



BEA WebLogic Platform™

用語集

著作権

Copyright © 2002 BEA Systems, Inc. All Rights Reserved.

限定的権利条項

本ソフトウェアおよびマニュアルは、BEA Systems, Inc. 又は日本ビー・イー・エー・システムズ株式会社（以下、「BEA」といいます）の使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができ、同契約の条項通りにのみ使用またはコピーすることができます。同契約で明示的に許可されている以外の方法で同ソフトウェアをコピーすることは法律に違反します。このマニュアルの一部または全部を、BEA からの書面による事前の同意なしに、複製、複製、翻訳、あるいはいかなる電子媒体または機械可読形式への変換も行うことはできません。

米国政府による使用、複製もしくは開示は、BEA の使用許諾契約、および FAR 52.227-19 の「Commercial Computer Software-Restricted Rights」条項のサブパラグラフ (c)(1)、DFARS 252.227-7013 の「Rights in Technical Data and Computer Software」条項のサブパラグラフ (c)(1)(ii)、NASA FAR 補遺 16-52.227-86 の「Commercial Computer Software--Licensing」条項のサブパラグラフ (d)、もしくはそれらと同等の条項で定める制限の対象となります。

このマニュアルに記載されている内容は予告なく変更されることがあり、また BEA による責務を意味するものではありません。本ソフトウェアおよびマニュアルは「現状のまま」提供され、商品性や特定用途への適合性を始めとする（ただし、これらには限定されない）いかなる種類の保証も与えません。さらに、BEA は、正当性、正確さ、信頼性などについて、本ソフトウェアまたはマニュアルの使用もしくは使用結果に関していかなる確約、保証、あるいは表明も行いません。

商標または登録商標

BEA、Jolt、Tuxedo、および WebLogic は BEA Systems, Inc の登録商標です。BEA Builder、BEA Campaign Manager for WebLogic、BEA eLink、BEA Manager、BEA WebLogic Commerce Server、BEA WebLogic Enterprise、BEA WebLogic Enterprise、BEA WebLogic Express、BEA WebLogic Integration、BEA WebLogic Personalization Server、BEA WebLogic Platform、BEA WebLogic Portal、BEA WebLogic Server、BEA WebLogic Workshop、および How Business Becomes E-Business は BEA Systems, Inc の商標です。

その他の商標はすべて、関係各社がその権利を有します。

BEA WebLogic Platform 用語集

パート番号	日付	ソフトウェアのバージョン
なし	2002年6月	7.0

用語集

この用語集は、BEA WebLogic Platform のドキュメントで使用される用語の意味を定義したものです。URL 以外に青く表示されている用語については、この用語集で定義されています。

< イメージ マップ >

A

アボート (**abort**)

データベース レコードなどの保護されたすべてのリソースに割り当てる値がトランザクションの最初と変わらないようにトランザクションを停止すること。

アクセス制御 (**access control**)

ソフトウェア リソースの使用に対し、プリンシパル (人間またはソフトウェアのエンティティ) に与える機能の制限のセット。これらの制限により、リソースへの無認可使用を防止できます。

アクセス制御情報 (**access control information: ACI**)

リソースへのアクセス要求を開始するユーザに関する情報。ACI を使用して、アクセス制御の実行に関する決定を行います。

アクセス制御リスト (**access control list: ACL**)

プリンシパルによるリソース (ネットワーク サービスなど) へのアクセスを認可または拒否するために使用するデータ構造。ACL の各エントリには、特定のプリンシパルに関するパーミッション設定が含まれています。WebLogic Server における ACL の実装は `java.security.acl` パッケージに基づいています。

WebLogic Integration 環境では、さまざまな種類のリソースをさまざまなタイプのプリンシパルに対して使用できます。

WebLogic Integration 環境では、さまざまな種類のリソースをさまざまなタイプのプリンシパルに対して使用できます。

-
- **WebLogic Integration** のユーザには、3 種類のリソース (管理とコンフィグレーションに使用する **Java Server Pages (JSP)**、**Administration Console**、および **JDBC 接続プール**) へのアクセスが認可されます。
 - **トレーディング パートナ**には、転送サーブレットと **WebLogic Server JDBC 接続プール**へのアクセスの認可を与えることができます。
 - **WebLogic Integration** は転送サーブレットへのアクセスが認可されています。

レルム (realm)、パーミッション (permission)、プリンシパル (principal)、および認証 (authentication) を参照。

ACID プロパティ (ACID properties)

トランザクション処理システムの基本特性。

- **原子性** : トランザクション中にデータベースに対して行ったすべての変更は永続するか、またはすべての変更が元の状態に戻ります。
- **一貫性** : トランザクションが正しく実行されると、データベースが前の有効なステートから新しい有効なステートに変化します。
- **隔離性** : あるトランザクションで行われた変更内容は、そのトランザクションが完了するまで他のトランザクションに対しては表示されません。
- **持続性** : 完了したトランザクションの結果は永続的で、システム障害が起きた場合も復旧が可能です。

トランザクション (transaction) を参照。

ACL

アクセス制御リスト (access control list: ACL) を参照。

アクション (action)

WebLogic Integration でサポートされている **Business Process Management** の各機能で実行される動作の基本単位。アクションはワークフロー (workflow) とタスク (task) の動作を定義し、タスク、決定 (decision)、およびイベントの一部として定義されます。**WebLogic Integration** には、さまざまなアクションが標準で用意されています。

シナリオアクション (scenario action) を参照。

ActiveX コントロール (ActiveX controls)

コンポーネント オブジェクト モデル (COM) のテクノロジーを使用して、他の COM サービスやコンポーネントとの相互運用性を確保するためのコンポーネント。

Component Object Model (COM) を参照。

広告 (ad)

各種フォーマットで表すプロモーション用メッセージ。使用できるフォーマットは、HTML テキスト、画像 (GIF、JPG、PNG)、または Shockwave ムービーです。

広告は、広告用のコンテンツ管理システムをクエリするメカニズムの広告プレースホルダとともに使用し、結果を JSP で表示します。ビジネスアナリストは BEA E-Business Control Center を使用して、広告プレースホルダのクエリを定義し、ビジネス エンジニア (BE) は、ビジネスアナリストが広告を表示したい場所にプレースホルダのタグを作成します。また、ビジネスアナリストは、キャンペーンのシナリオを作成し、広告プレースホルダにクエリを追加することもできます。

ビジネスアナリスト (Business Analyst: BA)、ビジネス エンジニア (Business Engineer: BE)、および E-Business Control Center™ を参照。

広告識別子 (ad identifier)

相対パスを含む広告の固有名。広告識別子固有のコンテンツは、さまざまなコンテンツ管理システムにおいて変化させることができます。

コンテンツ管理システム (content management system) および広告 (ad) を参照。

アダプタ (adapter)

エンタープライズ情報システムと統合サーバ間のインタフェースを提供するソフトウェア コンポーネント。

裁決 (adjudication)

矛盾する認可決定を解消するためのメソッド。

Administration Console

BEA ソフトウェアに組み込まれているアプリケーションを設定、モニタするために管理者が使用する HTML インタフェース。

WebLogic Server Administration Console および WebLogic Integration B2B Console を参照。

拡張プログラム間通信機能 (Advanced Program to Program Communications: APPC)

ポータブル、ワークステーション、ミッドレンジ コンピュータ、ホスト コンピュータなどの異なるコンピュータのプログラム間的高速通信を SNA 上で可能にするソフトウェア プロトコル。このプロトコルは、オープンであり、公開されています。LU 6.2 としても知られています。

Systems Network Architecture (SNA) を参照。

Advisor

Personalization Server のサービスとコンポーネントを結びつけ、パーソナライズされたコンテンツをユーザに配信する Personalization Server の機能。Advisor は、コンテンツをユーザ プロファイル情報に合わせることでこれを実現します。具体的には、ユーザ管理コンポーネントでユーザ プロファイルから情報を収集し、その情報を Rules サービスに発行し、コンテンツ管理コンポーネントで使われるドキュメント管理システムにクエリを発行します。Advisor においてクエリの結果が JSP ページで表示されるか、Java サーブレットで表示されるかは、開発者が決定します。

Advisor には、JSP タグ ライブラリ、および EJB 開発者が EJB ステートレスセッション Bean にアクセスするのに使用する API が含まれています。エンタープライズ JavaBeans (Enterprise JavaBeans: EJB)、アプリケーションプログラミング インタフェース (application programming interface: API)、ユーザ プロファイル (user profile)、Web アプリケーション (Web application: Webapp)、および開発者 (developer) を参照。

匿名 (anonymous)

認証されていない WebLogic Server 環境のリソースにアクセスするプリシパルのタイプ。

Apache XML パーサ (Apache XML Parser)

Apache Software Foundation により提供されている、W3C XML、DOM、および SAX 標準を実装したサードパーティのパーサ。

API

アプリケーションプログラミング インタフェース (application programming interface: API) を参照。

APPC

拡張プログラム間通信機能 (Advanced Program to Program Communications: APPC) を参照。

アプレット (applet)

通常は HTML ページに埋め込まれ、Java 対応の Web ブラウザで表示される、クライアントサイドの Java プログラム。

アプリケーション (application)

エンドユーザがコンピューティングタスクを実行するために集合的に使用する 1 つまたは複数のソフトウェアプログラム。

アプリケーションアダプタ (application adapter)

エンタープライズ情報システムと統合サーバ間のインタフェースを提供するソフトウェアコンポーネント。アプリケーションアダプタは、アプリケーションのすべての機能に対するシステムレベルのインタフェースを表します。

アプリケーションコンポーネント (application component)

- 1) アプリケーションサーバでデプロイ、管理、実行される、EJB、JSP、サーブレットなどのサーバサイドのコンポーネント。
- 2) コンポーネントは、アプリケーションサーバによって Web クライアントで利用できるようになり、Java アプレットや DHTML などの Web クライアント層で実行されます。

アプリケーションサーバ (application server) を参照。

Application Integration

BEA Systems, Inc によって開発された、エンタープライズ間およびエンタープライズ内統合のための J2EE Connector Architecture 標準ベースの統合ブローカ。

アプリケーションプログラミング インタフェース (application programming interface: API)

- 1) 特定のシステムのソフトウェア製品をサポートするための、機能を含むアプリケーションレベルの環境。
- 2) アプリケーション内で、開発者がクライアント / サーバ要求を開始、完了するためのコードのセット。
- 3) サービスを起動する方法を定義した呼び出し規約。ソフトウェアプログラムが別のサービスを利用する際に使用する、適切に定義されたプログラミング インタフェース (エントリ ポイント、呼び出しパラメータ、戻り値) のセット。

アプリケーション サーバ (application server)

- 1) アプリケーション ロジックの共有と処理、およびバックエンド リソースへの接続を提供するサーバ。リソースには、データベース、ERP アプリケーション、従来型のメインフレーム アプリケーションが含まれます。また、アプリケーションサーバには、ユーザ インタフェースを開発したり、アプリケーションを Web プラットフォームにデプロイしたりするためのツールもあります。BEA WebLogic Server は Java アプリケーションサーバです。
- 2) WebLogic Workshop 環境の場合 — 開発者が (通常はコンポーネントを介して) プロジェクトのビジネス ロジックを分離し、3 層アプリケーションを開発するために設計されたサーバ。また、多くのアプリケーションサーバには、トランザクション管理、クラスタ化とフェイルオーバー、ロードバランシングなどの機能もあります。
- 3) WebLogic Workshop 環境の場合 — Java 2 Enterprise Edition (J2EE) プラットフォームに対応した Java アプリケーションサーバ。

エンタープライズ リソース プランニング (enterprise resource planning: ERP) を参照。

アプリケーションビュー (application view)

アプリケーションの機能に対するビジネスレベル インタフェース。アプリケーションビューは単一のビジネス目的のためにコンフィグレーションされ、そのビジネス目的に関連するサービスだけを含みます。また、アプリケーションビューはビジネス目的をサポートするイベントとサービスの両方を表し、これにより、ビジネス ユーザは、アプリケーションを使用したすべての通信でアプリケーションビューを使用できます。

非同期 (asynchronous)

発生時間が異なるために、イベントの発生時間の関連性が予測できないイベントの特性。

Web サービスなどの分散型アプリケーションアーキテクチャにおいては、クライアントがメソッド (またはメッセージを送信する) サーバを呼び出し、サーバが応答します。サーバの応答を待つ間に、クライアントが他の処理を実行できないようにブロックされている場合、クライアントはサーバと同期しているため、対話は同期して記述されます。

サーバが応答の準備をしている間にもクライアントが他の処理を続行できるように対話が設計されている場合、応答の準備が整ったときにサーバがクライアントに通知され、対話は非同期で記述されます。

非同期のアーキテクチャは、イベント駆動型のシナリオで使用すると便利です。イベント駆動型のシナリオでは、イベントを常に実行でき、そのつど、送信者がそのイベントを処理できます。

添付ファイル (attachment)

ビジネス メッセージ (business message) と一緒に送信される情報。

属性 (attribute)

- 1) オブジェクトと値の間で識別可能な関連性。
- 2) (CORBA) OMG IDL を使用する場合は、パブリック クラス フィールドまたはデータ メンバーに類似した OMG IDL インタフェースの該当部分を参照してください。コンパイラは、OMG IDL 属性を C++ または Java プログラミング言語のアクセサおよび修飾子メソッドにマップします。たとえば、インタフェース `ball` に属性 `color` が含まれているとします。idltojava コンパイラは `color` を取得する C++ または Java プログラミング言語のメソッドを作成し、属性が読み取り専用でない限り、`color` を設定するメソッドを作成します。CORBA の属性は JavaBeans プロパティと厳密に対応しています。

認証 (authentication)

1) プリンシパルの ID を検証するのにサーバが使用するプロセス。プリンシパルは、ユーザまたはコンピュータなどです。認証は、片方向認証または双方向のクライアント / サーバ認証のいずれかで実行できます。

2) 受信者に情報のソースを保障するテクノロジー。

セキュア ソケット レイヤ (secure sockets layer: SSL) およびアクセス制御リスト (access control list: ACL) を参照。

認可 (authorization)

特定のエンティティ (ユーザまたはプロセス) がアクセスするサービスを決定し、それらのサービスにアクセスするためのパーミッションをエンティティに与えるプロセス。

WebLogic Integration 環境では、トレーディング パートナの認証はロール (role) ベースのセキュリティ機能です。ここでは、トレーディング パートナは、特定の会話 (conversation) において、トレーディング パートナのサブスクリプション (subscription) で定義されたような一定のタイプのメッセージを送受信できます。トレーディング パートナは転送サーブレットと JDBC プールへのアクセスが認可されています。

また、WebLogic Integration ソフトウェアは、メッセージをトレーディング パートナに送信するために転送サーブレットへのアクセスが認可されています。

ユーザは、JSP へのアクセスと、JDBC プールへの間接的なアクセスが認可されています。WebLogic Server アクセス制御リスト (access control list: ACL) は、ユーザのリソースへのアクセスを認可、または拒否するように設定できます。

B

B2B

企業間電子商取引 (B2B e-commerce) を参照。

企業間電子商取引 (B2B e-commerce)

電子トランザクションを介した企業間 (*Business-to-Business: B2B*) の直接売買の方法。

B2B Integration

エンタープライズ間のサプライチェーン統合などのアプリケーション統合をサポートする **WebLogic Integration** の機能。

企業間電子商取引 (B2B e-commerce) を参照。

B2B プラグイン (B2B plug-in)

B2B Integration とともに使用できるように **Business Process Management (BPM)** を拡張する **WebLogic Integration Studio** コンポーネント。

B2C

企業対消費者。

B2C 統合 (B2C integration)

企業内の情報を企業外の既知または未知のユーザやエンティティに知らせるプロセス。

BEA E-Business Control Center™

E-Business Control Center™ を参照。

BEA WebLogic Integration

WebLogic Integration を参照。

BEA WebLogic Personalization Server

WebLogic Personalization Server を参照。

BEA WebLogic Portal

WebLogic Portal を参照。

BEA WebLogic Server

WebLogic Server を参照。

Bean 管理のトランザクション (bean-managed transaction)

Enterprise JavaBean (EJB) がトランザクションの境界を制御するトランザクション。Bean 管理のトランザクションでは、**JTA** を使用して制御を指定します。EJB コードはトランザクションを管理します。これは、あるメソッドで開始し、別のメソッドで終了することができます。

エンタープライズ JavaBeans (Enterprise JavaBeans: EJB)、Java Transaction API (JTA)、トランザクション (transaction)、およびコンテナ管理によるトランザクション (container-managed transaction) を参照。

ビッグ エンディアン (**big endian**)

最上位バイトが最下位アドレスを持つバイナリ フォーマット。このフォーマットは、IBM 370 システムとほとんどの RISC 設計で使用されています。

バイナリ データ (**binary data**)

- 1) ビットのシーケンスとしてコード化されるが、必ずしも印刷可能な文字 (テキスト) のシーケンスで構成されないデータのファイル フォーマット。
- 2) 実行可能なマシン コードです。

バインド (**bind**)

- 1) アプリケーション オブジェクトやネーミング コンテキスト オブジェクトに名前を関連付けること。
- 2) クライアント アプリケーションをアプリケーション オブジェクトに接続すること。

BizTalk™

XML スキーマの発行と XML メッセージの使用のために **BizTalk Framework** を定義する業界イニシアティブ。Microsoft によって開始され、テクノロジーベンダ (SAP、CommerceOne、Ariba) およびテクノロジーユーザ (BASDA など) を含む広範囲の組織によってサポートされています。BizTalk は標準規格の制定機関ではなく、標準規格ユーザのコミュニティです。

<http://www.biztalk.org> を参照。

BPM

Business Process Management (BPM) を参照。

分岐 (branching)

入力プロセッサがパイプラインを正しく実行することにより返されるオブジェクトの値に基づき、Webflow を異なる送り先に送信するためのメカニズム。

ノード (node)、パイプライン (pipeline)、Webflow、および入力プロセッサ (input processor) を参照。

ブラウザ クライアント (browser client)

Web ブラウザを使用して他のトレーディング パートナと通信するトレーディング パートナ。

BTP

ビジネス トランザクション プロトコル (business transaction protocol: BTP) を参照。

バッファリング (buffering)

バッファをサービスのメソッドに追加して、サービスがクライアントにすぐに戻ることを確認する方法。このメカニズムにより、サーバが他の要求を処理するのをクライアントが待機する必要がなくなります。バッファ付きメソッドに呼び出しを入力すると、待ち状態となり、サーバが要求でオーバーフローするのを避けることができます。

Bulk Loader

- 1) WebLogic Integration 環境の場合 — レポジトリ データを XML ファイルにエクスポートし、XML データをレポジトリにインポートし、レポジトリからデータを削除するユーティリティ。これを使用して、レポジトリ データをある環境から別の環境、たとえば、サポートしているデータベースから別のデータベースに移行できます。Bulk Loader は複数の JDBC データベースをサポートしています。
- 2) WebLogic Portal 環境の場合 — ドキュメントのメタデータをディレクトリやファイル構造から参照実装データベースにロードするコマンドライン アプリケーション。BulkLoader はドキュメント ベースを解析し、すべてのドキュメント メタデータをロードするので、Content Management コンポーネントはドキュメントを検索できます。BulkLoader は、HTML ドキュメントだけでなくすべてのドキュメント タイプをサポートしています。

ビジネスアナリスト (**Business Analyst: BA**)

e ビジネスの効果的なモデルを考える非技術スタッフ。通常は、技術専門のビジネス エンジニアと共同で作業します。ビジネスアナリストは、個々のキャンペーンの戦略と目標を立て、**BEA E-Business Control Center** を使用してキャンペーンの実行と評価にあたります。

ビジネス エンジニア (**Business Engineer: BE**) および **E-Business Control Center™** を参照。

ビジネスドキュメント (**business document**)

ビジネス メッセージ (**business message**) のペイロード (**payload**) の XML の部分。

ビジネス エンジニア (**Business Engineer: BE**)

ビジネスアナリストと開発者のスキルを併せ持つスタッフ。BE の業務を以下に示します。

- Web サイト全体、またはその大部分の業務を管理します。
- 大幅な、または若干の変更のプロセスを調整します。
- ビジネス効率全般とシステム パフォーマンスの両方を積極的にモニタします。
- ユーザ プロファイル、ユーザ パーミッションなどを管理するために、e ビジネスの Web サイトの構造的で主要な更新を担当します。
- Web サイトの機能性を常に保持するためのプログラミング業務を後方支援します。

開発者 (**developer**) およびビジネスアナリスト (**Business Analyst: BA**) を参照。

ビジネスロジック (**business logic**)

- 1) 税金や支払いの計算などのビジネスに特化した機能を実行する、**Commerce Services** で使用される原理と説明。
- 2) ユーザとグループを電子商取引サイトの適切なコンテンツに一致させるなどのビジネスに特化した機能を実行する、**WebLogic Personalization Server** で使用される原理と説明。

ビジネス メッセージ (business message)

会話 (conversation) におけるトレーディング パートナ間の通信の基本単位。マルチパート MIME メッセージです。ビジネスドキュメント (business document)、添付ファイル (attachment)、および メッセージ ヘッダー (message header) で構成されています。

ビジネス処理 (business operation)

WebLogic Integration のユーザ定義の処理を EJB を起動するか、Java のクラスを呼び出すことにより追加するメソッド。

Business Process Management (BPM)

ワークフロー (workflow) を定義およびモニタできる WebLogic Integration オートメーション ツール。

ビジネス プロセス (business process)

ワークフロー (workflow) を参照。

ビジネス プロトコル (business protocol)

ネットワークにより、企業間でビジネス情報を電子的にやり取りするためのルール。ビジネス プロトコルは、ビジネス メッセージ (business message) の構造、メッセージの処理方法、およびそれらを適切な受信側にルーティングするための方法を規定します。WebLogic Integration トレーディング パートナは、RosettaNet ビジネス プロトコルを使用してビジネス メッセージ (business message) を送受信します。

ビジネス プロトコル定義 (business protocol definition)

