.jacl* スクリプトを使用する場合に、*-extensionname* の代わりに *-ex* と入力することができます。ただし、*-e* は、*-extensionname* や *-end* のいずれかを表している可能性があるため無効です。

コマンドの構文および使用方法に関するヘルプを表示するには、以下のようにコマンドの後にワード *help* を入力します。

```
wsadmin -f scriptname.jacl help
```

Simple Objects Access Protocol (SOAP) プロトコルで wsadmin ツールを使用する場合は、操作が完了する前にコマンドがタイムアウトすることがあります。例えば、イベント・サーバーからたくさんのイベントを照会したりページしたりする場合にこれが起こることがあります。タイムアウトが起こると、wsadmin ツールは以下のエラー・メッセージを表示し、SOAP RPC 呼び出しが失敗したことを示します。

```
SOAP RPC 呼び出しが失敗しました: 呼び出し (Failed to make the SOAP RPC call: invoke)
```

このエラー・メッセージが表示された場合、接続タイプに RMI、宛先ポートに 2809 を指定し、コマンドを再度実行します。例えば、以下のコマンドは、RMI 接続を使用してイベント・サーバーからイベントをページします。

```
wsadmin -conntype rmi -port 2809 eventpurge.jacl -seconds 0
```

wsadmin ツールの *-conntype* パラメーターについて詳しくは、WebSphere Application Server の資料を参照してください。

eventcatalog.jacl スクリプト:

コマンド行インターフェースを使用して、イベント・カタログにアクセスできます。

目的

イベント・カタログでイベント定義またはソース・カテゴリーをリストし、イベント定義のインポートおよびエクスポートを行います。

```
wsadmin -f eventcatalog.jacl [-serverName server_name] [-listdefinitions |  
-listcategories | -exportdefinitions | -importdefinitions] [-file filename] [-name  
event_def_name] [-pattern] [-resolve] [-replace]
```

説明

eventcatalog.jacl スクリプトによって、コマンド行を使用してイベント・カタログの内容にアクセスすることができます。また、イベント定義のインポートおよびエクスポートもサポートします。

パラメーター

-serverName *server_name*

EventServer アプリケーションがデプロイされたアプリケーション・サーバーの名前。この引数はオプションですが、EventServer アプリケーションがデプロイされている WebSphere ノードで複数のアプリケーション・サーバーが実行中のときは指定する必要があります。そうしない場合、Common Event Infrastructure エンタープライズ Bean を見つけることができないため、コマンドは失敗します。

-listdefinitions

指定したイベント定義を、読み取り可能な形式でリストします。この場合、名前の昇順でソートされます。リストは、**-file** パラメーターで指定したファイルに書き込まれます。このパラメーターを指定しない場合、リストは、標準出力に書き込まれます。

-listcategories

定義したイベント・ソース・カテゴリとこれに含まれるイベント・クラスをリストします。この場合、ソース・カテゴリの昇順でソートされます。リストは、**-file** パラメーターで指定したファイルに書き込まれます。このパラメーターを指定しない場合、リストは、標準出力に書き込まれます。

-exportdefinitions

指定したイベント定義を、インポートに適した形式でリストします。リストは、eventdefinition5_0_1.xsd XSD スキーマに準拠した XML 文書として書き込まれます。このスキーマは、events-client.jar ファイルにパッケージされています。リストは、**-file** パラメーターで指定したファイルに書き込まれます。このパラメーターを指定しない場合、リストは、標準出力に書き込まれます。

-importdefinitions

イベント定義のリストをファイルから読み取り、そのイベント定義をイベント・カタログに追加します。インポートするイベント定義のリストは、eventdefinition.xsd XSD スキーマに準拠した XML 文書として書き込まれる必要があります。

-file *filename*

リストまたはエクスポート操作の場合は、出力を書き込むファイルの名前です。インポート操作の場合は、インポートするイベント定義が含まれるファイルになります。このパラメーターは、インポート操作では必須ですが、リスト操作およびエクスポート操作ではオプションです。リスト操作またはエクスポート操作でこのパラメーターが指定されていない場合、出力は標準出力に書き込まれます。

-name *event_def_name*

リストまたはエクスポートするイベント定義を識別する名前です。**-pattern** パラメーターが指定されていない場合、**-name** パラメーターにより特定の単一イベント定義が識別されます。**-pattern** が指定されている場合は、**-name** が、イベント定義名の比較の対象となるパターンを指定します。このパターンで、パーセント文字 (%) はゼロ文字以上の任意のシーケンスを表し、下線 () は任意の 1 文字を表します。他のすべての文字は、リテラル文字として取り扱われます。

このパラメーターは、 **-listdefinitions** および **-exportdefinitions** オプションが指定されている場合のみ有効です。 **-listcategories** または **-importdefinitions** オプションを指定した場合は無効です。

-pattern

-name パラメーターで指定した値を、パターンとして扱うように指定します。このパラメーターは、 **-listdefinitions** および **-exportdefinitions** オプションが指定されている場合のみ有効です。 **-listcategories** または **-importdefinitions** オプションを指定した場合は無効です。

-resolve

リストまたはエクスポートするイベント定義を解決することを指定します。解決済みのイベント定義には、継承の階層内の祖先から継承したプロパティおよび拡張データ・エレメントの説明が含まれます。このパラメーターが指定されていない場合、イベント定義のリストには未加工のイベント定義のみが含まれます。

このパラメーターは、 **-listdefinitions** および **-exportdefinitions** オプションが指定されている場合のみ有効です。 **-listcategories** または **-importdefinitions** オプションを指定した場合は無効です。

-replace

インポートするイベント定義で、同じ名前の既存のイベント定義が置き換えられることを指定します。このパラメーターが指定されていない場合、既存のイベント定義とインポートされるイベント定義の間で名前の衝突が起これ、その結果エラーが発生して、イベント定義はインポートされません。

このパラメーターは、 **-importdefinitions** オプションが指定されている場合のみ有効です。 **-listdefinitions**、**-listcategories**、または **-exportdefinitions** オプションを指定した場合は無効です。

例

この例では、次のように、`insurance_claim_start` という名前の単一の解決済みイベント定義の内容が表示され、結果が標準出力に書き込まれます。

```
wsadmin -f eventcatalog.jacl -listdefinitions -name insurance_claim_start -resolve
```

この例では、次のように、名前がストリング `insurance_claim_start` で始まるイベント定義のセットがエクスポートされ、結果が XML ファイルに書き込まれます。

```
wsadmin -f eventcatalog.jacl -exportdefinitions -file d:%myexport.xml  
-name insurance_claim_start% -pattern
```

この例では、次のように、ファイル `myimport.xml` からイベント定義のセットがインポートされ、同じ名前の既存の定義と置き換えられます。

```
wsadmin -f eventcatalog.jacl -importdefinitions -file d:%myimport.xml -replace
```

この例では、定義したイベント・ソース・カテゴリーすべてとこれらに含まれるイベントのリストが表示されます。結果は、次のように、標準出力に書き込まれます。

```
wsadmin -f eventcatalog.jacl -listcategories
```

emitevent.jacl スクリプト:

コマンド行インターフェースを使用して、イベント・サーバーにイベントを実行依頼することができます。

目的

イベントをイベント・サーバーに送信します。

```
wsadmin -f emitevent.jacl [-xml url] [-serverName server_name] [-msg message]  
[-severity severity] [-extensionname extension_name] [-emitter profile_name]  
[-synchronous | -asynchronous]
```

説明

emitevent.jacl スクリプトによって、イベント・サーバーにイベントをサブミットするためのコマンド行インターフェースが提供されます。イベント・コンテンツは、ソース XML ファイルを提供するか、またはコマンド行でプロパティ値を指定して提供することができます。

このスクリプトで生成されるイベントは、以下のデフォルト・コンテンツを持っています。

```
<CommonBaseEvent creationTime=current_system_time  
version="1.0.1"> <sourceComponentId component="emitevent.jacl"  
componentIdType="Application" location=local_hostname  
locationType="Hostname"  
subComponent="com.ibm.events.cli.util.EmitEventCliHelper"  
componentType="http://www.ibm.com/namespaces/autonomic/Tivoli  
/EventInfrastructure"/> <situation categoryName="ReportSituation">  
  <situationType xsi:type="ReportSituation" reasoningScope="EXTERNAL"  
    reportCategory="CLI"/>  
</situation>  
</CommonBaseEvent>
```

current_system_time パラメーターは、XML dateTime ストリングとして指定される、イベントが生成されたシステム時刻です。

パラメーター

-xml *url*

サブミットするイベントが含まれる XML 文書のロケーションを指定する URL。この XML 文書は、Common Base Event のバージョン 1.0.1 XSD スキーマに準拠している必要があります。URL スキーム (http:// など) が指定されていない場合は、ローカル・ファイルと見なされます。このパラメーターはオプションです。

2 つのサンプル XML ファイル `eventsample1.xml` および `eventsample2.xml` は、`install_root/events/samples` ディレクトリーにあります。

-serverName *server_name*

EventServer アプリケーションがデプロイされたアプリケーション・サーバーの名前。この引数はオプションですが、EventServer アプリケーションがデプロイされている WebSphere ノードで複数のアプリケーション・サーバーが実行中のときは指定する必要があります。そうしない場合、Common Event Infrastructure エンタープライズ Bean を見つけることができないため、コマンドは失敗します。

-msg *message*

イベントのメッセージ・プロパティに使用する値。メッセージにスペースが含まれている場合は、この値を引用符で囲んでください。このパラメーターはオプションです。XML ファイルに加えてこのパラメーターを指定した場合、**-msg** パラメーターの値は、XML ファイルで `msg` プロパティに対して指定したすべての値をオーバーライドします。

-severity *severity*

イベントの重大度プロパティに使用する値。このパラメーターはオプションです。XML ファイルに加えてこのパラメーターを指定した場合、**-severity** パラメーターの値は、XML ファイルで `severity` プロパティに対して指定したすべての値をオーバーライドします。

-extensionname *extension_name*

イベントの `extensionName` プロパティに使用する値。拡張名にスペースが含まれている場合は、この値を引用符で囲んでください。このパラメーターはオプションです。XML ファイルに加えてこのパラメーターを指定した場合、

-extensionname パラメーターの値は、XML ファイルで `extensionName` プロパティに対して指定したすべての値をオーバーライドします。

