

バージョン 6.0.1



製品概説

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、43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 (製品番号 5655-N53) バージョン 6、リリース 0、モディフィケーション 1、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せん。

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
Product Overview
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

製品の概要	1
WebSphere Process Server の概要	1
このリリースの新機能	2
製品ファミリーの概要	3
WebSphere Process Server のアーキテクチャーの概要	4
サービス指向アーキテクチャー・コア	5
サポート・サービス	9
サービス・コンポーネント	11
インポートおよびエクスポートとアダプター	14
クラスターおよびワークロード管理	15
エンタープライズ・サービス・バスの概要	16
バス環境の概要	16
クライアント	18
メディエーション・モジュール	19
メディエーション・プリミティブ	22
サービス・メッセージ・オブジェクト	24
統合アプリケーションの開発およびデプロイメント	29
WebSphere Process Server におけるアプリケーションの管理	30
WebSphere Process Server におけるセキュリティー	31
WebSphere Process Server におけるパフォーマンス・モニター	31
サンプルとチュートリアル	31
チュートリアル	32
サンプル (サンプル・ギャラリー) へのアクセス	32
標準への準拠	34
アクセシビリティ機能	34
連邦情報処理標準	35
Common Criteria	35
インターネット・プロトコル・バージョン 6.0	36
グローバル化	36
特記事項	43
プログラミング・インターフェース情報	45
商標	45

製品の概要

概説および概要の資料へのリンクを使用して、IBM WebSphere Process Server の概要を理解することができます。

WebSphere Process Server 資料 (PDF 形式)

WebSphere® Process Server のハードウェア要件およびソフトウェア要件は、WebSphere Process Server system requirements の Web サイトにあります。

WebSphere Process Server の情報ロードマップは、IBM® developerWorks® Web サイトの WebSphere Business Integration information roadmaps の下にあります。

概要の資料には、製品の概要、新機能、WebSphere Process Server と共に動作するその他の IBM 製品に関する情報、アーキテクチャーやコンポーネントの技術概要、さらにサンプル、標準への準拠、およびグローバル化に関する情報が含まれています。

WebSphere Process Server の概要

IBM WebSphere Process Server は、実証されたビジネス・インテグレーション概念、アプリケーション・サーバー・テクノロジー、および最新のオープン・スタンダードから発展した、次世代のビジネス・プロセス統合サーバーです。

IBM WebSphere Process Server は、サービス指向アーキテクチャー (SOA) をサポートしているので、異なるテクノロジーを使用したビジネス・インテグレーションに必要なビジネス・アプリケーションにとって理想的なプラットフォームです。

WebSphere Integration Developer ツール・セットを使用すれば、Service Component Architecture (SCA) プログラミング・モデルや Service Data Objects (SDO) データ・モデルなどの単純化された統合メカニズムを使用してビジネス・インテグレーション・ソリューションを作成することができます。SDO ビジネス・オブジェクトは、SCA コンポーネントを使用して定義、変換、送付、およびマップできます。

WebSphere Adapters は、バックエンドのエンタープライズ情報システム (EIS) への接続を提供します。WebSphere Process Server では、ビジネス・インテグレーション・アプリケーションは Web サービスに基づくビジネス・ロジックおよびプロセスを定義できます (BPEL (Business Process Execution Language)、ヒューマン・タスク、およびビジネス・ルールなど)。ビジネス・インテグレーション・ソリューションのランタイム・モニターに関して、WebSphere Process Server は Common Event Infrastructure (CEI) を提供します。CEI は、これらのアプリケーションで発生する可能性があるさまざまなイベントのモニターを集中化します。

WebSphere Process Server では、標準ベースの統合アプリケーションを SOA 構造でデプロイすることができます。SOA は、常用しているビジネス・アプリケーションを使用して、そのアプリケーションをサービスと呼ばれる個々のビジネス機能やビジネス・プロセスに細分するアプリケーション・フレームワークです。SOA は、アプリケーションが提供するコンポーネントとサービスに関して、基礎にあるコンポーネント、サービス、およびコンポーネント間接続のインプリメンテーションに

関係なく、ソフトウェア・システムの構造を概念的に記述するものです。SOA に基づく疎結合統合アプリケーションは、柔軟性と敏捷性を備えています。統合ソリューションは、プラットフォーム、プロトコル、および製品から独立してインプリメントできます。SOA について詳しくは、IBM Web サイトの Service Oriented Architecture (SOA) を参照してください。

ハードウェアおよびソフトウェア要件

WebSphere Process Server でサポートされるハードウェアおよびソフトウェアについての公式の説明を見るには、WebSphere Process Server system requirements の Web サイトにアクセスしてください。

情報ロードマップ

WebSphere Process Server は、WebSphere Integration Developer で開発したデプロイ・ソリューションを WebSphere Process Server に使用するソリューション・デプロイヤー、管理コンソールを使用してソリューションを管理するソリューション管理者、管理コンソールを使用してソリューションを操作するオペレーターなど、さまざまなユーザーが使用できます。

製品のインフォメーション・センターの内部と外部にある使用可能な情報源にナビゲートするために、WebSphere Process Server ロードマップは IBM developerWorks からオンラインで使用可能です。これらのロードマップには、ユーザー・ロールに基づく高水準のゴールがリストされ、WebSphere Process Server でゴールに到達するために有用な文書リソースが示されます。

WebSphere Process Server の情報ロードマップを見るには、IBM developerWorks の WebSphere Business Integration information roadmaps ページにアクセスしてください。

このリリースの新機能

IBM WebSphere Process Server for z/OS バージョン 6.0.1 には、いくつかの新機能があります。

IBM WebSphere Process Server for z/OS バージョン 6.0.1 へようこそ。この製品には、以下の新機能があります。

- Service Component Architecture - 既存の IT を利用する単純化された統合フレームワーク。
- IBM WebSphere Integration Developer バージョン 6.0 (コンポーネントの開発、アセンブリー、統合テスト、およびデプロイメント用のビジュアル・エディターが付属した新しい簡易開発ツール) で開発されたアプリケーションのためのデプロイメント環境。
- ヒューマン・タスク、ロール・ベースのタスク割り当て、多層エスカレーションなど、あらゆるスタイルの統合に対するサポート。
- ビジネス・プロセスを最低限のプログラミング・スキルで、アプリケーションを再デプロイすることなく変更できる能力。
- ビジネス・シナリオに基づいてインターフェースを動的に選択するための、ビジネス・ルール、ビジネス状態マシン、およびセレクター。

- IBM WebSphere Adapters の広範囲の統合およびサポート。
- WebSphere Process Server に付属の IBM WebSphere Partner Gateway の制限付きライセンスによる、企業間取引 (B2B) アプリケーションのサポート。

6.0.1+ 機能拡張には以下のものがあります。

- ワークロード管理のためのサーバーのクラスター化。
- エンタープライズ・サービス・バス統合機能: 標準ベースのメッセージング・インフラストラクチャーを提供するサービス指向、メッセージ指向、およびイベント・ドリブンのテクノロジー。

製品ファミリーの概要

IBM WebSphere Process Server は他のいくつかの IBM 製品と連動します。例えば、WebSphere Integration Developer、WebSphere Application Server、WebSphere Adapters、WebSphere Business Integration Adapters、WebSphere Application Server Toolkit、WebSphere Business Modeler、WebSphere Business Monitor、Rational Application Developer、Rational Software Architect、WebSphere Partner Gateway、および WebSphere Portal がこれに該当します。

IBM WebSphere Process Server は、堅固な J2EE 1.4 インフラストラクチャーと、WebSphere Application Server バージョン 6.0 で提供される関連するプラットフォーム・サービスを基にしています。WebSphere Process Server は、WebSphere Application Server Network Deployment バージョン 6.0 上に構築されます。また、WebSphere Process Server は、WebSphere Application Server バージョン 6.0 のインフラストラクチャーおよびプラットフォーム・サービスとも連動します。WebSphere Application Server 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。

IBM WebSphere Integration Developer は、WebSphere Process Server の開発環境です。WebSphere Integration Developer について詳しくは、WebSphere Integration Developer インフォメーション・センターを参照してください。

WebSphere Process Server および WebSphere Integration Developer には、統合アプリケーションのモデル化、構築、デプロイ、インストール、構成、実行、モニター、および管理を可能にする追加の機能が組み込まれています。WebSphere Integration Developer は、IBM WebSphere Business Modeler バージョン 6.0 および IBM WebSphere Business Monitor バージョン 6.0 を補完するもので、IBM Rational Application Developer バージョン 6.0、または IBM Rational® Software Architect バージョン 6.0 と共に使用すると、統合された独自の強力な統合開発プラットフォームを作成することができます。これらの製品について詳しくは、WebSphere Business Modeler インフォメーション・センター、WebSphere Business Monitor インフォメーション・センター、Rational Application Developer インフォメーション・センター、および Rational Software Architect インフォメーション・センターを参照してください。

IBM WebSphere Adapters バージョン 6.0、および IBM WebSphere Business Integration Adapters (IBM WebSphere Business Integration Framework バージョン 2.6 をベースとする) では、既存のエンタープライズ情報システムのインフラストラクチャーと、WebSphere Process Server 上でデプロイしたアプリケーションの統合が

可能です。これらの製品について詳しくは、WebSphere Adapters インフォメーション・センターおよび WebSphere Business Integration Adapters インフォメーション・センターを参照してください。

さらに、CICS® Transaction Gateway および IBM WebSphere Host Access Transformation Services を含む IBM エンタープライズ近代化ポートフォリオを使用して、既存のアプリケーションをエンタープライズ・プロセスで再利用できるように拡張することができます。これらの製品について詳しくは、CICS Transaction Gateway インフォメーション・センターおよび WebSphere Host Access Transformation Services インフォメーション・センターを参照してください。

IBM WebSphere Application Server Toolkit は、WebSphere Process Server における Web サービスのアセンブル、テスト、およびデプロイに役立つ一連の基本ツールです。詳しくは、WebSphere Application Server for z/OS インフォメーション・センターにある WebSphere Application Server Toolkit の資料を参照してください。

WebSphere Process Server で使用する IBM WebSphere Partner Gateway では、企業間取引 (B2B) アプリケーションをサポートします。WebSphere Process Server には、WebSphere Partner Gateway の制限付きライセンスが付属しています。WebSphere Partner Gateway について詳しくは、WebSphere Partner Gateway インフォメーション・センターを参照してください。

IBM WebSphere Portal は、さまざまな管理機能へのアクセスを可能にし、ポートレットが WebSphere Process Server 内のビジネス・プロセスやその他の Service Component Architecture サービスにアクセスできるようにします。WebSphere Portal について詳しくは、WebSphere Portal 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 のアーキテクチャーの概要

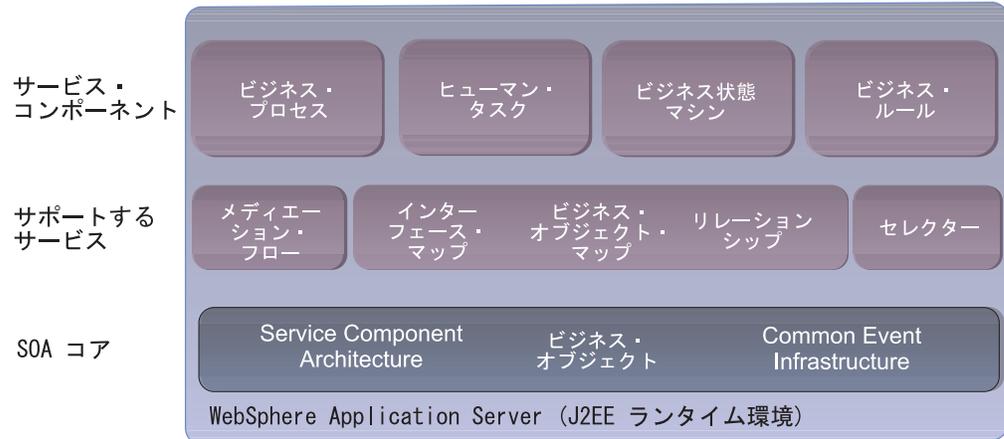
IBM WebSphere Process Server は、WebSphere Application Server の全機能を使用して、複合的なアプリケーション・プラットフォームに統合機能を結合し、完全集中型の標準に基づいたビジネス・プロセス・エンジンを持つ統合プラットフォームを提供します。