ビジネス プロトコル (business protocol) を実装するロジック プラグインのセット。

ビジネス サービス (business service)

会話定義 (conversation definition) とのインタフェース。トレーディング パートナは、交信を希望する他のトレーディング パートナに対してビジネス サービスを提供します。

ビジネストランザクションプロトコル (**business transaction protocol: BTP**)

B2B の電子商取引で一般的な長期の複雑なトランザクションを管理する、オープンかつ精密に定義されたメソッドの標準。BTP は、インターネット上での複雑なマルチステップの B2B トランザクションを表現し、シームレスに管理するための XML ベースのポキャブラリ プロトコルです。BTP を使用すると、トレーディング パートナは、複雑な XML メッセージ交換を長期にわたる疎結合の会話 (conversation) として管理できます。

企業間電子商取引 (**business-to-business (B2B) e-commerce**)

企業間電子商取引 (B2B e-commerce) を参照。

企業間統合 (**business-to-business (B2B) integration**)

B2B Integration を参照。

企業消費者間統合 (**business-to-consumer integration**)

B2C 統合 (B2C integration) を参照。

バイトコード (**bytecode**)

Java 仮想マシン (Java virtual machine: JVM) のあるコンピュータで実行 (インタプリタ) できる Java プログラムをコンパイルしたフォーマット。

Java 仮想マシン (Java virtual machine: JVM) を参照。

C

キャブ (**cab**)

単一のハイパーテキスト転送プロトコル (HTTP) 要求のアプレットを実行するのに必要なクラス、および他のファイルをロードするのに使用する圧縮ファイル。Microsoft Internet Explorer のバージョン 4.0 以降で実行しているアプレットでのみ使用できます。

アプレット (applet)、Hypertext Transfer Protocol (HTTP)、および JAR ファイル (JAR file) を参照。

コールバック (**callback**)

サービスによって呼び出すことのできるクライアントで定義されたメソッド。コールバックにより、クライアントとサービス間の非同期の双方向交換をサポートできます。たとえば、サービスが時間のかかる処理

を実行するときに、クライアントの要求を単純な戻り値ですぐに確認し、コールバックを使用してあとで処理の完全な結果を戻すことができます。コールバックは、対話に参加している必要があります。

コールバック ハンドラ (**callback handler**)

対応するコールバックを受信するときにサービスによって実行されるメソッド。コールバックを含むコントロールによって定義されます。たとえば、タイマーコントロールは、**onTimeout** メソッドをコールバックハンドラとして使用できます。タイマーが起動するときに実行するコードを追加するオプションもあります。

呼び出し側のプリンシパル (**caller principal**)

メソッドの呼び出し中にアプリケーション コンポーネント インスタンスに関連付けられているプリンシパル。たとえば、**EJB** インスタンスは **getCallerPrincipal** メソッドを呼び出して、現在のセキュリティ コンテキストに関連するプリンシパルを取得できます。

プリンシパル (**principal**) およびエンタープライズ **JavaBeans (Enterprise JavaBeans: EJB)** を参照。

キャンペーン (**campaign**)

ビジネスの戦略目標をともに達成するシナリオの名前付きのグループ。キャンペーンは、複数の **Personalization**、**Commerce**、および **Campaign** サービスを調整して、電子商取引の **Web** サイトにおけるマーケティング目標を作成、追跡できます。

シナリオ (**scenario**) を参照。

キャンペーン ディスカウント (**campaign discount**)

一部のユーザのみを対象にすることを条件として、製品、注品、注文に関連する出荷の価格を値下げすること。**BEA E-Business Control Center** では、割引率を定義してから、後でオンライン顧客に対する価格を決定することもできます。

E-Business Control Center™ を参照。

キャンペーン終了条件 (**campaign end criteria**)

キャンペーンの目標。目標に到達した場合は、それが予定された終了日より前でも、キャンペーンを終了することがあります。この目標は、**BEA E-Business Control Center** で指定されます。

E-Business Control Center™ を参照。

Campaign サービス (Campaign services)

ビジネスアナリストとビジネスエンジニアが電子商取引の Web サイトにおけるマーケティング目標を作成、および追跡できるように WebLogic Portal が提供するサービスのセット。E-Business Control Center を使用してキャンペーンを設定したり管理したりできます。

ビジネスアナリスト (Business Analyst: BA)、ビジネスエンジニア (Business Engineer: BE)、および E-Business Control Center™ を参照。

キャンペーン スポンサー (campaign sponsor)

キャンペーンを委託する組織。多くの場合には、その組織 (または組織の一部) が e ビジネスの Web サイトを所有、運営しています。

BEA E-Business Control Center では、スポンサは、キャンペーンを検索、報告および解析するための基準として使用されます。

BEA E-Business Control Center を使用してキャンペーン スポンサーを指定できます。

E-Business Control Center™ を参照。

キャンペーン ステート (campaign state)

キャンペーンのステート。アクティブまたは非アクティブのいずれかになります。

キャンペーンを保存すると、アクティブステートになります。この場合、キャンペーンが指定した開始時刻と終了時刻の間で、適切な条件を満たすと実行できます。

実行中のアクティブなキャンペーンを無効にしたり、指定した終了日や目標に達する前に実行しているキャンペーンを中止すると、そのキャンペーンは非アクティブなステートとなります。E-Business Control Center のキャンペーンリスト (Control Center の [Explorer] ウィンドウに表示されます) では非アクティブ化されたキャンペーンは、リストの他のアイテムよりも薄い色で表示されます。

E-Business Control Center™ を参照。

Catalog Manager サービス (Catalog Manager service)

複数の製品カタログ サービスにアクセスする単一ポイント。Category Manager サービス、Product Item Manager サービス、Custom Data Manager サービス、および Catalog Query Manager サービスがこれに含まれます。

サービス (service) および製品カタログ (product catalog) を参照。

Catalog Query Manager サービス (Catalog Query Manager service)

Commerce Services 製品カタログをクエリするサービス。キーワード検索と属性に基づく検索の 2 種類の方法でカタログを検索できます。キーワード検索は、複数のキーワードを使用した単純な検索で、属性に基づく検索の場合、アイテム属性の複雑なブール式を使用して値を求めることができます。

サービス (service) および製品カタログ (product catalog) を参照。

Category Manager サービス (Category Manager service)

Commerce Services 製品カタログの階層構造を管理するサービス。階層の作成および変更、またアイテムをカテゴリにマッピングするための複雑なインタフェースを定義します。

サービス (service) および製品カタログ (product catalog) を参照。

CD-ROM

コンパクト ディスク読み込み専用メモリ (CD-ROM) を参照。

証明書 (certificate)

特定の公開鍵に名前や他の属性を関連付けるデジタル文。認証局 (certificate authority: CA) によりデジタルで署名されます。本物の文のみを署名する認証局を信頼することにより、その証明書に名前が含まれるユーザが持つ公開鍵を信頼できます。

証明書認証 (certificate authentication: CA) および認証局 (certificate authority: CA) を参照。

証明書認証 (certificate authentication: CA)

デジタル証明書を使用することにより、サーバがクライアントの ID を信頼するためのメソッド。通常、証明書認証は、パスワードによる認証よりも信頼性が高いと見なされます。これは、証明書認証がユーザの所有物 (プライベート キー) とユーザが持っている情報 (プライベート キーを保護するパスワード) に基づいているためです。

認証局 (certificate authority: CA)、証明書 (certificate)、および認証 (authentication) を参照。

証明書認証プロバイダ (certificate authenticator)

デジタル証明書 (digital certificate) を WebLogic Server ユーザにマップするための WebLogic Server で定義された標準インタフェース。受信した証明書を検証した後、証明書認証プロバイダは、証明書に基づいて、その証明書に関連して送信されるトレーディング パートナを、対応する WebLogic Server ユーザにマップします。

認証局 (certificate authority: CA)

公開鍵証明書を発行する既知の信頼できるエンティティ。認証局は、公証人と同様に、ユーザの実際の ID を証明するものです。

証明書認証 (certificate authentication: CA) および証明書 (certificate) を参照。

認証局 (certification authority)

VeriSign などの信頼できる既知の組織。証明書所有者の ID を別のパーティに対して認証するためにデジタル証明書 (digital certificate) を発行します。

チェーニング (chaining)

Webflow をあるプロセッサ ノード (入力プロセッサまたはパイプライン) から別のプロセッサ ノードに移動するプロセス。このメカニズムを使用すると、プレゼンテーション ノード間でプロセッサ ノードを使用する必要がなくなります。チェーニングでは、正しく実行できたプロセッサ ノードの結果のステートを別のプロセッサ ノードに送ります。

ノード (node)、パイプライン (pipeline)、Webflow、および入力プロセッサ (input processor) を参照。

チェックアウト (check out)

顧客が購入するアイテムを確認すると、支払いと出荷に関する情報を提示する対話型プロセス。その後、**Commerce Services** は顧客情報を検証し、クレジットカードのトランザクションを行って、出荷と税金支払いの必要条件をログに記録します。チェックアウトプロセスにより、**Registering Users** および **Managing Customer** サービスと、**Managing Purchases** および **Processing Orders** サービスが実行されます。

顧客の登録および顧客サービスの管理 (**Registering Customers and Managing Customer services**) および購入の管理と注文サービスの処理 (**Managing Purchases and Processing Orders services**) を参照。

CICS

顧客情報制御システム (**Customer Information Control System: CICS**) を参照。

暗号 (cipher)

暗号法では、暗号化されたメッセージを作成するのに使用する符号化システム。

暗号スイート (cipher suite)、暗号文 (cipher text)、セキュアソケットレイヤ (secure sockets layer: SSL)、および暗号 (cipher) を参照。

暗号スイート (cipher suite)

セキュアソケットレイヤ (secure sockets layer: SSL) の暗号化メソッド。次の3種類のアルゴリズムを使用して、通信の完全性を保護できます。それらは、交換アルゴリズム、対称暗号化アルゴリズム、セキュアハッシュアルゴリズムです。

セキュアソケットレイヤ (secure sockets layer: SSL) および暗号 (cipher) を参照。

暗号文 (cipher text)

暗号法では、暗号化されるテキスト。

クラス (class)

オブジェクト指向のプログラミングで使用されるオブジェクトのカテゴリ。クラスは、特定の種類のオブジェクトの実装を定義します。クラス定義は、インスタンスとクラスの変数とメソッドを定義し、インタ

フェースとクラス実装、およびクラスの直近のスーパークラスを指定します。スーパークラスが明示的に指定されていない場合、**Object**であると想定されます。

クラスライブラリ (class library) およびオブジェクト指向プログラミング (object-oriented programming: OOP) を参照。

クラスライブラリ (class library)

クラスと呼ばれるクライアントプログラミングツールのセット。これらのツールは **Java** または **C++** プログラム、または **Web** ページに埋め込み可能な **Java** アプレットで使用できます。

クラス (class) を参照。

CLASSPATH

実行時に必要な実行可能なクラスファイルを選択するために **Java** 仮想マシン (**JVM**) で検索するファイルシステムディレクトリまたは **Java** アーカイブファイルのパスのリスト。このリストは、オペレーティングシステムの環境変数 (**CLASSPATH**) か仮想マシンに送信されるコメントラインスイッチ (**-classpath**) により提供されます。サーブレットエンジンや **EJB** コンテナなどのアプリケーションサーバコンテナには、クラスパス情報の追加レベルを含めることもできます。

クラスライブラリ (class library) およびクラス (class) を参照。

クリックスルー (clickthrough)

広告をクリックする行為。クリックスルーの回数を 1 つの評価基準として使用して、キャンペーンの目標に到達できたかどうかを判断します。

クライアント (client)

1) 次の手順を実行するプログラム。

- ユーザインタフェースを使用してユーザからサービスの要求を収集します。
- これらの要求をサーバに送信します。
- サーバの応答を受け取り、ユーザに渡します。

クライアントが、ターゲットサーバと同じドメインに属しているマシン上にある場合には、そのクライアントは **ネイティブクライアント** と呼ばれます。クライアントがドメイン外のマシンにある場合、そのク

クライアントはリモート クライアントまたはワークステーション クライアントと呼ばれます。

2) (CORBA) 分散型オブジェクトにおいてオペレーションを実行するコード。

クライアント、Web サービス (client, Web service)

要求を Web サービスに送り、データを戻すエンティティ。Web サービスで通信可能な方法であれば、あらゆる言語で書くことができ、あらゆるプラットフォームで実行できます。ほとんどの Web サービスは、HTTP などのインターネット プロトコルで要求を受け取ることができ、それらの要求を SOAP 仕様に基づいて書式設定された XML メッセージにすることができます。

クライアント コネクタ インタフェース (client connector interface: CCI)

J2EE ベースのアプリケーションでバックエンドのシステムに接続し、それらにアクセスできるインタフェース。CCI は、クライアント アプリケーションとバックエンド システム間のデータ フローを管理します。コンテナまたはアプリケーション サーバのアクティビティはアダプタでは表示されません。

クライアント / ネットワーク (client/network)

異機種クライアントからの、ネットワーク上の多くの異なるサービスに対する要求が、高性能でインテリジェントな中間サーバやサーバのクラスタ (pure-Java WebLogic Server など) によって透過的に実現されるネットワーク。

Web サービス (Web service) およびクライアント / サーバ (client/server) を参照。

クライアント / サーバ (client/server)

コンピュータの処理がクライアント (デスクトップ PC) とサーバ (セントラル コンピュータ) において分散されているネットワーク アーキテクチャ。

クライアント / ネットワーク (client/network) を参照。

CLOB

文字ラージ オブジェクト。

クラスタ (cluster)

単一サーバよりもよりパワフルで信頼性の高いアプリケーションプラットフォームを提供するために、複数のサーバが集まって形成しているグループ。クラスタは、単一サーバとしてクライアントに表示されますが、実際は1つのサーバとして動作しているサーバのグループです。適切に設計され設定されているクラスタは、可用性と拡張性の両方を備えています。クラスタに新しいプロセスとマシンを動的に追加することで、クラスタをシャットダウンせずにさらに大容量のロードを処理することができます。個々のサーバは、定期的にクラスタから削除されるので、クラスタのパフォーマンスに影響を与えることなく保持できます。サーバ (server) を参照。

クラスタ化可能なスタブ (clusterable stub)

クラスタ化可能なスタブは、呼び出しをサービス インスタンスに委託するラッパーです。サービスは、別のサーバに対しては透過的に処理されるので、ロード バランシングとフェイルオーバーを高度に処理することができます。RMIC の `-clusterable` のフラグ、または EJB デプロイメント記述子の同様のプロパティを設定することによりクラスタ化可能なスタブを生成します。

ロード バランシング (load balancing)、スタブ (stub)、およびフェイルオーバー (failover) を参照。

COBOL Copybook Importer

COBOL Copybook を読み込み、COBOL Copybook のデータ構造を反映するメッセージフォーマットを生成するエンティティ。

コード ページ (code page)

このドキュメントでは、フィールド データをコード化する文字。

CODE タグ (CODE tag)

- 1) アプレットに対するクラスの完全パッケージ名。
 - 2) HTML アプレット タグの属性の1つです。
- パッケージ名 (package name) およびアプレット (applet) を参照。

CODEBASE タグ (CODEBASE tag)

サーバホストでのアプレットのクラスを検索するベース URL。アプレットの完全 URL を取得するには、CODEBASE をアプレットの CODE 名とともに使用します。

CODE タグ (CODE tag)、Uniform Resource Locator (URL)、およびアプレット (applet) を参照。

コラボレーションアグリーメント (collaboration agreement)

トレーディング パートナが実行に同意した対話の定義と、これらの対話の実行方法を定めた仕様。この仕様には、転送、メッセージング、セキュリティ制約、プロセス仕様へのバインディングに関する詳細が含まれます。

協調的ワークフロー (collaborative workflow)

WebLogic Integration の B2B Integration コンポーネントとともにビジネスを遂行する間のロールを実装するトレーディング パートナで使用するワークフロー。

CollaboratorMBean

実行時環境でトレーディング パートナを表す MBean インタフェース。WebLogic Integration 管理アプリケーションは CollaboratorMBean オブジェクトを使用してトレーディング パートナをモニタします。

COM

Component Object Model (COM) を参照。

COM ビュー (COM view)

コンポーネント オブジェクト モデル (Component Object Model: COM) 標準に準拠するようなオブジェクト表現。必要なすべてのインタフェース実装を含みます。

Component Object Model (COM) を参照。

Commerce サービス (Commerce services)

製品の説明を表示し、顧客からの注文を管理する Web アプリケーションのための開発キット。JSP テンプレートのセット、データベーススキーマ、Java コンポーネント、コンフィグレーションファイルを含みます。

Web アプリケーション (Web application: Webapp) および JSP テンプレート (JSP template) を参照。

コミット (commit)

- 1) 変更が記録され、安定するようにトランザクションを完了させること。保護されていたリソースは開放されます。
- 2) トランザクションを更新し、メッセージを他のトランザクションに表示する宣言またはプロセス。トランザクションをコミットすると、すべての結果は公開され、永続的なものとなります。コミットメント後は、トランザクションの結果を自動的に戻すことはできません。
トランザクション (transaction) を参照。

共通クライアント インタフェース (Common Client Interface: CCI)

アプリケーション コンポーネントの標準的なクライアント API を定義し、共通クライアント API を使用してアプリケーション コンポーネントとエンタープライズ アプリケーション統合 (Enterprise Application Integration: EAI) フレームワークで EIS を駆動させます。J2EE コネクタアーキテクチャは EIS アクセスの CCI を定義します。

エンタープライズ情報システム (Enterprise Information System: EIS)、J2EE コネクタ アーキテクチャ (J2EE Connector Architecture)、およびアプリケーションプログラミング インタフェース (application programming interface: API) を参照。

共通オブジェクト要求ブローカ アーキテクチャ (common object request broker architecture: CORBA)

分散オブジェクト間で通信するための、オブジェクト管理グループ (Object Management Group: OMG) の標準。CORBA は IDL (インタフェース定義言語) と ORB (オブジェクト要求ブローカ) を使用して、IIOP (インターネット ORB 間プロトコル) と通信します。

Internet Interoperability Protocol (IIOP)、Object Request Broker (ORB)、およびインタフェース定義言語 (interface definition language: IDL) を参照。

コンパクト ディスク読み込み専用メモリ (CD-ROM)

レーザーによって光学的に読み込まれるデータおよび特殊な条件下以外では修正できないデータのディスク。

コンポーネント (component)

アプリケーションの一部。

Component Object Model (COM)

ネットワーク環境でソフトウェア コンポーネントを連携させるためのサービスの集合。

COM ビュー (COM view) を参照。

条件 (condition)

割引を提供する場合または電子メールを送信する場合に、広告が表示される条件を決定する評価基準。BEA E-Business Control Center では、選択した条件を記述するテキストが [New Action] ウィンドウのアクション領域に表示されます。

E-Business Control Center™ を参照。

コンフィグレーション可能なエンティティ (configurable entity)

- 1) 実行時に設定できる任意のプロパティを持つオブジェクトを記述するインタフェース。
- 2) このインタフェースのベース エンティティ EJB 実装の総称。

コンフィグレーション (configuration)

コンピュータまたはネットワーク上のハードウェア、ハードウェア オプション、ソフトウェア、およびソフトウェア設定の集合。

コンフィグレーションする (configure) およびコンフィグレーション セット (configuration set) を参照。

コンフィグレーション セット (configuration set)

コンフィグレーション区分において特定のコンフィグレーションを参照するために使われる名前または番号。各コンフィグレーション セットは、コンフィグレーションがアクティブなときに使用されるサービスを定義します。

コンフィグレーションする (configure) およびコンフィグレーション (configuration) を参照。

コンフィグレーションする (configure)

コンピュータまたはネットワークの用途に合わせてハードウェアまたはソフトウェアをカスタマイズすること。

コンフィグレーションセット (configuration set) およびコンフィグレーション (configuration) を参照。

Confirmed_Delivery_to_Destination

すべての受信側トレーディング パートナによるメッセージの受信を検証するサービス品質 (Quality of Service: QoS) オプション。再試行 (Retry Attempts) および持続性 (Durability) などのオプション (永続性 (Persistent) オプション) を使用すると、配信の信頼性をより高めることができます。

Confirmed_Delivery_to_Destination は、送信側および管理者に追加情報も提供します。これには、受信側の配信ステータス、管理者用の完全なメッセージトラッキング (message tracking)、**WebLogic Integration の B2B Integration** コンポーネントからの配信完了時の確認が含まれます。このオプションを使用できるかどうかは、使用されているビジネスプロトコル (business protocol) により決まります。

Confirmed_Delivery_to_Hub

WebLogic Integration の B2B Integration コンポーネントによるメッセージの受信を検証するデフォルトのサービス品質 (Quality of Service: QoS) オプション。このオプションは、多くの場合、システム パフォーマンスが問題となるときに使用されます。このオプションでは、受信側の配信ステータスは提供されず、メッセージトラッキング (message tracking) も行われません。トレーディング パートナの選択プロセスは送信側に確認が到達した後で行われるため、潜在的な受信側の情報は使用できません。このオプションを使用できるかどうかは、使用されているビジネスプロトコル (business protocol) により決まります。

接続 (connection)

リソース マネージャへの接続性を提供し、アプリケーション クライアントによるリソース マネージャへの接続、トランザクションの実行、リソース マネージャによって提供されるサービスへのアクセスを可能にすること。接続にはトランザクション対応とトランザクション非対応の 2 種類があります。接続の例には、データベース接続や SAP R/3 接続があります。

リソース マネージャ (resource manager) を参照。

接続ポート (connection port)

Webflow では、ノードの端の小さなグラフィック機器。イベントや例外がそのノードに接続したポイントをマークします。これは、ノードの接続ポートを移動するのに役立つことがあります。

Webflow およびノード (node) を参照。

コンストラクタ (constructor)

WebLogic Server 環境の場合 — オブジェクトを作成する疑似メソッド。Java におけるコンストラクタとは、インスタンスのメソッドのうちそのクラスと同じ名前を持つものです。Java のコンストラクタは `new` キーワードを使用して呼び出されます。