-emitter *profile_name*

エミッターを取得するときに使用するエミッター・ファクトリー・プロファイルの Java Naming and Directory Interface (JNDI) 名。このパラメーターはオプションです。このパラメーターを指定しない場合、デフォルトのエミッター・ファクトリー・プロファイル (`/com/ibm/events/configuration/emitter/Default`) が使用されます。

-synchronous | **-asynchronous**

イベント送信に使用される同期モード。このパラメーターはオプションです。指定しない場合、エミッターに対して構成されている同期モード指定が使用されます。

例

以下の例では、重大度が 30、拡張名が `test_event` (その他のすべてのプロパティはデフォルト値です) のイベントをイベント・サーバーに送信しています。

```
wsadmin -f emitevent.jacl -severity 30 -extensionname test_event
```

以下の例では、`eventsample1.xml` で指定したプロパティを使用してイベントを送信しています。

```
wsadmin -f emitevent.jacl -xml ../samples/eventsample1.xml
```

eventquery.jacl スクリプト:

コマンド行インターフェースを使用して、イベント・データベースを照会することができます。

目的

イベント・データベース内のイベントをリストするレポートを生成します。

```
wsadmin -f eventquery.jacl [-serverName server_name] [-globalinstanceid  
global_instance_id | -group event_group] [-severity severity] [-extensionname  
extension_name] [-start start_time] [-end end_time] [-number number]  
[-ascending|-descending]
```

説明

eventquery.jacl スクリプトによって、イベント・データベースが照会され、結果をリストするレポートが生成されます。イベントは、イベント・グループ、重大度、または拡張名に基づいて照会することができます。また、指定した期間内に作成されたイベントを照会することもできます。

パラメーター

-serverName *server_name*

EventServer アプリケーションがデプロイされたアプリケーション・サーバーの名前。この引数はオプションですが、EventServer アプリケーションがデプロイされている WebSphere ノードで複数のアプリケーション・サーバーが実行中のときは指定する必要があります。そうしない場合、Common Event Infrastructure エンタープライズ Bean を見つけることができないため、コマンドは失敗します。

-globalinstanceid *global_instance_id*

照会するイベントのグローバル・インスタンス ID。このパラメーターか、**-group** のいずれかが必要ですが、両方は必要ありません。

-group *event_group*

照会するイベントが含まれるイベント・グループ。 *event_group* の値は、Common Event Infrastructure 構成で定義したイベント・グループの名前でなければなりません。このパラメーターか、**-globalinstanceid** のいずれかが必要ですが、両方は必要ありません。

-severity *severity*

レポートに組み込むイベントの重大度。 *severity* の値は整数である必要があります。指定した値に等しい重大度のイベントのみがレポートに組み込まれます。このパラメーターはオプションです。

-extensionname *extension_name*

レポートに組み込むイベントの拡張名。このパラメーターで、照会するイベントを特定のタイプに限定します。 *extensionName* プロパティーが *extension_name* に等しいイベントのみがレポートに組み込まれます。このパラメーターはオプションです。

-start *start_time*

レポートに組み込むイベントの最も早い時刻。指定した日時より後に生成されたイベントに照会を制限する場合、このパラメーターを使用します。このパラメーターは、XML *dateTime* データ型に基づいて指定した日時である必要があります。基本形式は *CCYY-MM-DDThh:mm:ss* で、オプションで、この後に時間帯インディケーターが続きます。例えば、東部標準時の 2004 年 1 月 1 日正午は、*2004-01-01T12:00:00-05:00* のようになります。 *dateTime* データ型については、www.w3.org の XML スキーマ (*XML schema at www.w3.org*) を参照してください。

-end *end_time*

レポートに組み込むイベントの最も遅い時刻。指定した日時より前に生成されたイベントに照会を制限する場合、このパラメーターを使用します。このパラメーターは、XML `dateTime` データ型に基づいて指定した日時である必要があります。詳しくは、**-start** パラメーターの説明を参照してください。

-number *number*

レポートに組み込むイベントの最大数。このパラメーターは整数でなければなりません。データベース内で一致するイベントの数が指定した値を超える場合は、レポートは切り捨てられます。つまり、レポートが昇順でソートされている場合には、最新の一致イベントが省略されます。レポートが降順でソートされている場合には、最も古い一致イベントが省略されます。

-ascending-descending

レポート内のイベントをソートする発生順の順序。これは、必ず次の値のいずれかになります。

ascending

昇順 (発生順)。古いイベントから順に並べます。これはデフォルト値です。

descending

降順 (発生順の逆)。新しいイベントから順に並べます。

例

以下の例では、データベース内の、**全イベント**というイベント・グループに属し、2004年2月17日に生成されたすべてのイベントがリストされます。レポートは、次のように、日時順の逆でソートされます。

```
eventquery.jacl -group "All events" -start "2004-02-17T00:00:00-05:00"
-end "2004-02-17T23:59:59-05:00" -order DESC
```

eventbucket.jacl スクリプト:

コマンド行インターフェースを使用して、イベント・データベース・バケット構成を変更できます。

目的

イベント・データベース・バケット構成を表示または変更します。

```
wsadmin -f eventbucket.jacl [-status] [-change] [-serverName server_name]
```

説明

eventbucket.jacl スクリプトは、イベント・データベース・バケット構成を表示または変更します。バケットは、イベント・データベースから古いイベント・データをパージするための高速パージ・ユーティリティによって使用されます。このコマンドを実行して、現在のバケット構成を判別したり、アクティブおよび非アクティブ・バケットをスワップすることができます。

注: WebSphere セキュリティーが使用可能にされている場合、ユーザー ID を `eventAdministrator` ロールにマップして、イベント・データベース・バケット構成を表示または変更する必要があります。

パラメーター

-status

現在のバケット構成についての情報を表示できます。例えば、アクティブ・バケットの設定、およびバケットのチェック間隔 (データ・ストア・プラグインが、どのバケットがアクティブか判定する頻度) を表示できます。

-change

アクティブ・バケットを非アクティブ・バケットに、非アクティブ・バケットをアクティブ・バケットにするためにバケットをスワップします。非アクティブ・バケットは空でないと、このオプションは使用できません。

-serverName *server_name*

EventServer アプリケーションがデプロイされたアプリケーション・サーバーの名前。この引数はオプションですが、EventServer アプリケーションがデプロイされている WebSphere ノードで複数のアプリケーション・サーバーが実行中のときは指定する必要があります。そうしない場合、Common Event Infrastructure エンタープライズ Bean を見つけることができないため、コマンドは失敗します。

例

以下の例は、現在のバケット構成を表示します。

```
wsadmin -f eventbucket.jacl -status
```

以下の例は、アクティブ・バケットおよび非アクティブ・バケットをスワップします。

```
wsadmin -f eventbucket.jacl -change
```

eventpurge.jacl スクリプト:

コマンド行インターフェースを使用して、イベント・データベースからイベントをパージすることができます。

目的

イベント・データベースからイベントをパージします。

```
wsadmin -f eventpurge.jacl [-serverName server_name] [-seconds seconds | -end  
end_time] [-group event_group] [-severity severity] [-extensionname extension_name]  
[-start start_time] [-size size]
```

説明

eventpurge.jacl スクリプトは、イベント・データベースからイベントをパージします。イベント・データベースからすべてのイベントをパージすることも、一定の基準を満たすイベントに限定してパージすることもできます。

注: WebSphere セキュリティーが使用可能に設定されている場合、ユーザーID を eventAdministrator ロールにマップして、イベントを削除する必要があります。

パラメーター

-serverName *server_name*

EventServer アプリケーションがデプロイされたアプリケーション・サーバーの名前。この引数はオプションですが、EventServer アプリケーションがデプロイされている WebSphere ノードで複数のアプリケーション・サーバーが実行中のときは指定する必要があります。そうしない場合、Common Event Infrastructure エンタープライズ Bean を見つけることができないため、コマンドは失敗します。

-seconds *seconds*

ページするイベントの最小経過期間。*seconds* の値は整数でなければなりません。指定した秒数より長時間経過したイベントのみがページされます。**-end** パラメーターを指定しない場合、このパラメーターが必要です。

-end *end_time*

削除するイベント・グループの終了時刻。指定時刻以前に生成されたイベントのみ削除されます。*end_time* 値は、XML *dateTime* フォーマット (CCYY-MM-DDThh:mm:ss) で指定する必要があります。例えば、東部標準時の 2006 年 1 月 1 日正午は、2006-01-01T12:00:00-05:00 のようになります。*dateTime* データ型について詳しくは、www.w3.org の XML スキーマ (XML schema at www.w3.org) を参照してください。

-seconds パラメーターを指定しない場合、このパラメーターが必要です。

-group *eventGroup*

ページするイベントが含まれるイベント・グループ。*event_group* の値は、Common Event Infrastructure 構成で定義したイベント・グループの名前でなければなりません。このパラメーターはオプションです。

-severity *severity*

ページするイベントの重大度。*severity* の値は整数である必要があります。重要度が指定した値と等しいイベントのみページされます。このパラメーターはオプションです。

-extensionname *extension_name*

ページに組み込むイベントの拡張名。このパラメーターで、ページの対象を特定タイプのイベントに限定します。*extensionName* プロパティーが *extension_name* に等しいイベントのみがページされます。このパラメーターはオプションです。

-start *start_time*

削除するイベント・グループの開始時刻。指定時刻後に生成されたイベントのみ削除されます。*start_time* 値は、XML *dateTime* フォーマット (CCYY-MM-DDThh:mm:ss) で指定する必要があります。このパラメーターはオプションです。

-size *size*

単一トランザクション内でページするイベントの数。*size* の値は整数でなければなりません。この数のイベントがページされた後、コマンドによってトランザクションがコミットされてから、新規トランザクションが継続されます。このパラメーターはオプションです。

例

以下の例では、生成されてから 10 分以上経過している、重大度が 20 (重大でない) のすべてのイベントがデータベースからパージされます。

```
eventpurge.jacl -group "All events" -severity 20 -seconds 600
```

WebSphere Process Server 管理のトラブルシューティング

トラブルシューティングは、問題の原因を検出して除去するプロセスです。このトピック群は、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のに役立ちます。