IBM WebSphere Process Server は、均一の呼び出しプログラミング・モデルおよび均一のデータ表現モデルに基づいたサービス指向アーキテクチャー (SOA) 統合プラットフォームです。

WebSphere Process Server の基本ランタイム・インフラストラクチャーは WebSphere Application Server です。SOA コアの一部である Service Component Architecture およびビジネス・オブジェクトは、均一の呼び出しプログラミング・モデルおよびデータ表現プログラミング・モデルを提供します。SOA コアには、WebSphere Process Server 上で稼働するアプリケーションのモニターと管理のためのイベントを生成する Common Event Infrastructure が含まれています。サポート・サービスは、基本のビジネス・オブジェクトおよび変換フレームワークを WebSphere Process Server に提供します。サービス・コンポーネントは、複合アプリケーションに必要な機能コンポーネントを表します。

WebSphere Process Server の強力な基盤 (WebSphere Application Server および SOA コア) とサービス・コンポーネントの組み合わせによって、WebSphere Process Server で稼働する高度な複合アプリケーションを短時間で開発しデプロイすることができます。

すべての統合形式を処理する 1 つのコンポーネント・ベースのフレームワーク

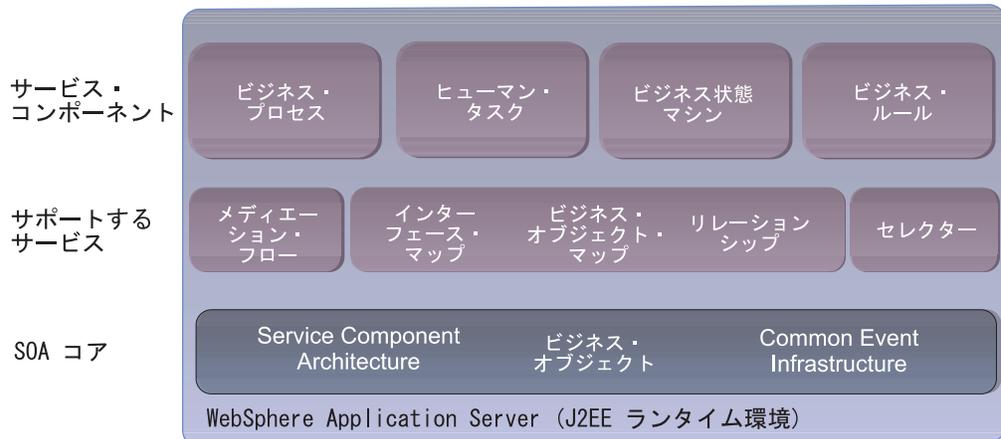


サービス指向アーキテクチャー・コア

WebSphere Process Server のサービス指向アーキテクチャー (SOA) コアは、WebSphere Process Server で稼働するアプリケーションに対して、均一な呼び出しプログラミング・モデルとデータ表現プログラミング・モデルの両方を提供し、さらにはモニター機能および管理機能も提供します。

SOA は、アプリケーションが提供するコンポーネントとサービスに関して、基礎にあるコンポーネント、サービス、およびコンポーネント間接続のインプリメンテーションに関係なく、ソフトウェア・システムの構造を概念的に記述するものです。SOA コアの一部である Service Component Architecture およびビジネス・オブジェクトは、IBM WebSphere Process Server にデプロイされるアプリケーションに対して、均一の呼び出しプログラミング・モデルおよびデータ表現プログラミング・モデルを提供します。SOA コアには、WebSphere Process Server 上のアプリケーションをモニターおよび管理するためのイベントを生成する Common Event Infrastructure も含まれています。

すべての統合形式を処理する 1 つのコンポーネント・ベースのフレームワーク



Service Component Architecture

Service Component Architecture は、ビジネス・トランザクションのすべてのエレメント (Web サービス、エンタープライズ情報システム (EIS) サービス資産、ビジネス・ルール、ワークフロー、データベースなどへのアクセス) をサービス指向の方式で提供します。

Service Component Architecture によりビジネス・ロジックがインプリメンテーションから切り離されるので、インプリメンテーションの詳細を知らなくても統合アプリケーションのアセンブルに集中できます。ビジネス・プロセスのインプリメンテーションは、サービス・コンポーネントに含まれます。

サービス・コンポーネントは、IBM WebSphere Integration Developer ツールを使用してグラフィカルにアセンブルできるので、インプリメンテーションを後で追加できます。Service Component Architecture プログラミング・モデルは、Java™ および J2EE、または特定のシナリオにおけるその他のインプリメンテーションについて開発者が知っておくべき知識の範囲を、今日、他のプログラム言語でビジネス・アプリケーションを開発しているすべての開発者になじみの深い、言語概念のコア・セットに狭めます。これにより、開発者はすばやく容易にテクノロジーを統合できます。

従来のアプリケーション開発環境から切り替えた開発者は、非常に小さな学習曲線に向き合うこととなります。つまり、このプログラミング・モデルを使用することで、早く生産性を上げることができます。また、Service Component Architecture のプログラミング・モデルを使用すると、経験豊富な J2EE 開発者の生産性がさらに向上します。

Service Component Architecture は、いくつかの標準サービス・インプリメンテーション・タイプをサポートします。

- Java オブジェクト。Java クラスをインプリメントします。Java プログラム言語の場合のように、実行時の Java コンポーネントのインスタンスは Java オブジェクトと呼ばれます。

- ビジネス・プロセス・コンポーネント。ビジネス・プロセスをインプリメントします。そのインプリメンテーション言語は、Business Process Execution Language (BPEL) および IBM 拡張です。
- ヒューマン・タスク・コンポーネントは、ビジネス・プロセスまたは統合アプリケーションにおいて、通常は人が実行するタスクを表し、インプリメントします。
- ビジネス状態マシン・コンポーネント。一連の状態を持つ成果物をアプリケーションが処理する場合に使用されます。状態マシンは、特定の時間に成果物が実行できることを定義します。
- ビジネス・ルール・コンポーネント。コンテキストに基づいてビジネス・プロセスの結果を判別するもので、if-then ルール、デシジョン・テーブル、または決定木として設計できます。ビジネス・プロセス内のビジネス・ルールにより、アプリケーションはビジネス条件の変化に迅速に対応できます。ルールはビジネス・プロセス自体から独立しているので、プロセスを再実行せずにいつでも変更できます。

サービス修飾子は、WebSphere Process Server ランタイム環境のサービス・クライアントとサービス間の対話を管理します。サービス修飾子とは、伝送優先順位、経路の信頼性のレベル、トランザクション管理、およびセキュリティー・レベルのためにアプリケーションに必要な一連の通信特性を定義する、サービス品質仕様です。アプリケーションは、サービス修飾子を指定することにより、ランタイム環境へのサービス品質ニーズと通信します。サービス品質修飾子は、WebSphere Integration Developer 内のアセンブリー・エディターでコンポーネントを結合するときに指定できます。WebSphere Process Server 上で実行されている場合、これらの仕様は、ターゲット・コンポーネントとのクライアントの対話方法を決定します。指定された修飾子に応じて、ランタイムは追加の必要処理を提供する場合があります。

WebSphere Process Server ソリューションは、トランザクション、セキュリティー、およびワークロード管理用の、基礎となる WebSphere Application Server 機能に依存して、拡張が容易な統合環境を提供します。

ビジネス・プロセスの場合、WebSphere Process Server は 2 フェーズ・コミット・プロセスを使用して、複数のリソース・マネージャーを含むトランザクション用のサポートを提供し、原子性、一貫性、独立性および耐久性 (ACID) プロパティーを保証します。この機能は、短期実行フロー (単一トランザクション) および長期実行フロー (複数トランザクション) の両方に使用できます。ビジネス・プロセス内の複数のステップを 1 つのトランザクションにグループ化するには、WebSphere Integration Developer でトランザクションの境界を変更します。

すべてのサービス起動で 2 フェーズ・コミット・トランザクションがサポートされるわけではないので、WebSphere Process Server にはリカバリー機能も組み込まれています。統合アプリケーションの実行途中で障害が発生した場合、サーバーは障害を検出し、管理者が失敗したイベントを障害のあるイベント・マネージャーから管理できるようにします。

サービス・データ・オブジェクトおよびビジネス・オブジェクト

サービス・データ・オブジェクトおよびビジネス・オブジェクトは、Service Component Architecture で定義されているコンポーネント間に流れるデータを定義します。

WebSphere Application Server 機能の一部であり、WebSphere Process Server に組み込まれているサービス・データ・オブジェクト (SDO) は、データ・アプリケーション開発用のフレームワークを提供します。このフレームワークは、J2EE データ・プログラミング・モデルを単純化します。

WebSphere Process Server には、拡張された SDO であるビジネス・オブジェクトが組み込まれています。ビジネス・オブジェクトは、サービス・データ・オブジェクトと呼ばれるデータ・アクセス・テクノロジーに基づいています。SDO には、さまざまなデータ (例えば、JDBC ResultSet や XML スキーマで記述されたデータ) を記述するための汎用手段が準備されています。ビジネス・オブジェクトには、統合ソリューションにとって非常に重要ないくつかの拡張機能があり、これらは Service Component Architecture サービス間で交換されるデータを詳細に記述するために使用されます。ビジネス・オブジェクトは、WebSphere Process Server のサービス指向アーキテクチャー (SOA) コアの一部です。

ビジネス・オブジェクトは、ビジネス・エンティティ (従業員など)、データに対するアクション (作成操作や更新操作)、およびデータの処理に対する指示を表す一連の属性です。統合アプリケーションのコンポーネントは、ビジネス・オブジェクトを使用して、情報を交換したりアクションを起動したりします。ビジネス・オブジェクトはさまざまな種類のデータを表すことができるので、柔軟性があります。例えば、従来の統合サーバーのデータ正規化モデルのサポートに加えて、同期 EJB セッション Bean ファサードまたは同期ビジネス・プロセスから戻されたデータを表し、それを IBM WebSphere Portal ポートレットや JSF コンポーネントに結合することができます。

ビジネス・オブジェクトは、ビジネス・エンティティを表す、つまりリテラル・メッセージ定義を文書化するための基本メカニズムであり、スカラー・プロパティを持つ単純な基本オブジェクトから、大規模で複雑なオブジェクトの階層またはグラフまでのすべてを使用可能にします。

WebSphere Process Server では、ビジネス・オブジェクトのフレームワークは、以下のエレメントで構成されます。

- ビジネス・オブジェクト定義
- ビジネス・グラフ定義
- ビジネス・オブジェクト・メタデータ定義
- ビジネス・オブジェクト・サービス (サービス API)

ビジネス・オブジェクト定義は、ビジネス・オブジェクトのタイプを指定する、名前、順序付けられた属性のセット、プロパティ、バージョン番号、およびアプリケーション固有のテキストです。ビジネス・グラフ定義は、ビジネス・グラフ内のビジネス・オブジェクトに関連する変更の要約情報やイベントの要約情報の送達などの追加機能を提供するために、単純なビジネス・オブジェクトまたはビジネス・オブジェクトの階層に追加されるラッパーです。ビジネス・オブジェクト・メタデータ定義は、WebSphere Process Server 上で実行されているときに、ビジネス・オ

プロジェクト定義に追加して、値を拡張することができるメタデータです。このメタデータは、xs:annotation エlement および xs:appinfo エlement としてよく知られている、ビジネス・オブジェクトの XML スキーマ定義に追加されます。ビジネス・オブジェクト・サービスは、WebSphere Application Server のサービス・データ・オブジェクトにより提供される基本機能のトップにある一連の機能です。例えば、作成、コピー、等価、直列化などのサービスです。

WebSphere Application Server サービス・データ・オブジェクトについて詳しくは、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。

Common Event Infrastructure

Common Event Infrastructure および Common Base Event をサポートすることにより、WebSphere Process Server で稼働するアプリケーションのビジネス・プロセスのトラッキング、監査、およびモニターが拡張されます。

Common Event Infrastructure は、IBM WebSphere Process Server ランタイム環境において、異なる多数のプログラミング環境からイベントを永続的に保管し、検索する機能を提供します。イベントは、Common Base Event モデル (イベントの構造を定義する標準の XML ベース・フォーマット) を使用して表現されます。