インスタンス (instance)、Java、メタデータ インタフェース (metadata interface)、オブジェクト (object)、およびクラス (class) を参照。

コンテナ (container)

WebLogic Server などのアプリケーション サーバの一部。アプリケーション コンポーネントをデプロイし、実行時サポートを提供します。コンテナを使用すると、サポートされているコンポーネントをモニタおよび管理するサービスだけでなく、それらのコンポーネントもモニタおよび管理できます。コンテナは以下のいずれかです。

- リソースアダプタのホストとなるコネクタ コンテナ
- JSP、サーブレット、および静的 HTML ページのホストとなる Web コンテナ
- EJB コンポーネントのホストとなる EJB コンテナ
- スタンドアロンアプリケーション クライアントのホストとなるアプリケーション クライアント コンテナ

Java Server Pages (JSP)、サーブレット (servlet)、およびエンタープライズ JavaBeans (Enterprise JavaBeans: EJB) を参照。

コンテナ管理によるトランザクション (**container-managed transaction**)

エンタープライズ **JavaBean** コンテナがトランザクション境界を制御するトランザクション。コンテナ管理によるトランザクションでは、制御はデプロイメント記述子で指定します。**Bean** メソッドが呼び出されると、コンテナは、同じメソッドで始まり同じメソッドで終了するトランザクションを管理します。エンティティ **Bean** はコンテナ管理によるトランザクションを使用する必要があります。

トランザクション (**transaction**) および **Bean** 管理のトランザクション (**bean-managed transaction**) を参照。

コンテンツ管理システム (**content management system**)

コンテンツのメタデータなど、コンテンツ オブジェクト (アークティクル、ドキュメント、画像など) の集合を管理するシステム。**Interwoven** や **Documentum** などが販売しているコンテンツ管理システムは、通常、コンテンツの検索などのサービスおよびコンテンツ オーサリング サービスを提供します。コンテンツ オーサリング サービスには、コンテンツ オブジェクトの作成、編集、バージョン管理のサービスとワークフロー サービスが含まれます。

メタデータ (**metadata**) を参照。

コンテンツ マネージャ (**Content Manager**)

JSP タグと **EJB** の両方を介してコンテンツにアクセスできる、主要な **WebLogic Personalization Server** コンポーネント。**JSP** タグにより、開発者は検索式構文を使用して直接コンテンツ データベースをクエリし、それにより、コンテンツ オブジェクトを使用できます。コンテンツ管理ツールは、ルール マネージャとルール サービス、ユーザ プロファイル、プロパティ設定マネージャとともに使用して、電子商取引のサイトでパーソナライゼーションされたコンテンツ配信を作成できるように設計されています。コンテンツ管理コンポーネントの機能へのプライマリ インタフェースは **ContentManager** セッション **Bean** です。

JSP タグ (**JSP tag**)、プロパティ セット管理 (**property set management**)、ユーザ プロファイル (**user profile**)、およびエンタープライズ **JavaBeans** (**Enterprise JavaBeans: EJB**) を参照。

コンテンツ選択 (content selector)

条件セットの下で、ドキュメントのコンテンツ管理システムをクエリするメカニズム。BEA E-Business Control Center では、ビジネスアナリストが、クエリを実行するコンテンツ選択をトリガする条件を指定します。また、BEA E-Business Control Center を使用して、コンテンツ選択を実行するクエリも作成します。ビジネスエンジニアは、ビジネスアナリストがドキュメントを表示したい場所にコンテンツ選択の JSP タグを作成します。

ビジネスアナリスト (Business Analyst: BA)、ビジネスエンジニア (Business Engineer: BE)、および E-Business Control CenterTM を参照。

制御 (control)

Web サービスで使用され、サービスが他の種類のアプリケーションやコンポーネントと通信できるようにするコンポーネント。たとえば、データベース制御により、Web サービスはデータベースのデータを要求できます。サービス制御を使用すると、Web サービスが他の Web サービスをより簡単に呼び出すことができます。

会話 (conversation)

- 1) WebLogic Integration 環境の場合 — WebLogic Integration 環境の場合 — WebLogic Integration 環境の場合 — トレーディング パートナ間の一連のメッセージ交換。会話は、コラボレーション アグリーメント (collaboration agreement) で定義されます。メッセージ交換の構成は、トレーディング パートナ間のコラボレーション アグリーメントの一部である会話定義 (conversation definition) によって決まります。
- 2) WebLogic Workshop 環境の場合 — WebLogic Workshop 環境の場合 — クライアントと Web サービスの間で発生する一連の対話。Web サービスの会話の単一のインスタンスには関連する状態があり、会話の有効期間中はその状態が持続します。この状態には、少なくとも会話 ID が含まれ、これを使用してメッセージの入力やサービスの中止を特定の会話インスタンスと関連付けます。たとえば、1,000 人に対する 1,000 の信用調査を処理する場合、WebLogic Workshop では自動的にそれぞれに固有の会話をオープンにし、送受信するメッセージを適切な会話と関連付けます。

会話コーディネータ (**conversation coordinator**)

対応するビジネスプロトコル (business protocol) のルールに基づいて会話ライフサイクル (conversation life cycle) を管理し、複数の組織的な境界にまたがり長時間にわたる持続性のある会話 (conversation) を管理するサービス。会話コーディネータは、会話 ID を生成し、トレーディング パートナを会話に登録し、ビジネスおよびシステム メッセージをクライアントに配信し、会話に関するステータス情報を維持し、ビジネスプロトコル (business protocol) の実行のために会話コンテキストを提供します。

会話定義 (**conversation definition**)

会話 (conversation) のパラメータを指定する値の集合。

会話開始者 (**conversation initiator**)

会話 (conversation) を開始および終了するトレーディング パートナ。

会話ライフサイクル (**conversation life cycle**)

トレーディング パートナ間でメッセージが交換される期間。

会話参加者 (**conversation participant**)

ビジネス メッセージ (business message) を受信および処理するトレーディング パートナ。

会話終了 (**conversation termination**)

WebLogic Integration がサポートしている B2B Integration または Business Process Management (BPM) 機能の一部として実行されるメッセージ交換のインテリジェントで制御された終了。

- BPM クライアントの場合、会話 (conversation) は、完了ノードの会話終了プロパティ (成否) に定義されているように、会話開始者 (conversation initiator) が完了 (done) 状態に達したときに終了します。会話参加者 (conversation participant) は、会話に対する自身の参加を終了できますが、会話を終了できるのは会話開始者ワークフローだけです。
- B2B Integration クライアントの場合、会話終了は、会話開始者によって実行される `Conversation.terminate()` 処理によって行わ

れます。この処理の結果、各会話参加者は
`ConversationHandler.onTerminate()` メッセージを受信します。

ConversationHandler

トレーディング パートナが会話 (conversation) のビジネス メッセージ (business message) を送受信するためのインタフェース。関連するトレーディング パートナが会話に参加するには、会話ハンドラを、1 つまたは複数の会話タイプに対して登録する必要があります。

ConversationMBean

会話 (conversation) を表す MBean インタフェース。WebLogic Integration 管理アプリケーションは、**ConversationMBean** オブジェクトを使用して実行時環境の会話をモニタします。

コピーブック (copybook)

多くのソースプログラムにコピーできるよう設計されたソースコードの共通部分。主に **IBM DOS** メインフレーム プログラミングで使用されます。メインフレーム **DOS (DOS/VS** や **DOS/VSE** など) では、コピーブックはソース ライブラリにブックとして格納されました。ライブラリは複数のブックで構成され、各ブックの名前の先頭にはプログラミング言語を表す接頭辞 (たとえば、アセンブラの場合は **A.name**、**COBOL** の場合は **C.name** など) が付けられていました。これは、**DOS** が複数のライブラリまたはプライベート ライブラリをサポートしていなかったためです。この用語は主に **COBOL** プログラマによって使用されていますが、ほとんどのメインフレーム言語でサポートされています。**IBM OS** シリーズはコピーブックという用語は使用せず、代わりに、**区分データセット**、つまり **PDS** として実装されたライブラリなどのファイルを参照していました。コピーブックは、**C** または **C++ include** ファイルと機能的に同じものです。

CORBA

共通オブジェクト要求ブローカ アーキテクチャ (common object request broker architecture: **CORBA**) を参照。

相関 (correlation)

WebLogic Workshop により Web サービスに自動的に提供される機能。Web サービスに多くの同期クライアントが含まれている場合は、要求に対応してどの応答がどのクライアントに送られるのかを常に追跡する必要があります。その場合、次のコールバックが含まれます。コールバックを介してクライアントに送信しなければならない Web サービスでイベントが発生すると、WebLogic Workshop の相関機能により自動的にコールバック メッセージが配信されます。

相関 ID (Correlation ID)

メッセージ ID 文字列を含むサービス品質 (Quality of Service: QoS) オプション。これは、1 つのアプリケーション内の異なるビジネス メッセージ (business message) を相互に関連付けるのに使用できます。このオプションを使用できるかどうかは、使用されているビジネス プロトコル (business protocol) により決まります。

資格 (credential)

追加のサービスに対してプリンシパルを認証できるセキュリティ情報を格納または参照するデータ エンティティ。プリンシパルは認証に基づいて、あるいは自らの資格の使用を許可する別のプリンシパルから資格を取得します。後者はプリンシパルの委託と呼ばれます。資格の形態にはパスワード、X509 デジタル証明書または証明書チェーン、あるいはユーザを検証するその他のトークンなどがあります。

ユーザ (user) およびレルム (realm) を参照。

カスタム データ マネージャ サービス (Custom Data Manager service)

製品項目のカスタム属性に永続性を与えるインタフェースを定義するサービス。カスタム属性は、ProductItem インタフェースで定義されない属性です。

サービス (service) を参照。

カスタム セキュリティ プロバイダ (Custom Security providers)

セキュリティ ベンダまたはアプリケーション開発者によって開発されるセキュリティ コンポーネント。カスタム セキュリティ プロバイダは SSPI (Security Service Provider Interfaces) の実装です。

WebLogic Server セキュリティプロバイダ (WebLogic Server security providers) および Security Service Provider Interfaces (SSPI) を参照。

顧客情報制御システム (**Customer Information Control System: CICS**)

IBM がライセンスするプログラム。リモートの端末で入力されたトランザクションを、ユーザが記述したアプリケーションプログラムによって同時に処理できるようにします。

顧客プロフィール (**customer profile**)

顧客の Web サイトへのアクセスに伴って (または他のオフラインソースから) 収集、保存、および更新される一連の情報。顧客プロフィールは各顧客に固有であり、製品の好みから、入力フォームに顧客が入力した統計データに至るまでのさまざまな情報が含まれます。顧客プロフィールに保存された各情報 (またはそれらの情報を組み合わせたもの) を使用して顧客セグメントを作成できます。

顧客セグメント (customer segment) を参照。

顧客プロフィールサービス (**Customer Profile service**)

顧客がユーザプロフィールを変更することによる、ビジネスロジックとプレゼンテーションロジックの実装。

サービス (service)、ユーザプロフィール (user profile)、およびビジネスロジック (business logic) を参照。

顧客セグメント (**customer segment**)

製品の好みや年齢など顧客プロフィールから収集した共通の特性に基づいて、顧客をグループに分類すること。

顧客プロフィール (customer profile) を参照。

D

デーモン (**daemon**)

バックグラウンドで自動的かつ連続的に動作するプログラム。

DASD

直接アクセス記憶装置 (Direct Access Storage Device: DASD) を参照。

Data Integration

XML ファイルと非 XML ファイルのフォーマットを変換する WebLogic Integration コンポーネント。

Data Integration プラグイン (data integration plug-in)

Data Integration 機能とともに使用できるように Business Process Management (BPM) 機能を拡張する WebLogic Integration Studio コンポーネント。

データプライバシー (data privacy)

トレーディング パートナ間の通信のメッセージをリンク レベルで暗号化する WebLogic Integration の SSL ベースのセキュリティ機能。ID に名前とパスワードを使用している場合は、ユーザにはデータ プライバシ (secure sockets layer: SSL) がありません。データ プライバシを使用するには、デジタル証明書 (digital certificate) が必要です。

データトランスフォーメーション (data transformation)

WebLogic Integration 環境の場合 — XML フォーマット間で提供されるデータのマッピング。たとえば、RosettaNet ドキュメントを ebXML ドキュメントにマップするときに、データトランスフォーメーションが実行されます。

データ変換 (data translation)

WebLogic Integration 環境の場合 — バイナリ データと XML を変換するプロセス。

データユニバーサルナンバリング システム (data universal numbering system: DUNS)

グローバルな電子商取引に携わる企業に、国際的に認知された識別子として 9 桁の数字を割り当てるための仕様。ビジネス識別子は、外部利用のために複数のビジネスプロトコルによってトレーディング パートナに割り当てられます。これらは、内部の WebLogic Integration トレーディング パートナ名とは異なります。RosettaNet ビジネスプロトコルの実装では、DUNS 番号を使用しなければなりません。

データベース (database)

互いに関係する、それぞれ独立したデータ項目の集合体。1 つ以上のアプリケーションから利用できるように冗長性を伴わずにまとめて格納されています。

データベース管理システム (database management system: DBMS)

ユーザがデータベース テーブル内のデータを構造化、操作できるようにするプログラムの集合。DBMS は、多数のユーザがいる環境でのプライバシー、回復処理、およびデータの整合性を保証します。

データベース (database) を参照。

データ型 (datatype)

変数内のデータの型を示すプログラミング上の分類。一般的に使用されるデータ型には、整数、文字、ブールなどのさまざまな形式があります。ほとんどのプログラミング言語では、データ オブジェクトまたはフィールドのデータ型を宣言する必要があります。データ型は使用するプログラミング言語によって異なります。

dbKona

WebLogic Server 環境の場合 — どのような JDBC ドライバとも併用できる、データベース アクセス用の高レベルな Java オブジェクト群。

JDBC を参照。

DBMS

データベース管理システム (database management system: DBMS) を参照。

決定 (decision)

遷移が発生するときに評価される条件を含むワークフロー コンポーネントの Business Process Management (BPM) 用語。評価結果 (真または偽) により、フロー制御が渡される適切な依存ノードが決まります。

ワークフロー (workflow) を参照。

デコーダ (decoder)

プロトコル固有のメッセージヘッダー (message header) の処理、送信トレーディング パートナの識別、会話 (conversation) への送信トレーディング パートナの追加、送信者への返信の準備、およびスケジューリング サービスへのメッセージの転送を行うコンポーネント。

復号化 (decryption)

暗号テキスト (暗号化されたデータ) と暗号鍵を元にして、プレーン テキスト (暗号化される前の元データ) を生成するプロセス。

暗号化 (encryption) を参照。

デフォルト (default)

ユーザが値を指定しない場合にプログラムが使用する変数の値。

遅延同期メッセージ配信 (deferred synchronous message delivery)

メッセージが送られるとアプリケーションに制御が戻されるメッセージ配信メソッド。メッセージトークン (message token) がアプリケーションに戻され、アプリケーションは、後でそれにアクセスして、メッセージ配信のステータスをチェックできます。トークンにアクセスした後、アプリケーションは指定した時間、または後続のイベントが発生するまで待機し、すべての可能性のある送り先から、メッセージのタイムアウト、またはメッセージが送信された会話 (conversation) の終了を知らせる通知を受け取ります。

委託 (delegation)

プリンシパル A がプリンシパル B に、多くの場合制限付きで自らの身元または権限の使用を認めること。

区切り記号 (delimiter)

フィールドまたはデータ グループの末尾を示すバイト シーケンス。

配信チャネル (delivery channel)

ビジネス メッセージ (business message) を 1 つのトレーディング パートナに配信するための仕様。各トレーディング パートナには、それぞれがサポートするビジネスプロトコル (business protocol) ごとに配信チャネルがあります。

サービス拒否 (DoS) 攻撃 (Denial of Service (DoS) attack)

セキュリティ攻撃の一種。ユーザまたは組織が通常は利用できるはずのリソースのサービスが利用できなくなります。たとえば、企業の Web サイトが攻撃によって運用停止に追い込まれ、復旧に要する時間や収益の面で多大な損失を被ることがあります。

デプロイメント (deployment)

- 1) アプリケーションを分散型環境に配置し、使用可能な状態にするプロセス。このプロセスには、インストール、コンフィグレーション、アプリケーションの各部位の管理などの作業が伴うことがあります。
- 2) WebLogic Workshop 環境の場合 — Web サービスを開発環境からプロダクション環境に移行させるプロセス。WebLogic Workshop の実行時ソフトウェアは、WebLogic Server に統合されているため、WebLogic Workshop Web サービスを簡単にデプロイできます。ターゲットの JWS ファイル (およびサポート ファイル) をプロダクションサーバにコピーするだけです。

デプロイメント記述子 (deployment descriptor)

EJB の実行時プロパティをコンフィグレーションするために必要な情報を EJB サーバに提供する XML ファイル。また、システム管理者は、EJB に対するパーミッションを定義したり、EJB が使用するリソースをコンフィグレーションしたりすることができます。デプロイメント記述子は、Sun Microsystems, Inc. JavaSoft の文書型定義 (Document Type Definition: DTD) に準拠しています。

エンタープライズ JavaBeans (Enterprise JavaBeans: EJB)、XML、および文書型定義 (Document Type Definition: DTD) を参照。

記述子 (descriptor)

Campaign Services Web アプリケーションのデプロイメント要件を指定する XML ファイル。記述子はプロパティと値のフレーズで構成されています。

広告プレースホルダに対し、新しい MIME タイプを追加するには、WebLogic Personalization Server Web アプリケーションのデプロイメント記述子を変更する必要があります。テキスト エディタを使用してデプロイメント記述子を変更できます。

開発者 (developer)

組織において、コードを記述することにより Web アプリケーションを作成、保守する技術専門家の役割を示す総称。Java 2 Platform Enterprise Edition (J2EE) 仕様では、開発者はアプリケーション コンポーネント プロバイダと呼ばれます。

開発者には、組織の他の役割を担う個人が共有できない特殊なアクセス権限が与えられます。

開発者は、その専門分野により、Java/EJB 開発者、HTML/JSP 開発者、アプリケーションのアセンブラ/デプロイヤ、システム管理者などに分類されます。

デジタル証明書 (digital certificate)

トレーディング パートナを認証するために公開鍵暗号システムとともに WebLogic Integration で使用されるデジタル ID カード。デジタル証明書には所有者の公開鍵が含まれています。これは認証局 (certification authority) によりデジタル署名がされています。

デジタル署名 (digital signature)

2つのエンティティ間で交換されるデータのセキュリティを、各エンティティの ID を検証することによって保護する際に使用されるビット文字列。特にこの文字列は、レコードの送信側から送られたデータと、転送中に変更されなかったデータの検証に使用されます。

デジタル署名は、エンティティの署名付きデータと秘密鍵から計算されます。デジタル署名は、確認に使用される公開鍵が信頼できる場合のみ信頼できます。

直接アクセス記憶装置 (Direct Access Storage Device: DASD)

1つまたは複数のディスク、またはメインフレーム マシンのディスクドライブのセット。

直接プログラミング モデル (direct programming model)

ルックアップ プロセスで使用されるモデル。これを使用するとレプリカ対応スタブが生成され、実行のたびに異なるサービス プロバイダによって処理される場合があります。直接モデルは、ステートレスなオブジェクトの場合に適しています。

ステートレス オブジェクト (stateless object)、レプリカ対応スタブ (replica-aware stub)、ターゲット プログラム モデル (targeted programming model)、および間接プログラミング モデル (indirect programming model) を参照。

割引優先順位 (discount priority)

割引の相対的な重要度を 1 ~ 20 の範囲の値で指定したもの (1 が最も優先順位が高くなります)。指定した項目に 2 つの類似した割引を適用するイベントでは、各割引に付けられた優先順位を使用して競合を解消します。

BEA E-Business Control Center は、複数の割引に同じ優先順位が割り当てられている状況を検出しません。2 つの割引の優先順位が同じであると、いずれかの割引が無作為に選択されます。

E-Business Control Center™ を参照。

割引値 (discount value)

割引の割合、またはドルで示した金額。すべての割引値は、割引額の割合、割引額、または固定金額の 3 つの形式のうち、いずれかの形式で指定されます。

表示優先順位 (display priority)

広告グループを表示する相対的な順序。複数の広告グループが同時に同じプレースホルダに表示される可能性のある場合の、他のグループの広告との相対順位です。デフォルトの広告グループは、キャンペーン広告グループよりも自動的に優先順位が低くなります。

BEA E-Business Control Center は、複数の広告に同じ優先順位が割り当てられている状況を検出しません。2 つの広告クエリが同じ優先順位の場合は、両方がプレースホルダで実行される確率は等しくなります。

E-Business Control Center™ を参照。

分散型アプリケーション (distributed application)

ネットワークを介して通信を行う別々のコンピュータに存在し、複数の部分に分かれているアプリケーション (クライアントとサーバなど)。

Web アプリケーションは本来、分散型アプリケーションです。

Web アプリケーション (Web application: Webapp) を参照。

分散コンピューティング (distributed computing)

異なるコンピュータで実行され、ネットワーク経由で通信する複数のユニットにアプリケーションを分離する、アプリケーション設計および実装の戦略。たとえば、アプリケーションをユーザ インタフェース ユニット、処理ユニット、記憶ユニットの3つの分散したユニットに分離できます。

分散型オブジェクト (distributed object)

ネットワーク上のどこにでも配置できるオブジェクト。分散型オブジェクトは、メソッド呼び出しを介してリモート クライアントからアクセス可能な、独立したコードの断片としてパッケージ化できます。分散型オブジェクトを作成するための言語とコンパイラは、クライアントから完全に透過的です。クライアントは分散型オブジェクトの位置や、オブジェクトを実行しているオペレーティング システムの種類を知っている必要はありません。

分散オブジェクト モデル (distributed object model: DOM)

アプリケーションの分割方法、つまり、ロジックがそれをサポートする層 (多層または2層) にどのように割り振られるかを記述する抽象概念。2層 (two-tier) および多層 (multitier) を参照。