Failed Event Manager のトラブルシューティング

このトピックでは、Failed Event Manager の使用中に発生する可能性がある問題について説明します。

注: このトピックでは、システムで失敗したイベントを検出、変更、再サブミット、または削除するために、Failed Event Manager を使用方法については説明しません。失敗したイベントの管理について詳しくは、インフォメーション・センターの『WebSphere Process Server の失敗したイベントの管理』を参照してください。

次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という問題がある	『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	『「日付ごと」タブの値は、正しく入力されないと自動的にデフォルトの値に変更されません。』
期限切れのイベントを削除できない	357 ページの『「期限切れイベントを削除」機能を実行すると Failed Event Manager が中断するように見える』

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で「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。

「日付ごと」タブの値は、正しく入力されないと自動的にデフォルトの値に変更されます。

「検索」ページの「日付ごと」タブには、「開始日」と「日付ごと」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要です。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と全く同じようにフォーマットする必要があります。値の形式に不整合があると (例えば、年数に 2 桁ではなく 4 桁の数字を使用したり、時間値を省略したりなど)、Failed Event Manager が以下の警告メッセージを発行してフィールドのデフォルト値を置き換えます。

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、「開始日」フィールドは、東部標準時 (EST) のタイム・ゾーンの en_US ロケールに配置されたマシンでは、12/31/69 7:00 PM がデフォルト値になります。「日付ごと」フィールドのデフォルト値は常に、ユーザーのロケールおよびタイム・ゾーンにあわせてフォーマット設定された現在日付と時刻になります。

この問題を回避するには、上記の各フィールドで示した例に従って、日付と時刻を常に慎重に入力するようにしてください。

「期限切れイベントを削除」機能を実行すると Failed Event Manager が中断するよう見える

現行の検索結果に失敗したイベントが多く存在する状態、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状態で「期限切れイベントを削除」ボタンを使用すると、Failed Event Manager が無期限に中断したように見える場合があります。

この状態では、Failed Event Manager は実際に中断しているわけではありません。実際には大量のデータ・セットの処理中であり、コマンドが完了し次第すぐに結果をリフレッシュします。

Business Process Choreographer 構成のトラブルシューティング

このトピックを参照して、ビジネス・プロセス・コンテナまたはヒューマン・タスク・コンテナの構成に関連した問題を解決します。

このセクションの目的は、ビジネス・プロセス・コンテナまたはヒューマン・タスク・コンテナの構成が期待どおりに機能しない理由を理解したり、問題を解決したりする際に役立つ情報を提供することです。以下のタスクは、ビジネス・プロセス・コンテナまたはヒューマン・タスク・コンテナの構成中に発生する可能性のある問題の判別および解決方法に焦点を当てています。

Business Process Choreographer のログ・ファイル

ここでは、Business Process Choreographer 構成のログ・ファイルの保管場所について説明します。

プロファイルの作成

Business Process Choreographer のプロファイル・アクションでは、logs ディレクトリーの *bpcaugment.log* ファイルに書き込みを行います。

プロファイル・ウィザードでサンプル構成オプションを選択すると、bpeconfig.jacl スクリプトが呼び出され、アクションは logs ディレクトリーの bpeconfig.log ファイルに記録されます。

管理スクリプト

wsadmin を使用して実行されるすべての Business Process Choreographer スクリプトは、wsadmin.traceout ファイルに記録されます。ただし、このファイルは wsadmin が起動されるごとに上書きされるので、wsadmin を再度起動する前に必ずこのログ・ファイルを保管してください。

構成関連スクリプト

bpeconfig.jacl、taskconfig.jacl、clientconfig.jacl、および bpeunconfig.jacl の各スクリプト・ファイルは、それぞれ bpeconfig.log、taskconfig.log、clientconfig.log、および bpeunconfig.log という名前でログ・ファイルを logs ディレクトリーに書き込みます。また、wsadmin.traceout ファイルも確認します。

管理ユーティリティー・スクリプト

ProcessChoreographer ディレクトリーの util サブディレクトリーにある管理スクリプトは、独自のログ・ファイルを書き込みません。wsadmin.traceout ファイルおよびアプリケーション・サーバーのログ・ファイルを確認します。

構成チェッカー

ProcessChoreographer ディレクトリーにある bpecheck.jacl スクリプト・ファイルを使用して、共通の構成問題を確認できます。結果は、logs ディレクトリーの bpecheck.log ファイルに書き込まれます。

Business Process Choreographer のトレースの使用可能化

ここでは、サポートに連絡する前に行うべきことについて説明します。

トレースの使用可能化

Business Process Choreographer トレースでは、標準の WebSphere Process Server トレース・メカニズムを使用します。このメカニズムは通常の方法で使用可能に設定する必要があります。

トレース仕様は次のとおりです。

```
com.ibm.bpe.*=all=enabled;com.ibm.ws.staffsupport.*=all=enabled
```

ここで com.ibm.bpe は、ビジネス・プロセス、およびヒューマン・タスクのほとんどの局面をトレースします。ヒューマン・タスクの残りの局面であるスタッフ・プラグインは、com.ibm.ws.staffsupport によってトレースされます。

サポートに送信するもの

トレースを使用可能に設定後、問題シナリオを再作成して以下のファイルを準備します。

- SystemOut.log

- SystemErr.log
- trace.log

これらのファイルは次のディレクトリーにあります。
install_rootprofiles/profile_name/logs/

タスク・コンテナ・アプリケーションの開始に失敗する

ejb/htm/TaskContainerStartupBean という名前の開始 Bean がアプリケーションを強制的に停止しました。

症状

次のエラーが SystemOut.log ファイルに書き込まれます。

```
WSVR0037I: Starting EJB jar: taskejb.jar
NMSV0605W: A Reference object looked up from the context "java:"
with the name "comp/env/scheduler/DefaultUserCalendarHome"
was sent to the JNDI Naming Manager and an exception resulted.
Reference data follows:
Reference Factory Class Name: com.ibm.ws.naming.util.IndirectJndiLookupObjectFactory
Reference Factory Class Location URLs:
Reference Class Name: java.lang.Object
Type: JndiLookupInfo
Content: JndiLookupInfo:
jndiName="com/ibm/websphere/scheduler/calendar/DefaultUserCalendarHome";
providerURL=""; initialContextFactory=""
:
StartBeanInfo E STUP0005E: Startup bean named ejb/htm/TaskContainerStartupBean
forced application to stop.
ApplicationMg W WSVR0101W: An error occurred starting, TaskContainer_utxt1b10Node01_server1
ApplicationMg A WSVR0217I: Stopping application: TaskContainer_utxt1b10Node01_server1
EJBContainerI I WSVR0041I: Stopping EJB jar: taskejb.jar
```

理由

このエラーは、TaskContainer アプリケーションの開始時に SchedulerCalendars アプリケーションが使用不可になっている場合に発生します。

解決方法

SchedulerCalendars アプリケーションを手動でインストールするか、既にインストール済みの場合は、新規のターゲット・マッピングを追加します。

デフォルト・プロファイルでは、SchedulerCalendars アプリケーションが、WebSphere システム・アプリケーションとして自動的に使用可能になります。ただし、カスタム・プロファイルでは自動的に使用可能にはなりません。

bpeconfig.jacl スクリプトは、SchedulerCalendars アプリケーションのインストールを試行しますが、常に可能とは限りません。

管理コンソールのインストール・ウィザードを使用して ND 環境の Business Process Choreographer を構成する場合は、SchedulerCalendars アプリケーションを手動でインストールする必要があります。

Business Process Choreographer データベースおよびデータ・ソースのトラブルシューティング

このタスクを使用して、Business Process Choreographer データベースおよびデータ・ソースの問題を解決します。

ビジネス・プロセス・コンテナとヒューマン・タスク・コンテナは、両方ともデータベースを必要とします。データベースがなければ、ビジネス・プロセスおよびヒューマン・タスクを含むエンタープライズ・アプリケーションは機能しません。

• DB2® を使用している場合:

- DB2 Universal JDBC ドライバー・タイプ 4 を使用しており、Business Process Choreographer データ・ソースで接続をテストするとき、またはサーバーを始動するときに、"com.ibm.db2.jcc.a.re: XAER_RMERR : The DDM parameter value is not supported. DDM parameter code point having unsupported value : 0x113f DB2ConnectionCorrelator: NF000001.PA0C.051117223022" などの DB2 内部エラーが発生する場合は、以下のアクションを実行します。

1. データ・ソースのクラスパス設定を確認します。デフォルト・セットアップでは、WebSphere 変数 `{DB2UNIVERSAL_JDBC_DRIVER_PATH}` により、`universalDriver_wbi` ディレクトリーにある WebSphere Process Server 組み込み DB2 Universal JDBC ドライバーを指すことができます。
2. ドライバーのバージョンは、DB2 サーバー・バージョンと互換性がない場合があります。ご使用のデータベース・インストールのオリジナル `db2jcc.jar` ファイルを使用しており、WebSphere Process Server 組み込み DB2 Universal JDBC ドライバーではないことを確認します。必要な場合は、オリジナル `db2jcc.jar` ファイルを指すように、WebSphere 変数 `{DB2UNIVERSAL_JDBC_DRIVER_PATH}` の値を変更します。
3. サーバーを再始動します。

- DB2 インスタンスの `db2diag.log` ファイルに、以下に示す ADM5503E のようなメッセージが含まれている場合:

```
2004-06-25-15.53.42.078000 Instance:DB2 Node:000
PID:2352(db2syscs.exe) TID:4360 Appid:*LOCAL.DB2.027785142343
data management sqlEscalateLocks Probe:4 Database:BPEDB
```

ADM5503E ロック・インテント "X" に対する表 "GRAALFS .ACTIVITY_INSTANCE_T" での "10" ロックのエスカレーションは失敗しました。SQLCODE は "-911" です。

LOCKLIST 値を増やします。例えば、この値を 500 に設定するには、以下の DB2 コマンドを入力します。

```
db2 UPDATE DB CFG FOR BPEDB USING LOCKLIST 500
```

これにより、パフォーマンスが大きく向上します。

- デッドロックを回避するには、データベース・システムが十分なメモリー (特にバッファ・プール用) を使用するように構成されていることを確認します。DB2 の場合は、DB2 Configuration Advisor を使用して、ご使用の構成に合った値を判別します。
- データ・ソースのインプリメンテーション・クラス `COM.ibm.db2.jdbc.DB2XADataSource` について述べるエラーが発生した場合:

- server.policy ファイルで使用されているすべての WebSphere 環境変数が、正しく設定されていることを確認します。例えば、DB2_INSTALL_ROOT と DB2_JDBC_DRIVER_PATH です。
- ご使用の JDBC プロバイダーのクラスパス定義が正しいこと、および項目が 2 つ存在しないことを確認します。
- コンポーネント管理認証別名が、*cellName/BPEAuthDataAliasDbType_Scope* に設定されていることを確認します。ここで、*cellName* はセルの名前、*DbType* はデータベース・タイプ、および *Scope* は定義のスコープです。
- リモートの DB2 for z/OS[®] データベースを使用している場合で、アプリケーション・サーバーがリモート・データベースとの最初の XA トランザクションを開始しようとするときに、SystemOut.log ファイルに SQL コード 30090N がある場合は、以下を実行します。
 - インスタンス構成変数 SPM_NAME が、8 文字以内のホスト名を持つローカル・サーバーを指していることを確認します。ホスト名が 8 文字より長い場合は、etc/hosts ファイルで短い別名を定義します。
 - その他の場合、無効な同期点マネージャー・ログ項目が sqllib/spmlog ディレクトリーにある可能性があります。sqllib/spmlog ディレクトリーの項目のクリアを試行して再始動します。
 - SPM_LOG_FILE_SZ の値を増やすことを考慮します。
- Cloudscape を使用している場合:
 - Linux または UNIX システムで「開かれたファイルが多すぎます (Too many open files)」エラーが発生した場合は、使用可能なファイル・ハンドルの数を例えば 4000 以上に増やします。使用可能なファイル・ハンドルの数を増やす方法について詳しくは、ご使用のオペレーティング・システムの資料を参照してください。
 - Cloudscape ツールを起動しようとする「Java クラスが見つかりません (Java class not found)」という例外が発生する場合は、Java 環境がセットアップされているか、および classpath 環境変数に以下の JAR ファイルが含まれているか確認します。
 - db2j.jar
 - db2jtools.jar
 - db2jcc.jar
 - db2jcvview.jar
 - Cloudscape ツール (ij または cvview など) を使用して Cloudscape データベースに接続できず、以下の例外が発生する場合:

エラー XJ040: データベース「c:¥WebSphere¥AppServer¥profiles¥profile_name¥databases¥BPEDB」を開始できませんでした。
(Failed to start database 'c:¥WebSphere¥AppServer¥profiles¥profile_name¥databases¥BPEDB')

詳細は次の例外を参照してください (see the next exception for details.)。

エラー XSDB6: Cloudscape の別のインスタンスが既にデータベース c:¥WebSphere¥AppServer¥profiles¥profile_name¥databases¥BPEDB をブートしている可能性があります。

(Another instance of Cloudscape may have already booted the database
c:¥WebSphere¥AppServer¥profiles¥profile_name¥databases¥BPEDB)

Cloudscape データベースにアクセスできるアプリケーションは一度に 1 つだけなので、これらのツールを使用する前に WebSphere Application Server を停止する必要があります。

- ビジネス・プロセスまたはヒューマン・タスクを含むエンタープライズ・アプリケーションのインストール中に、データベース・エラーが発生する場合。エンタープライズ・アプリケーションをインストールすると、プロセス・テンプレートとタスク・テンプレートは Business Process Choreographer データベースに書き込まれます。ビジネス・プロセス・コンテナが使用するデータベース・システムが稼動しており、アクセス可能であることを確認します。
- 国別文字の使用に問題がある場合。Unicode 文字セットをサポートするようデータベースが作成されていることを確認します。
- データベースの中でテーブルまたはビューが検出されない場合。データ・ソースの認証別名を構成するときには、データベース表の作成に使用したユーザー ID (またはデータベース表を作成するスクリプトの実行に使用したユーザー ID) と同じユーザー ID を指定する必要があります。

Business Process Choreographer キュー・マネージャーおよび JMS プロバイダーのトラブルシューティング

これを使用して、キュー、キュー・マネージャー、および Java Message Service (JMS) プロバイダーに関連した Business Process Choreographer の問題を解決します。

Business Process Choreographer は、高信頼性メッセージングを使用します。メッセージング・サービスは、WebSphere に組み込まれた JMS プロバイダーか、別途インストールした製品 WebSphere MQ です。以下に、起こりうる問題に対する解決方法をいくつか示します。

ビジネス・プロセスとヒューマン・タスクのトラブルシューティング

このトピックを参照して、ビジネス・プロセスおよびヒューマン・タスクに関連した問題を解決します。

以下のタスクは、ビジネス・プロセスまたはタスクの実行中に発生する可能性のある問題のトラブルシューティングに焦点を当てています。

ビジネス・プロセス・アプリケーションとヒューマン・タスク・アプリケーションのインストールのトラブルシューティング

ND 環境で、ビジネス・プロセス、ヒューマン・タスク、またはその両方を含むアプリケーションをインストールするときに、デプロイメント・マネージャーの SystemErr.log ファイルに例外が記録されます。

症状

ND 環境で、ビジネス・プロセス、ヒューマン・タスク、またはその両方を含むアプリケーションをインストールするときに、デプロイメント・マネージャーの SystemErr.log ファイルで以下の例外が検出されます。

```
SystemErr R com.ibm.ws.management.commands.sib.SIBAdminCommandException:
CWSJA0012E: Messaging engine not found.
at com.ibm.ws.management.commands.sib.SIBAdminCommandHelper.createDestination
(SIBAdminCommandHelper.java:787)
at com.ibm.ws.management.commands.sib.CreateSIBDestinationCommand.afterStepsExecuted
(CreateSIBDestinationCommand.java:459)
```

```
at com.ibm.websphere.management.cmdframework.provider.AbstractTaskCommand.execute
(AbstractTaskCommand.java:547)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdminHelper.call(SIBAdminHelper.java:136)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdminHelper.createSIBDestination
(SIBAdminHelper.java:112)
at com.ibm.ws.sca.internal.deployment.sib.SIBAdmin.createDestination(SIBAdmin.java:327)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.createDestination
(SIBDestinationTask.java:263)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.preInstallModule
(SIBDestinationTask.java:71)
at com.ibm.ws.sca.internal.deployment.SCATaskBase.installModule(SCATaskBase.java:57)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.processArtifacts
(SIBDestinationTask.java:228)
at com.ibm.ws.sca.internal.deployment.sib.SIBDestinationTask.install
(SIBDestinationTask.java:287)
at com.ibm.ws.sca.internal.deployment.SCAInstallTask.performInstallTasks
(SCAInstallTask.java:116)
at com.ibm.ws.sca.internal.deployment.SCAInstallTask.performTask
(SCAInstallTask.java:61)
at com.ibm.ws.management.application.SchedulerImpl.run(SchedulerImpl.java:253)
at java.lang.Thread.run(Thread.java:568)
```

理由

「SCA.SYSTEM.cellName.Bus」バスのバス・メンバーが欠落しています。

解決方法

管理コンソールで、「サービス統合」 → 「バス」 → **SCA.SYSTEM.cellName.Bus** をクリックします。「トポロジー」セクションで、「バス・メンバー」をクリックします。ビジネス・プロセス・アプリケーションまたはヒューマン・タスク・アプリケーションをバス・メンバーとしてインストールするサーバーまたはクラスターを追加してから、該当サーバーまたはクラスターを再始動し、再度アプリケーションのインストールを試行します。

ビジネス・プロセスの実行のトラブルシューティング

ここでは、ビジネス・プロセスの実行に関する共通の問題の解決方法について説明します。

Business Process Choreographer Explorer を使用し、IBM® 技術サポート・ページでエラー・メッセージ・コードを検索できます。

1. エラー・ページで、「詳しくは、**検索してください**」リンクをクリックします。これにより、IBM 技術サポート・サイトでエラー・コードの検索を開始します。このサイトで提供される情報は英語のみです。
2. エラー・ページに表示されるエラー・メッセージ・コードをクリップボードにコピーします。エラー・コードの形式は CWWBcnnnc で、c は文字を、nnnn は 4 桁の数値を示します。「WebSphere Process Server technical support」ページに進みます。
3. エラー・コードを「**追加検索項目 (Additional search terms)**」フィールドに貼り付けて、「**実行 (Go)**」をクリックします。

特定の問題に対する解決方法を以降のトピックで説明します。

Microflow を含むアプリケーションを停止するときの ClassCastException:

Microflow を含むアプリケーションが停止した時間帯に、ClassCastException 例外が SystemOut.log ファイルに記述されます。

理由

アプリケーションが停止するとき、EAR ファイルに含まれるクラスはクラスパスから除去されます。ただし、それらのクラスを必要とする Microflow インスタンスがまだ実行されている可能性があります。

解決方法

以下のアクションを実行します。

1. まず、Microflow プロセス・テンプレートを停止します。この時点以降、そのテンプレートから新規の Microflow インスタンスを開始することはできなくなります。
2. 少なくとも Microflow が実行される最大期間が過ぎるのを待ち、実行中のインスタンスすべてを完了させます。
3. アプリケーションを停止します。

XPath 照会により配列から予期しない値が戻される:

XPath 照会を使用して配列のメンバーにアクセスすると、予期しない値が戻されます。

理由

この問題の一般的な理由は、配列の最初のエレメントの指標値がゼロになっていると考えられます。配列の XPath 照会では、最初のエレメントの指標値は 1 になっています。

解決方法

配列で使用する指標値がエレメント 1 で始まっていることを確認します。

アクティビティは処理不能の障害のため停止しました (メッセージ: CWWBE0057I):

システム・ログに CWWBE0057I メッセージがあります。プロセスは「実行」状態になっていますが、現行パスでナビゲーションが先に進みません。

理由

次のすべての状態が発生した場合は、invoke アクティビティ、インライン・ヒューマン・タスク、および Java スニペットを停止状態にします。

- アクティビティにより障害が発生した
- エンクロージング・スコープで障害が処理されない
- アクティビティの continueOnError 属性が false に設定されている