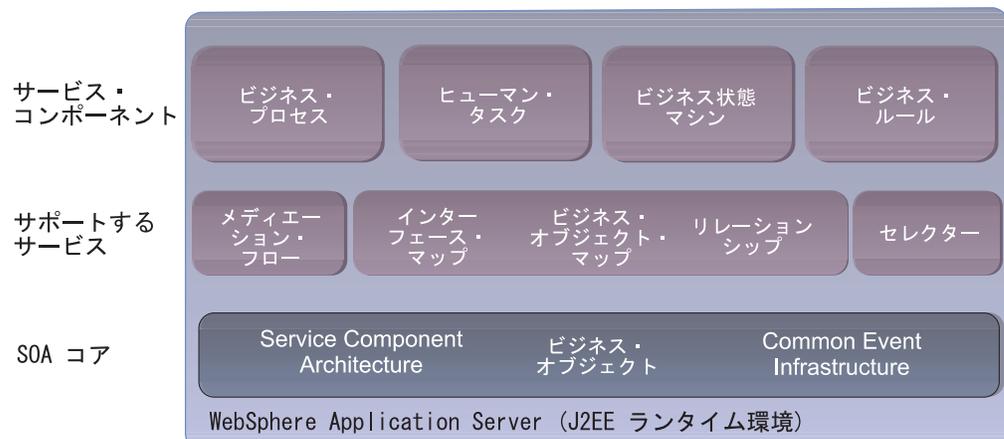
Common Event Infrastructure について詳しくは、「*Administering WebSphere Process Server*」PDF ファイルを参照してください。

サポート・サービス

IBM WebSphere Process Server のサポート・サービスは、コンポーネントおよび外部成果物を接続するためのいくつかの変換の問題に対処します。

メディエーション・フロー、インターフェース・マップ、ビジネス・オブジェクト・マップ、リレーションシップ、およびセレクターを使用して、IBM WebSphere Process Server で稼働するアプリケーションを統合することができます。

すべての統合形式を処理する 1 つのコンポーネント・ベースのフレームワーク



メディエーション・フロー

メディエーション・フローは、既存のサービス (プロバイダー) と、それらのサービスを使用するクライアント (要求側) との間で渡されるメッセージをインターセプトして変更します。

メディエーション・フローは、エクスポートとインポートの間に介在または介入して、メッセージ・ロギング、データ形式変更、ルーティングなどの機能を提供します。メディエーション・フローは、IBM WebSphere Integration Developer で作成され、WebSphere Process Server のメディエーション・モジュールの一部としてデプロイされます。

関連概念

エンタープライズ・サービス・バスの概要

WebSphere Process Server のエンタープライズ・サービス・バス機能は、サービス指向、メッセージ指向、およびイベント・ドリブンのテクノロジーの統合をサポートすることにより、エンタープライズ・サービス・バスを迅速に運用開始したい企業に対して標準ベースのメッセージング・インフラストラクチャーを提供します。

インターフェース・マップ

インターフェース・マップは、異なるインターフェースを持つコンポーネント間の差異を調整します。

インターフェース・マップは、他の Service Component Architecture (SCA) コンポーネントのインターフェース間の差異を解決および調整して、コンポーネント同士が通信できるようにする、IBM WebSphere Process Server のサポート・サービス・コンポーネントです。インターフェース・マップは、IBM WebSphere Integration Developer のモジュール設計者が変換や他の基本操作によって複数インターフェース間の差異を調整できるようにする first-class パターンを取り込みます。インターフェース・マップは、モジュール (SCA モジュールとも呼ばれる) の一部として WebSphere Process Server にデプロイされます。

ビジネス・オブジェクト・マップ

ビジネス・オブジェクト・マップは、ソース・ビジネス・オブジェクトとターゲット・ビジネス・オブジェクトの間のマッピングをサポートします。

ビジネス・オブジェクト・マップは、ソース・ビジネス・オブジェクトのサービス・コンポーネントの値に基づいてターゲット・ビジネス・オブジェクトのサービス・コンポーネントに値を割り当てる、IBM WebSphere Process Server のサポート・サービス・コンポーネントです。

開発者は IBM WebSphere Integration Developer でビジネス・オブジェクト・マップを作成します。

リレーションシップ

リレーションシップは、複数のデータ型のデータ間の関連を設定する、WebSphere Process Server アプリケーションのサポート・サービスです。

IBM WebSphere Process Server の Relationship Manager は、リレーションシップ・データを手動で操作するためのツールです。このツールを使用して、自動リレーシ

ョンシップ管理で検出されたエラーを訂正したり、より完全なリレーションシップ情報を提供したりします。特に、このツールには、リレーションシップ・インスタンス・データの変更だけでなく検索の機能もあります。Relationship Manager によって、リレーションシップ・ランタイム・データ (参加者およびそのデータを含む) の構成、照会、表示、およびオペレーションの実行が可能になります。リレーションシップの定義は、Relationship Designer を使用して作成します。実行時、リレーションシップのインスタンスには、異なるアプリケーションからの情報を関連付けるデータが取り込まれます。

Relationship Manager について詳しくは、「*Administering WebSphere Process Server*」PDF ファイルを参照してください。

セレクター

セレクターは、IBM WebSphere Process Server で実行中のアプリケーション内のコンポーネントの処理時の柔軟性を提供します。

セレクターは、実行時にサービス・コンポーネントの処理時点で柔軟性を提供するサポート・サービスです。セレクターを使用すると、1 つの呼び出しによって、選択基準に基づいて異なるターゲットを呼び出すことができます。

セレクターを使用すると、ビジネス・ルール以上のいっそうの柔軟性が加わります。ビジネス・ルールは、ビジネスの基本的な部分です。ビジネス・ルールは、アプリケーションの一般的な処理や、アプリケーションを介してデータを取得する特定のサービスの呼び出しを駆動します。例えば、学校が始まる 2 週間前に、学校に関連する商品の新学期前の特別価格を提供するルールがあるとします。セレクターを使用すると、1 つの呼び出しによって、選択基準に基づいて異なるターゲットを呼び出すことができます。例えば、学校が始まる直前の場合、以前に新学期前に提供した価格が呼び出されます。ただし、シーズンが学校の終了直後である場合、夏休み準備用に提供した価格が呼び出されます。

アプリケーションはいつでも同じものを呼び出すため、移植可能です。ビジネス・ルールは常に変わりません。しかし、セレクターのため、実際の処理が異なります (別のサービス・コンポーネントを呼び出します)。

セレクター (セレクター・コンポーネントとも呼ばれる) について詳しくは、「*Administering WebSphere Process Server*」PDF ファイルを参照してください。

サービス・コンポーネント

IBM WebSphere Process Server で実行されるすべての統合成果物 (例えば、ビジネス・プロセス、ビジネス・ルール、およびヒューマン・タスク) は、明確に定義されたインターフェースを持つコンポーネントとして表されます。Service Component Architecture (SCA) 内で、サービス・コンポーネントはサービス・インプリメンテーションを定義します。

すべての統合成果物は、サービス・コンポーネント (SCA コンポーネントとも呼ばれる) として表されるので、IBM WebSphere Process Server では非常に柔軟性のある環境が作成されます。SCA コンポーネントにはそれぞれインターフェースがあります。これらのコンポーネントを結合して、WebSphere Process Server にデプロイされるモジュールを形成できます。これにより、アプリケーションの任意の部分を

変更する際に、他の部分に影響を与えないようにすることができます。例えば、承認のヒューマン・タスクを自動承認のビジネス・ルールで置き換えるには、ビジネス・プロセスやビジネス・プロセスの呼び出し元を変更せずに、アセンブリー・ダイアグラム内のコンポーネントを置き換えるだけで済みます。

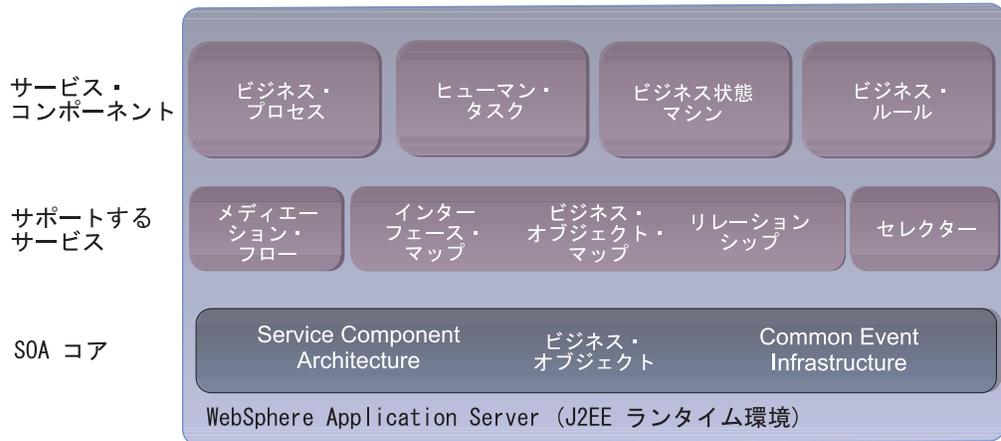
以下のプログラミング構成体を使用することにより、コンポーネントは既存のアプリケーションと対話できます。

- Java Bean
- Enterprise Java Bean
- Web サービス
- JMS メッセージ

さらに、WebSphere Business Integration Framework バージョン 2.6 をベースとする IBM WebSphere Adapters バージョン 6.0 および WebSphere Business Integration Adapters を使用すれば、コンポーネントはエンタープライズ情報システム (EIS) 上の他のアプリケーションと対話できます。

ランタイム・インフラストラクチャー・サポート・サービスおよびサービス指向アーキテクチャー・コアに加えて、WebSphere Process Server は、統合アプリケーションですぐに使用できる多様な SCA サービス・コンポーネントを提供します。

すべての統合形式を処理する 1 つのコンポーネント・ベースのフレームワーク



ビジネス・プロセス

ビジネス・プロセスは、エンタープライズ・サービス統合の主な手段を提供するサービス・コンポーネントです。

ビジネス・プロセスは、より大きなビジネス・ゴールを達成するために組織が使用するシステムまたは手順です。ビジネス・プロセスを分解すると、実際は一連の個別タスクであり、各タスクは特定の順序で実行されることがわかります。IBM WebSphere Process Server で稼働するアプリケーションに不可欠な部分として、ビジネス・プロセスはエンタープライズ・サービス統合の主な手段を提供します。

ビジネス・プロセス・コンポーネントは、完全にサポートされる Web Services BPEL (Business Process Execution Language) エンジンを実装します。WebSphere Process Server には、WebSphere Application Server の上にあるビジネス・プロセス・コレオグラフィー・エンジンが含まれます。拡張性の高いインフラストラクチャーにおいて、長期実行および短期実行のビジネス・プロセスに対する高度なサポートを持つ単純な開発モデルで、複雑なビジネス・プロセスを開発およびデプロイすることができます。WebSphere Integration Developer バージョン 6.0 で BPEL モデルを作成することも、WebSphere Business Modeler バージョン 6.0 で作成したビジネス・モデルから BPEL モデルをインポートすることもできます。

Web Services Business Process Execution Language (BPEL) が使用され、ビジネス・プロセス・フローのコレオグラフィーを行います。ビジネス・プロセス統合サービスは BPEL4WS バージョン 1.1 を基盤に構築され、さらに来るべき WS-BPEL バージョン 2.0 仕様の主要機能が追加されています。

ビジネス・プロセスについて詳しくは、「WebSphere Process Server *Installing*」PDF ファイルを参照してください。

ヒューマン・タスク

ヒューマン・タスクはスタンドアロンのサービス・コンポーネントであり、従業員への作業の割り当てやその他のサービスの起動に使用できます。

Human Task Manager (IBM WebSphere Process Server で使用可能) は、タスクの随時作成およびトラッキングをサポートします。ユーザー情報およびグループ情報へのアクセスには、(オペレーティング・システムのリポジトリおよび WebSphere ユーザー・レジストリーのほかに) 既存の LDAP ディレクトリーを使用できます。WebSphere Process Server では、E メール通知を含むヒューマン・タスクのマルチレベル・エスカレーションがサポートされます。また、ヒューマン・タスクを管理するための Web クライアントと、カスタム・クライアントの作成やヒューマン・タスク機能の他の Web アプリケーションへの組み込みに使用できる、一連の Java Server Faces (JSF) コンポーネントも組み込まれています。