分散型プログラム リンク (Distributed Program Link: DPL)

Java の遠隔メソッド呼び出しの概念とほぼ相当する CICS の概念。

顧客情報制御システム (Customer Information Control System: CICS) を参照。

分散トランザクション (distributed transaction)

複数のトランザクション マネージャが関連するトランザクション。分散トランザクション環境では、クライアントアプリケーションは複数のサーバに要求を送信し、その結果、複数のリソース マネージャでリソースが更新されます。トランザクションを完了するには、各参加コンポーネント (クライアント、サーバ、およびリソース マネージャ) のトランザクション マネージャをポーリングして、ドメイン内の各参加コンポーネントのコミットプロセスを調整する必要があります。

トランザクション マネージャ (transaction manager)、リソース マネージャ (resource manager)、およびトランザクション (transaction) を参照。

分散型トランザクション処理 (**distributed transaction processing: DTP**)

複数のアプリケーションプログラムが、複数のリソース (データベースなど) を均整のとれた方式で更新するような処理形態。プログラムおよびリソースは、ネットワーク上の 1 つまたは複数のコンピュータに配置できます。

DLL

ダイナミック リンク ライブラリ (dynamic link library: DLL) を参照。

DMS

ドキュメント管理システム (document management system: DMS) を参照。

ドキュメント (**document**)

コンテンツ管理システムに保存され、Web サイトに表示されるファイル。たとえば、HTML ファイル、画像ファイル、ブラウザ プラグインで表示するファイルなどがあります。

コンテンツ管理システム (content management system) を参照。

ドキュメント定義 (**document definition**)

有効なドキュメントの必要条件を指定する、文書型定義 (Document Type Definition: DTD) などのスキーマ。WebLogic Integration ドキュメント定義は XML DTD にあります。各ドキュメント定義には、システム ID (DTD システム識別子) と、ドキュメント定義の場所を指定する URL という 2 つの属性があります。

ドキュメント交換 (**document exchange**)

ドキュメントを交換するメソッドの定義。ドキュメント交換では、ビジネスプロトコル (business protocol) と複数の実行時パラメータを定義します。

ドキュメント管理システム (**document management system: DMS**)

ドキュメントの集合を管理するシステム。通常は、ドキュメント検索などのサービスと、作成、編集、バージョン管理、ワークフローなどのドキュメント オーサリング サービスを提供します。

ドキュメント参照実装 (document reference implementation)

WebLogic Personalization Server のデフォルトの優れたドキュメント管理ソリューション。この実装は、BulkLoader ユーティリティを使用して、コンテンツ メタデータをデータベースに、またコンテンツ スキーマを XML ファイルにロードします。これは、WebLogic Personalization Server ドキュメント管理サービス プロバイダ インタフェース (Service Provider Interface: SPI) の実装の 1 つです。これを使用してサードパーティ製のドキュメント管理システムを WebLogic Personalization Server に統合できます。

文書型定義 (Document Type Definition: DTD)

XML または SGML のいずれかで記述された関連するメッセージまたはファイル用にフォーマット (文法と構文) を定義するファイル。具体的には、DTD ファイルは、XML や SGML ドキュメントのマークアップ タグが、ドキュメントを扱うアプリケーションでどのように解釈されるかを定義し、それにより、ドキュメントが、意図したとおりに表示または印刷されます。この定義は、Standard Generalized Markup Language (SGML) のルールに従っています。DTD は W3C XML 仕様の一部です。スキーマ (schema) を参照。

DOM

分散オブジェクト モデル (distributed object model: DOM) を参照。

ドメイン (domain)

単一のコンフィグレーション ファイルで定義されたサーバ、サービス、インタフェース、マシン、および関連するリソース マネージャの集まり。リソース マネージャ (resource manager) を参照。

完了 (done)

ワークフロー全体を完全なものとしてマーキングする ワークフロー (workflow) コンポーネントの Business Process Management (BPM) 用語。この用語の発行後はタスク (task) の処理は行われません。

DPL

分散型プログラム リンク (Distributed Program Link: DPL) を参照。

DTD

文書型定義 (Document Type Definition: DTD) を参照。

DUNS

データユニバーサルナンバリングシステム (data universal numbering system: DUNS) を参照。

持続性 (Durability)

ノードに障害が発生したときに、持続性のあるメッセージストアを使用してメッセージ配信を保証するかどうかを指定する サービス品質 (Quality of Service: QoS) オプション。このオプションには、ネームスペース (namespace) (デフォルト) と永続性 (Persistent) があります。このオプションを使用できるかどうかは、使用されているビジネスプロトコル (business protocol) により決まります。

動的データ交換 (Dynamic Data Exchange: DDE)

アプリケーション同士が一連のメッセージを通じて情報を交換できる、Microsoft Windows プラットフォームで利用可能な通信形態。DDE メッセージの送信と応答を行う 2 つのアプリケーションのことを「DDE で会話している」といいます。

動的データ型 (dynamic datatype)

コードのコンパイル時点でそのメモリ サイズが不明なデータ型。動的データ型のメモリ サイズは、コードが実行される時点ではじめて判明します。

ダイナミック リンク ライブラリ (dynamic link library: DLL)

Windows アプリケーションから利用できる実行可能関数またはデータのライブラリ。Java プログラムで DLL を使用することは、その Java プログラムにネイティブ コードが含まれていることを示します。WebLogic Server ではデータベース ベンダによって公開された、2 層ネイティブドライバ用の DLL を使用しています。

動的ロール割り当て (dynamic role association)

ビジネス要求のコンテキストとターゲットに基づいて、プリンシパルにロールを事後バインドする能力。

E

E-Business Control Center™

パーソナライゼーションされた Web サイトの作成と保守を円滑に行うツール。BEA E-Business Control Center には、ユーザがさまざまなタスクを実行するための直観的なグラフィカルユーザ インタフェースがあります。

ebXML

electronic business XML (ebXML) を参照。

ECMAScript

JavaScript と Jscript を組み合わせた標準化された言語。JavaScript は Netscape Communications、Jscript は Microsoft Corporation によって開発されました。ECMAScript は European Computer Manufacturers Association (ECMA) と呼ばれる標準化機構により策定されました。

EDI

電子データ交換 (Electronic Data Interchange: EDI) を参照。

EDI VAN

電子データ交換付加価値ネットワーク (Electronic Data Interchange Value-Added Network: EDI VAN) を参照。

EIS

エンタープライズ情報システム (Enterprise Information System: EIS) を参照。

EJB

エンタープライズ JavaBeans (Enterprise JavaBeans: EJB) を参照。

electronic business XML (ebXML)

ビジネスデータの電子交換をサポートするモジュラーフレームワーク用の仕様。国際連合 (UN/CEFACT) と OASIS の共同構想により開発されたグローバルな使用方法です。

電子データ交換 (**Electronic Data Interchange: EDI**)

電子商取引で使用されるビジネス メッセージ (business message) の形式と、そのようなメッセージの使用を扱う法律用語の業界標準。

電子データ交換付加価値ネットワーク (**Electronic Data Interchange Value-Added Network: EDI VAN**)

トレーディング パートナ間で EDI メッセージをルーティングするサードパーティによる中継機能。

エンコーダ (**encoder**)

必要なビジネス プロトコル (business protocol) をサポートするために必要に応じてメッセージを変換し、そのメッセージを転送サービスに転送するコンポーネント。

コード化 (**encoding**)

アーキテクチャ固有のデータを、XDR (External Data Representation) コード化のように、異なった複数のアーキテクチャ間で送受信できる形式に変換すること。

暗号化 (**encryption**)

認可を受けたユーザによる元データへのアクセスは可能にしたまま、未認可の開示を防ぐ (または妨げる) 目的で、アルゴリズムを使用してデータにスクランブルをかけるプロセス。暗号化されたファイルを読み取るには、受信者が、その暗号を解くための秘密鍵かパスワードへのアクセス権を持っている必要があります。暗号化されていないデータをプレーン テキスト、暗号化されていないテキストを暗号テキストと呼びます。

暗号化鍵ペア (**encryption key pair**)

情報を暗号化するために使用される公開鍵と、情報を復号化するために使用される秘密鍵で構成される暗号鍵ペア。

エンド ポイント

トレーディング パートナの URL。

エンタープライズ アプリケーション統合 (**Enterprise Application Integration: EAI**)

複数のアプリケーションによる無制限の情報共有。組織内または組織間の情報の移動および交換を可能にする一連のテクノロジーです (『Linthicum 2000』、354 ページ)。

エンタープライズ情報システム (**Enterprise Information System: EIS**)

企業向けの情報インフラストラクチャを提供するソフトウェア システム。EIS は一連のサービスをクライアントに提供します。これらのサービスは、ローカル インタフェースおよびリモート インタフェース、またはそのいずれかを介してクライアントで使用できます。EIS には以下の例があります。

- SAP R/3 や PeopleSoft などのエンタープライズ リソース プランニング (**Enterprise Resource Planning: ERP**) システム
- CICS などのメインフレーム トランザクション処理システム
- Oracle などのレガシー データベース システム

エンタープライズ情報システム (EIS) リソース (**Enterprise Information System (EIS) resource**)

EIS 固有の機能をクライアントに提供すること。EIS リソースには以下の例があります。

- データベース システムのレコードまたはレコード セット
- エンタープライズ リソース プランニング (**Enterprise Resource Planning: ERP**) システムのビジネス オブジェクト
- トランザクション処理システムのトランザクションプログラム

エンタープライズ **JavaBeans (Enterprise JavaBeans: EJB)**

多層クライアント / サーバ システム向けのコンポーネント アーキテクチャを定義する Java API。具体的には、EJB は、オブジェクト指向の分散型エンタープライズレベルアプリケーションの開発とデプロイメントのためのアーキテクチャを指定します。EJB アーキテクチャを使用して作成されたアプリケーションは、拡張可能で、トランザクションに使用可能な安全なものです。

Java 2 Enterprise Edition (J2EE) を参照。

エンタープライズ リソース プランニング (enterprise resource planning: ERP)

計画、製造、販売、マーケティングなど、ビジネスのすべての局面を統合するビジネス管理システム。

エンティティ Bean (entity bean)

データベース レコードを表し、そのレコードのデータで呼び出せるメソッドを含むエンタープライズ JavaBean。

エンティティ プロパティ マネージャ (Entity Property Manager)

プロパティの永続性を処理するステートレスのセッション Bean。

環境変数 (environment variable)

アプリケーションのパラメータを表すアプリケーション定義の文字列。ユーザはそれにカスタム値を割り当て、そのパラメータに依存しているコードを記述しなおすことなく、必要に応じて後でその値を変更することができます。環境変数は、ホスト アプリケーションを起動すると利用できます。

ERP

エンタープライズ リソース プランニング (enterprise resource planning: ERP) を参照。

イベント (event)

- 1) WebLogic Integration 環境の場合 — ソフトウェア システム (通常は EIS) で発生する事象。外部の当事者があるイベントに関係している場合、イベント オブジェクトが生成され、その当事者にイベント通知 (event notification) として配信されます。
- 2) WebLogic Portal 環境の場合 — 顧客の動作またはシステムの動作。
 - 顧客の動作 — たとえば、BEA E-Business Control Center を使用すると、あらかじめ定義された多数の顧客の動作 (イベント) から動作を選択してシナリオのアクションをトリガできます。追跡される顧客の動作は、コンテンツの表示からハイパーリンクのクリックまで含まれます。
 - システムの動作 — イベント サービスは、キャンペーン アクティビティ、セッション イベント、ルールの停止など広範囲のイベン

トを追跡します。これらのイベントはログに記録され、分析ツールによってさらに深く分析することができます。

シナリオ アクション (scenario action) および E-Business Control Center™ を参照。

イベント ノード (event node)

ワークフロー (workflow) の Business Process Management (BPM) 用語。イベント ノードを実行するために、ワークフローは XML メッセージがイベントをトリガするのを待ちます。トリガが発生すると、イベント ノード内に定義されているさまざまなサブアクションが実行されるか、またはワークフロー変数が設定されます。

イベント通知 (event notification)

システム (通常 EIS) 内のイベントの一方向非同期通知。イベント通知は、イベントを表すイベント オブジェクトをそのタイプのイベントのサブスクライバに配信することによって達成されます。

イベント オブジェクト (event object)

システム (通常 EIS) 内のイベント (event) を表すデータ オブジェクト。イベント オブジェクトには通常、イベントとそれが発生するコンテキストを説明するのに十分なデータが含まれています。多くの場合、イベント オブジェクトは単にイベントと呼ばれます。これは、イベント オブジェクトが単に、イベントが渡された後のイベントを表し、イベントの永続的な結果であるからです。イベント オブジェクトのタイプとは、同じ原因を持つすべてのイベントを示す名前です。イベント オブジェクトには、そのイベントのデータが含まれる XML ドキュメントが含まれます。

例外 (exception)

プログラムまたはマイクロプロセッサが、エラーを処理するための別のルーチンに分岐するきっかけとなる状況 (多くの場合エラー)。実行時エラーを処理する Java メソッドです。

例外探索 (exception walking)

実行できない遷移を解決するために、Webflow が代替の実行フローを検索するプロセス。

Webflow を参照。

明示的なプロパティ (explicit properties)

別のテーブルや別のデータベースに保存されるプロパティで、カスタムのエンティティプロパティマネージャの実装によりアクセスできるプロパティ。

暗黙的なプロパティ (implicit properties) およびエンティティプロパティマネージャ (Entity Property Manager) を参照。

拡張プロパティ (extended property)

リポジトリ内のエンティティに関連付けることができるユーザ定義の要素、属性、またはテキスト。このリリースでは、拡張プロパティは、*WebLogic Integration B2B Console*、または *Bulk Loader* ユーティリティによってトレーディング パートナにのみ関連付けることができます。これらのプロパティにより、トレーディング パートナの標準的な定義済み属性のアプリケーション拡張が可能になります。

extensible markup language (XML)

カスタマイズされたマークアップ言語を定義するのに使用できるメタ言語 (言語を記述するための言語)。これは、*Standard Generalized Markup Language (SGML)* のサブセットで構成されています。

XML によって、ユーザ定義のドキュメント タイプの開発、およびそのようなタイプのドキュメントのデータを使用できるプログラムの作成が容易になります。XML は急速に、データフォーマットとドキュメントを定義、検証、共有するための世界標準となっています。

XML はテキストベースであり (バイナリ フォーマットで記述されない)、バイナリ マーカではなく構文を使用してデータを整理するので、異種および互換性のないシステムとプラットフォーム間でデプロイできます。

XML は *WebLogic Integration* ソリューションの重要なコンポーネントです。

eXtensible Stylesheet Language (XSL)

XML ドキュメントのフォーマットを指定するための言語。

eXtensible Stylesheet Language Transformations (XSLT)

XMLドキュメントを別のXMLドキュメントに変換するために設計されたXML言語。XSLTドキュメントまたはスタイルシートは、XMLドキュメントのノードに対して実行されるデータ変換を定義します。XSLTを使用して、XMLドキュメントをさまざまなテキストフォーマット(XML、HTML、PDFなど)に変換できます。

F

ファクトリ (factory)

- 1) 他の分散型CORBAオブジェクトへのオブジェクト参照を戻す分散型CORBAオブジェクト。ファクトリはサーバアプリケーション内に配置されます。
- 2) CORBAオブジェクトへのオブジェクト参照を取得するためにクライアントによって使用されるインタフェース。ファクトリへのオブジェクト参照は、Factory Finderインタフェースへのオブジェクト参照を使用してクライアントにより取得されます。Factory Finderインタフェースは、システムにより公開され、クライアントのブートストラップ処理の一環としてクライアントから利用できるようになります。

共通オブジェクト要求ブローカーアーキテクチャ (common object request broker architecture: CORBA) を参照。

フェイルオーバー (failover)

サービスの起動に障害が発生したときに、ユーザの操作を必要とせずにサービスの要求を代替のプロバイダに転送することにより、その障害に応答するシステムの機能。つまり、障害が発生したときに制御をバックアップコンポーネントに転送するシステムの機能です。

初期起動が失敗してもフェイルオーバー機能により起動できるサービスのタイプには、サービスプロバイダのロックアップ、サーバの障害のイベントを回復する機能、およびアクセス可能なサーバでサービスの別のインスタンスを検索する機能が含まれます。

クラスタ化可能なスタブ (clusterable stub) を参照。

フィールド (field)

Weblogic Server 環境では、以下を示します。

- 1) レコードの指定した領域。データの特定のカテゴリに使用されます。

-
- 2) セグメント内の参照できるデータの最小単位。
 - 3) セグメントの指定された部分。
 - 4) データベース テーブル内の単一のデータ項目。
 - 5) ウィンドウ内でデータが表示される部分。

WebLogic Integration 環境の場合：データの基本単位としてアプリケーションに解釈されるバイト シーケンス。

ファイル共有クライアント (**file-sharing client**)

FTP を使用して他のトレーディング パートナと通信するトレーディング パートナ。

ファイアウォール (**firewall**)

内部ネットワークとインターネットの間のトラフィックをモニタし、内部ネットワークに出入りできるネットワーク トラフィックのタイプを規制するソフトウェア。

FIX

Financial Information Exchange (FIX) プロトコル。リアルタイムの電子セキュリティ トランザクションを行うために開発されたメッセージの標準。

G

ガベージ コレクション (**garbage collection: GC**)

Java プログラムを実行中に、未使用のオブジェクトと変数をメモリから再生する機能。この機能は通常、バックグラウンドで自動的に実行されますが、プログラマが指定した時刻に実行するよう要求することもできます。

GC

ガベージ コレクション (**garbage collection: GC**) を参照。

一般保護例外 (**general protection fault: GPF**)

プログラムが別のプログラムで使用しているメモリ領域にアクセスしようとした結果発生する Windows プログラムのエラー状況。

グローバルトランザクション (global transaction)

複数のサーバまたは複数のリソース マネージャ インタフェースが、作業の基本単位としてのトランザクションを調整するために使用するトランザクション。グローバルトランザクションは、それぞれが単一のリソース マネージャにアクセスする複数のローカルトランザクションで構成される場合もあります。

ACID プロパティ (ACID properties) および分散トランザクション (distributed transaction) を参照。

GlobalConversationMBean

会話定義 (conversation definition) を表す MBean インタフェース。B2B Integration 管理アプリケーションは、GlobalConversationMBean オブジェクトを使用して実行時環境の会話 (conversation) をモニタします。

GPF

一般保護例外 (general protection fault: GPF) を参照。

グラフィカル ユーザ インタフェース (graphical user interface: GUI)

ウィンドウにメニューやグラフィカル シンボルを表示する高水準のインタフェース。これにより、ユーザは、一連のシステム コマンドを入力せずに、選択や、プロンプトが表示されるときの情報入力などのタスクを実行できます。たとえば、BEA WebLogic Server Administration Console や WebLogic Integration B2B Console では、管理者は Web ブラウザを使用して BEA アプリケーションをコンフィグレーションおよび制御することができます。

Administration Console を参照。

グループ (group)

1) WebLogic Server 環境の場合 — 何らかの特性を共有するユーザの集合。ACL はグループにパーミッションを割り当てることができます。WebLogic Server レルムではすべてのパーミッションが肯定的であるため、グループにパーミッションを与えることは、そのグループのメンバーである各ユーザにパーミッションを与えることと同じです。否定的パーミッションをサポートするレルムでは、個々のパーミッションがグループのパーミッションをオーバーライドします。

2) WebLogic Integration 環境の場合 — 単一のユニットとして扱われるフィールドやグループの集合。

レルム (realm)、パーミッション (permission)、およびアクセス制御リスト (access control list: ACL) を参照。

グループ選択 (group choice)

フィールドまたは他のグループで構成されるグループのどちらかが必要です。すべてのフィールド (または他のグループ) は、それらに含まれる実際のバイナリ データに関して相互に排他的でなければなりません。

グループ ポータル (group portal)

指定したユーザ グループに対して定義されたポータルの特定のビュー。単一のポータル Web アプリケーション内で複数のグループ ポータルを作成できます。グループ ポータルは、レイアウトやポートレットなどのポータル リソースを共有できますが、各グループの必要性に合わせて個別にコンフィグレーションをすることもできます。

グループ プロファイル (group profile)

ユーザや他のグループの集まり。グループ プロファイルには一連の属性を含めることもでき、プロフィール内の各ユーザやサブグループはそれらの属性を継承できます。

ユーザ プロファイル (user profile) およびユーザ管理コンポーネント (User Management component) を参照。

GUI

グラフィカル ユーザ インタフェース (graphical user interface: GUI) を参照。

H

ヒューリスティック コミット (heuristic commit)

トランザクションに関連するすべての更新がコミットされた場合のヒューリスティックな終了。

ヒューリスティックな終了 (heuristic completion) を参照。

ヒューリスティックな終了 (heuristic completion)

分散トランザクションの終了段階で、更新をコミットするかロールバックするかをリソースが一方的に決定すること。この決定により、分散されたデータは不確定な状態のままになることがあります。ヒューリスティックな終了の原因として、ネットワークの障害やトランザクションのタイムアウトが考えられます。ヒューリスティックな決定を参照。
分散トランザクション (distributed transaction)、rollback、およびコミット (commit) を参照。

ヒューリスティックな決定 (heuristic decision)

ヒューリスティックな終了 (heuristic completion) を参照。

ヒューリスティック障害 (heuristic hazard)

トランザクションに関連する少なくとも 1 つの更新がロールバックされたかコミットされたかがリソース マネージャにわからないようなヒューリスティックな終了の結果。

ヒューリスティックな終了 (heuristic completion) を参照。

混合ヒューリスティック (heuristic mixed)

トランザクションに関連する更新の一部がロールバックされ、その他がコミットされた場合のヒューリスティックな終了の結果。

ヒューリスティックな終了 (heuristic completion) を参照。

ヒューリスティックなロールバック (heuristic rollback)

トランザクションに関連するすべての更新がロールバックされた場合のヒューリスティックな終了の結果。

ヒューリスティックな終了 (heuristic completion) を参照。

階層構造 (hierarchy)