解決方法

この問題を解決するには、次の 2 つのレベルの処置が必要です。

1. 管理者は、停止したアクティビティ・インスタンスを手動で修復する必要があります。例えば、停止したアクティビティ・インスタンスを強制完了するか、強制再実行します。
2. 障害の理由を調査する必要があります。場合によっては、モデリング・エラーによって障害が発生することもあり、その場合はモデル内で修正する必要があります。

例えば、WebSphere Scheduler デフォルト・カレンダーを使用し、アクティビティに「タイムアウト」によって有効期限が定義されている場合は、時間が正しい形式で定義されていることを確認します。特に、時間の数字と単位の間空白がないことを確かめます。正しく指定されたタイムアウト期間の例を次に示します。

- 1minute
- 2hours 4minutes 1second
- 1day 1hour

Microflow が補正されない:

Microflow がサービスを呼び出したときにプロセスが失敗しましたが、元に戻すサービスが呼び出されません。

解決方法

Microflow の補正を起動するには、さまざまな条件を満たす必要があります。次の点を確認します。

1. Business Process Choreographer Explorer にログオンし、「失敗した補正」をクリックして、補正サービスが失敗しており、修復する必要があるかどうかを確認します。
2. Microflow の補正は、Microflow のトランザクションがロールバックした場合にのみ起動されます。この場合に該当するかどうか確認してください。
3. Microflow の compensationSphere 属性を「必須」に設定する必要があります。
4. 補正サービスが実行されるのは、対応する転送サービスが Microflow のトランザクションに関わっていない場合のみです。転送サービスがナビゲーション・トランザクションに関わっていないことを確認してください。例えば、プロセス・コンポーネントの参照時には、Service Component Architecture (SCA) の修飾子 suspendTransaction を True に設定します。

長期実行プロセスが停止しているように見える:

長期実行プロセスは実行中の状態になっていますが、何も動作していないように見えます。

理由

このような振る舞いをするにはさまざまな理由が考えられます。

1. ナビゲーション・メッセージを再実行した回数が多すぎるため、保存キューまたは保留キューに移動された。
2. Service Component Architecture (SCA) インフラストラクチャーからの応答メッセージが繰り返し失敗した。

3. プロセスが、イベント、タイムアウト、あるいは長期実行呼び出しまたはタスクが戻るのを待っている。
4. プロセスのアクティビティーが停止状態になっている。

解決方法

上述の理由それぞれに対して、異なる修正アクションが必要になります。

1. 管理についての PDF に記載のとおり、保存キューまたは保留キューにメッセージがあるかどうかを確認します。
2. 管理コンソールの失敗したイベント管理ビューに何か表示されているかどうかを確認します。
 - Service Component Architecture (SCA) 応答メッセージからの失敗したイベントがある場合は、そのメッセージを再活動化します。
 - それ以外は、長期実行アクティビティーを強制完了するか、強制再試行します。
3. 停止状態のアクティビティーが存在するかどうかを調べ、存在する場合はそれらのアクティビティーを修復します。システム・ログに CWWBE0057I メッセージがある場合は、『メッセージ: CWWBE0057I』に記載のとおり、モデルを訂正することも必要になります。

別の EAR ファイルの同期サブプロセスの呼び出しが失敗する:

長期実行プロセスが別のプロセスを同期して呼び出し、サブプロセスが別のエンタープライズ・アーカイブ (EAR) ファイルに配置されている場合は、そのサブプロセスの呼び出しは失敗します。

例えば、次のような例外が発生します。

```
com.ibm.ws.sca.internal.ejb.util.EJBStubAdapter com.ibm.ws.sca.internal.ejb.util.EJBStubAdapter#003
Exception:
java.rmi.AccessException: CORBA NO_PERMISSION 0x49424307 No; nested exception is:
org.omg.CORBA.NO_PERMISSION: The WSCredential does not contain a forwardable token.
Please enable Identity Assertion for this scenario.
vmcid: 0x49424000 minor code: 307 completed: No
at com.ibm.CORBA.iiop.UtilDelegateImpl.mapSystemException(UtilDelegateImpl.java:202)
at javax.rmi.CORBA.Util.mapSystemException(Util.java:84)
```

理由

別の EAR ファイルの同期サブプロセスを呼び出す場合は、Common Secure Interoperability バージョン 2 (CSIv2) ID アサーションを使用可能に設定する必要があります。

解決方法

CSIv2 インバウンド認証と CSIv2 アウトバウンド認証を構成します。

実行中に予期しない例外が発生しました (メッセージ: CWWBA0010E):

キュー・マネージャーが実行されていないか、Business Process Choreographer 構成に誤ったデータベース・パスワードが含まれています。

解決方法

次の点を確認します。

1. `systemout.log` ファイルに `"javax.jms.JMSEException: MQJMS2005: failed to create MQQueueManager"` が含まれている場合、キュー・マネージャーを開始してください。
2. `Business Process Choreographer` 構成に保管されているデータベース管理者パスワードが、データベースで設定されているパスワードと一致していることを確認してください。

不明のイベント (メッセージ: CWWBE0037E):

プロセス・インスタンスにイベントを送信、または新規プロセス・インスタンスを開始しようとして、「CWWBE0037E: 不明のイベント」例外が発生しました。

理由

このエラーの共通の理由は、メッセージをプロセスに送信しても、`receive` または `pick` アクティビティーが既にナビゲート済みであるために、メッセージを再度このプロセス・インスタンスでコンシュームできないというものです。

解決方法

この問題を訂正するには、次の指示に従います。

- イベントが既存のプロセス・インスタンスによってコンシュームされることになっている場合は、対応する `receive` または `pick` アクティビティーをまだナビゲートしていない既存のプロセス・インスタンスと一致する、関連セット値を渡す必要があります。
- イベントが新規プロセス・インスタンスを開始させることになっている場合は、関連セット値が既存のプロセス・インスタンスと一致してはいけません。

ビジネス・プロセスで関連セットを使用する方法については、「[technote 1171649](#)」を参照してください。

プロセス・インスタンスを検索できないか、または作成できません (メッセージ: CWWBA0140E):

プロセス・インスタンスにイベントを送信しようとして、「`CreateRejectedException`」メッセージを受け取ります。

理由

このエラーの一般的な理由は、`createInstance` 属性が `no` に設定され、このアクティビティーで使用される関連セットのメッセージと一緒に渡される値が既存のプロセス・インスタンスと一致しないために、新規プロセス・インスタンスをインスタンス化できない `receive` または `pick` アクティビティーにメッセージが送信されるというものです。

解決方法

この問題を訂正するには、既存のプロセス・インスタンスと一致する相関セット値を渡す必要があります。

ビジネス・プロセスでの相関セットの使用法について詳しくは、「Correlation sets in BPEL processes」を参照してください。

Java スニペットの未初期化変数または `NullPointerException`:

ビジネス・プロセスで未初期化変数を使用すると、さまざまな例外が発生します。

症状

次のような例外が発生します。

- 変数の内容を読み取ったり操作したりする Java スニペットまたは Java 式の実行時に、`NullPointerException` がスローされます。
- `assign`、`invoke`、`reply`、または `throw` アクティビティーの実行時に、BPEL の標準障害「`uninitializedVariable`」(メッセージ `CWWBE0068E`) がスローされます。

理由

ビジネス・プロセスのすべての変数は、プロセスが開始されるときに `NULL` 値を持っており、それらの変数は事前初期化されません。Java スニペットまたは Java 式の内で未初期化変数を使用すると、`NullPointerException` が発生します。

解決方法

変数は、使用する前に初期化する必要があります。これは、`assign` アクティビティーで実行できます。例えば、変数は `assign` の `to-spec` に現れる必要があります。あるいは、Java スニペットの内側で変数を初期化することが可能です。

応答例外の欠落 (メッセージ: `CWWBE0071E`):

Microflow または長期実行プロセスを実行すると、`MissingReplyException` (メッセージ: `CWWBE0071E`) が発生するか、この例外がシステム・ログまたは `SystemOut.log` ファイルで検出されます。

理由

両方向オペレーションでは応答を送信する必要があります。このエラーは、プロセスが `reply` アクティビティーをナビゲートせずに終了する場合に生成されます。この状態は、以下の事情の下で発生します。

- `reply` アクティビティーがスキップされた。
- 障害が発生し、対応する障害ハンドラーに `reply` アクティビティーが含まれていない。
- 障害が発生し、対応する障害ハンドラーが存在しない。

解決方法

モデルを訂正し、プロセスの終了前に `reply` アクティビティーが必ず実行されるようにします。

並列パスが順次化される:

flow アクティビティの内側に 2 つ以上の並列 invoke アクティビティがありますが、invoke アクティビティが順次実行されます。

解決方法

- 真の並列処理を実現するには、それぞれのパスを別々のトランザクションに置く必要があります。すべての並列 invoke アクティビティの「トランザクションの振る舞い」属性を、「事前コミット (commit before)」または「所有が必要 (requires own)」に設定します。
- データベース・システムとして Cloudscape を使用している場合、プロセス・エンジンは並列パスの実行を直列化します。この振る舞いを変更することはできません。

ネストされたデータ・オブジェクトを別のデータ・オブジェクトにコピーするとソース・オブジェクトの参照が破棄される:

データ・オブジェクト Father には、別のデータ・オブジェクト Child が含まれています。Java スニペットの内側では、Child を含むオブジェクトがフェッチされ、データ・オブジェクト Mother の副構造で設定されます。データ・オブジェクト Father での Child への参照は消失します。

理由

Child への参照は、Father から Mother に移されます。

解決方法

上記のようなデータ形式変更を Java スニペットで実行する場合は、別のオブジェクトに割り当てられる前にそのデータ・オブジェクトをコピーします。以下のコード・スニペットはその方法を示しています。

```
BOCopy copyService = (BOCopy)ServiceManager.INSTANCE.locateService
    ("com/ibm/websphere/bo/BOCopy");
DataObject Child = Father.get("Child");
DataObject BCopy = copyService.copy(Child);
Mother.set("Child", BCopy);
```

CScope が使用不可である:

長期実行プロセスでの Microflow の開始またはナビゲーション・ステップの実行が失敗し、「事後条件違反 !(cscope != null) (postcondition violation !(cscope != null))」というアサーションが表示されます。