ロール・ベースのタスク割り当て、起動、およびエスカレーションを可能にする、ヒューマン・タスク・サービス

ヒューマン・タスクについて詳しくは、「WebSphere Process Server *Installing*」PDF ファイルを参照してください。

ビジネス状態マシン

ビジネス状態マシンは、イベントに対応してオブジェクトまたは対話が存続中に起こす一連の状態、応答、およびアクションを指定するサービス・コンポーネントです。

ビジネス状態マシンは、ビジネス・プロセスのモデル化に別の方法を提供します。これにより、一連のビジネス・プロセス・モデルではなく、状態やイベントに基づいて、ビジネス・プロセスを表す選択肢を得られます。

WebSphere Process Server のビジネス状態マシンのモニターについて詳しくは、「WebSphere Process Server *Monitoring*」PDF ファイルを参照してください。

ビジネス・ルール

ビジネス・ルールは、ビジネス内で満たされなければならないポリシーまたは条件を宣言するサービス・コンポーネントです。

ビジネス・プロセスの柔軟性を高めるビジネス・ルールビジネス・ルールは、コンテキストに基づいてプロセスの結果を決定するので、ビジネス・プロセス内でビジネス・ルールを使用するとアプリケーションはビジネス条件の変化に迅速に対応できます。

ビジネス・ルールのオーサリングは、IBM WebSphere Integration Developer でサポートされています。IBM WebSphere Process Server には Business Rules Manager が含まれています。これは、ビジネス・アナリストがビジネス・ニーズに応じてビジネス・ルールを更新するための Web ベースのランタイム・ツールであり、ルールを更新しても、他のコンポーネントまたは Service Component Architecture (SCA) サービスに影響を及ぼしません。

WebSphere Process Server のビジネス・ルールの管理について詳しくは、「*Administering WebSphere Process Server*」PDF ファイルを参照してください。

インポートおよびエクスポートとアダプター

IBM Service Component Architecture 内の機能のインポートおよびエクスポートでは、サービス・モジュールの外部インターフェースまたは WebSphere Process Server のアクセス・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、同一アプリケーション内の他のモジュールに対しても、エンタープライズ情報システム (EIS) 上の他のアプリケーションに対しても実行できます。

インポートは、モジュールの外側のサービスを識別し、モジュール内からそれらのサービスを呼び出せるようにします。エクスポートは、モジュール内のコンポーネントが、サービスを外部のクライアントに提供できるようにします。

モジュール・レベルのインポートまたはエクスポートは、モジュールが他のモジュールにアクセスできるようにします。システム・レベルのインポートまたはエクスポートでは、アプリケーションが、ローカル・コンポーネントであるかのように EIS システム上のアプリケーションにアクセスできます。これにより、IBM WebSphere Adapter および WebSphere Business Integration Adapter との連動が可能になります。

WebSphere Adapters バージョン 6.0 および WebSphere Business Integration Adapters (WebSphere Business Integration Framework バージョン 2.6 をベースとする) では、EIS の統合に対するサービス指向の手法を提供します。

WebSphere Adapter は、J2EE Connector Architecture (JCA 1.5) に準拠しています。JCA は、EIS の接続性のための J2EE 標準です。EIS のインポートおよび EIS のエクスポートでは、SCA コンポーネントに対して、モジュール外部のサービスの統一した表示を提供します。これにより、各コンポーネントが一貫性のある SCA プログラミング・モデルを使用して、さまざまな外部 EIS システムと通信することができます。インポートされた RAR ファイルを元に、WebSphere Integration

Developer 内で WebSphere Adapter がアセンブルされ、その後エンタープライズ・アーカイブ (EAR) ファイルとしてエクスポートされて、WebSphere Process Server 上でデプロイされます。

WebSphere Adapter には、以下のものがあります。

- IBM WebSphere Adapter For Flat Files バージョン 6.0
- IBM WebSphere Adapter for JDBC バージョン 6.0
- IBM WebSphere Adapter for PeopleSoft Enterprise バージョン 6.0
- IBM WebSphere Adapter for Siebel Business Applications バージョン 6.0
- IBM WebSphere Adapter for SAP Applications バージョン 6.0

これらの製品について詳しくは、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 Business Integration Adapter は、アプリケーションが統合ブローカーを介してビジネス・データを交換できるようにするためのソフトウェア、アプリケーション・プログラミング・インターフェース、およびツールの集合で構成されています。ビジネス統合プロセスに参加するために、各ビジネス・アプリケーションそれぞれに、アプリケーション固有のアダプターが必要です。現行の WebSphere Business Integration Adapter Framework and Development Kit System Manager ツール群を使用して、アダプターをインストールし、構成し、テストすることができます。WebSphere Integration Developer では、既存のビジネス・オブジェクトやコネクター構成ファイルのインポートや、成果物の生成、WebSphere Process Server のソリューションのアセンブルが可能です。WebSphere Business Integration Adapter の操作可能なコマンドは、WebSphere Process Server 管理コンソールに組み込まれています。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 Adapters インフォメーション・センターを参照してください。

インポートおよびエクスポートにはバインディング情報が必要であり、この情報によって、モジュールからのデータのトランスポート手段が指定されます。WebSphere Integration Developer のアSEMBリー・エディターは、インポートおよびエクスポートのセットアップ、サポートされるバインディングのリスト、およびそれらの作成の単純化を行います。バインディング情報は、プロパティ・ビューに表示されます。

クラスターおよびワークロード管理

クラスターは、まとめて管理されるサーバーのセットであり、ワークロード管理に関係します。同じノードの一部である複数のサーバーは同じホスト・マシン上に置く必要があるのに対し、クラスターのメンバーである複数のサーバーは別々のホスト・マシン上に置くことができます。

サーバーのクラスターを作成する (サーバーをクラスタリングする) と、クラスターを構成するサーバー間で要求が分散されるので、アプリケーションは追加の処理能力を使用できるようになります。

クラスターに属するサーバーはそのクラスター・セットのメンバーなので、それらすべてのサーバー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

いなければなりません。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・メンバーの各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。

クラスターを作成するときは、既存のアプリケーション・サーバー・テンプレートのコピーを作成します。このテンプレートは、たいいていの場合、以前に構成したアプリケーション・サーバーです。そのサーバーをクラスターのメンバーとするオプションが提示されます。しかしながら、このサーバーはテンプレートとしてのみ使用可能にしておくことをお勧めします。というのも、クラスター・メンバーを除去するには、サーバーを削除するしか方法がないからです。クラスターを削除するときは、そのクラスターのメンバーだったサーバーも削除します。クラスターのメンバーを保持する方法はありません。オリジナルのテンプレートをそのまま保持していれば、構成を再作成する必要がある場合にそのテンプレートを再利用できます。

クラスターについて詳しくは、WebSphere Application Server for z/OS インフォメーション・センターの『概要: クラスター』を参照してください。

エンタープライズ・サービス・バスの概要

WebSphere Process Server のエンタープライズ・サービス・バス機能は、サービス指向、メッセージ指向、およびイベント・ドリブンのテクノロジーの統合をサポートすることにより、エンタープライズ・サービス・バスを迅速に運用開始したい企業に対して標準ベースのメッセージング・インフラストラクチャーを提供します。

この機能は、WebSphere Application Server Network Deployment、バージョン 6.0.2 で提供される堅固な J2EE 1.4 インフラストラクチャーおよび関連プラットフォーム・サービスに基づいています。

関連概念

10 ページの『メディエーション・フロー』

メディエーション・フローは、既存のサービス (プロバイダー) と、それらのサービスを使用するクライアント (要求側) との間で渡されるメッセージをインターセプトして変更します。

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活用される) 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・プロパティーで構成され、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。

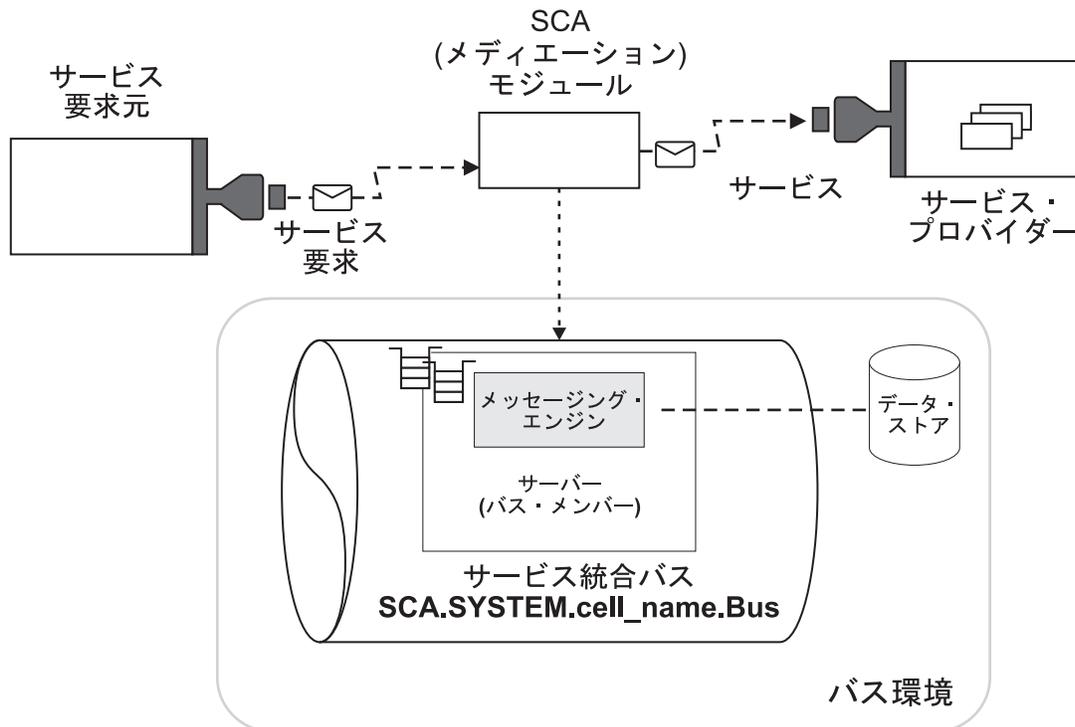


図 1. 1 つのサーバーが SCA.SYSTEM サービス統合バスに割り当てられたバス環境。: バス・メンバーとして、サーバーには 1 つのメッセージング・エンジンが割り当てられており、このエンジンには状態データおよびメッセージを保管するためのデータ・ストアがあります。この図は、メディエーション・モジュールがバス環境にデプロイされ、バス・メンバーに割り当てられていることも示しています。

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。例えば以下のように構成できます。

- 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。

- 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、そ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され、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。

関連概念

26 ページの『サーバーおよびバス環境の管理』

WebSphere ESB を使用すれば、管理者は、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 ESB サーバーおよびサービス統合バスの環境を作成できます。

クライアント

WebSphere ESB は、エンタープライズ・サービス・バスの接続を拡張する Message Service クライアントを提供します。

- Message Service Clients for C/C++ および .NET によって、非 Java アプリケーションは WebSphere ESB に接続できるようになります。

Message Service Clients for C/C++ および .NET は、Java Message Service (JMS) API と同じインターフェースのセットを持つ XMS という API を提供します。Message Service Client for C/C++ には 2 つの XMS インプリメンテーションがあり、一方は C アプリケーション用、もう一方は C++ アプリケーション用です。Message Service Client for .NET には、完全に管理された XMS インプリメンテーションがあり、これは任意の .NET 対応言語で使用できます。

- Web Services Client for C++ は、ライブラリーのセットおよび Java ツールを提供します。これらを使用すれば、既存の Web サービス記述言語 (WSDL) ファイルから ANSI C++ Web サービス・クライアント・アプリケーションを作成できます。

Web Services Client for C++ ライブラリーおよび Java ツールを使用して、既存の WSDL ファイルから作成された ANSI C++ Web サービス・クライアント・アプリケーションは、同様に構成された他のアプリケーションと、SOAP プロトコルによる TCP/IP を使用した HTTP 経由で通信できます。

WebSphere Application Server Network Deployment から J2EE クライアント・サポート (Web サービス・クライアント、EJB クライアント、および JMS クライアントを含む) をインストールして使用することもできます。J2EE クライアント・サポートのインストールについて詳しくは、WebSphere Application Server Network Deployment インフォメーション・センターの「**WebSphere Application Server のアプリケーション・クライアントのインストール**」を参照してください。

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るので、メッセージの内容または書式を変更できます。さらにメディエーション・モジュールは、メッセージ・ロギングなどの機能や、ユーザーの要件に合うように調整されたエラー処理を提供します。

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、またはサービス・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