- 1) 項目を順位付けするシステム。
- 2) データベースにおけるセグメントのツリー。ルートを起点とし、上位の要素に従属するセグメント タイプへと枝分かれしていきます。あるセグメント タイプが、他の複数のセグメント タイプに従属することはありません。

ホーム インタフェース (home interface)

エンタープライズ Bean の 2 種類のインタフェース (ホーム インタフェースとリモート インタフェース) のうちの 1 つ。ホーム インタフェースは、エンタープライズ Bean を作成および削除するための任意の数のメソッドを定義します (定義しない場合もあります)。セッション Bean のホーム インタフェースが作成および削除メソッドを定義する一方で、エンティティ Bean のホーム インタフェースは作成、ファインダ、および削除メソッドを定義します。

リモート インタフェース (remote interface) を参照。

ホップ タイムスタンプ (hop timestamp)

WebLogic Integration の B2B Integration コンポーネントで到着時に各メッセージにタイムスタンプを付ける機能。アプリケーションからの入力はありません。タイムスタンプは、デバッグとパフォーマンスの測定に役立ちます。このオプションを使用できるかどうかは、使用されているビジネスプロトコル (business protocol) により決まります。

ホスト (host)

- 1) ネットワークに接続され、通信スイッチとして機能する以外のサービスを提供するコンピュータ。
- 2) データ通信システムにおけるプライマリ コンピュータまたは制御側コンピュータ。

ホット デプロイメント (hot deployment)

Web アプリケーション サーバを再起動せずに Web アプリケーションを起動または再コンフィグレーションし、再起動すること。

Web アプリケーション サーバ (Web application server) および Web アプリケーション (Web application: Webapp) を参照。

HTML

Hypertext Markup Language (HTML) を参照。

htmlKona

HTML ページまたはサーブレットを動的に生成するための、WebLogic の Java オブジェクト。

HTTP

Hypertext Transfer Protocol (HTTP) を参照。

HTTP プロキシ (HTTP proxy)

送信する HTTP リクエスト用のインターネットと、受信する要求用のさまざまな IP 送り先を示す中継機能。

HTTP リクエスト (HTTP request)

対応するシナリオのアクションによって指定された JSP URI を起動するためにキャンペーン サービスによって使用される要求の形式。これには、シナリオとシナリオのアクションをトリガした顧客を識別するパラメータが含まれています。これは、HTTP フォーマットで表されます。

シナリオ アクション (scenario action) および Campaign サービス (Campaign services) を参照。

HTTP サーバ (HTTPD)

Hypertext Transfer Protocol (HTTP) を使用してデータを転送するサーバ。

Hypertext Transfer Protocol (HTTP) を参照。

HTTPD

HTTP サーバ (HTTPD) を参照。

Hypertext Markup Language (HTML)

ブラウザに表示されるファイルのフォーマットの指定に使用する標準コード。

Hypertext Transfer Protocol (HTTP)

TCP プロトコルファミリの一部で、TCP/IP に基づいてリモート ホストからハイパーテキスト ファイルを取得するのに使用されるもの。このプロトコルは、以下の関連するエンティティも定義します。

- HTTPD — データ転送に HTTP を使用するサーバ。
- HTTPS — 安全な Web サーバにアクセスするのに使用される Hypertext Transfer Protocol (HTTP)。

transmission control protocol (TCP) を参照。

I

I18N

インターナショナルライゼーション (internationalization I18N) を参照。

IDE

統合開発環境 (integrated development environment: IDE) を参照。

多重呼び出し不変 (idempotent)

$f(f(x)) = x$ が成り立つ場合に、多重呼び出し不変と呼ばれる関数 f 。コンピュータサイエンスでは、プロシージャを同じ引数で繰り返し呼び出したときに結果が同じであれば、そのプロシージャを多重呼び出し不変と定義します。

これは、複製オブジェクトの再試行を取り扱う際に特に有効な特性です。リモートメソッドの呼び出し中に通信障害が発生した場合、その障害の発生が、呼び出されたメソッドがサーバ上で完了する前か後かをいつも判断できるとはかぎりません。メソッドがサーバ上で完了した後に障害が発生したのであれば、再試行すると、同じメソッドを2回(別のサーバ上であっても)呼び出すことになります。

そのメソッドが多重呼び出し不変でない場合は、このような動作が間違った結果を引き起こす恐れがあります。多重呼び出し不変である場合は、同じメソッドを同じ引数で何度呼び出しても、1回呼び出すことと何ら変わりはありません。

アイデンティティ (identity)

プリンシパルに割り当てられる一意のセキュリティ属性の集合。2つのプリンシパルのアイデンティティが同じになることはありません。プリンシパルは、それぞれが一意で種類の異なる複数のアイデンティティを持つことができます。

IDL

インタフェース定義言語 (interface definition language: IDL) を参照。

IIOP

Internet Interoperability Protocol (IIOP) を参照。

impl

実装 (implementation) を参照。

実装 (implementation)

サポートされているインタフェースのすべての操作と属性の動作を定義する、名前付きのオブジェクトまたはクラス。たとえば、`my_object` は `my_impl` を実装します。これは、RMI でリモート インタフェースを呼び出すのに使われます。

暗黙的なプロパティ (implicit properties)

WebLogic Personalization Server プロパティ テーブルに格納されたプロパティ。ゲッターおよびセッター メソッドに対応しません。

明示的なプロパティ (explicit properties) を参照。

インプレッション (impressions)

これまでに広告を見たユーザー数。

IMS

Information Management System (IMS) を参照。

間接プログラミング モデル (indirect programming model)

レプリカ対応スタブが返されるルックアップで使用されるモデル。実行のたびに必ず同じサービス プロバイダによって処理されます。このプログラミング モデルは、ステートフル オブジェクトの場合に用いなければなりません。

直接プログラミング モデル (direct programming model)、レプリカ対応スタブ (replica-aware stub)、ステートフル オブジェクト (stateful object)、ターゲット プログラム モデル (targeted programming model) を参照。

Information Management System (IMS)

OS/390 または MVS/ESA プラットフォーム上の大規模な商用アプリケーション システム向けに使用される、IBM のトランザクション処理モニタ および階層データベース製品。2つの主要コンポーネントは、階層データベース マネージャとキューベースのトランザクション マネージャです。後者は端末からアプリケーション プログラムに要求をルーティングします。

インフラストラクチャ (**infrastructure**)

特定のコンピュータ システムにおいて、より高水準の別のコンポーネントの集合をサポートするコンポーネントの集合 (基礎サービス群)。より高水準のコンポーネント群は一般に、システム全体で使用される機能を提供します。

開始側プリンシパル (**initiating principal**)

アプリケーションと直接対話するエンド ユーザを示すセキュリティプリンシパル。エンド ユーザは、Web クライアントまたはアプリケーションクライアントのどちらかを使用して認証を行うことができます。プリンシパル (**principal**) を参照。

入力プロセッサ (**input processor**)

HTML フォームで送信された顧客データを検証し、それらのデータをパイプラインセッションに格納して後でパイプライン コンポーネントで使用できるようにするプロセッサ。

パイプライン セッション (**pipeline session**) およびパイプライン コンポーネント (**pipeline component**) を参照。

インスタンス (**instance**)

オブジェクトのクラスやコンピュータ プロセスなど、抽象概念またはテンプレートを具体化した個々の要素。

インスタンス変数 (**instance variable**)

特定のオブジェクトに関連付けられるデータ項目。クラスの各インスタンスには、クラスで定義されたインスタンス変数のコピーがあります。

インスタンス化する (**instantiate**)

クラスに属するオブジェクトの特定の変種を定義して名前を付け、オブジェクトに物理的な位置を与えることによってインスタンスを作成すること。

インストルメンテーション (**instrumentation**)

- 1) 管理対象リソースの属性にアクセスする手段を提供し、属性の値を取得または修正できるようにする機能。
- 2) エージェントによって、管理要求に応えるために行われる管理対象リソースへのアクセス。

統合開発環境 (**integrated development environment: IDE**)

既存アプリケーションにおいて、プログラミングのタスクをサポートするソフトウェア環境。Borland の *JBuilder*、Sybase の *PowerJ*、WebGain の *Visual Cafe* などがその例です。

統合 (**integration**)

サービスを要求、またはサービス要求に応えることによって、情報の共有または独立した処理を行うアプリケーションの能力。効果的に統合されたシステムでは、すべての部位が目的を持ち、部位同士が効率的に結合してシステム全体の目的を達成します。

対話仕様 (**interaction specification**)

サービスに関する二次的なメタデータを含む静的 XML ドキュメント。たとえば、DBMS サービスアダプタの対話仕様には、データベース上のプロセスを呼び出すための SQL 文が含まれます。

インタフェース定義言語 (**interface definition language: IDL**)

CORBA 分散オブジェクト環境でオブジェクトを定義する構文。具体的には、オブジェクトが遂行するサービスと、データをそのオブジェクトに渡す方法を記述しています。

共通オブジェクト要求ブローカアーキテクチャ (common object request broker architecture: CORBA) を参照。

国際標準化機構 (**International Standards Organization: ISO**)

コンピュータネットワーク通信やその他多くの技術の標準を制定、保持する国際的な組織。

インターナショナルライゼーション (**internationalization I18N**)

画面に表示されるテキストで使用する自然言語を対象者の地理的なロケーションと文化に合わせるために、ソフトウェアインタフェース(エンドユーザに対して表示されるソフトウェアの部分)を修正すること。多くの場合、HTML インタフェースで使用される JSP の言語カスタマイズを参照します。

I18N という語は、*internationalization* という単語の *i* と *n* の間の 18 文字を示します。

インターネット (Internet)

プロトコルの TCP/IP ファミリを介して通信する、より小規模な分散ネットワークからなる世界規模のコンピュータネットワーク。

transmission control protocol (TCP) およびイントラネット (intranet) を参照。

Internet Interoperability Protocol (IIOP)

World Wide Web 上で ORB と通信するために CORBA クライアントが使用するプロトコル。

Object Request Broker (ORB) および共通オブジェクト要求ブローカーアーキテクチャ (common object request broker architecture: CORBA) を参照。

インターネット プロトコル (Internet Protocol: IP)

TCP ファミリに属するプロトコルの 1 つ。IP ではパケット (パケット交換ネットワーク上で送受信されるメッセージの小片) の形式およびアドレス体系が規定されています。

パケット (packet) および transmission control protocol (TCP) を参照。

インターネット プロトコル アドレス (IP アドレス) (Internet Protocol Address: IP address)

TCP/IP ネットワークでノードを一意に識別する数値。IP アドレスは通常、0 ~ 255 の範囲の 4 つの 10 進数をドットで区切った形で表されます。たとえば、123.205.23.99 のように表します。

インターネット プロトコル マルチキャスト (IP マルチキャスト) (Internet Protocol multicast: IP multicast)

マルチキャスト バックボーン上に分散配置されたサーバ群にデータを送出すること。これは、すべてのオペレーティング システムと大半のルータでサポートされているパケット配信サービスであり、WebLogic Cluster で使用されています。

インターネット プロトコル (Internet Protocol: IP) を参照。

相互運用性 (interoperability)

エンティティ間で要求を交換する能力。

イントラネット (intranet)

LAN、WAN、または場合によってはインターネットを経由して接続されるプライベート ネットワーク。

ワイド エリア ネットワーク (wide area network: WAN)、インターネット (Internet)、およびローカル エリア ネットワーク (Local Area Network: LAN) を参照。

呼び出し (invocation)

分散型オブジェクトのメソッド呼び出しを実行するプロセス。呼び出しが行われるとき、ネットワーク上でのオブジェクトの位置が認識されている場合と認識されていない場合があります。オブジェクトのインタフェースがコンパイル時に認識されるときは、CORBA の静的呼び出しが使われます。この呼び出しでは、呼び出し側でクライアント スタブが、呼び出されるサービスの側ではサーバ スケルトンがそれぞれ使用されます。インタフェースがコンパイル時点で未知である場合、CORBA の動的呼び出しを使用する必要があります。

IP

インターネット プロトコル (Internet Protocol: IP) を参照。

IP マルチキャスト (IP multicast)

インターネット プロトコル マルチキャスト (IP マルチキャスト) (Internet Protocol multicast: IP multicast) を参照。

ISO

国際標準化機構 (International Standards Organization: ISO) を参照。

J

J2EE

JAAS および Java 2 Enterprise Edition (J2EE) を参照。

J2EE コネクタ (J2EE Connector)

ベンダが供給するリソース アダプタ。J2EE 準拠のアプリケーションサーバをエンタープライズ情報システム (enterprise information systems: EIS) と統合するために使用されます。

Java 2 Enterprise Edition (J2EE)、J2EE コネクタ アーキテクチャ (J2EE Connector Architecture)、およびリソース アダプタ (resource adapter) を参照。

J2EE コネクタ アーキテクチャ (J2EE Connector Architecture)

J2EE 準拠のアプリケーションサーバをエンタープライズ情報システム (enterprise information systems: EIS) と統合するためのアーキテクチャ。このアーキテクチャは、EIS ベンダが供給するリソース アダプタと、リソース アダプタがプラグインする WebLogic Server などのアプリケーションサーバの 2 つの部分で構成されます。このアーキテクチャでは、アプリケーションサーバのプラグインとして機能するためにリソース アダプタがサポートする必要があるトランザクション、セキュリティ、接続管理などの規約を定義します。

J2EE コネクタ アーキテクチャは EIS アクセス用の Common Client Interface (CCI) も定義します。CCI は、異種 EIS と対話するためのクライアント API を定義します。

J2EE コネクタ (J2EE Connector) および Java 2 Enterprise Edition (J2EE) を参照。

J2ME

Java 2 Platform, Micro Edition (J2ME) を参照。

JAAS

Java Authentication and Authorization Service (JAAS) を参照。

.jar

JAR ファイル (JAR file) を参照。

JAR ファイル (JAR file)

多数のファイルを 1 つに集約するのに使用される Java アーカイブと呼ばれるファイルフォーマット。このフォーマットは圧縮 (zip) ファイルです。単一の HTTP リクエストのアプリレットを実行するのに必要なクラスとその他のファイルをロードするのに役立ちます。

JAR ファイルには .jar の拡張子が付けられます。JAR ファイルは、Netscape 3.0 またはそれ以降で動作するアプリレットにのみ使用できます。

アプレット (applet)、Hypertext Transfer Protocol (HTTP)、キャブ (cab)、および Java を参照。

Java

Sun Microsystems, Inc. によって開発されたオブジェクト指向プログラミング言語。「一度書けばどこでも動く (write once, run anywhere)」の原則を特徴としています。

Java 2 Enterprise Edition (J2EE)

ネットワーク、インターネット、および Web 上で動作するように設計された、分散型で拡張可能なエンタープライズレベルのアプリケーションを開発、デプロイするための Java 中心の環境。プラットフォームに依存しない特徴があります。J2EE プラットフォームは、多層の Web ベースアプリケーションを開発するための機能を提供する一連のサービス、API、プロトコルで構成されています。

J2EE により、プログラマは、標準化したモジュラー コンポーネントに基づいてエンタープライズ アプリケーションを簡単に設計できます。また、J2EE は、これらのコンポーネントに対して、アプリケーションの多くの動作を自動的に管理するサービスを提供します。結果として、エンタープライズアプリケーションの開発者は、多くの共通機能に対して複雑なカスタム コードを書く必要がなくなります。

J2EE の主要なコンポーネントの 1 つが EJB テクノロジーです。

WebLogic Portal スイートは J2EE と EJB に基づくものです。

エンタープライズ JavaBeans (Enterprise JavaBeans: EJB) を参照。

Java 2 Platform, Micro Edition (J2ME)

コンシューマ向けのワイヤレス機器用に開発された Java プラットフォーム技術。

Java Authentication and Authorization Service (JAAS)

サービスにおいてユーザを認証し、ユーザに対するアクセス制御の適用方針を規定するための Java パッケージ。

Java Cryptography Extension (JCE)

強力な暗号、鍵の生成と照合、およびメッセージ認証コード (Message Authentication Code) アルゴリズムを使用して暗号化を行うためのフレームワークを規定するパッケージ集合。

Java Database Connectivity (JDBC)

Sun Microsystems, Inc. が策定した、Java からリレーショナルデータベースにアクセスするための仕様。最新リリースは 2.0 です。詳細については、<http://java.sun.com/products/jdbc/index.html> を参照。

Java Enterprise API を参照。

Java Development Kit (JDK)

JavaSoft が提供するソフトウェア開発キット。アプレットおよびアプリケーションを Java で記述するために使用します。

Java 仮想マシン (Java virtual machine: JVM) を参照。

Java Enterprise API

クロスプラットフォームの Java エンタープライズ ソフトウェアの開発を容易にする、JavaSoft のアプリケーションプログラミング インタフェースの集合。BEA は、多くの Java Enterprise API (WebLogic EJB、WebLogic HTTP Servlets、WebLogic JDBC、WebLogic JNDI、WebLogic RMI など) を実装してきました。現在ではさらに多くの API を実装しています。

WebLogic Server HTTP サーブレット (WebLogic Server HTTP servlets)、WebLogic Server jDriver、WebLogic Server JNDI、アクセス制御リスト (access control list: ACL)、WebLogic Server RMI、および WebLogic Server EJB を参照。

Java ヒープ メモリ (Java heap memory)

Java プログラムが実行時に使うためのメモリ領域。

Java Management Extensions (JMX)

Sun Microsystems, Inc. が策定した、管理アプリケーションのための標準 API。WebLogic Integration 管理 API は JMX API に基づいています。

Java Message Service (JMS)

- 1) エンタープライズ メッセージング システムにアクセスするための標準 API。メッセージ指向ミドルウェア (Message-Oriented Middleware: MOM) と呼ばれるエンタープライズ メッセージング システムは、企業全体で重要なビジネス データやイベントを非同期的に交換するための、信頼性のある柔軟なサービスを提供します。
- 2) JMS API はこれに、Java プログラミング言語による、ポータブルなメッセージベース アプリケーションの開発を可能にする共通 API およびプロバイダ フレームワークを付加します。

WebLogic Server JMS は、Java JMS 仕様バージョン 1.0.2 の実装です。
メッセージ指向ミドルウェア (Message-Oriented Middleware: MOM) を参照。

Java Naming and Directory Interface (JNDI)

- 1) 名前によってオブジェクトとリソースを検索するための標準 Java API。
- 2) アプリケーション サーバが、名前付きリソースとファイル システムをあらゆる種類のユーザにどのように知らせ、アクセスさせるべきかを記述した仕様。

WebLogic JNDI は、JavaSoft JNDI に基づいています。

WebLogic Server JNDI および Java Enterprise API を参照。

Java Runtime Environment (JRE)

JVM だけを内蔵している自己完結型 Java 環境。JRE がインストールされていると、ユーザは Java プログラムを実行できます。

ZAC (zero administration client) を使ってアプリケーションと一緒に JRE (さまざまなベンダからいくつかのバージョンが提供されています) を配布し、アプリケーションが想定する Java 環境がユーザ側にあることを保証することができます。

Java 仮想マシン (Java virtual machine: JVM)、Zero Administration Client (ZAC)、および Java Development Kit (JDK) を参照。

Java スクリプト (Java scriptlet)

Java Server Page (JSP) 仕様で使用されるスクリプト言語で有効なコード断片を含むスクリプト記述の要素。JSP 仕様は言語ページの属性が *java* である場合の有効なスクリプトの概念を定義します。

Java セキュリティ マネージャ (Java Security Manager)

Java 仮想マシン (Java virtual machine: JVM) 用のセキュリティ マネージャ。セキュリティ マネージャは Java API とともに機能し、`java.lang.SecurityManager` クラスを通じてセキュリティ境界を定義することにより、プログラマが各自の Java アプリケーション用にカスタムセキュリティ ポリシーを確立できるようにします。

Java Server Pages (JSP)

サーブレット クラスを拡張する Java コンポーネント。Java を併用できる HTML インタフェースの迅速なサーバサイド開発を可能にします。

Java サーブレット (Java servlet)

ブラウザに出力を表示するサーバサイドの Java プログラム。通常は、HTTP リクエストに応答して実行されます。

Java システム プロパティ (Java system property)

Java プログラムが実行される環境に関する情報をまとめた、ネイティブプログラムの環境変数に似たパラメータ。

BEA 製品で使用するプロパティは、クラス `java.util.Properties` に従っており、このクラスが Java システム プロパティの使い方を定義しています。

Java Transaction API (JTA)

1) WebLogic Integration 環境の場合 — アプリケーションと J2EE サーバがトランザクションにアクセスできるようにする API。

2) WebLogic Server 環境の場合 — 高度なアプリケーション トランザクション インタフェースと XA への Java マッピング。JTA により、アプリケーションがユーザのトランザクションの境界を制御できます。

XA インタフェースにより、外部トランザクション マネージャが、2 フェーズ コミット X/Open XA プロトコルを使用して複数のリソース マネージャによって実行される処理のトランザクション境界を制御できます。

JTA は、`javax.transaction` および `javax.transaction.XA` パッケージで定義されています。

トランザクション マネージャ (transaction manager)、リソース マネージャ (resource manager)、XA インタフェース (XA interface)、およびトランザクション (transaction) を参照。

Java Transaction Service (JTS)

以下のいずれかの機能を提供するサービス。

- 1) JTA をサポートするトランザクション マネージャの実装を指定し、API 以下のレベルでの **OMG Object Transaction Service (OTS) 1.1** 仕様の **Java** マッピングを実装します。
- 2) 分散トランザクションの間で違反やその他の障害が波及しないように、それらのトランザクションの制御を維持する方法を提供します。**WebLogic Platform** は、内部的に **JTS** を使ってトランザクション処理機能を実現していますが、その **JTS** 実装のどの部分も公開 API としてエクスポートしていません。

Java Enterprise API を参照。

Java 仮想マシン (Java virtual machine: JVM)

バイトコード命令セット、レジスタ群、スタック、ガベージコレクション対応ヒープ、およびメソッド格納領域からなる **Java** インタプリタ。**Java** コードは、特定のプラットフォーム (**UNIX**、**Windows**) 上の **JVM** で処理されます。