理由

特定の状態ではプロセス・エンジンは補正サービスを使用しますが、これは使用不可でした。

解決方法

管理についての PDF に記載のとおり、補正サービスを使用可能に設定します。

プロセス関連またはタスク関連メッセージの操作

ディスプレイに表示またはログ・ファイルに書き込まれる Business Process Choreographer メッセージの詳細情報を取得する方法について説明します。

Business Process Choreographer に属するメッセージのプレフィックスは、プロセス関連メッセージの場合は *CWWB*、タスク関連メッセージの場合は *CWTK* です。これらのメッセージのフォーマットは、*PrefixComponentNumberTypeCode* です。タイプ・コードは、以下のとおりです。

- I** 情報メッセージ
- W** 警告メッセージ
- E** エラー・メッセージ

プロセスおよびタスクが実行されると、メッセージは Business Process Choreographer Explorer に表示されるか、SystemOut.log ファイルおよびトレースに追加されます。これらのファイルで提供されるメッセージ・テキストを使用しても問題を解決できない場合は、WebSphere Application Server 症状データベースを使用して詳細な情報を検索できます。Business Process Choreographer メッセージを表示するには、WebSphere ログ・アナライザーを使用して activity.log ファイルを確認します。

1. WebSphere ログ・アナライザーを開始します。

次のスクリプト `install_root/bin/waslogbr.sh` を実行します。

2. オプション: 「ファイル」 > 「データベースの更新」 > 「WebSphere Application Server 症状データベース」をクリックして、症状データベースの最新バージョンを確認します。
3. オプション: アクティビティ・ログをロードします。
 - a. アクティビティ・ログ・ファイルを選択します。
 - `install_root/profiles/profile_name/logs/activity.log` ファイル
 - b. 「開く (Open)」をクリックします。

Business Process Choreographer Explorer のトラブルシューティング

このトピックを参照して、Business Process Choreographer Explorer に関連した問題を解決します。

以下の情報を使用して、Business Process Choreographer Explorer に関連した問題を解決します。

- ブラウザーを使って Business Process Choreographer Explorer にアクセスしようとすると、エラー・メッセージ「HTTP 404 - File not found」が表示される場合は、次の操作を試してください。
 - 管理コンソールを使用して、Web クライアント・アプリケーション `BPCExplorer_node_name_server_name` がサーバー上に実際にデプロイされ、実行されていることを確認します。
 - 管理コンソールのアプリケーションのページにある「デプロイメント記述子の表示」下で、コンテキスト・ルートが `/bpc` になっていることを確認します。

- Business Process Choreographer Explorer の使用中にエラー・メッセージが表示される場合は、エラー・ページで「詳しくは、**検索してください**」リンクをクリックします。これにより、IBM 技術サポート・サイトでエラー・コードの検索を開始します。このサイトで提供される情報は英語のみです。Business Process Choreographer Explorer エラー・ページに表示されるエラー・メッセージ・コードをクリップボードにコピーします。エラー・コードの形式は CWWBcnnnc で、c は文字を、nnnn は 4 桁の数値を示します。「WebSphere Process Server technical support」ページに進みます。エラー・コードを「**追加検索項目 (Additional search terms)**」フィールドに貼り付けて、「**実行 (Go)**」をクリックします。
- EngineMissingReplyException メッセージが表示される場合、これはプロセス・モデルに関する問題の症状です。この問題の解決方法について詳しくは、『ビジネス・プロセスおよびヒューマン・タスクの管理のトラブルシューティング』を参照してください。
- Business Process Choreographer Explorer にログオンできるものの、一部の項目が表示されない場合、または特定のボタンが使用できない場合、これはユーザーの権限に問題があることを示しています。

この問題の解決方法として、次の方法が考えられます。

- 管理コンソールを使用して、セキュリティーをオンにする。
- 正しい ID を使用して Business Process Choreographer Explorer にログオンしているか確認する。プロセス管理者ではないユーザー ID でログオンすると、すべての管理ビューおよびオプションは非表示または使用不可になります。
- WebSphere Integration Developer を使用して、ビジネス・プロセスに定義されている権限設定を確認または変更する。
- エラー・メッセージ WWBU0024E 理由:「命名例外」で、ローカルのビジネス・プロセス EJB への接続を確立できませんでした。(WWBU0024E Could not establish a connection to local business process EJB with a reason: "Naming Exception".) このエラーは、ビジネス・プロセス・コンテナが停止していることを示す可能性があります。アプリケーション BPEContainer_InstallScope が実行中であることを確認してください。ここで、InstallScope は cluster_name または hostname_servername です。

関連タスク

363 ページの『ビジネス・プロセスの実行のトラブルシューティング』

ここでは、ビジネス・プロセスの実行に関する共通の問題の解決方法について説明します。

ビジネス・プロセスおよびヒューマン・タスクの管理のトラブルシューティング

ここでは、ビジネス・プロセスに共通の問題を解決する方法について説明します。

次の情報は、ビジネス・プロセスの問題をデバッグするのに役立ちます。

ビジネス・プロセス・アプリケーションにまだプロセス・インスタンスがある間にそのアプリケーションを停止しようとする、管理コンソールは、応答を停止します。アプリケーションを停止する前に、ビジネス・プロセスを停止して新規インスタンスが作成されないようにし、次のいずれかの操作を実行する必要があります。

- すべての既存のプロセス・インスタンスが、正常な方法で終了するのを待つ。

- すべてのプロセス・インスタンスを強制終了して削除する。

これらの操作を行った後でのみ、プロセス・アプリケーションを停止できます。この問題を回避する方法については、「technote 1166009」を参照してください。

プロセス関連およびタスク関連の監査証跡情報の使用

ビジネス・プロセスおよびヒューマン・タスクのイベント・タイプとデータベース構造について説明します。

ビジネス・プロセス・コンテナとタスク・コンテナの一方または両方で、ロギングを使用可能にする必要があります。

ロギングが使用可能に設定されている場合には、ビジネス・プロセスまたはヒューマン・タスクの実行時に重要なステップが発生すると常に、情報が監査ログまたは Common Event Infrastructure (CEI) ログに書き込まれます。CEI について詳しくは、モニターについての PDF を参照してください。以降のトピックでは、ビジネス・プロセスおよびヒューマン・タスクのイベント・タイプとデータベース構造について説明します。

ビジネス・プロセスの監査イベント・タイプ:

ここでは、ビジネス・プロセスの処理中に監査ログに書き込むことができるイベントのタイプについて説明します。

イベントをログに記録するには、次の条件を満たす必要があります。

- ビジネス・プロセス・コンテナ用の対応する監査ロギング・タイプが使用可能になっている
- プロセス・モデルの対応するエンティティ用にイベントが必ず使用可能になっている

次の表に、ビジネス・プロセスの実行中に発生する可能性がある監査イベントのコード一覧を示します。

表 13. プロセス・インスタンス・イベント

監査イベント	イベント・コード
PROCESS_STARTED	21000
PROCESS_SUSPENDED	21001
PROCESS_RESUMED	21002
PROCESS_COMPLETED	21004
PROCESS_TERMINATED	21005
PROCESS_RESTARTED	21019
PROCESS_DELETED	21020
PROCESS_FAILED	42001
PROCESS_COMPENSATING	42003
PROCESS_COMPENSATED	42004
PROCESS_TERMINATING	42009
PROCESS_FAILING	42010

表 13. プロセス・インスタンス・イベント (続き)

監査イベント	イベント・コード
PROCESS_CORRELATION_SET_INITIALIZED	42027
PROCESS_COMPENSATION_INDOUBT	42030
PROCESS_WORKITEM_DELETED	42041
PROCESS_WORKITEM_CREATED	42042
PROCESS_COMPENSATION_FAILED	42046
PROCESS_EVENT_RECEIVED	42047
PROCESS_EVENT_ESCALATED	42049
PROCESS_WORKITEM_TRANSFERRED	42056

表 14. アクティビティ・イベント

監査イベント	イベント・コード
ACTIVITY_READY	21006
ACTIVITY_STARTED	21007
ACTIVITY_COMPLETED	21011
ACTIVITY_CLAIM_CANCELED	21021
ACTIVITY_CLAIMED	21022
ACTIVITY_TERMINATED	21027
ACTIVITY_FAILED	21080
ACTIVITY_EXPIRED	21081
ACTIVITY_LOOPED	42002
ACTIVITY_SKIPPED	42005
ACTIVITY_TERMINATING	42008
ACTIVITY_FAILING	42011
ACTIVITY_OUTPUT_MESSAGE_SET	42012
ACTIVITY_FAULT_MESSAGE_SET	42013
ACTIVITY_STOPPED	42015
ACTIVITY_FORCE_RETRIED	42031
ACTIVITY_FORCE_COMPLETED	42032
ACTIVITY_UNDO_STARTED	42033
ACTIVITY_UNDO_SKIPPED	42034
ACTIVITY_UNDO_COMPLETED	42035
ACTIVITY_MESSAGE_RECEIVED	42036
ACTIVITY_LOOP_CONDITION_TRUE	42037
ACTIVITY_LOOP_CONDITION_FALSE	42038
ACTIVITY_WORKITEM_DELETED	42039
ACTIVITY_WORKITEM_CREATED	42040
ACTIVITY_ESCALATED	42050
ACTIVITY_WORKITEM_REFRESHED	42054
ACTIVITY_WORKITEM_TRANSFERRED	42055

表 15. 変数関連イベント

監査イベント	イベント・コード
VARIABLE_UPDATED	21090

表 16. 制御リンク・イベント

監査イベント	イベント・コード
LINK_EVALUATED_TO_TRUE	21034
LINK_EVALUATED_TO_FALSE	42000

表 17. プロセス・テンプレート・イベント

監査イベント	イベント・コード
PROCESS_INSTALLED	42006
PROCESS_UNINSTALLED	42007

表 18. スコープ・インスタンス・イベント

監査イベント	イベント・コード
SCOPE_STARTED	42020
SCOPE_SKIPPED	42021
SCOPE_FAILED	42022
SCOPE_FAILING	42023
SCOPE_TERMINATED	42024
SCOPE_COMPLETED	42026
SCOPE_COMPENSATING	42043
SCOPE_COMPENSATED	42044
SCOPE_COMPENSATION_FAILED	42045
SCOPE_EVENT_RECEIVED	42048
SCOPE_EVENT_ESCALATED	42051

ヒューマン・タスクの監査イベント・タイプ:

ここでは、ヒューマン・タスクの処理中に監査ログに書き込むことができるイベントのタイプについて説明します。

イベントをログに記録するには、次の条件を満たす必要があります。

- ヒューマン・タスク・コンテナ用の対応する監査ロギング・タイプが使用可能になっている
- タスク・モデルの対応するエンティティ用にイベントが必ず使用可能になっている

次の表に、ヒューマン・タスクの実行中に発生する可能性がある監査イベントのコード一覧を示します。

表 19. タスク・インスタンス・イベント

監査イベント	イベント・コード
TASK_CREATED	51001
TASK_DELETED	51002
TASK_STARTED	51003
TASK_COMPLETED	51004
TASK_CLAIM_CANCELLED	51005
TASK_CLAIMED	51006
TASK_TERMINATED	51007
TASK_FAILED	51008
TASK_EXPIRED	51009
TASK_WAITING_FOR_SUBTASK	51010
TASK_SUBTASKS_COMPLETED	51011
TASK_RESTARTED	51012
TASK_SUSPENDED	51013
TASK_RESUMED	51014
TASK_COMPLETED_WITH_FOLLOW_ON	51015
TASK_UPDATED	51101
TASK_OUTPUT_MESSAGE_UPDATED	51103
TASK_FAULT_MESSAGE_UPDATED	51104
TASK_WORKITEM_DELETED	51201
TASK_WORKITEM_CREATED	51202
TASK_WORKITEM_TRANSFERRED	51204
TASK_WORKITEM_REFRESHED	51205

表 20. タスク・テンプレート・イベント

監査イベント	イベント・コード
TASK_TEMPLATE_INSTALLED	52001
TASK_TEMPLATE_UNINSTALLED	52002

表 21. エスカレーション・インスタンス・イベント

監査イベント	イベント・コード
ESCALATION_FIRED	53001
ESCALATION_WORKITEM_DELETED	53201
ESCALATION_WORKITEM_CREATED	53202
ESCALATION_WORKITEM_TRANSFERRED	53204
ESCALATION_WORKITEM_REFRESHED	53205

ビジネス・プロセス用監査証跡データベース・ビューの構造:

AUDIT_LOG_B データベース・ビューは、ビジネス・プロセスについての監査ログ情報を提供します。

監査証跡の内容を読み取るには、データベース表とビューの読み取りをサポートする SQL または他の管理ツールを使用します。

監査イベントは、プロセス・エンティティと関連があります。監査イベントのタイプは、そのイベントが参照するエンティティによって異なります。監査イベントには、次のタイプがあります。

- プロセス・インスタンス・イベント (PIE)
- アクティビティ・インスタンス・イベント (AIE)
- 変数関連イベント (VAR)
- 制御リンク・イベント (CLE)
- プロセス・テンプレート・イベント (PTE)
- スコープ関連イベント (SIE)

監査イベント・タイプのコード一覧については、372 ページの『ビジネス・プロセスの監査イベント・タイプ』を参照してください。

以下の表に、AUDIT_LOG_B 監査証跡ビューの構造を示します。表では、列名とイベント・タイプをリストし、列について簡単に説明しています。

インライン・タスクは、AUDIT_LOG_B 監査証跡ビューに記録され、TASK_LOG 監査証跡ビューには記録されません。例えば、インライン参加タスクを要求すると ACTIVITY_CLAIMED イベントが生成されますが、タスク関連イベントは生成されません。

表 22. AUDIT_LOG_B 監査証跡ビューの構造

名前	PTE	PIE	AIE	VAR	CLE	SIE	説明
ALID	x	x	x	x	x	x	監査ログ・エントリーの ID。
EVENT_TIME	x	x	x	x	x	x	イベントが発生したときのタイム・スタンプ (協定世界時 (UTC) 形式)。
EVENT_TIME_UTC	x	x	x	x	x	x	イベントが発生したときのタイム・スタンプ (協定世界時 (UTC) 形式)。
AUDIT_EVENT	x	x	x	x	x	x	発生したイベントのタイプ。
PTID	x	x	x	x	x	x	現行イベントに関連したプロセスのプロセス・テンプレート ID。
PIID		x	x	x	x	x	現行イベントに関連したプロセス・インスタンスのプロセス・インスタンス ID。
VARIABLE_NAME				x			現行イベントに関連した変数の名前。
SIID						x	イベントに関連したスコープ・インスタンスの ID。
PROCESS_TEMPL_NAME	x	x	x	x	x	x	現行イベントに関連したプロセス・テンプレートのプロセス・テンプレート名。
TOP_LEVEL_PIID		x	x	x	x	x	現行イベントに関連したトップレベル・プロセスの ID。
PARENT_PIID		x	x	x	x	x	親プロセスのプロセス・インスタンス ID。親が存在しない場合は NULL。

表 22. AUDIT_LOG_B 監査証跡ビューの構造 (続き)

名前	PTE	PIE	AIE	VAR	CLE	SIE	説明
VALID_FROM	x	x	x	x	x	x	現行イベントに関連したプロセス・テンプレートの有効開始日付。
VALID_FROM_UTC	x	x	x	x	x	x	現行イベントに関連したプロセス・テンプレートの有効開始日付 (協定世界時 (UTC) 形式)。
ATID			x				現行イベントに関連したアクティビティ・テンプレートの ID。
ACTIVITY_NAME			x			x	イベントが発生したアクティビティ名。
ACTIVITY_KIND			x				<p>アクティビティが発生したときのアクティビティの種類。考えられる値は次のとおりです。</p> <p>KIND_EMPTY 3 KIND_INVOKE 21 KIND_RECEIVE 23 KIND_REPLY 24 KIND_THROW 25 KIND_TERMINATE 26 KIND_WAIT 27 KIND_COMPENSATE 29 KIND_SEQUENCE 30 KIND_SWITCH 32 KIND_WHILE 34 KIND_PICK 36 KIND_FLOW 38 KIND_SCRIPT 42 KIND_STAFF 43 KIND_ASSIGN 44 KIND_CUSTOM 45 KIND_RETHROW 46</p> <p>これらは、ActivityInstanceData.KIND_* で定義される定数です。</p>

表 22. AUDIT_LOG_B 監査証跡ビューの構造 (続き)

名前	PTE	PIE	AIE	VAR	CLE	SIE	説明
ACTIVITY_STATE			x				<p>イベントに関連したアクティビティの状態。考えられる値は次のとおりです。</p> <p>STATE_INACTIVE 1 STATE_READY 2 STATE_RUNNING 3 STATE_SKIPPED 4 STATE_FINISHED 5 STATE_FAILED 6 STATE_TERMINATED 7 STATE_CLAIMED 8 STATE_TERMINATING 9 STATE_FAILING 10 STATE_WAITING 11 STATE_EXPIRED 12 STATE_STOPPED 13</p> <p>これらは、 ActivityInstanceData.STATE_* で定義される定数です。</p>
CONTROL_LINK_NAME					x		<p>現行リンク・イベントに関連したリンクの名前。</p>
PRINCIPAL		x	x	x	x	x	<p>プリンシパルの名前。 PROCESS_DELETED イベントの場合には設定されません。</p>
VARIABLE_DATA				x			<p>variable updated イベント用の変数のデータ。</p>
EXCEPTION_TEXT		x	x			x	<p>アクティビティまたはプロセスが失敗する原因となった例外メッセージ。次の場合に適用されます。</p> <p>PROCESS_FAILED ACTIVITY_FAILED SCOPE_FAILED</p>
DESCRIPTION		x	x	x	x	x	<p>潜在的に解決される可能性のある置換変数を含むアクティビティまたはプロセスの説明。</p>
CORR_SET_INFO		x					<p>プロセスの開始時に初期化された相関セットのストリング表現。 processCorrelationSetInitialized イベント (42027) により出力されます。</p>

表 22. AUDIT_LOG_B 監査証跡ビューの構造 (続き)

名前	PTE	PIE	AIE	VAR	CLE	SIE	説明
USER_NAME		x	x				<p>作業項目が変更されたユーザーの名前。次のイベントの場合に適用されません。</p> <ul style="list-style-type: none"> • プロセス・インスタンスの作業項目が削除された • アクティビティ・インスタンスの作業項目が削除された • プロセス・インスタンスの作業項目が作成された • アクティビティ・インスタンスの作業項目が作成された

表 22. AUDIT_LOG_B 監査証跡ビューの構造 (続き)

名前	PTE	PIE	AIE	VAR	CLE	SIE	説明
ADDITIONAL_ INFO		x	x			x	<p>このフィールドの内容は、以下のイベントのタイプによって決まります。</p> <p>ACTIVITY_WORKITEM_TRANSFERRED、 PROCESS_WORK_ITEM_TRANSFERRED 作業項目を受け取ったユーザーの名前。</p> <p>ACTIVITY_WORKITEM_CREATED、 ACTIVITY_WORKITEM_REFRESHED、 ACTIVITY_ESCALATED 作業項目を作成または更新する対象となったユーザーすべてのリスト (「,」区切り)。リストのユーザーが 1 人のみの場合は、USER_NAME フィールドにこのユーザーの名前が入力されます。リストのユーザーが 1 人のみの場合は、USER_NAME フィールドにこのユーザーのユーザー名が入力され、ADDITIONAL_INFO フィールドは空 (NULL) になります。</p> <p>PROCESS_EVENT_RECEIVED、 SCOPE_EVENT_RECEIVED 選択可能な場合は、イベント・ハンドラーが受信したオペレーションのタイプ。使用形式は次のとおりです。'{'ポート・タイプ・ネーム・スペース '}' ポート・タイプ名 ':' オペレーション名。このフィールドは、「onAlarm」イベントの場合には設定されません。</p>

ヒューマン・タスク用監査証跡データベース・ビューの構造:

TASK_AUDIT_LOG データベース・ビューは、ヒューマン・タスクについての監査ログ情報を提供します。

インライン・タスクは、AUDIT_LOG_B ビューに記録され、これ以外のすべてのタスク・タイプは TASK_AUDIT_LOG ビューに記録されます。

監査証跡の内容を読み取るには、データベース表とビューの読み取りをサポートする SQL または他の管理ツールを使用します。

監査イベントは、タスク・エンティティと関連があります。監査イベントのタイプは、そのイベントが参照するエンティティによって異なります。監査イベントには、次のタイプがあります。