ジュールなどの SCA モジュールを作成します。WebSphere Integration Developer を使用すれば、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ています。メディエーション・モジュールに含めることができるメディエーション・フロー・コンポーネントは 1 つまでです。
- 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・プリミティブを含めることも、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実現するメディエーション・プリミティブの提供される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、サービス要求のプロトコルを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、一部の変更操作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また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。

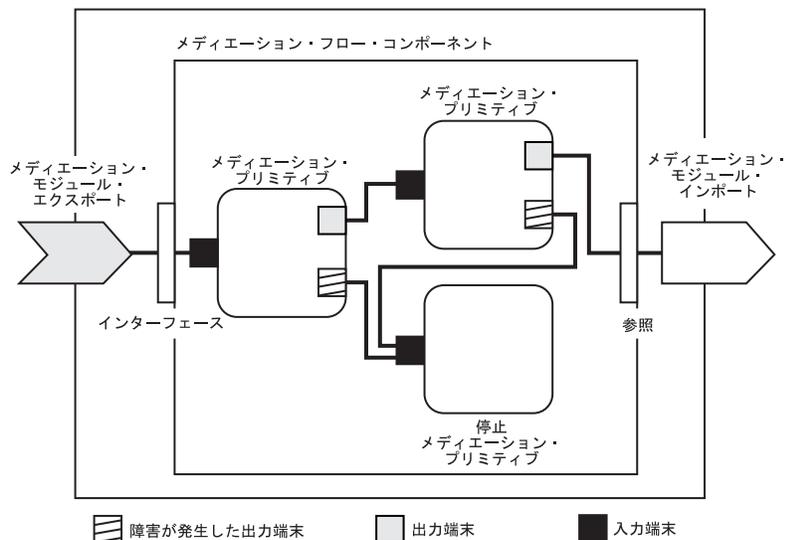


図2. メディエーション・モジュールの簡単な例： メディエーション・モジュールには 1 つのメディエーション・フロー・コンポーネントが含まれます。メディエーション・フロー・コンポーネントにはメディエーション・プリミティブが含まれます。

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

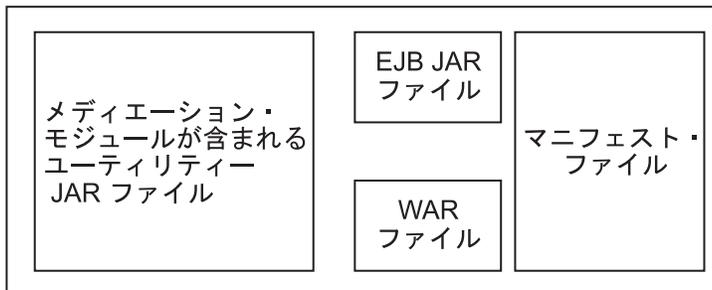


図3. メディエーション・モジュールが含まれる EAR ファイルの簡単な例：EAR ファイルには JAR ファイルが含まれます。ユーティリティー JAR ファイルにはメディエーション・モジュールが含まれます。

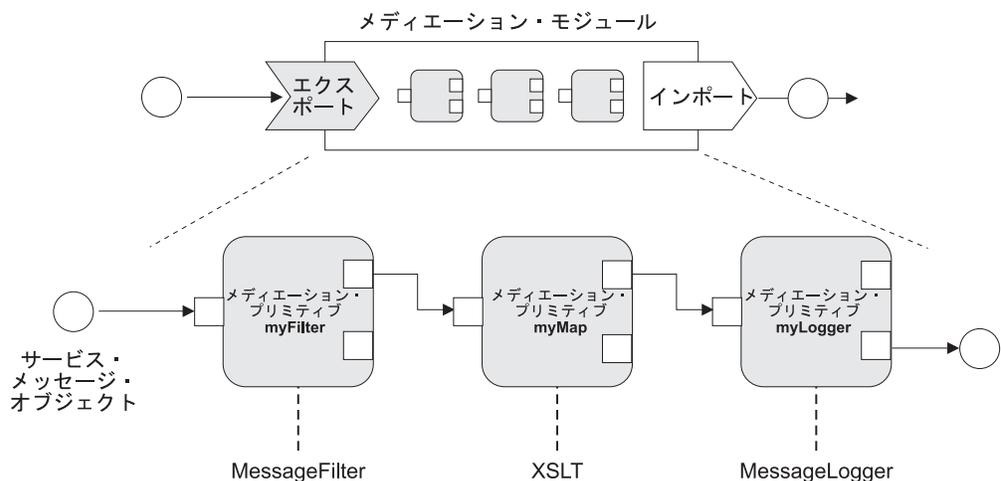
メディエーション・プリミティブ

メディエーション・コンポーネントは、サービス・コンポーネント間のメッセージ・フローを操作します。メディエーション・コンポーネントの機能は、標準サービス・インプリメンテーション・タイプをインプリメントするメディエーション・プリミティブによってインプリメントされます。

メディエーション・コンポーネントには、1 つ以上のフローがあります (例、要求用と応答用に 1 つずつ)。

WebSphere ESB は、提供されたメディエーション・プリミティブのセットをサポートしており、このメディエーション・プリミティブ・セットは、WebSphere ESB にデプロイされたメディエーション・モジュールの標準メディエーション機能をインプリメントします。特殊なメディエーション機能が必要な場合は、ユーザー独自のカスタム・メディエーション・プリミティブを開発できます。

メディエーション・プリミティブには、サービス・メッセージ・オブジェクトによって表されるメッセージを処理する 1 つの「in」オペレーションを定義します。メディエーション・プリミティブには、他のコンポーネントまたはモジュールにメッセージを送信する「out」オペレーションも定義できます。



メディエーション・プリミティブは一般に、単一オペレーション・レベルで機能して、要求 (入力) と応答 (出力) のメディエーションが可能です。場合によっては、オペレーションの単一パラメーターのレベルまでメディエーション・プリミティブで指定できます。例えば、セレクターはオペレーション・レベルまたはパラメーター・レベルで作動できます。

WebSphere Integration Developer を使用すると、メディエーション・プリミティブからメディエーション・コンポーネントをグラフィカルにモデル化し組み立てて、さらにメディエーション・コンポーネントからメディエーション・モジュールを組み立てることができます。

以下のメディエーション・プリミティブ・セットが WebSphere ESB でサポートされています。

Custom Mediation

カスタム・ロジックを実行します。Custom Mediation プリミティブは、ユーザーが提供する外部 Service Component Architecture (SCA) コンポーネントを呼び出すことができます。

- 呼び出されるオペレーションは、両方向オペレーションでなければなりません。
- ターゲット SCA コンポーネントは、Custom Mediation プリミティブと同じメディエーション・モジュールに存在していなければなりません。

Database Lookup

ユーザーが提供するデータベースからの情報を使用して、メッセージを変更します。

- Database Lookup メディエーション・プリミティブが使用できるようにデータベース、データ・ソース、およびサーバー認証の設定をセットアップする必要があります。
- Database Lookup メディエーション・プリミティブは、1 つのテーブルからのみ読み取ることができます。
- 指定されたキー列には、固有な値が含まれていなければなりません。
- 値列のデータは、Java プリミティブまたは Java ストリング (あるいは Java プリミティブまたは Java ストリングにキャスト可能) でなければなりません。

Fail フロー内で障害を生成します。

Message Filter

メッセージ内容に基づいて、さまざまなパスにメッセージを送付します。

Message Logger

データベースにメッセージを記録します。メッセージは XML として保管されるので、データは XML 対応アプリケーションで後処理できます。

- データベース・スキーマは IBM によって定義されています。
- WebSphere ESB のデフォルト・インストールを行うと、Cloudscape データベースおよびデータ・ソースが作成されます。デフォルトでは、WebSphere Integration Developer は、この Cloudscape データベースを使用するように Message Logger メディエーション・プリミティブを構成します。

- 管理コンソールを使用してユーザー独自のデータベースおよびデータ・ソースを作成する場合、WebSphere ESB では、テーブル・スキーマを記述するデータ定義言語 (ddl) ファイルが提供されています。この Table.ddl ファイルは以下の場所に格納されています。

install_root/utl/EsbLoggerMediation/database_type/Table.ddl ここで、*database_type* はデータベースのタイプを指します (例、CLOUDSCAPE_V50)。ユーザー独自のデータベースを作成して、データ・ソースにデフォルト JNDI 名を使用する場合は、デフォルト・データ・ソースを除去する必要があります。

Stop 例外を生成せずに、フロー内の特定のパスを停止します。

XSLT メッセージを変換します。

- XSLT メディエーション・プリミティブは、メッセージのヘッダーまたは本体を変更できます。
- XSLT (Extensible Stylesheet Transformations) 1.0 変換を使用してメッセージを変換します。この変換は、メッセージの XML 直列化を処理します。

CEIEmitter

処理されるメッセージの CEI イベントを出力します。

SplitPath

ターゲット・サービス (または他のメディエーション) の選択、特定ターゲットへの送付、およびルーティング・パスの変更を行います。

BOMapper

ターゲット・サービス (または他のメディエーション) の選択、特定ターゲットへの送付、およびルーティング・パスの変更を行います。

サービス・メッセージ・オブジェクト

サービス・メッセージ・オブジェクト (SMO) は、拡張サービス・データ・オブジェクト (SDO) です。SMO は、サービス間で交換されるメッセージを処理および操作するための抽象化層を提供します。

SMO モデル

SMO モデルは、SDO `DataObject` を使用してメッセージを表したパターンです。SMO には、以下のデータ・グループの表現が含まれます。

- メッセージのビジネス・ペイロード。ペイロードは、サービス・エンドポイント間で交換されるアプリケーション・データです。
- メッセージに関連付けられたヘッダー情報。例えば、JMS API を使用してメッセージが伝達された場合は Java Message Service (JMS) ヘッダーになります。
- コンテキスト情報 (メッセージ・ペイロード以外のデータ)。

この情報はすべて SDO `DataObject` としてアクセスされます。また、SMO の全体構造を指定するスキーマ宣言があります。このスキーマは WebSphere Integration Developer によって生成されます。

すべての SMO の基本構造は同じです。この構造は、`ServiceMessageObject` と呼ばれるルート・データ・オブジェクトで構成されており、ルート・データ・オブジェ

クト内にヘッダー、本体、およびコンテキスト・データを表す他のデータ・オブジェクトが格納されています。SMO 本体にはメッセージ・ペイロードが格納されます。ヘッダーには、特定のインポートまたはエクスポート・バインディングから生じる情報が格納されます。例えば JMS バインディングなどです。

SMO は、メッセージ・ヘッダーおよびメッセージ・ペイロードへのアクセスおよび変更を行うためのインターフェースを提供します。SMO によって、さまざまなタイプのメッセージの論理コンテンツを表現できます。