Java Development Kit (JDK) を参照。

JavaBeans

Sun Microsystems による対話の定義に従って他の **Java** オブジェクトと対話する **Java** オブジェクト。**JavaBeans** は、移植性が高く、プラットフォームに依存しません。

Java Enterprise API を参照。

Javadoc

- 1) **Java** ソース コードのコメントから **HTML** フォーマットの **API** ドキュメントを生成する、**Sun Microsystems, Inc.** が提供するツール。たとえば、**Java API** リファレンス ドキュメントは **Javadoc** ツールでフォーマットされています。
- 2) このツールで作成されたドキュメントの一般的な用語。

アプリケーションプログラミング インタフェース (application programming interface: API) を参照。

JCE

Java Cryptography Extension (JCE) を参照。

JCL

ジョブ制御言語 (Job Control Language: JCL) を参照。

JDBC

Java Database Connectivity (JDBC) を参照。

JDBC-ODBCブリッジ (JDBC-ODBC bridge)

ODBC クライアント ライブラリによって汎用のデータベース接続を提供するソフトウェア コンポーネント。このブリッジは、Java から任意のリレーショナル データベースへの接続をサポートしています。Intersolve と JavaSoft の共同プロジェクトとして作成されました。

Open Database Connectivity (ODBC) を参照。

JDK

Java Development Kit (JDK) を参照。

JIT

Just-In-Time コンパイラ (just-in-time compiler: JIT) を参照。

JMS

Java Message Service (JMS) を参照。

JMX

Java Management Extensions (JMX) を参照。

JNDI

Java Naming and Directory Interface (JNDI) を参照。

ジョブ制御言語 (Job Control Language: JCL)

MVS、OS/390、および VSE オペレーティング システム上でジョブ (作業の単位) を定義するための言語。

結合 (join)

イベント、タスク、決定をリンクさせることにより、**Business Process Management (BPM)** イベント、タスク (task)、および決定 (decision) での遷移を制御する **Workflow** コンポーネント。

結合は、**AND** または **OR** の 2 タイプのいずれかになります。

- 結合が **AND** の場合、サクセサまたは結合が起動される前に、リンクされたすべてのコンポーネントを満たしていなければなりません。
- 結合が **OR** の場合、サクセサまたは結合を起動しフローを続行するためには、その結合によりリンクされるコンポーネントのいずれか 1 つを満たしている必要があります。

ワークフロー (workflow) を参照。

Joy

BEA Systems, Inc. によって開発されたユーザ インタフェース ツールキット。**XML** ドキュメントを入力として使用し、ドキュメントのコンテンツからアプリケーションのグラフィカルユーザ インタフェースを作成します。**Joy** は、拡張可能で高度なカスタマイズが可能なユーザ インタフェースの構築をサポートします。

JRE

Java Runtime Environment (JRE) を参照。

JSP

Java Transaction API (JTA) および Java Server Pages (JSP) を参照。

JSP タグ (JSP tag)

かぎカッコ (< 文字列 >) 間に表示される文字列。**JSP** 要素内のマークアップ指示です。さまざまなスクリプト言語で使用できます。

アプリケーションプログラミング インタフェース (application programming interface: API) を参照。

JSP タグ ライブラリ (JSP tag library)

JSP にインポートされた後、特殊なタスクを集团的に実行するためにグループ化されたタグのセット。

JSP タグ (JSP tag) および Java Server Pages (JSP) を参照。

JSP テンプレート (JSP template)

WebLogic Portal の機能を表示する JSP。ビジネス要件に合わせてこれらのテンプレートを変更し、独自のサイトを開発する際のモデルとして使用できます。

JTA

Java Transaction API (JTA) を参照。

JTS

Java Transaction Service (JTS) を参照。

Just-In-Time コンパイラ (just-in-time compiler: JIT)

生成されたマシン コードをメモリに格納し、可能であればそれを再利用するコンパイラ。その結果、Java アプリケーションのパフォーマンスが向上する可能性があります。

デバッグの際には、JIT をオフにすることで、スタック トレースでコードの行番号が表示されます。

JVM

Java 仮想マシン (Java virtual machine: JVM) を参照。

K

Keystore

キーストアの WebLogic Server の実装。WebLogic Server は、カスタムのキーストアもサポートしていますが、必要に応じて独自のものを作成することもできます。

キーストア (keystore)

秘密鍵のデータベースと信頼できるエンティティからの証明書。

キーストア内の各秘密鍵には、対応する公開鍵を認証する証明書チェーンが関連付けられます。

キーワード (keyword)

以下の目的のいずれかのために使用される文字の文字列。

-
- パラメータを識別する
 - コマンド オペランドの一部となる

L

L10N

ローカライゼーション (localization: L10N) を参照。

LAN

ローカル エリア ネットワーク (Local Area Network: LAN) を参照。

大規模メッセージ サポート

管理者が特定のサイズより大きいすべてのメッセージをメモリ内 (デフォルト) ではなくディスクに格納することを要求するための **WebLogic Integration** 機能。特に指定しない限り、大規模メッセージ サポートの対象となるメッセージのサイズは最低 **3KB** です。

LDAP

Lightweight Directory Access Protocol (LDAP) を参照。

レガシー アプリケーション (legacy application)

長い年月と費用を投じて開発され、開発した組織において長期にわたって使用されているカスタム アプリケーション。1 つ以上のそのようなアプリケーションが、集中管理された環境に存在する場合があります。大半のレガシー アプリケーションはメインフレーム上で動作しますが、その一方で、ミニコンピュータおよび一部の大型 UNIX システムで実行できるように作成されたアプリケーションをレガシー システムとみなす場合もあります。また、ストープパイプ アプリケーションとも呼ばれます。

有効期間、Web サービス (lifetime, Web service)

Web サービス開発者が制御する期間に **WebLogic Server** に存在する **WebLogic Workshop Web** サービスのインスタンス。この期間は以下の要因で決まります。

- Web サービスが会話を管理しない場合は、インスタンスは呼び出しのたびに作成され、破棄されます。

- Web サービスが会話を管理する場合は、各会話に有効期間が与えられます。
- 指定したタイムアウト間隔の会話の間で対話が発生しない場合は、会話インスタンスが解放されます。
- 1 つ以上の特定のメソッドまたはコールバックが呼び出されると、会話が終了するように明示的または暗示的に指定することもできます。

Lightweight Directory Access Protocol (LDAP)

データベースまたはサーバにあるディレクトリ(電話番号や請求書送付先住所などの情報の集まり)にアクセスしたり、検索したりするための標準プロトコルの集合。これらのディレクトリは、企業内の多くのアプリケーションからアクセスできるように、複数のシステムに物理的に分散させることができます。

LDAP は X.500 標準に基づいていますが、X.500 に比べると非常にシンプルです。また X.500 と異なり、LDAP はあらゆる種類のインターネット アクセスに必要な TCP/IP をサポートします。

LDAP は証明書の X.509 標準と密接に結び付いていることから、証明書を発行するための理想的な手段です。

X.509 および証明書 (certificate) を参照。

リテラル (literal)

整数、浮動小数点、文字値の基本的な表記。たとえば、3.0 は倍精度の浮動小数点リテラル、「a」は文字リテラルです。

リトルエンディアン (little endian)

下位アドレスのバイトがより低い重要度のバイナリ フォーマット。このフォーマットは、Intel および VAX プロセッサで使用されています。

ロード バランシング (load balancing)

特定のデバイスに過度の負荷が集中しないように、処理および通信アクティビティをコンピュータ ネットワーク間で均等に分散させること。クライアントからの要求に応答する作業は、個々のサーバの負荷に応じて振り分けられます。ロード バランシングには、プロバイダの ID と状態をホストからホストへ移動させる能力が必要です。

クラスタ化可能なスタブ (clusterable stub) を参照。

ローカルエリア ネットワーク (**Local Area Network: LAN**)

比較的狭い領域内で、ワークステーションおよびパーソナル コンピュータを内部接続する設定。

ワイド エリア ネットワーク (wide area network: WAN) を参照。

ローカル トランザクション (**local transaction**)

単一のデータベースまたはファイルにアクセスし、トランザクションへの参加者として単一のローカル リソース マネージャによって制御されるトランザクション。

分散トランザクション (distributed transaction) を参照。

ローカライゼーション (**localization: L10N**)

インターナショナルライゼーションの原則に従って、実行時にロケール固有の言語と構成 (日付フォーマットなど) を使用すること。ソフトウェアがこれらの原則に従っていることを確認するには、以下のことを行います。

- テキスト、数値、アイコンなど、項目に関するハード コード化されたすべての値をソース コードから削除する。
- 上記の手順で削除した値を、特定のロケールに適切なものを提供するリソース バンドルに置き換える。

インターナショナルライゼーション (internationalization I18N) を参照。

ログ ファイル (**log file**)

操作の間に発生したイベントを記録するメッセージ ファイル。ログ ファイルは操作中に頻繁に更新され、システムの操作およびエラーを追跡するために役立ちます。

ログ メッセージ (**log message**)

ローカル ログに記録される特定の状況の通知。BEA 製品では、報告される状況の重大度レベルに基づいて、**fatal** (致命的)、**error** (エラー)、**warning** (警告)、および **info** (情報) の 4 種類のログ メッセージが生成されます。ログ メッセージにはタイムスタンプが付記されます。

ロギング (logging)

システム ログ メッセージ (system log message) およびログ メッセージ (log message) を参照。

M

Managed Bean (MBean)

JMX アーキテクチャのインスツルメンテーションレベルの一部として管理操作の属性を含む Java オブジェクト (JavaBean)。WebLogic Integration では、MBean クラスは実行時情報のモニタに使用されます。MBean は、WebLogic Integration の内部で稼動する MBean サーバによって登録されます。MBean が作成されると、リポジトリからそれらの属性が取り込まれます。実行時に、WebLogic Integration は実行中のシステムの状態を反映するよう MBean 属性を更新します。MBean は、標準 MBean として実装されます。つまり、各クラスは独自の MBean インタフェースを実装します。

ConversationMBean、GlobalConversationMBean、および CollaboratorMBean を参照。

管理対象の環境 (managed environment)

エンタープライズ情報システム (enterprise information systems: EIS) にアクセスする J2EE ベースの Web 対応多層アプリケーションの操作環境の定義。アプリケーションは、コンテナにデプロイされる 1 つまたは複数のアプリケーション コンポーネント (EJB、JSP、サーブレット) で構成されます。コンテナは以下のいずれかです。

- JSP、サーブレット、および静的 HTML ページのホストとなる Web コンテナ
- EJB コンポーネントのホストとなる EJB コンテナ
- スタンドアロンアプリケーション クライアントのホストとなるアプリケーション クライアント コンテナ

管理情報ベース (Management Information Base: MIB)

システムを構成するクラスとその属性の完全な定義を提供するシステムコンポーネント。システムの管理情報ベース (MIB) は、システム全体を対象とした汎用 MIB と、ドメインやワークステーションなどの主要コンポーネントごとの個別 MIB で構成できます。

購入の管理と注文サービスの処理 (Managing Purchases and Processing Orders services)

ショッピング カート、出荷、支払い、税金、注文の概要 / 確認などの Commerce Services を組み合わせたもの。

サービス (service) および顧客の登録および顧客サービスの管理 (Registering Customers and Managing Customer services) を参照。

介在者の攻撃 (man-in-the-middle attack)

部外者がネットワークにマシンを忍び込ませ、通信の当事者間のすべてのメッセージを傍受し、時には改ざんして再送するような攻撃のこと。

MBean

Managed Bean (MBean) を参照。

メッセージ (message)

アプリケーション間でデータを送信するために WebLogic Server で使用される決まった方法。メッセージには、アプリケーションプロセスについての統計またはステータス情報や、受信側に対する指示などが含まれます。メッセージは、次の 2 つの部分で構成されます。

- メッセージ ID データを含むヘッダー
- ユーザ定義の情報を含む本文

ログ メッセージ (log message)、システム ログ メッセージ (system log message)、トリガ商品 (trigger item)、およびビジネス メッセージ (business message) を参照。

メッセージ カタログ (message catalog)

固有の識別子でインデックス付けされたテキスト メッセージの集合の説明。

メッセージ コンテキスト (message context)

メッセージの受信側が、メッセージの構造と内容、およびメッセージが送信された会話 (conversation) を調べるための情報。具体的には、メッセージ コンテキストは、関連する会話、メッセージ定義 (message definition)、メッセージ部分、各メッセージ部分のコンテンツタイプ、および各メッセージ部分のドキュメント定義 (document definition) を識別します。

メッセージ定義 (message definition)

ビジネス コンテンツ、つまりビジネス メッセージ (business message) のペイロード (payload) の仕様。メッセージ定義は順序付けされたメッセージ部分で構成されており、その内容はバイナリまたは XML です。XML メッセージ部分はビジネス ドキュメントを定義し、ドキュメント定義 (document definition) を必要とします。バイナリ メッセージ部分は添付ファイルを定義し、他の情報を必要としません。ペイロードの詳細については、添付ファイル (attachment) およびビジネスドキュメント (business document) を参照。

メッセージ配信方法 (message delivery method)

同期メッセージ配信 (synchronous message delivery) および遅延同期メッセージ配信 (deferred synchronous message delivery) を参照。

メッセージ ダイジェスト (message digest)

テキストをある桁数の単一の文字列の形式で表したものの。一方方向ハッシュ関数と呼ばれる式によって作成されます。秘密鍵を使用してメッセージ ダイジェストを暗号化することにより、電子的な認証手段であるデジタル署名が作成されます。

メッセージ ダイジェスト アルゴリズム (message digest algorithm)

任意長のメッセージを、メッセージ ダイジェストまたはハッシュ値と呼ばれる固定長の文字列に縮約する方式。メッセージ ダイジェスト アルゴリズムには、特定のメッセージ ダイジェストに対応するメッセージを計算によって導き出すことはできない、あるいは、2 つの異なったメッセージから同じメッセージ ダイジェストが生成されることはないという特性があります。メッセージ ダイジェスト アルゴリズムの例には MD5 や SHA-1 があります。

メッセージ エンベロープ (message envelope)

WebLogic Integration の B2B Integration 機能を介して転送されるビジネス メッセージ (business message) のコンテナ。ビジネス メッセージが到着すると、B2B Integration は指定された受信側ごとにメッセージ エンベロープを作成します。ビジネス メッセージ以外にも、メッセージ エンベロープには高度なルーティング情報と処理情報 (送信側と受信側の URL など) が含まれます。

メッセージ フォーマット (message format)

Format Builder によって作成されるバイナリ フォーマットの記述。

メッセージ フォーマット 言語 (message format language: MFL)

バイナリ データ (binary data) のネイティブ表現と階層を記述する BEA の XML 言語。MFL はバイナリ データを XML で記述したものです。

メッセージ ヘッダー (message header)

送信側と受信側、会話 (conversation)、サービス品質 (Quality of Service: QoS) を識別する情報などの属性が含まれるビジネス メッセージ (business message) の一部分。

メッセージ マニピュレータ (message manipulator)

メッセージ マニピュレータ インタフェース `com.bea.b2b.wlpi.MessageManipulator` を実装し、B2B Integration と Business Process Management (BPM) の統合を支援する Java クラス。メッセージ マニピュレータは、BPM 変数を使用して B2B Integration ビジネス メッセージ (business message) の作成と処理を行います。

メッセージ トークン (message token)

アプリケーションがメッセージを送信した後にそのアプリケーションに返されるトークン。このトークンは、メッセージ ID、メッセージが送信された会話 (conversation) の ID、メッセージが適切に送信されたかどうか、すべての受信側が確認応答を送信するまでの経過時間、サービス品質 (Quality of Service: QoS) 配信オプションでこの情報を送信するかどうか、初期選択後の受信側送り先数 (RPC 仕様が適用)、最終選択後の受信側送り先数などの情報を提供します。

メッセージ トラッキング (message tracking)

メッセージが、B2B Integration メッセージング システムの定義済みのさまざまなメッセージ トラッキングのロケーションを通過するのに伴って、メッセージのステータスに関する情報を提供する機能。WebLogic Integration 管理者は、メッセージ トラッキング情報を利用してデバッグ およびシステムのボトルネックの識別を行うことができます。メッセージ トラッキングのロケーションを利用できるかどうかは、WebLogic Integration システムのコンフィグレーション、メッセージのサービス品質 (Quality of Service: QoS)、およびビジネス プロトコル (business protocol) により決まります。

メッセージ駆動型 Bean (message-driven bean)

JMS メッセージ キューから受信する非同期メッセージを処理する EJB。メッセージ駆動型 Bean は、プールからインスタンスを選択してメッセージを処理します。

Java Message Service (JMS)、非同期 (asynchronous)、およびエンタープライズ JavaBeans (Enterprise JavaBeans: EJB) を参照。

メッセージ指向ミドルウェア (Message-Oriented Middleware: MOM)

異機種混在の複数のプラットフォームにアプリケーションを分散可能にすることにより、アプリケーションの相互運用性、ポータビリティ、および柔軟性を向上させるクライアント/サーバ インフラストラクチャ。

メタデータ (metadata)

他のデータの目的、意味、ロケーションを示すのに使用されるデータ。コンテンツ管理システムなどのシステムによって、コンテンツを記述するメタデータに基づいてコンテンツを検索し、ユーザのコンテンツをクエリしたり、一致させたりすることができます。

データベース スキーマ (schema) と XML DTD はメタデータの例です。

コンテンツ管理システム (content management system) を参照。

メタデータ インタフェース (metadata interface)

データについての情報 (特定のオブジェクトについての説明的情報) にアクセスするインタフェース。

メソッド (method)

オブジェクト指向プログラミングにおいて、クラスの一部として定義され、そのクラスのすべてのオブジェクトに組み込まれるプログラム化された手順。クラス(とそれに属するオブジェクト)には複数のメソッドを定義できます。オブジェクトのメソッドは、そのオブジェクトが認識しているデータにのみアクセスできます。これにより、アプリケーション内のオブジェクト集合間でのデータ整合性が保証されます。メソッドは複数のオブジェクトで再利用できます。

オブジェクト (object)、オブジェクト指向プログラミング (object-oriented programming: OOP)、およびクラス (class) を参照。

MIB

管理情報ベース (Management Information Base: MIB) を参照。

最小購入要件 (minimum purchase requirement)

顧客が、指定された項目ごとの割引を受けるために購入しなければならない最小項目数。

MOM

メッセージ指向ミドルウェア (Message-Oriented Middleware: MOM) を参照。

マルチスレッド処理 (multithreading)

(スレッドと呼ばれる)アプリケーションのさまざまな部位を同時に実行できる、オペレーティングシステムの能力。この仕組みによりアプリケーションは複数のタスクを同時に実行できます。

多層 (multitier)

クライアント (1層)、クライアントがアクセスしなければならないすべてのネットワークサービス (1層以上)、ブローカ、アプリケーションサーバ、およびその他の種類のリソース サプライヤ (1層以上) からなる、ネットワーク接続された環境またはアプリケーションのこと。2層は一般に、クライアント/サーバ環境またはアプリケーションを指すのに使われます。

2層 (two-tier) およびクライアント/ネットワーク (client/network) を参照。

N

名前バインディング (name binding)

オブジェクト参照との名前に関連付け。名前バインディングはネーミング コンテキストに格納されます。

バインド (bind) を参照。

名前解決 (name resolution)

名前をオブジェクト参照に変換するプロセス。

ネームスペース (namespace)

1) WebLogic Integration 環境の場合 — 要素タイプと属性名として XML ドキュメントで使用される文字列の集合。各集合は、URI 参照によって識別されます。

2) WebLogic Portal 環境の場合 — WebLogic Portal 環境の場合 — Webflow をより小さな管理しやすいモジュールに分割するのに使用できる項目。たとえば、あるネームスペースで定義されたパイプライン コンポーネントは、別のネームスペースで定義された変数にアクセスでき、その変数の値を第三者のネームスペースで定義した JSP にリダイレクトできます。

Webflow を参照。

ネイティブ メソッド (native method)

Java 以外の言語で書かれたメソッドや関数を呼び出すための Java メカニズム。

WebLogic Server jDriver を参照。

ノード (node)

Webflow メカニズムの状態を示すアイコン。ノード タイプ (表示またはプロセッサ) により、発生しうる多数の定義済みイベント (Web ページのリンクをクリックする訪問者など) があります。特定のイベントが発生すると、Webflow は、そのフローの続行のために次に呼び出すノードを決定します。ノードを呼び出してフローを続行することを遷移といいます。

プロセッサ ノード (processor node)、Webflow、およびプレゼンテーション ノード (presentation node) を参照。

非管理対象の環境 (nonmanaged environment)

2層アプリケーションの操作環境の定義。アプリケーションクライアントは、リソースアダプタを直接使用して、2層アプリケーションの第2層を定義するエンタープライズ情報システム (enterprise information system: EIS) にアクセスします。

非永続性 (nonpersistent)

デフォルトの持続性 (Durability) サービス品質 (Quality of Service: QoS) オプション。このオプションでは、メッセージは WebLogic Integration システムのどこにも格納されません。このオプションで送信されたメッセージは、送り先への送信途中でシステム全体または一部に障害が発生した場合に回復させることができません。このオプションで配信されたメッセージはシステムリソースを少量しか使用しないので、スループットが向上します。

否認防止性 (nonrepudiation)

メッセージが送信または受信されたことを示す法的証明を提供するメカニズム。送信側の否認防止性は、メッセージが送信されたことを法的に証明します。これによって、受信されたメッセージがその送信側にリンクされます。受信側の否認防止性は、メッセージが受信されたことを法的に証明します。これによって、処理されたメッセージがその受信側にリンクされます。

O

OASIS

Organization for the Advancement of Structured Information Standards (OASIS) を参照。

オブジェクト (object)

その状態、動作、およびアイデンティティによって定義されるエンティティ。これらの属性 (プロパティとも呼ばれる) はオブジェクトのオブジェクトシステムによって定義されます。