- タスク・インスタンス・イベント (TIE)
- タスク・テンプレート・イベント (TTE)
- エスカレーション・インスタンス・イベント (EIE)

以下の表に、TASK_AUDIT_LOG 監査証跡ビューの構造を示します。表では、列名とイベント・タイプをリストし、列について簡単に説明しています。

インライン・タスクは、AUDIT_LOG_B 監査証跡ビューに記録され、TASK_AUDIT_LOG 監査証跡ビューには記録されません。例えば、インライン参加タスクを要求すると ACTIVITY_CLAIMED イベントが生成されますが、タスク関連イベントは生成されません。

表 23. TASK_AUDIT_LOG 監査証跡ビューの構造

名前	TIE	TTE	EIE	説明
ALID	x	x	x	監査ログ・エントリーの ID。
AUDIT_EVENT	x	x	x	発生したイベントのタイプ。監査イベント・コードの一覧については、374 ページの『ヒューマン・タスクの監査イベント・タイプ』を参照してください。
CONTAINMENT_ CTX_ID	x	x		収容コンテキストの ID (ACOID、PTID、または PIID など)。
ESIID			x	現行イベントに関連したエスカレーション・インスタンスの ID。
ESTID			x	現行イベントに関連したエスカレーション・テンプレートの ID。
EVENT_TIME	x	x	x	イベントが発生したときの時刻 (協定世界時 (UTC) 形式)。
FAULT_NAME	x			障害メッセージの名前。この属性は、次のイベントの場合に適用されます。 TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FAULT_NAME_SPACE	x			障害メッセージ・タイプのネーム・スペース。この属性は、次のイベントの場合に適用されます。 TASK_FAILED TASK_FAULT_MESSAGE_UPDATED
FOLLOW_ON_TKIID	x			後続のタスク・インスタンスの ID。
NAME	x	x	x	イベントに関連付けられたタスク・インスタンス、タスク・テンプレート、またはエスカレーション・インスタンスの名前。

表 23. TASK_AUDIT_LOG 監査証跡ビューの構造 (続き)

名前	TIE	TTE	EIE	説明
NAMESPACE	x	x		イベントに関連付けられたタスク・インスタンス、タスク・テンプレート、またはエスカレーション・インスタンスのネーム・スペース。
NEW_USER				転送済み作業項目の新規所有者。この属性は、次のイベントの場合に適用されます。
	x			TASK_WORKITEM_CREATED
	x			TASK_WORKITEM_TRANSFERRED
			x	ESCALATION_WORKITEM_CREATED
			x	ESCALATION_WORKITEM_TRANSFERRED
OLD_USER				転送済み作業項目の前の所有者。この属性は、次のイベントの場合に適用されます。
	x			TASK_WORKITEM_TRANSFERRED
	x			TASK_WORKITEM_DELETED
			x	ESCALATION_WORKITEM_TRANSFERRED
			x	ESCALATION_WORKITEM_DELETED
PARENT_CONTEXT_ID	x			タスクの親コンテキスト (例えば、アクティビティ・テンプレートやタスク・インスタンス) の ID。サブタスクおよび後続タスクの場合にのみ設定されます。
PARENT_TASK_NAME	x			親タスク・インスタンスまたはテンプレートの名前。サブタスクおよび後続タスクの場合にのみ設定されます。
PARENT_TASK_NAMESP	x			親タスク・インスタンスまたはテンプレートのネーム・スペース。サブタスクおよび後続タスクの場合にのみ設定されます。
PARENT_TKIID	x			親タスク・インスタンスの ID。
PRINCIPAL	x	x	x	出したリクエストによってイベントがトリガーされたプリンシパルの名前。
TASK_KIND	x	x		タスクの種類。考えられる値は次のとおりです。 KIND_HUMAN 101 KIND_ORIGINATING 103 KIND_PARTICIPATING 105 KIND_ADMINISTRATIVE 106

表 23. TASK_AUDIT_LOG 監査証跡ビューの構造 (続き)

名前	TIE	TTE	EIE	説明
TASK_STATE	x			<p>タスクまたはタスク・テンプレートの状態。タスク・テンプレートの場合に考えられる値は次のとおりです。</p> <p>STATE_STARTED 1 STATE_STOPPED 2</p> <p>タスク・インスタンスの場合に考えられる値は次のとおりです。</p> <p>'1' : 'STATE_INACTIVE' '2' : 'STATE_READY' '3' : 'STATE_RUNNING' '5' : 'STATE_FINISHED' '6' : 'STATE_FAILED' '7' : 'STATE_TERMINATED' '8' : 'STATE_CLAIMED' '12' : 'STATE_EXPIRED' '101' : 'FORWARDED'</p>
TKIID	x		x	タスク・インスタンスの ID。
TKTID	x	x		タスク・テンプレートの ID。
TOP_TKIID	x			トップ・タスク・インスタンスの ID。
VALID_FROM		x		現行イベントに関連したタスク・テンプレートの有効開始日付。
WORK_ITEM_REASON	x		x	<p>作業項目の割り当て理由。考えられる値は次のとおりです。</p> <p>POTENTIAL_OWNER 1 EDITOR 2 READER 3 OWNER 4 POTENTIAL_STARTER 5 STARTER 6 ADMINISTRATOR 7 POTENTIAL_SENDER 8 ORIGINATOR 9 ESCALATION_RECEIVER 10 POTENTIAL_INSTANCE_CREATOR 11</p> <p>理由は、作業項目に関連したすべてのイベントの場合に設定されます。例えば、ESCALATION_RECEIVER はエスカレーション作業項目に関連したイベントの場合に設定される一方で、他の理由はタスク作業項目関連のイベントに適用されます。</p>

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・マネージャーに問題が生じた場合には、ログイン・エラー、ログインの競合、アクセスの競合などについて検証します。

ログイン・エラー

ログイン時に、ログイン・エラー・メッセージを受信します。

以下はログイン・エラー・メッセージです。

ログインを処理できません。ユーザー ID とパスワードを調べて、やり直してください。

このエラーは、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発生します。

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る場合にのみ発生します。

1. エラー・メッセージで「OK」をクリックします。

ログイン・ページに戻ります。

2. 有効なユーザー ID とパスワードを入力してください。

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か確認してください。

ユーザー ID とパスワードのスペルが正しいか確認してください。

ユーザー ID とパスワードが正しいかどうかシステム管理者に確認してください。

3. 「ログイン」ボタンをクリックします。

ログイン・エラーが解決したら、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。エラーが解決できない場合は、システム管理者にお問い合わせください。

ログイン競合エラー

このイベントは、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。

以下はログイン競合メッセージです。

現在、別のユーザーが同じユーザー ID でログインしています。次のオプション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。

選択できるオプションは 3 つあります。

- 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て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非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。

進行中の作業を継続して行う場合、このオプションを使用します。前のセッションの保管していない非公開のローカル変更はすべて失われません。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

一人のユーザーがデータベース内のビジネス・ルールを更新している時に、別のユーザーが同時に同じルールを更新しようとするするとアクセス競合が発生します。

ローカル側の変更をデータベースに公開すると、このエラーが報告されます。

以下はアクセス競合エラーを訂正する操作です。

- 「ルール」ページを公開します。
- エラーの原因となっているビジネス・ルールのソースを検出し、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、自身の変更は必要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継続する場合、データベースからエラーが発生したルール・ページを再ロードする必要があります。これは、エラーが発生したルール・ページのローカル変更が使用できなくなるためです。エラー状態でない他のルール・ページのローカル変更は引き続き使用できます。
- エラーが報告されたルールの「公開および復帰」ページで「再ロード」をクリックし、ルール・ページを再ロードします。

Common Base Event ブラウザーのトラブルシューティング

主に 4 つの条件下では、Common Base Event ブラウザーにアクセスすることができません。

条件

「サーバーが見つかりません (Cannot find server)」

WebSphere Process Server (またはネットワーク・サーバー) が使用不可です。イベント・ブラウザー URI を起動しようすると、「サーバーが見つかりません (Cannot find server)」ブラウザー・ページが戻され、サーバーが使用不可であることが示されます。この場合は、IBM ヘルプ・デスクに連絡して、問題の原因を判別する必要があります。

「ファイルが見つかりません」

WebSphere Process Server は使用可能ですが、イベント・ブラウザ・アプリケーションがサーバーにインストールされていないか、始動していません。イベント・ブラウザ URI を起動しようとする、と、「ファイルが見つかりません」ブラウザ・ページが戻され、サーバーは使用可能であっても、そのサーバーでは URI が選択不可であることが示されます。この場合は、IBM ヘルプ・デスクに連絡して、問題の原因を判別する必要があります。

ログオン・パネルが表示される

WebSphere Process Server とイベント・ブラウザは使用可能ですが、正しいロールをマップしてイベント・ブラウザにアクセスできるようになっていません。ログオン・パネルによってプロンプトが出されます。ユーザー ID とパスワードを入力してログインしようとしても、ログインに失敗します。この場合は、IBM ヘルプ・デスクに連絡して、イベント・ブラウザを起動するための正当な許可を取得する必要があります。

「イベント・データの取得 (Get event data)」パネルにエラー・メッセージが表示される WebSphere Process Server とイベント・ブラウザが使用可能で、ユーザーもアクセスするための正当な許可を持っていますが、Common Event Infrastructure サーバーが使用不可です。「イベントの取得」ボタンをクリックすると、イベント・ブラウザの「イベントの取得」パネルにエラー・メッセージが表示されます。エラー情報は、メッセージ・ログに記録されません。

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。

〒106-0032
東京都港区六本木 3-2-31
IBM World Trade Asia Corporation
Licensing

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation 577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができませんが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

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。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

プログラミング・インターフェース情報は、プログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

警告: 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

商標

以下は、IBM Corporation の商標です。IBM、IBM LOGO、AIX、CICS、Cloudscape、DB2、DB2 Connect、DB2 Universal Database、developerWorks、IMS、Informix、iSeries、Lotus、Lotus Domino、MQSeries、MVS、OS/390、Passport Advantage、pSeries、Rational、Redbooks、Tivoli、WebSphere、z/OS、zSeries

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または登録商標です。

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 z/OS バージョン 6.0.1



Printed in Japan