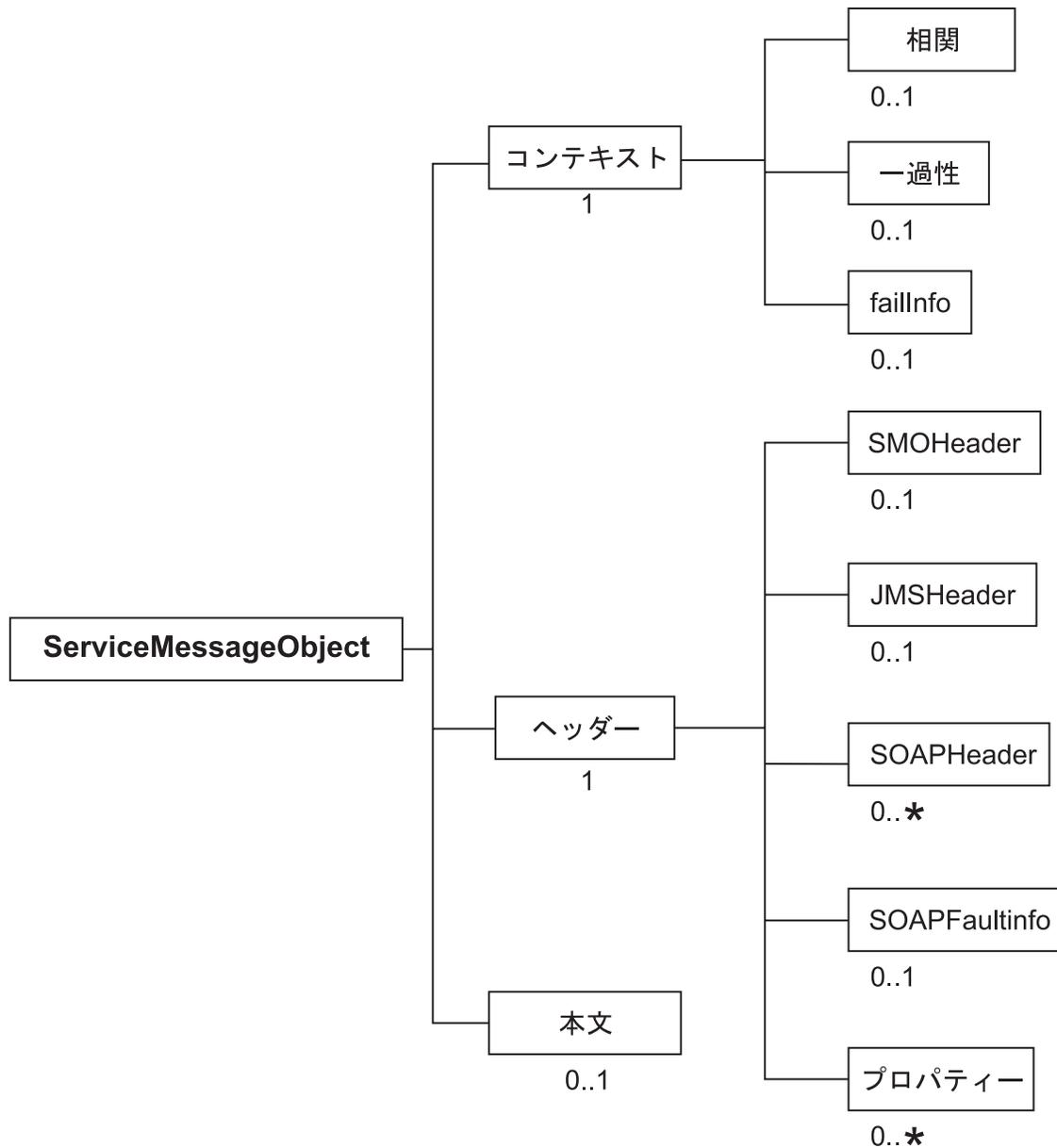


図 4. SMO 構造の概要： ServiceMessageObject のヘッダー、コンテキスト、および本体

WebSphere ESB と SMO

WebSphere ESB は、対話エンドポイント間で送信途中にあるメッセージを操作します。WebSphere ESB 内部では、メディエーション・フローはメッセージを SMO として処理します。

メッセージは複数のソースから送信される可能性があるため、SMO は異なる種類のメッセージ・ヘッダーを保持できなければなりません。WebSphere ESB によって処理されるメッセージ・ヘッダーの種類は以下のとおりです。

- Web Services メッセージ・ヘッダー
- Service Component Architecture (SCA) メッセージ・ヘッダー
- Java Message Service (JMS) メッセージ・ヘッダー
- WebSphere Adapter メッセージ・ヘッダー

WebSphere ESB SMO ランタイム

WebSphere ESB は SMO オブジェクトを作成します。この SMO オブジェクトはメディエーション・フローに使用できます。

ランタイムが作成する SMO オブジェクトの一部は、ランタイムが提供するクラスによってインプリメントされます。例えば、ServiceMessageObject クラスは WebSphere ESB によって提供されます。SMO ヘッダー・クラスの一部もランタイムによって提供されます。ランタイムが作成するその他の SMO オブジェクトは、開発者が作成したクラスによってインプリメントされます。

メディエーション・フローを作成するときに、WebSphere Integration Developer は各端末 (入力、出力、または失敗) のメッセージ本体のタイプと、オプションでコンテンツ情報のタイプを指定します。WebSphere ESB は、この情報を使用してメッセージを指定のタイプの SMO オブジェクトに変換します。

サーバーおよびバス環境の管理

WebSphere ESB を使用すれば、管理者は、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 ESB サーバーおよびサービス統合バスの環境を作成できます。

バス環境の管理には、WebSphere ESB のセットアップが含まれます。その際に、より大規模なシステムの一部としてセットアップする場合がありますが、一般的には実稼働環境または実際のテスト環境としてセットアップします。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。中心となるのは、サービス統合バスおよびサーバーの管理であり、クラスタリングや高可用性の構成によってワークロードのバランスを取る方法です。

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。

WebSphere ESB デプロイメント・マネージャー・プロファイルの作成を選択すれば、複数のサーバーおよびサーバー・クラスターの使用が可能になります。最初は

デプロイメント・マネージャー・セル内のサーバーを 1 つとし、オプションで、サーバーまたはサーバー・クラスターを複数に増やすことにより、容量および拡張可用性を追加することもできます。

必要な場合は他のバスを追加すれば、ESB に接続するアプリケーションのデプロイや、WebSphere MQ と統合が可能になります。

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。「サーバーおよびバス」タスク選択フィルターの適用を選択すれば、タスクの範囲をバス環境管理の関連タスクに限定できます。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・モジュールおよびサービス・アプリケーションのデプロイメントをサポートするために必要な 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、およびローカル・トポロジーのリスト表示
- 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 プロファイルなど) の作成
- ESB ランタイムの操作制御

これは、サーバー・アプリケーションおよびメディエーション・モジュールのデプロイメントをサポートするために必要なサーバーおよびバス環境の管理に携わ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。

より広範囲の管理用タスクに関しては、「すべて」タスク選択フィルターの適用を選択することもできます。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 の管理に携わる管理者に最適です。

管理コンソールのほかにも、コマンド、スクリプト、および管理プログラムを使用してサーバーおよびバス環境を管理できます。

関連概念

16 ページの『バス環境の概要』

バス環境は、1 つ以上のサービス統合バス、ESB サーバー、およびそのリソースで構成されており、セルとノードからなる論理的な管理可能ドメインに編成されます。

サービス・アプリケーションの開発およびデプロイ

WebSphere ESB にデプロイされると、サービス間でメッセージ・ベースの通信が可能になり、対話エンドポイント間で送信途中のメッセージを処理および操作できるようなサービス・アプリケーションを開発できます。

サービス・アプリケーションの開発

WebSphere Integration Developer は、WebSphere ESB 用の独立した開発環境です。WebSphere Integration Developer を使用すると、メディエーション・プリミティブからメディエーション・コンポーネントをグラフィカルにモデル化し組み立てて、さらにメディエーション・コンポーネントからメディエーション・モジュールを組み立てることができます。

- SCA メディエーション・コンポーネントのインターフェースがインポートされない場合は、Simplified Interface Editor を使用してインターフェースを作成できます。このエディターを使用して、メディエーション・モジュールのインターフェース (オペレーションやパラメーター) を指定および編集することができます。
- Mediation Flow Editor を使用して、メディエーションのエンドポイント上のオペレーション間をマップし、このアプリケーションに必要なメディエーション・フローのセットを定義することができます。事前定義メディエーション・プリミティブのセットを使用して、メディエーション・フローを視覚的に構成できます。
- Business Object Editor を使用して、メディエーションに使用されるメッセージを構成することができます。
- 他のエディターを使用して、ビジネス・ニーズに合うように開発環境を拡張することができます。例えば以下のようになります。
 - カスタム・メディエーション・プリミティブを作成および編集して、それらを Mediation Flow Editor に追加する。
 - メッセージ記述子を作成および編集する。

他のアプリケーション開発ツールを使用して、いくつかのサービス・コンポーネントを開発し、それらを WebSphere Integration Developer にインポートして、WebSphere ESB へのデプロイメント用にモデル化、編集、テスト、およびパッケージ化することができます。

サービス・アプリケーションのデプロイ

デプロイとは、テスト環境または実稼働環境でアプリケーションを使用可能にすることです。どちらの環境でもデプロイは同じ意味ですが、それぞれの環境のデプロイメント・タスクにはいくつかの違いがあります。

サービス・アプリケーションの開発後、アプリケーションをテスト・サーバーにデプロイしてテストしてから、実動要件にコミットするのがベスト・プラクティスです。アプリケーションをテスト環境にデプロイする場合、および WebSphere ESB へのデプロイメント用にサービス・アプリケーションを標準エンタープライズ・アプリケーション・パッケージとしてパッケージする場合は、WebSphere Integration Developer を使用します。

アプリケーションを実稼働環境にインストールしてデプロイする場合は、WebSphere ESB を使用します。WebSphere ESB では、標準 WebSphere 管理コンソールを使用して、ロール・ベースの管理ビューを表示できるので、これによりソリューション管理者はサービス統合パッケージのコンポーネントを簡単にデプロイおよび管理できます。

関連概念

『統合アプリケーションの開発およびデプロイメント』

WebSphere Process Server における統合アプリケーションの開発およびデプロイメント用オプションには、WebSphere Integration Developer の開発環境での作業、Service Component Architecture API を使用した作業、および WebSphere Process Server を使用したテスト環境または実稼働サーバー環境でのアプリケーションの使用可能化があります。

統合アプリケーションの開発およびデプロイメント

WebSphere Process Server における統合アプリケーションの開発およびデプロイメント用オプションには、WebSphere Integration Developer の開発環境での作業、Service Component Architecture API を使用した作業、および WebSphere Process Server を使用したテスト環境または実稼働サーバー環境でのアプリケーションの使用可能化があります。

IBM WebSphere Integration Developer は、WebSphere Process Server の開発環境です。WebSphere Integration Developer での統合アプリケーションの開発について詳しくは、WebSphere Integration Developer インフォメーション・センターを参照してください。

WebSphere Integration Developer 開発環境に加えて、開発者には Service Component Architecture API が公開されています。Service Component Architecture API について詳しくは、「WebSphere Process Server *Developing and Deploying Modules*」PDF ファイルを参照してください。

モジュール (WebSphere Process Server にデプロイされた場合は Service Component Architecture (SCA) モジュールとも呼ばれる) によって、ランタイム環境にデプロイされるエンタープライズ・アーカイブ (EAR) ファイルにパッケージされる成果物が決まります。

WebSphere Integration Developer 内では、アセンブリ・エディターを使用して、サービスをモジュールにグループ化したり、モジュールによって外部の利用者に公開されるサービスを指定したりできます。その後で、モジュールは接続され、完全な統合ソリューションが形成されます。

Service Component Architecture を使用すると、統合ロジックをモジュールにカプセル化できるので、モジュール内のサービスが変更されても、変更されたモジュールのインターフェースが同じである限り、ソリューション内の他のモジュールに影響はありません。

デプロイとは、テスト環境または実稼働環境でアプリケーションを使用可能にすることです。どちらの環境でもデプロイは同じ意味ですが、それぞれの環境のデプロイメント・タスクにはいくつかの違いがあります。アプリケーションへの変更は、実稼働環境にコミットする前にテスト・サーバーでテストするのが最も望ましいので、WebSphere Process Server を使用してアプリケーションを実稼働環境にデプロイする前に、WebSphere Integration Developer を使用してアプリケーションをテスト環境にデプロイします。

多数のアプリケーション・ファイルをデプロイする必要がある場合は、バッチ・ファイルを使用することもできます。バッチ・ファイルについては、「WebSphere Process Server *Developing and Deploying Modules*」PDF ファイルを参照してください。

関連概念

27 ページの『サービス・アプリケーションの開発およびデプロイ』
WebSphere ESB にデプロイされると、サービス間でメッセージ・ベースの通信が可能になり、対話エンドポイント間で送信途中のメッセージを処理および操作できるようなサービス・アプリケーションを開発できます。

WebSphere Process Server におけるアプリケーションの管理

IBM WebSphere Process Server の管理には、アプリケーションやリソース自体の作業だけでなく、アプリケーションやリソースのデプロイ先となる環境の準備、モニター、および変更も含まれます。

アプリケーションの管理については、「*Administering*」PDF ファイルを参照してください。

WebSphere Process Server には、ランタイム環境の管理用として以下のインターフェースがあります。

- 管理コンソール

管理コンソールは、WebSphere Process Server で稼動する各種のアプリケーション、サービス、およびアプリケーションのリソースをモニター、更新、停止、および開始できる、ブラウザ・ベースのインターフェースです。また、管理コンソールは、リレーションシップの処理に使用したり、失敗した WebSphere Process Server イベントの検索および解決に使用したりできます。