Object Linking and Embedding (OLE)

オブジェクトのリンクと埋め込みをサポートするドキュメント標準。ActiveX コントロール (ActiveX controls) を参照。

Object Management Group (OMG)

業界ガイドラインおよびオブジェクト管理仕様を策定し、オブジェクト指向アプリケーション開発のための共通フレームワークを提供する国際的組織。OMG の CORBA (Common Object Request Broker Architecture) は、CORBA オブジェクト モデルを仕様化します。これは、アプリケーションまたはシステムの全体設計をオブジェクトとして反映するモデルです。

Object Request Broker (ORB) を参照。

オブジェクト参照 (object reference)

分散型 ORB システムの内部でオブジェクトのインスタンスを一意に指定する識別子。

オブジェクト (object) およびインスタンス (instance) を参照。

Object Request Broker (ORB)

CORBA アプリケーションでクライアントが用いる共通インタフェース。ORB は、オブジェクトが含まれている適切なサーバに要求を送り、その結果をクライアントにリダイレクトします。

共通オブジェクト要求ブローカ アーキテクチャ (common object request broker architecture: CORBA) を参照。

オブジェクト指向プログラミング (object-oriented programming: OOP)

データ、データの構造についての情報、およびデータを処理する関数を 1 つのオブジェクトにまとめる、プログラミング手法の一種。オブジェクト間には関係を作成できます。

ODBC

Open Database Connectivity (ODBC) を参照。

OLE

Object Linking and Embedding (OLE) を参照。

OLTP

オンライン トランザクション処理 (online transaction processing: OLTP) を参照。

オンライン トランザクション処理 (**online transaction processing: OLTP**)

通常、2 フェーズ コミット (two-phase commit: 2PC) と呼ばれる処理を通じて各データベース トランザクションの整合性を保証する方法。

Open Database Connectivity (ODBC)

リレーショナル データベースにアクセスするための Microsoft の標準。
JDBC-ODBC ブリッジ (JDBC-ODBC bridge) および Java Database Connectivity (JDBC) を参照。

オープン フレームワーク (**open framework**)

開発者がプラグアンドプレイ方式でソフトウェア コンポーネントを容易に取り外し、または取り替えできるソフトウェア インフラストラクチャ。

Open Software Distribution (OSD)

W3C に提案され、ネットワークを介したソフトウェアの自動配布を容易にするために用いられるファイルフォーマット仕様。OSD では、コンポーネント ファイルとそれらの間の依存関係を詳細に記述した、ソフトウェア配布キットの XML エンコード済みマニフェストの使い方を明記しています。

extensible markup language (XML) を参照。

オープン システム (**open system**)

複数のコンピュータ ベンダ間で仕様化された共通の標準を実装したシステム。通信用のオープン システム標準を実装することにより、異なったベンダのコンピュータ同士が互いに通信できます。

Open Systems Interconnect Transaction Processing (OSI TP)

異なったコンピュータ上で動作するクライアントとサービス ルーチンの間で、対話を確立し、メッセージを受け渡しするために使用されるサービスとプロトコルを規定した ISO 標準 (ISO/IEC 10026-2)。または、ISO/IEC 10026-2 標準のソフトウェア実装を指します。

Open Systems Interconnection (OSI)

種類の異なるコンピュータ システム間での通信に関する仕様を策定するコンソーシアム。

ORB

Object Request Broker (ORB) を参照。

注文割引 (order discount)

顧客の注文合計に対して適用される割引。

注文限度 (order limit)

1 回の注文で割引できる回数の制限。注文限度は、割引を行う項目や割引を適用する項目に設定できます。

Organization for the Advancement of Structured Information Standards (OASIS)

XML などの公開標準に準拠した、製品やプラットフォームに依存しないフォーマットを迅速に採用するよう活動している国際的なコンソーシアム。OASIS では、DTD、スキーマ、ネームスペースなど、XML 仕様のためのリポジトリとして機能しリソースを提供する Web サイトを運用しています。OASIS の Web サイトの URL は www.xml.org です。

孤立アイテム (orphaned item)

どのカテゴリにも関連付けされていない、製品カタログの品目。
製品カタログ (product catalog) を参照。

OSD

Open Software Distribution (OSD) を参照。

OSI

Open Systems Interconnection (OSI) を参照。

OSI TP

Open Systems Interconnect Transaction Processing (OSI TP) を参照。

総限度 (割引) (overall limit (on discounts))

割引を適用する場合の、注文の回数制限。

P

P3P

Platform for Privacy Preferences Project (P3P) を参照。

パッケージ (package)

Java におけるクラスやインタフェースの論理的グループ分け。 .java ファイルの先頭行に **package** キーワードを宣言することで行います。パッケージ名 (package name) を参照。

パッケージ名 (package name)

Java におけるクラスやインタフェースの論理的グループ分けの名称。 .java ファイルの先頭行に **package** キーワードを宣言することで行います。完全修飾パッケージ名はユニークな名前を命名する方法 (たとえば `weblogic.jdbc.t3.Driver`) であり、コンパイルされたクラスファイルの階層構造と場所が反映されます。Java インタプリタや他のユーティリティがアクセスするためには、名前付きのパッケージの一部であるコンパイル済み Java クラスファイルが、ホストが仮想マシンの **CLASSPATH** で示す場所に存在する必要があります。パッケージ名は、アプレット タグの **CODE** 値を構成します。

CODE タグ (CODE tag) および **CLASSPATH** を参照。

パッケージ アプリケーション (packaged application)

開発するのではなく購入するアプリケーション。これらのアプリケーションには、最高のビジネス モデルを表す再利用可能なビジネス プロセスなどがあり、全面的な開発作業は必要ありません。

パケット (packet)

パケット交換ネットワークを介して伝送されるメッセージの断片。パケットには、データに加えて宛先アドレスの情報が含まれます。

パラメトリック認可 (parametric authorization)

保護されたリソースに対して、ビジネスの要求に関する状況や対象を考慮しながら認可における意思決定を行うことができること。

パーサ (parser)

ソースコード (プログラミング言語のコード) を、オブジェクトコード (マシンコード) に翻訳可能なコンポーネントに分割するプログラム。

Apache XML パーサ (Apache XML Parser) を参照。

パートナーインタフェースプロセス (partner interface process: PIP)

メッセージの編成方法を指定する RosettaNet プロトコルセットの 1 つ。

PIP は会話定義 (conversation definition) の RosettaNet プロトコルです。

パーティ (party)

会話定義 (conversation definition) におけるロール (role) をコラボレーションアグリーメント (collaboration agreement) のトレーディングパートナーにバインドするエンティティ。

ペイロード (payload)

任意の数のビジネスドキュメント (business document) と添付ファイル (attachment) で構成される、ビジネスメッセージ (business message) のビジネスコンテンツ。

支払いサービス (Payment service)

Commerce Services で代金決済の認可、支払い清算、および振り替えを行う方法を規定した、ビジネスロジックとプレゼンテーションロジックを実装したもの。

サービス (service) を参照。

商品ごとの割引 (per-item discounts)

製品カタログの個々のアイテムに適用する割引。商品ごとの割引では、最少購入条件を指定できます。

パーミッション (permission)

システムリソースへのアクセスを定義したもの。WebLogic Server リソースにアクセスするには、主要なアクセスに対応するパーミッションが明示的に付与されている必要があります。

アクセス制御リストで定義されているとおりに、ある処理をあるオブジェクトに対して実行できること。パーミッションは許可 (明示的にある処理をあるオブジェクトに対して実行できること) の場合や禁止 (明

示的にある処理をあるオブジェクトに対して実行できないこと) の場合があります。WebLogic Server 環境では、すべてのパーミッションは許可を示します。処理を禁止するには、ACL から処理を認可する設定を除外する必要があります。

プリンシパル (principal)、アクセス制御リスト (access control list: ACL)、およびレルム (realm) を参照。

永続 (persistence)

一時的に存在する情報や状態をファイルやデータベースなどのリソースに保存するプロセス。

永続性 (Persistent)

持続性 (Durability) メッセージを配信している過程において、送り元と送り先の間にある複数の場所にメッセージを格納し転送する際のサービスの品質 (Quality of Service: QoS) オプション。マシンに障害が発生した場合の保護機構として機能します。このプロセスは、メッセージの永続と呼ばれています。アプリケーションはシステムに対して、メッセージ単位または会話 (conversation) ベースでメッセージ永続させるように指示することができます。しかし、WebLogic Integration の B2B Integration コンポーネントは、メッセージが永続するためには回復可能である必要があります。永続性を達成すると、システムのスループットが犠牲になります。ネームスペース (namespace) メッセージに比べて永続性メッセージは、転送に時間がかかりリソースを消費します。

永続オブジェクト (persistent object)

プロセスとは独立して存在するオブジェクト。プロセス内でオブジェクト参照が作成されます。

Transport Layer Security (TLS) およびオブジェクト参照 (object reference) を参照。

パーソナライゼーション (personalization)

プロファイル基準や他の変数を元にして特定の個人またはグループにアプリケーションを合わせる Web コンテンツ開発者によって使用されるカスタマイズ手法。

Advisor を参照。

パーソナライゼーション アドバイザ (**personalization advisor**)

Advisor を参照。

パーソナライゼーション サーバ (**Personalization Server**)

WebLogic Server Remote を参照。

PIP

パートナ インタフェース プロセス (partner interface process: PIP) を参照。

パイプライン (**pipeline**)

一連のサービスを単一の名前付きサービスにバインドするための機構。たとえば **Commerce Services** では、パイプラインを使用して業務データの処理を管理します。一般的にパイプラインは、**Webflow** の処理結果を元に実行するビジネス ロジックの流れを制御します。**Webflow** および **Pipeline** エディタではパイプラインはパイプライン ノードとして示されます。**Webflow** および **Pipeline** エディタは **BEA E-Business Control Center** から入手できます。

入力プロセッサ (input processor)、サービス (service)、**Webflow**、**Webflow** および **Pipeline** エディタ、および **E-Business Control CenterTM** を参照。

パイプライン コンポーネント (**pipeline component**)

課税額の算出や注文品の発送のような処理のために使用する、サーバ側ビジネス ロジックの独立した単位。**Webflow** および **Pipeline** エディタを使用すれば、一連のパイプライン コンポーネントを指定して 1 つのパイプラインを作成できます。**Webflow** および **Pipeline** エディタは **E-Business Control Center** で入手できます。

パイプライン (pipeline)、**Webflow** および **Pipeline** エディタ、および **E-Business Control CenterTM** を参照。

パイプライン ネームスペース ファイル (**pipeline namespace file**)

1 つ以上のパイプラインをコンフィグレーションする XML ファイル。パイプライン ネームスペース ファイルは **Webflow and Pipeline editors** を使用して編集します。**Webflow** および **Pipeline** エディタは **E-Business Control Center** から入手できます。

Webflow および **Pipeline** エディタ およびパイプライン (pipeline) を参照。

パイプラインセッション (pipeline session)

現在のセッション (現在のショッピングカートなど) や一時的なデータ (顧客の入力内容に対するエラーメッセージなど) に関する情報用の保管場所。

ショッピングカート管理サービス (Shopping Cart Management service) を参照。

PKCSI

Public-Key Cryptography Standards (PKCS-1) を参照。

プレースホルダ (placeholder)

広告が表示される、事前定義済みの場所。プレースホルダはビジネスエンジニア (または他の技術リソース) によって Web ページに埋め込まれます。プレースホルダによって「このメッセージは Web サイトのどこに配置すれば良いか?」という質問に回答できます。プレースホルダの名前は即座に識別できるような名前にしてください。

ビジネスエンジニア (Business Engineer: BE) を参照。

プラットフォーム (platform)

アプリケーションをサポートする、システムハードウェアとソフトウェアの組み合わせ。

Platform for Privacy Preferences Project (P3P)

消費者のプライバシーを訪問先 Web サイトのプライバシー保護慣行と照らし合わせる方法を自動化するために規定した、新しい業界標準。これによって Web サイトにおけるプライバシー保護慣行を自動的にアクセスして簡単に解釈できるフォーマットに変換します。

プラグイン (plug-in)

機能をアプリケーションに追加するソフトウェアモジュール。

B2B プラグイン (B2B plug-in) を参照。

ポート (port)

- 1) マシンに依存している部分のコードを書き換えてプログラムを新しい環境で再コンパイルすることによって、あるハードウェアまたはソフトウェア環境から別の環境にプログラムを移すこと。例としては、別バージョンの **WebLogic Server** 上で動作するようにアプリケーションを修正することがあります。
- 2) **TCP/IP** ネットワークにおける論理的接続の終点。ポート番号によって関連するポートの種類を識別します。

ポート番号 (port number)

論理的な通信チャネルを識別し他の接続と区別する、**TCP/IP** ホスト上のエンティティ。**TCP/IP** サーバは、指定したポートに対して受信する接続リクエストをリスンします。**TCP/IP** クライアントではホストの **IP** アドレスとサーバのポート番号を指定してサーバとの接続を開始します。

ポータル (portal)

エンタープライズのデータとアプリケーションにアクセスする単一のポイントを用意しコンテンツを充実させた **Web** サイト。統合化されパーソナライゼーションされたビューを介して情報が従業員、顧客、およびビジネス パートナに対して表示されます。

ポートレット (portlet) を参照。

ポートレット (portlet)

- 1) **Web** サイトの訪問者の観点において、ポータル ページの小さなウィンドウを占める、特化したコンテンツ領域。たとえばポートレットには、旅行案内書、ビジネス ニュース、天気予報、スポーツのスコアなどがあります。ポータルの管理者が設定した特権に応じて、訪問者はポートレットのコンテンツ、見た目、および位置をカスタマイズできます。訪問者はまたポートレットのウィンドウを編集、最大化、最小化および自由に移動できます。
- 2) サーバ アプリケーションの観点において、ポータル ページで特定のテーマ (天気、ビジネス ニュースなど) のために静的なコンテンツと動的なコンテンツを定義した、**JSP** として実装されているコンポーネント。ポートレット **JSP** は、サーバのデータ エンティティやコンテンツ アダプタをアクセスすることで **HTML** を生成します。

WebLogic Portal はポータルおよびポートレットを作成するツールを提供します。

ポータル (portal) およびアプリケーションプログラミング インタフェース (application programming interface: API) を参照。

プレゼンテーション ノード (presentation node)

Web アプリケーションと対話を行っている人に関係するものを **Webflow** が表示している状態を示すアイコン。**Webflow** では必ずプレゼンテーション ノードの起動と停止を行う必要があります。

プロセッサ ノード (processor node) およびノード (node) を参照。

プリンシパル (principal)

1) **WebLogic Server** 環境では、エンタープライズにおいてデプロイされている機構による認証を条件としたエンティティ。プリンシパルはプリンシパル名を使用して識別され、認証データを使用して認証されます。プリンシパル名と認証データの内容とフォーマットは認証機構によって異なります。

しばしばアクセス制御リストが、レルム内部におけるパーミッションの定義に使用されます。ユーザとグループはともに **Principal** の型です。ある特別な場合を除いて、**ACL** には、**ACL** を管理するための権限をもつプリンシパル オーナが必ず最低 1 人は必要です。

2) **WebLogic Integration** 環境では、**WebLogic Integration** システムへのアクセスが必要なエンティティ。プリンシパルにはトレーディング パートナ、ユーザ、および **B2B Integration** があります。

ユーザ (user)、パーミッション (permission)、アクセス制御リスト (access control list: ACL)、グループ (group)、およびレルム (realm) を参照。

秘密鍵 (private key)

セキュアなメッセージを交換する当事者だけが知っている暗号化 / 復号化鍵。

プライベート プロセス (private process)

ビジネス組織内部で行われるビジネス プロセス。プライベート プロセスの定義と設計は組織特有であり、外部には見えません。プライベート プロセスは、プロセス エンジン (process engine) やバックエンド ビジネス システムとのインタフェースとして動作できます。

プロセス エンジン (process engine)

WebLogic Integration Studio を参照。

プロセッサ ノード (processor node)

Webflow が特化されたコンポーネントを起動して、サイトを表示させるバックエンド ビジネス ロジックやフォーム検証のようなアクティビティを処理している状態を示すアイコン。プロセッサ ノードは次の用途に利用できます。

- 入力プロセッサ
- パイプライン

プレゼンテーション ノード (presentation node) およびノード (node) を参照。

製品カタログ (product catalog)

以下に示す Commerce Services 機能の集合体。

- Dublin Core Open Standard に基づくデータベース スキーマ
- DBLoader と呼ばれる Bulk Loader プログラム
- ブラウザ ベースの管理画面
- JSP テンプレート
- Application Programming Interface (API)

アプリケーション プログラミング インタフェース (application programming interface: API) を参照。

製品アイテム マネージャ (Product Item Manager)

Commerce Services 製品カタログ内のアイテムを作成、取得、更新、および削除するサービス。

プロパティ (property)

名前と値の組によってメタデータを格納するためのしくみ。すべてのプロパティには次の情報が含まれています。すなわち、プロパティ名、データ型、プロパティが単一の値 (つまりデフォルト値が 1 つ) か複数レベル (つまり、デフォルト値の集まりがある) なのかを指定する選択モード、デフォルト値定義用の有効値範囲です。

プロパティセット (property set)

プロパティの論理グループ分けを記述するメタデータの集まり。プロパティセットによってプロパティを適切に分類でき、複数のプロパティが同じ名前を持つよう定義できます。

プロパティセット管理 (property set management)

プロパティを命名したり、プロパティをプロパティセット(命名可能)にグループ分けするのに使用する **WebLogic Portal** コンポーネント。プロパティはオブジェクトの属性を示します。たとえば、*backgroundColor* は **HTML** ページのプロパティです。

プロパティを作成する場合、属性値の型(たとえば、文字列型、数値型、日付型など)、値が1つなのか複数なのか、あるセットに値が限定されるのかどうか、およびデフォルト値を指定します。次のタイプに基づいてプロパティセットを作成できます。すなわち、**User/Group**、**Session**、および **Request** です。

名前が付いたプロパティとプロパティセットを、ユーザとグループ、**HTTP** セッションと **HTTP** リクエスト (**ConfigurableEntity** コンポーネントによって処理されます)、または **Content Manager** からアクセスできるコンテンツに適用できます。しかし、プロパティセットはコンテンツやドキュメントローダーユーティリティ、サードパーティ製コンテンツ管理ツールを管理しません。コンテンツメタデータを管理します。

たとえば、開発者は訪問者がポータルに応じてさまざまな背景色を指定できるようにしたい場合があるでしょう。AポータルのプロパティセットとBポータルのプロパティセットを作成すると、両方のポータル用に **backgroundColor** プロパティを用意できます。2つの **backgroundColor** プロパティは同じ名前を持ち、同じ定義にも異なる定義にもすることができます。

プロトコル (protocol)

- 1) 通信リンクを介して送受信するメッセージのフォーマットとタイミングを規定するルールのセット。ネットワークプロトコルの例として **TCP/IP** があります。
- 2) 情報の通信と交換のために2台のシステムが従うルールのセット。

配信 (provisioning)

ファイルをある地点から別の地点、一般的にはサーバからクライアントに送信するプロセス。サーバは J2EE プラットフォームを実装している場合があり、クライアントは J2ME プラットフォームを実装している場合があります。

プロキシ (proxy)

データ処理タスクを別のプログラムまたはデバイスに移すこと。

プロキシサーバ (proxy server) を参照。

プロキシサーバ (proxy server)

リクエストを別のサーバで処理するために別のサーバに送信するサーバ。WebLogic Server では HTTPProxyServlet で HTTP リクエストのプロキシ処理をサポートしています。WebLogic Server の Netscape Server Application Programming Interface (NSAPI) や Internet Server API (ISAPI) を介して、Netscape や Microsoft Internet Information Server (IIS) から WebLogic Server のインスタンスにプロキシ処理を行うことができます。プロキシサーバを使用していることはエンド ユーザには見えません。

プロキシ (proxy) および Hypertext Transfer Protocol (HTTP) を参照。

公開鍵 (public key)

秘密鍵とともに認証局により提供される値で、メッセージの暗号化と復号化に使用します。

公開鍵アルゴリズム (public key algorithm)

公開鍵または秘密鍵を使用してデータを暗号化または復号化するためのアルゴリズム。秘密鍵は一般に、メッセージダイジェストを暗号化するために使用されます。そのようなアプリケーションでは、公開鍵アルゴリズムはメッセージダイジェスト暗号化アルゴリズムと呼ばれます。公開鍵は一般に、コンテンツ暗号化鍵 (セッションキー) を暗号化するために使用されます。そのようなアプリケーションでは、公開鍵アルゴリズムはキー暗号化アルゴリズムと呼ばれます。公開鍵アルゴリズムの例には RSA があります。

公開鍵暗号化 (public key encryption)

非対称の一对の鍵を使用して、暗号化と復号化を行うためのメカニズム。公開鍵は広く配布され、容易に入手できます。秘密鍵は公に配布されることはなく、常に機密扱いとなります。

公開鍵セキュリティ (public key security)

WebLogic Server 環境を保護するために開発した、公開鍵暗号化技術を拡張したもの。公開鍵暗号化を使用することにより、WebLogic Server とアプリケーション クライアントとの間で、端末間でのデジタル署名およびデータ秘匿を実現します。PKCS-7 標準に準拠しています。

パブリック プロセス (public process)

複数のトレーディング パートナの間で実行されるビジネス プロトコル。パブリック プロセスは、複数のトレーディング パートナ間における正式な契約の一部です。正式な契約では、複数のトレーディング パートナ間で相互にやりとりするメッセージの内容と意味が規定されます。RosettaNet ビジネス プロトコルを使用する場合、ビジネス プロトコルはパートナ インタフェース プロセス (partner interface process: PIP) です。すべてのビジネス プロトコルでは、パブリック プロセスは協調的ワークフロー (collaborative workflow) によって実装されます。

Public-Key Cryptography Standard 7 (PKCS-7)