管理コンソールには、WebSphere Application Server およびその他のユーザー定義製品の管理機能もあります。

管理コンソールについては、「*Administering*」PDF ファイルを参照してください。

- Business Process Choreographer Explorer

Business Process Choreographer Explorer は、ビジネス・プロセスやヒューマン・タスクを管理するための管理機能の基本セットを提供する、スタンドアロンの 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、およびそ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操作することもできます。例えば、新規プロセス・インスタンスの開始、失敗アクティビティの修復と再開、作業項目の管理、完了したプロセス・インスタンスやタスク・インスタンスの削除が可能です。

- スクリプト・プログラム

WebSphere 管理 (wsadmin) スクリプト・プログラムは、非グラフィカルのコマンド・インタープリター環境であり、スクリプト言語での管理オプションの実行、

および実行用のスクリプト言語プログラムの実行依頼が可能です。管理コンソールと同じタスクがサポートされています。 `wsadmin` ツールは、実稼働環境および不在操作向けです。

- コマンド行ツール

コマンド行ツールは、特定のタスクを実行するために、オペレーティング・システムのコマンド行プロンプトから実行する簡単なプログラムです。このツールを使用して、アプリケーション・サーバーの始動と停止、サーバー状況の確認、ノードの追加または除去、およびその他のタスクを実行できます。WebSphere Process Server コマンド行ツールには `serviceDeploy` コマンドが含まれます。このコマンドは、WebSphere Integration Developer 環境からエクスポートされた `.jar`、`.ear`、`.war`、および `.rar` ファイルを処理して、これらのファイルを実動サーバーへのインストール用に準備します。

- 管理プログラム

Java Management Extensions (JMX) 仕様に基づく Java クラスおよびメソッドのセットを使用することにより、Service Component Architecture (SCA) およびビジネス・オブジェクトの管理がサポートされます。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の使用法を示す例、および個々のメソッド説明への参照が含まれています。

WebSphere Process Server におけるセキュリティ

IBM WebSphere Process Server は、IBM WebSphere Application Server のセキュリティに基づいたセキュリティのインフラストラクチャーとメカニズムを備えています。

セキュリティについて詳しくは、「WebSphere Process Server *Securing Applications and their Environment*」PDF ファイルを参照してください。

WebSphere Process Server におけるパフォーマンス・モニター

アプリケーションをモニターして、応答時間、サーバー上で実行中のシステムの基本的な正常性、およびアプリケーションのリソース使用量を把握することができます。

IBM WebSphere Process Server モニター機能には、Tivoli® Performance Viewer を使用したパフォーマンス・モニターと、Common Base Event および Common Event Infrastructure を使用したイベント・モニターが含まれます。

詳しくは、「WebSphere Process Server *Monitoring*」PDF ファイルを参照してください。

サンプルとチュートリアル

IBM WebSphere Process Server を使用してゴールに到達する方法を習得するために、チュートリアルとサンプルを含む教材を使用できます。

サンプルとチュートリアルは、IBM Education Assistant の Web サイトから入手できます。管理用タスクのためのチュートリアルは、IBM WebSphere Process Server インフォメーション・センターにあります。

チュートリアル

IBM Education Assistant を使用して、チュートリアルを検索できます。

IBM Education Assistant のチュートリアル

IBM Education Assistant サイトには、必要に応じて使用できるチュートリアルがあります。IBM Education Assistant について詳しくは、IBM Education Assistant Web ページを参照してください。

WebSphere Process Server チュートリアル

WebSphere Process Server 資料には、管理、セキュリティー・セットアップ、およびモニターに関するいくつかのタスクを支援するチュートリアル・トピックが含まれます。

WebSphere Process Server 管理、セキュリティー、およびモニターのタスクに関するチュートリアルについては、「*Administering WebSphere Process Server*」PDF ファイル、「*Securing Applications and their Environment*」PDF ファイル、および「*Monitoring*」PDF ファイルを参照してください。

サンプル (サンプル・ギャラリー) へのアクセス

統合アプリケーション成果物のサンプルは、サンプル・ギャラリーで入手できます。

サンプル・ギャラリーには、IBM WebSphere Integration Developer が生成する成果物や、IBM WebSphere Process Server 上でデプロイされる成果物などの単純な成果物のサンプルがあります。

WebSphere Process Server サンプル・ギャラリーをインストールおよび表示するには、以下のステップを実行します。

1. WebSphere Process Server for z/OS インストール・メディアのコンテンツをアンロードします。インストール・メディアをアンロードすると、サンプル・パッケージがシステムにロードされます。インストール・スクリプトを実行して、サンプル・ディレクトリーでファイルへの symlinks を作成します。

サンプルは、`install_root/samples` ディレクトリーにインストールされます。

WebSphere Process Server 上で実行されるアプリケーションには、ビジネス・オブジェクト、リレーションシップ定義、ビジネス・ルールなど、アプリケーションをインストールする前にデプロイしておく必要のある XML 成果物があります。WebSphere Process Server には、これらの成果物を作成およびデプロイするための `serviceDeploy` というユーティリティーがあります。

`install_root/samples/lib` 内の各サンプル・アプリケーション用のエンタープライズ・アーカイブ (EAR) ファイルに、これらの成果物が格納されています。

`sampleDeploy` ユーティリティーは、サンプルに必要な個別のパラメーターを指定した `serviceDeploy` を呼び出します。 `sampleDeploy` を実行すると、オリジナル

の EAR ファイルと同じディレクトリーに *sample_nameDeployed.ear* という名前の 2 つ目の EAR ファイルが作成されます。この新しい EAR ファイルには、オリジナルの EAR ファイル内に格納されていた Web アーカイブ (WAR) ファイルに加えて、デプロイされた成果物を格納する追加の Java アーカイブ (JAR) ファイルおよび WAR ファイルが格納されます。デプロイされた EAR ファイルは、WebSphere Process Server にエンタープライズ・アプリケーションとしてインストールすることができます。

詳しくは、「WebSphere Process Server for z/OS *Developing and Deploying Modules*」PDF ファイルの『*serviceDeploy* を使用したインストール可能な EAR ファイルの作成』を参照してください。

2. サーバーを始動します。
3. サンプル・ギャラリーは、Web ブラウザーから直接、リンク <http://xxxxxxx:9080/WSSamples/en/index.html> (xxxxxxx はターゲット z/OS システムのホスト名または IP アドレスを表す) を使用してアクセスすることができます。各サンプルをサーバーにデプロイしたりインストールしたりするアクションが、サンプルそのものの一部です。「サンプル・ギャラリー」に、最初に WebSphere Process Server のサンプルが、インストール可能なサンプルとしてリスト表示されます。
4. インストールしたい各サンプルをクリックして、ブラウザー・ウィンドウに表示される、各サンプルのインストールのための指示に従います。
5. 「サンプル・ギャラリー」内の「最新表示」をクリックすると、各サンプルがインストール済みサンプルとしてリスト表示されます。インストールしたサンプルごとに、サンプル名を選択してブラウザー・ウィンドウを開き、追加情報を表示したり、サンプルを実行するオプションを選択することができます。
6. **6.0.1+** クラスタリングされていない分散 WebSphere Process Server 環境でサンプルを実行する場合は、以下のステップを実行します。
 - a. デプロイメント・マネージャー・ノードがあるマシンで、コマンド `install_root/samples/bin/installwbi -node node_name -server server_name -samples SamplesGallery WBISamplesGallery` を実行します。
 - b. 管理コンソールの「エンタープライズ・アプリケーション」の下で、SamplesGallery と WBISamplesGallery を開始します。
 - c. ブラウザーを開いて、`http://host_name:host_port /WSSamples/index.jsp` にあるサンプル・ギャラリーにアクセスします。
 - d. サンプル・ギャラリー内の指示に従って、各サンプルをインストールおよび実行します。その際、`installwbi` コマンドで必ず `-node node_name -server server_name` パラメーターを使用するようにしてください。
7. **6.0.1+** クラスタリングされた分散 WebSphere Application Server 環境でサンプルを実行する場合は、以下のステップを実行します。
 - a. 管理コンソールで、「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。
 - b. 「参照」ボタンをクリックして、`install_root/samples/lib/SamplesGallery` ディレクトリー内の `SamplesGallery.ear` ファイルを探し出します。

- c. EAR ファイルをインストールし、その際にすべてのデフォルトを受け入れます。ただしターゲット・マッピング・パネルに関しては、サンプル・ギャラリーのインストール先となるサーバーまたはクラスターを指定できます。
- d. `install_root/samples/lib/SamplesGallery` 内の `WBISamplesGallery.ear` ファイルに関して、前のステップを繰り返します。
- e. インストールしたアプリケーションを開始します。
- f. ブラウザーを開いて、`http://host_name:host_port /WSsamples/index.jsp` にあるサンプル・ギャラリーにアクセスします。
- g. サンプル・ギャラリー内の指示に従って、各サンプルをインストールおよび実行します。その際は、クラスターをサポートしない `installwbi` コマンドの代わりに、管理コンソール上で「**新規アプリケーションのインストール**」を使用してください。 `install_root/samples/lib/sample_name` ディレクトリー内で各サンプルのデプロイ済み EAR ファイルを探し出します。

標準への準拠

この製品は、アクセシビリティに関する標準、情報処理標準、ソフトウェアのダウンロード・セキュリティに関する標準、およびインターネット・プロトコル標準など、いくつかの政府標準および業界標準に準拠しています。

アクセシビリティ機能

アクセシビリティ機能は、運動障害や視覚障害など、身体に障害を持つユーザーがソフトウェア・プロダクトを快適に操作できるようにするための機能です。

IBM WebSphere Process Server のインストール機能および管理機能には、以下のアクセシビリティ機能が組み込まれています。

- インストール

WebSphere Process Server は、グラフィカル形式とサイレント形式のいずれかでインストールできます。アクセス支援が必要な方には、サイレント・インストール・プログラムをお勧めします。

WebSphere Process Server for z/OS をコマンド行からサイレント・インストールします。このサイレント・インストールは、アクセス可能性のニーズを満たしています。

詳しくは、「WebSphere Process Server *Installing*」PDF ファイルの『製品のサイレント・インストール』を参照してください。

詳細については、「WebSphere Process Server for z/OS *Installing*」PDF ファイルの『Running the install script』を参照してください。

- 管理

管理コンソールは、製品との対話するための基本インターフェースです。このコンソールは、標準の Web ブラウザーに表示されます。管理者は、Microsoft® Internet Explorer または Netscape Navigator などの利用できる Web ブラウザーを使用して、以下のことを行うことができます。

- スクリーン・リーダー・ソフトウェアとデジタル・スピーチ・シンセサイザーの使用による、画面の表示内容の聴覚による認識。
- IBM ViaVoice® などの音声認識ソフトウェアの使用による、データ入力およびユーザー・インターフェースのナビゲート。
- マウスの代わりにキーボードを使用することによる機能の操作。

提供されているグラフィカル・インターフェースの代わりに、標準テキスト・エディターおよびスクリプト・インターフェースまたはコマンド行インターフェースを使用して、製品の機能を構成および管理することができます。

該当する場合は、特定の製品機能の資料には、その機能のアクセシビリティについての詳細が記載されています。

連邦情報処理標準

連邦情報処理標準 (FIPS) は、米国連邦情報・技術局 (NIST) が、連邦政府コンピューター・システムのために発行した標準およびガイドラインです。

IBM WebSphere Process Server は、連邦情報処理標準に準拠するすべての暗号化機能に関して、IBM WebSphere Application Server に依存しています。

FIPS は、セキュリティおよびインターオペラビリティなど、標準に関する連邦政府の切実な要求がある一方で、実施可能な業界標準または解決方法が存在しない場合のために開発されました。政府機関および金融機関はこれらの標準を使用し、製品が明記されたセキュリティ要件に適合していることを確認します。この標準について詳しくは、「National Institute of Standards and Technology」を参照してください。

WebSphere Application Server には、Java Secure Socket Extension (JSSE) および Java Cryptography Extension (JCE) を含む暗号モジュールが組み込まれています。この暗号モジュールは、FIPS 140-2 認証が行われています。WebSphere Application Server の資料では、FIPS 認証を受けた IBM JSSE モジュールおよび JCE モジュールは、それぞれ IBMJSSEFIPS および IBMJCEFIPS と呼ばれています。

FIPS を使用可能にするには、WebSphere Application Server for z/OS インフォメーション・センターのトピック『**連邦情報処理標準 Java セキュア・ソケット拡張機能ファイルの構成**』を参照してください。FIPS を使用可能にすると、サーバーの一部のコンポーネント (暗号スイート、暗号プロバイダー、ロード・バランサー、キャッシング・プロキシ、高可用性マネージャー、データ複製サービスなど) が影響を受けます。

Common Criteria

National Institute of Standards and Technology (NIST) が開発した Common Criteria を使用すると、システムで使用するソフトウェアを安全にダウンロードすることができます。

IBM WebSphere Process Server は、National Institute of Standards and Technology が開発した Common Criteria 標準に準拠しています。

IT 製品またはシステムが保持している情報は、企業の目的達成を可能にする重要なリソースです。また、個人ユーザーは、IT 製品またはシステムに入っている個人情報
は非公開で、必要なときに使用でき、勝手に変更されないことを期待していま
す。IT 製品またはシステムは、その機能を実行する一方で、情報を適切に管理し
て、望まない、あるいは許可されない配布、変更、消失といった障害から情報を確
実に保護する必要があります。IT セキュリティーという用語は、このような障害の
防止および緩和を表す場合に使用されます。

IT 利用者の多くは、IT 製品またはシステムのセキュリティが十分信頼できるか
どうかを判断するのに必要な知識や専門技術、情報源を持っていませんが、だから
といって、開発者の意見だけを信頼するわけではありません。IT 製品またはシス
テムのセキュリティの信頼性を高めるには、セキュリティ分析 (すなわち、セ
キュリティー評価) を行うという選択肢もあります。

Common Criteria の詳細情報は、WebSphere Application Server および WebSphere
Process Server の推奨される更新も含めて、IBM サポート・サイトから入手できま
す。

インターネット・プロトコル・バージョン 6.0

IBM WebSphere Process Server は、インターネット・プロトコル・バージョン 6.0
のすべての互換性に関して、WebSphere Application Server に依存しています。

IBM WebSphere Application Server バージョン 6.0 およびその JavaMail コンポーネ
ントは、インターネット・プロトコル・バージョン 6.0 (IPv6) をサポートしていま
す。

WebSphere Application Server におけるこの互換性について詳しくは、WebSphere
Application Server for z/OS インフォメーション・センターで、『IPv6 support』を
参照してください。

IPv6 について詳しくは、www.ipv6.org を参照してください。

グローバル化

グローバル化された製品は、言語や文化の障壁なく使用することができ、特定ロー
ケルに関して使用可能に設定できます。

IBM WebSphere Process Server バージョン 6.0.0 およびバージョン 6.0.1 は英語で
のみ使用可能ですが、すべてのローケルをサポートしています。つまり、データは
他の言語で入力できますが、メッセージおよびインターフェース・エレメントは英
語です。

WebSphere Process Server バージョン 6.0.1.1 は、その他の言語で使用できます。詳
細については、http://www.ibm.com/common/ssi/rep_ca/9/897/ENUS205-309/ で
WebSphere Process Server の発表レターを参照してください。

WebSphere Application Server で使用できる国際化対応サービスの詳細については、
WebSphere Application Server for z/OS インフォメーション・センターで、
『WebSphere 拡張機能』を参照してください。

双方向言語サポート

WebSphere Process Server で処理するデータは ILYNN (暗黙、左から右、オン、オフ、公称) の双方向言語形式である必要があります。これは、Windows® の双方向言語形式でもあります。WebSphere Process Server で実行するアプリケーションで使用されるその他のすべての双方向言語形式は、WebSphere Process Server に導入する前に変換する必要があります。

双方向属性

適切な双方向言語形式に合わせて設定する必要がある 5 つの属性があります。これらの属性と設定値を以下の表に示します。

文字位置	目的	値	説明	デフォルトの設定値
1	順序スキーマ	I または V	暗黙 (論理) または表示	I
2	方向	L または R	左から右、または右から左	L
3	対称スワッピング	Y または N	対称スワッピングのオン/オフ	Y
4	形状指定	Y または N	テキストの形状を指定するか、または指定しない	N
5	数値の形状指定	H、C、N	ヒンディ語、コンテキスト、公称	N

サポートされる双方向言語形式へのデータ変換を行うのは、クライアント・アプリケーション、外部コンポーネント (Web サービス、Stateless Session Bean、およびカスタム・コードなど)、または WebSphere Process Server で実行するためのソリューションの構築を行う任意のユーザーのいずれかです。

string の双方向言語変換の例については、『例: string 型データでの双方向変換の使用』を参照してください。

サービス・データ・オブジェクトの双方向言語変換の例については、『例: DataObject 型データでの双方向変換の使用』を参照してください。

注: ユーザー・インターフェース (ブラウザ) のロケール設定で、双方向言語の表示および編集形式を定義します。

双方向言語について詳しくは、IBM developerWorks に関するテクニカル・アーティクル (www.ibm.com/developerworks/websphere/library/techarticles/bidi/bidigen.html で入手可能) を参照してください。

例: DataObject 型データでの双方向 (bi-di) 変換の使用

双方向言語スクリプト内の情報を使用する場合、データの形式の変換が必要になる場合があります。以下に、DataObject 型データについて変換を実行するコーディングを段階的に説明した例を示します。

6.0.0 ONLY

IBM WebSphere Process Server バージョン 6.0.0 で、Windows、AIX[®]、または Linux[®] オペレーティング・システム版の JDK 1.4.1 を使用している場合は、これらのクラスをインストールする必要はありません。

モジュールでは、DataObject 型の情報を双方向形式から別の形式へ変換する必要があります。

注: 形式がよく分からない場合は、40 ページの『双方向 (bi-di) 形式のストリングの値』を参照してください。

1. bi-di エンジンのインプリメンテーションを含むすべての bi-di クラスを組み込みます。

```
import com.ibm.bidiTools.bdlayout.*;
```

2. DataObject 型オブジェクトを操作する必要のあるすべてのクラスを組み込みます。

```
import commonj.sdo.DataObject;  
import commonj.sdo.Type;  
import commonj.sdo.Property;
```

3. DataObject 型オブジェクトに含まれる各種のストリングを格納するストリング変数を定義します。このステップにより、DataObject の横移動を再帰的に実行しながら、type String の属性のフィルタリングを容易に行うことができます。

```
String STRING_STR_TYPE = "String";  
String NORM_STRING_STR_TYPE = "normalizedString";  
String TOKEN_STR_TYPE = "token";  
String LANG_STR_TYPE = "language";  
String NAME_STR_TYPE = "Name";  
String NMTOKEN_STR_TYPE = "NMTOKEN";  
String NCNAME_STR_TYPE = "NCName";  
String ID_STR_TYPE = "ID";  
String IDREF_STR_TYPE = "IDREF";  
String IDREFS_STR_TYPE = "IDREFS";  
String ENTITY_STR_TYPE = "ENTITY";  
String ENTITIES_STR_TYPE = "ENTITIES";
```

4. プロパティの型が String であるかどうかを検査する関数を定義します。

```
private static boolean isStringFamilyType (Property property) {  
    boolean rc = false;  
    if ((property.getType().getName().equalsIgnoreCase(STRING_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(NORM_STRING_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(TOKEN_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(LANG_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(NAME_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(NMTOKEN_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(NCNAME_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(ID_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(IDREF_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(IDREFS_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(ENTITY_STR_TYPE)) ||  
        (property.getType().getName().equalsIgnoreCase(ENTITIES_STR_TYPE)))  
        rc = true;  
    return rc;  
}
```

5. DataObject 全体に bi-di 変換を適用する再帰的関数を定義します。

注: 以下に、コード・ロジックが従う基本的な前提事項を説明します。

- bi-di 変換は、string 型のプロパティにのみ適用されます。
- DataObject 内の string 型のプロパティは、1 つの bi-di 形式で格納されます。

```
DataObject BiDiDataObjTransformationBO(DataObject boIn, String formatIn, String formatOut){
    Type type;
    Property property;

    if (boIn == null) return null;

    type = boIn.getType();
    List propertyList = type.getProperties();
    for (int propertyNumber = 0; propertyNumber < propertyList.size(); propertyNumber++){
        property = (Property) propertyList.get(propertyNumber);
        String propertyName = property.getName();
```

a. string でないプロパティはすべてスキップします。

```
        if (!isStringFamilyType(property))
            continue;

        if (property.isContainment()) {
            if (property.isMany()) {
                List childsList = boIn.getList(property);
```

b. 変換を再帰的に呼び出して、子オブジェクトを処理します。

```
                for (int childNumber = 0; childNumber < childsList.size();
                    childNumber++){
                    BiDiDataObjTransformationBO(connectionContext,
                    ((DataObject)childsList.get(childNumber)),formatIn, formatOut);
                }
            } else {
```

c. 変換を再帰的に呼び出して、含まれている任意のビジネス・オブジェクトの子オブジェクトを処理します。

```
                BiDiDataObjTransformationBO(connectionContext,
                ((DataObject)boIn.get(property)),formatIn, formatOut);
            }
        } else {
```

d. 単純な string 属性を変換します。

```
                String str = BiDiStringTransformation(
                (boIn.getString(propertyName)),formatIn, formatOut);
                boIn.setString(propertyName, str);
            }
        }
    }
    return boIn;
}
```

例: string 型データでの双方向 (bi-di) 変換の使用

双方向言語スクリプト内の情報を使用する場合、データの形式の変換が必要になる場合があります。以下に、string 型データを変換するコーディングを段階的に説明した例を示します。

6.0.0 ONLY

IBM WebSphere Process Server バージョン 6.0.0 の場合は、モジュールを開発しているサーバー上に、bi-di エンジンのインプリメンテーションを含むすべてのクラスがインストールされていることを確認してください。

Windows、AIX、または Linux オペレーティング・システム用の JDK 1.4.1 を使用する場合は、以下に示すクラスをインストールする必要はありません。

モジュールでは、ストリング情報を双方向形式から別の形式へ変換する必要があります。

注: 形式がよく分からない場合は、『双方向 (bi-di) 形式のストリングの値』を参照してください。

1. bi-di エンジンのインプリメンテーションを含むすべての bi-di クラスを組み込みます。

```
import com.ibm.bidiTools.bdlayout.*;
```

2. 変換するデータ・オブジェクトを格納するストリングと、入出力形式値を定義します。

入力形式は、ストリング・オブジェクトが現在格納されている bi-di 形式です。

出力形式は、ストリング・オブジェクトを格納する bi-di 形式です。

```
String strIn = new String("Hello world");
String formatIn = "ILYNN";
String formatOut = "VLYNN";
```

3. BidiStringTransformation 関数を呼び出します。

```
String strOut = BiDiStringTransformation(strIn, formatIn, formatOut);
```

```
String BiDiStringTransformation(String strIn, String formatIn, String formatOut) {
```

- a. 入力ストリングがヌルであるかどうかをテストします。

```
    if (strIn == null) return null;
```

- b. 変換を実行します。

```
    BidiFlagSet flagsIn;
    BidiFlagSet flagsOut;
    formatIn = formatIn.toUpperCase();
    formatOut = formatOut.toUpperCase();
```

```
    if (formatIn != null)
        flagsIn = new BidiFlagSet(formatIn.toCharArray());
    else
        flagsIn = new BidiFlagSet();
```

```
    if (formatOut != null)
        flagsOut = new BidiFlagSet(formatOut.toCharArray());
    else
        flagsOut = new BidiFlagSet();
```

```
    if (flagsIn.equals(flagsOut)) return strIn;
    String strOut = BiDiStringTransformation(strIn, flagsIn, flagsOut);
    return strOut;
}
```

双方向 (bi-di) 形式のストリングの値

双方向 (bi-di) 言語形式のストリングの値は、双方向スクリプトの別の形式への変換を制御します。

目的

以下の表を使用して、String 型または DataObject 型データを別の形式に変換する際に使用する入力ストリングまたは出力ストリングの正しい値を判別してください。

表 1. 双方向言語形式のSTRING値

文字位置	目的	指定可能な値	デフォルト値	意味
1	順序スキーマ	I	I	暗黙
		V		表示
2	方向	L	L	左から右
		R		右から左
		C		コンテキスト (左から右)
		D		コンテキスト (右から左)
3	対称スワッピング	Y	Y	対称スワッピングがオン
		N		対称スワッピングがオフ
4	形状指定	S	N	テキストの形状を指定する
		N		テキストの形状を指定しない
		I		語頭形の指定
		M		語中形の指定
		F		語尾形の指定
		B		独立形の指定
5	数値	H	N	ヒンディ語 (各国語)
		C		コンテキスト
		N		公称

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が、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