RSA Laboratories によって開発されている公開鍵暗号化標準のセットの 1 つ。この開発は、当初 Apple、Microsoft、Digital Equipment Corporation、Lotus、Sun、および MIT が参加していた非公式のコンソーシアムとの協働で進められています。PKCS-7 では、デジタル署名や暗号化など、暗号技術の強化を含むメッセージの一般的な構文が定義されています。

WebLogic Server の公開鍵セキュリティは PKCS-7 標準に準拠しています。

Public-Key Cryptography Standards (PKCS-1)

データ伝送の暗号化を実装するために、RSA Laboratories によって開発されている公開鍵暗号化標準のセットの 1 つ。

RSA およびセキュア ソケット レイヤ (secure sockets layer: SSL) を参照。

Q

QoS

サービス品質 (Quality of Service: QoS) を参照。

QPA

Query for Price and Availability (QPA) を参照。

サービス品質 (Quality of Service: QoS)

B2B Integration アクティビティの一部として行われるメッセージ配信における信頼性のレベルを定義するのに **WebLogic Integration** が使用するオプションのクラス。QoS によって、ネットワークのリンク障害やノード障害が発生した場合でも信頼性の高いメッセージ配信を行うためのオプションが用意されます。**WebLogic Integration** オプションによって永続してデータを格納することで、ノード障害が発生した場合に **WebLogic Integration** が **B2B Integration** の処理を回復できるので信頼性が改善されます。

Query for Price and Availability (QPA)

サプライヤ トレーディング パートナに一斉送信されるメッセージ。サプライヤ トレーディング パートナはバイヤ トレーディング パートナに提供価格の応答を行います。

キュー (queue)

時間の順序に従ってサーバにリクエストを配信するのを管理するための単純なデータ構造。キューの要素は、優先順位によって並び替えられる場合があります。クライアントはキューにアイテムを挿入し、サーバがアイテムをキューから即座か定期的またはまとめて削除します。

R

ra.xml

Sun Microsystems によって **J2EE Connector Architecture Specification** で規定されているように、リソースアダプタ用にコンフィグレーションとプロパティを記述する XML ファイル。

文書型定義 (Document Type Definition: DTD) を参照。

RAR

リソースアダプタアーカイブ (Resource Adapter Archive: RAR) を参照。

RDBMS

リレーショナルデータベース管理システム (Relational Database Management System: RDBMS) を参照。

レルム (realm)

- 1) WebLogic Server 環境の場合、セキュリティ属性のセット用のドメイン。レルムはセキュリティ情報を体系的に配置し、処理の範囲を定義します。レルムには、専用のプリンシパルとパーミッションのデータがあります。特定のセキュリティドメインは Java においてレルムインスタンスとして反映されます。レルムは、ユーザの認証方法と WebLogic のリソースへのアクセスを取得する方法を規定したものです。
- 2) WebLogic Integration 環境の場合、ACL、プリンシパル (principal) の名前、および関連するセキュリティサービスへのアクセスを提供する一連のセキュリティ機能のドメイン。セキュリティ処理と、WebLogic Integration ユーザ定義を規定するその他のセキュリティ関連情報の範囲のコンテキストがレルムで提供されます。ユーザの認証方法を規定します。WebLogic Integration で利用できるセキュリティ機能は、WebLogic Server で提供しているセキュリティ機能上に実現されています。
- 3) WebLogic Portal 環境では、ユーザとグループ情報へのアクセス、および認証サービスを提供するデータベースへのインタフェース。グループ (group)、ユーザ (user)、パーミッション (permission)、プリンシパル (principal)、アクセス制御リスト (access control list: ACL)、および資格 (credential) を参照。

レコード (record)

- 1) フィールドに入力された入出力データ。各フィールドには、レコードに関連した情報のアイテムがあります。レコードのセットはファイルを構成します。
- 2) あるプログラミング言語で使用される特別なデータ構造体。レコードデータ構造体は、整数値、数字、文字列など他のデータオブジェクトの組み合わせです。

回復 (recover)

トランザクションを完了させるために行う、コーディネータまたは参加コンポーネントからのリクエスト。

回復処理 (recovery)

障害発生後、トランザクション システムを直近のコミットされた一貫性のある状態に戻すプロセス。

分散システムでは、回復処理は分散コンポーネントの再同期が必要になる場合があります。システムが復旧すると、処理を再開して障害により異常終了したトランザクションを再度実行することができます。

参照 (reference)

事前定義によってその名前、型、および終了属性が決まる、グループまたはフィールド。

参照実装 (reference implementation)

ドキュメント参照実装 (document reference implementation) を参照。

顧客の登録および顧客サービスの管理 (Registering Customers and Managing Customer services)

登録、ログイン、および顧客プロフィール Commerce Services の集合体注文のステータスや履歴を顧客がチェックする際の JSP のセットと、開発者が他のサービスを作成するのに使用できる API のセットが含まれます。

アプリケーション プログラミング インタフェース (application programming interface: API) および API を参照。

登録 (registration)

会話定義 (conversation definition) で指定したロール (role) に対するビジネスメッセージ (business message) を受信するためにトレーディング パートナが行うリクエスト。登録リクエストは WebLogic Integration に送信され、トレーディング パートナがサブスクリプション (subscription) するよう要求します。

登録およびログイン サービス (**Registration and Login service**)

Web サイトの訪問者がユーザ プロファイルを作成し WebLogic Portal で認証する際のビジネス ロジックとプレゼンテーション ロジックを実装したものの。

サービス (service) を参照。

リレーショナル データベース管理システム (**Relational Database Management System: RDBMS**)

リレーショナル データベースの管理用システム。

信頼性 (**reliability**)

システム (またはシステムの一部) が、パフォーマンス仕様を満足しながら (予測されない副作用を除いて) 繰り返し何度も正しい出力を生成する度合い。

信頼性のあるメッセージング (**reliable messaging**)

マシン障害が発生してもメッセージが安全に送り先に到着するよう保証する、さまざまな サービス品質 (**Quality of Service: QoS**) オプションを用いてメッセージを配信するプロセス。信頼できるメッセージを使用する場合、次に示す機能が利用できます。すなわち、メッセージの受信確認、メッセージ ロギングとトラッキング、メッセージ訂正、再試行、およびメッセージ配信方法の選択です。

リモート インタフェース (**remote interface**)

エンタープライズ bean インタフェース用インタフェース。ホーム インタフェースとリモート インタフェースのうちの 1 つ。リモート インタフェースではクライアントによって呼び出しが可能なビジネス メソッドが定義されます。

ホーム インタフェース (home interface) を参照。

Remote Method Invocation (**RMI**)

リモートの JVM に格納されたオブジェクトのメソッドをあたかもローカル システムにあるオブジェクトのようにアプリケーションが起動できるメソッド。WebLogic Server RMI は JavaSoft 仕様を実装したものです。

Java 仮想マシン (Java virtual machine: JVM) および Java Enterprise API を参照。

remote procedure call (RPC)

データベースに格納され名前を使用して実行できるデータベース プロシージャ。適切なパーミッションが付与されたクライアントは、相手がプロシージャを実行して結果を戻すよう要求することができます。

レプリカ ハンドラ (replica handler)

レプリカ対応スタブのためにロードバランシングとフェイルオーバーを行うソフトウェア コンポーネント。レプリカ ハンドラは利用可能なレプリカのリストを保持し、レプリカ対応スタブでリモートの起動が行われたときに実行するレプリカを選択します。

WebLogic Server の組み込みレプリカ ハンドラを選択するには、**RMI** と **EJB** コンパイル用のさまざまなフラグを設定します。フラグによって、障害と再試行、およびサーバアフィニティなどのサービス特性の処理方法が決まります。

レプリカ対応スタブ (replica-aware stub)、スタブ (stub)、および Remote Method Invocation (RMI) を参照。

レプリカ対応スタブ (replica-aware stub)

呼び出しをサービス インスタンスに委託するラッパー。サービスが別のサーバで透過的に処理される、ロードバランシングとフェイルオーバーの状況に対処する方法に関する情報を持っています。レプリカ対応スタブはレプリカ ハンドラと呼ばれるコンポーネントを使用して、そのような処理をカプセル化します。

レプリカ対応スタブを生成するには、**RMIC** 用に `-replicaAware` フラグか類似の **EJB** デプロイメントを記述し、プロパティを設定してください。

Remote Method Invocation (RMI)、スタブ (stub)、およびレプリカ ハンドラ (replica handler) を参照。

リポジトリ (repository)

1) **WebLogic Server** 環境において、格納されている情報の種類を示すデータベース。主な格納先としては、**JDBC** によりアクセスするリレーショナルデータベースがあります。**BEA** リポジトリは、コンソールと呼ばれる **HTML** インタフェースによりコンフィグレーションしてアクセスします。他のユーティリティを使用すると、ユーザはリポジトリのデータを更新し処理できます。

2) **WebLogic Integration** 環境において、格納されている情報の種類を示すデータベース。情報の種類を次に示します。すなわち、トレーディング パートナ、会話 (conversation)、ドキュメント定義 (document definition)、XML スキーマ (XML schema)、および XSLT マッピングです。主な格納先としては、JDBC によりアクセスするリレーショナルデータベースがあります。リポジトリは、HTML インタフェースである **WebLogic Integration B2B Console** を介してコンフィグレーションしアクセスすることができます。Bulk Loader コーティリティを使用すれば、リポジトリにあるデータを更新して処理し、データを削除できます。

リポジトリ ディレクトリ (repository directory)

イメージやポートレット JSP などの共有リソースがあるディレクトリ。Java サブレット登録中にリポジトリ ディレクトリを指定します。ポートレット (portlet) および Java サブレット (Java servlet) を参照。

リソースアダプタ (resource adapter)

システムレベルのソフトウェアドライバ。エンタープライズ情報システム (enterprise information system: EIS) に接続するアプリケーションクライアントかアプリケーションサーバ (WebLogic Server など) が使用します。リソースアダプタは J2EE コネクタとして機能します。WebLogic J2EE コネクタアーキテクチャは、EIS ベンダおよびサードパーティアプリケーション開発者が開発し、Sun Microsystems の J2EE プラットフォーム仕様、バージョン 1.3 に準拠しているアプリケーションサーバにデプロイ可能なリソースアダプタをサポートしています。リソースアダプタには Java コンポーネントと、必要に応じて EIS と対話的に処理するのに必要なネイティブ コンポーネントが含まれています。

J2EE コネクタアーキテクチャ (J2EE Connector Architecture) および J2EE コネクタ (J2EE Connector) を参照。

リソースアダプタ アーカイブ (Resource Adapter Archive: RAR)

リソースアダプタを実行するのに必要となるクラスとその他のファイルをロードするための圧縮ファイル (.zip フォーマット)。

リソースアダプタ (resource adapter) を参照。

リソース マネージャ (**resource manager**)

データベース管理システムなどの情報やプロセスにアクセスするためのインタフェースおよび関連ソフトウェア。リソース マネージャでは、トランザクション処理機能とアクションの永続性が提供されます。それらはグローバルトランザクション内部でアクセスされ制御されるエンティティです。

トランザクション (transaction) を参照。

リソースプリンシパル (**resource principal**)

エンタープライズ情報システム (enterprise information system: EIS) インスタンスへの接続が確立されたセキュリティ コンテキストのセキュリティポリシー。

エンタープライズ情報システム (Enterprise Information System: EIS) およびプリンシパル (principal) を参照。

制限トレーディング パートナ (**restricted trading partner**)

ライセンス制限版の B2B Integration を実行するトレーディング パートナ。

再試行 (**Retry Attempts**)

サービス品質 (Quality of Service: QoS) オプション。これにより B2B Integration は、断続的なネットワーク障害をマスクするために、指定された回数だけ配信地点でメッセージの再送を試みます。B2B Integration は、事前に定義された待ち時間が過ぎた後、メッセージの再送を試みます。このオプションが利用できるかどうかは、使用する ビジネスプロトコル (business protocol) によって決まります。

RMI

Remote Method Invocation (RMI) を参照。

RNIF

RosettaNet Implementation Framework (RNIF) を参照。

ロール (**role**)

WebLogic Server 環境において、認可プリンシパルで実行可能なアクションのセットによって定義される組織の識別情報。

WebLogic Integration 環境において、B2B Integration Business Process Management (BPM) のどちらかと関連しているアクティビティを定義したものの。

- **B2B Integration** 機能の場合は、会話 (conversation) の間でトレーディング パートナが実行可能な購買行為や販売行為などの活動に関連した活動のセットを定義したもの。ルールは、トレーディング パートナが会話において送信または受信したドキュメントの条件で定義します。会話にはそれぞれ複数のルールがあり、各ルールを協調して行うワークフロー (workflow) で定義します。
- **Business Process Management (BPM)** 機能の場合は、ユーザのグループが共有する、責任の範囲、能力、または認可レベル。1 人のユーザが複数のルールに属することができます。

ロールベースのアクセス制御 (**Role Based Access Control: RBAC**)

ロールと呼ばれる組織の識別子を介してリソースへのアクセスを仲介する、セキュリティ機構のクラス。

ロールバック (**roll back**)

トランザクションで更新したリソースすべてを元の状態に戻すような方法でトランザクションを終了させること。

rollback

トランザクションを終了させ、トランザクション中に指定されたリソースに対するすべての変更を取り消すまたは元に戻すイベント。

RosettaNet

- 1) 電子商取引のためのビジネス インタフェースを定義し標準化することを目的とした、ハイテク企業の非営利コンソーシアム。
- 2) RosettaNet コンソーシアムが規定するビジネスと転送プロトコル。

RosettaNet Implementation Framework (RNIF)

PIP を実行する相互運用可能なソフトウェア アプリケーションを作成するためのガイドライン。

RPC

remote procedure call (RPC) を参照。

RSA

データ伝送を暗号化するための公開鍵暗号化アルゴリズム。RSA は RSA Security, Inc. の製品であり、Web サーバとブラウザ間のデータ伝送を暗号化するために SSL で使用しているアルゴリズムの 1 つです。

暗号化 (encryption) およびセキュア ソケット レイヤ (secure sockets layer: SSL) を参照。

ルールセット (rule set)

WebLogic Personalization Server ルールが含まれている XML ファイル。すべてのルールは次のどちらかになります。ContentSelector または Classifier です。各ルール セットにはルールがない場合も、複数のルールがある場合もあります。ルール セットはプロパティセットからプロパティを使用して、ユーザとグループ プロファイルを評価します。

ルール マネージャ (Rules Manager)、XML、およびプロパティ セット (property set) を参照。

ルール マネージャ (Rules Manager)

Web サイトを閲覧するエンド ユーザにパーソナライゼーションしたコンテンツを提供するために、編集用サービスと実行時サービスを用意するコンポーネントのセット。このパーソナライゼーション コンポーネントではビジネス ルールを使用して、ユーザとグループに適切なコンテンツの照合処理を行います。

WebLogic Personalization Server の Rules Management Framework では、具体化したルールに基づいてパーソナライゼーションしたコンテンツを提供するために、編集機能、デプロイメント機能、実行時処理機能が用意されています。このコンポーネントは次の 2 つの主要部分から構成されています。ビジネス アナリストがビジネス ルールを定義しデプロイするための編集用 GUI と、定義したビジネス ルールを評価するための実行時サービスです。

ルール マネージャ EJB は、ベースとなる BEA Rules Engine へのスケラブルでスタートレスの J2EE エントリ ポイントです。これにより、E-Business Control Center で定義したビジネス ルールを実行するのに必要な実行時サービスが提供されます。

ユーザ プロファイル (user profile)、XML、およびルールセット (rule set) を参照。

S

スケーラビリティ (scalability)

ソフトウェア ソリューションをさまざまな規模の問題に適用すること。理想的には、1つのソリューションで単純なものから複雑なものまで多様な問題に十分対応できる必要があります。しかし実際は、それほど複雑でない問題は比較的単純なソリューションで解決できます。

シナリオ (scenario)

キャンペーンにおける主要な構成ブロック。シナリオはシナリオ アクションで構成されます。シナリオ アクションが条件を満たす場合は、そのすべてが実行されます。具体的には、特定のイベントが発生した場合、または顧客が指定した顧客セグメントに属する場合が該当します。シナリオ アクション (scenario action) およびキャンペーン (campaign) を参照。

シナリオ アクション (scenario action)

シナリオのコンポーネント (シナリオは任意の数のアクションで構成されます)。アクションは広告、電子メール、または割引などです。広告の場合、広告用のコンテンツ管理システムを照会します。プレースホルダに広告を表示する場合があります。

シナリオ (scenario)、プレースホルダ (placeholder)、および広告 (ad) を参照。

スケジューリング サービス (scheduling service)

メッセージに対して確実に適切なルーティング、キュー処理、および順番制御が行われるように支援する **WebLogic Integration 機能 (B2B Integration 機能)**に関連しています)。

スキーマ (schema)

XML ドキュメント用に有効な内容を定義するドキュメント。スキーマ定義は DTD に比べて具体的であり、内容をより詳細に制御できます。

extensible markup language (XML) および文書型定義 (Document Type Definition: DTD) を参照。

セキュアソケットレイヤ (secure sockets layer: SSL)

Netscape によって開発されたインターネット転送レベルの技術。アプリケーション間にデータプライバシー (data privacy) を提供し、公開鍵技術に基づいて、認証およびデータ暗号化の使用をサポートします。

- WebLogic Server 環境においてセキュリティ サービスは X.509 証明書とアクセス制御リスト (access control list: ACL) をサポートしており、参加コンポーネントを認証しネットワーク サービスへのアクセスを管理します。
- WebLogic Integration 環境において、双方向 SSL では、通信を行う両方のアプリケーションから デジタル証明書 (digital certificate) を取得する必要があります。

Public-Key Cryptography Standards (PKCS-1)、暗号化 (encryption)、RSA、認証 (authentication)、暗号スイート (cipher suite)、およびアクセス制御リスト (access control list: ACL) を参照。

セキュリティ (security)

データの破壊や窃盗を防止するために利用可能な機構のセット。

認可 (authorization)、データプライバシー (data privacy)、セキュアソケットレイヤ (secure sockets layer: SSL)、および認証 (authentication) を参照。

セキュリティ属性 (security attributes)

主体を統制するシステム ポリシーのベースを構成する、主体 (ユーザや他の実体に関係ありません) の特性。

プリンシパル (principal) および認証 (authentication) を参照。

セキュリティポリシー (security policy)

- 1) 任意のシステムを保護するために提供される、保護タイプの定義。アクセス制御や監査など、多くの種類のセキュリティポリシーを利用できます。
- 2) 要求のコンテキストに基づいてプリンシパルに付与される権限またはパーミッション。

セキュリティプリンシパル (security principal)

アプリケーションやシステムのためにセキュリティ機構にとって既知であり認証可能なエンティティ (ユーザやアプリケーションなど)。

セキュリティプロバイダ (security provider)

セキュリティ サービス (認証、認可、監査、PKI など) を WebLogic Server に提供するコード。セキュリティ サービスは BEA Systems 提供によるものや顧客が作成したものがああります。

セキュリティレルム (security realm)

ユーザ、グループ、および ACL を論理的にグループ化したもの。WebLogic Server 7.0 互換性モードを使用しているとき、WebLogic Server サーバのリソースを保護するのに使用します。

Security Service Provider Interfaces (SSPI)

WebLogic Server セキュリティ サービスに統合されるセキュリティ サービスを複数のベンダが提供できるようにする、Java パッケージのセット。

シリアライズ (serialize)

ある JVM から別の JVM に渡すことができるよう、オブジェクトをバイトストリームにコード化すること。

Java 仮想マシン (Java virtual machine: JVM) を参照。

サーバ (server)

クライアントのリクエストに応答し要求を満たすアプリケーション。サーバの例としては、リレーショナル データベースへの接続を提供する DBMS サーバや、HTTP が要求する HTML ページとサービスを提供する HTTP サーバがあります。

Hypertext Transfer Protocol (HTTP)、Web サービス (Web service)、および HTTP サーバ (HTTPD) を参照。

サーバプロファイル (server profile)

クライアントがサーバに接続するために必要な情報。

サーバサイド コード (server-side code)

クライアントの JVM ではなくてサーバの JVM で実行する Java コード。例としては、Java-Servlet-API (HTTP) サーブレット、WebLogic Server 起動クラス、EJBean、および WebLogic Remote クラスがあります。

Java 仮想マシン (Java virtual machine: JVM)、WebLogic Server Remote、サーブレット (servlet)、および Hypertext Transfer Protocol (HTTP) を参照。

サービス (service)

- 1) **Weblogic Integration** 環境では、EIS 用のアダプタが提供する対話処理によって実装した、エンタープライズ情報システム (Enterprise Information System: EIS) 内にある名前が付いたビジネス機能。サービスでは、双方向で EIS と同期または非同期通信を行うメカニズムが提供されます。
- 2) **WebLogic Portal** 環境では、Web アプリケーションが提供する、データの処理と蓄積、データの相互交換、またはデータの表示のこと。たとえば、出荷と呼ばれるサービスでは次の 3 つの機能が実行されます。すなわち、顧客の注文に関連する出荷情報を記録し、出荷コストを算出し、注文の出荷ステータスを追跡する機能です。

エンタープライズ情報システム (Enterprise Information System: EIS) を参照。

service control

ユーザの Web サービス内から別の Web サービスへの通信を容易にする機能。WSDL から対象となるサービス用にサービス コントロールを作成できます。

サービス プロバイダ インタフェース (service provider interface: SPI)

エンタープライズ情報システム (Enterprise Information System: EIS) との接続性を提供して管理し、トランザクション境界を確立し、受信待ちと伝送リクエストを行うイベント用フレームワークを用意する、オブジェクトが含まれるプログラム。すべての J2EE コネクタアーキテクチャ準拠アダプタは、`javax.resource.spi` パッケージにおけるインタフェース用の実装を用意する必要があります。

サーブレット (servlet)

通常、HTTP リクエストに応じて実行され、それに対するブラウザ内での出力を生成するサーバサイド Java アプレット。動的なコンテンツを生成し、リクエストと応答の典型的な方法でユーザが Web クライアントと対話処理ができるようにして、Web サーバの機能を拡張したもの。

サーバサイド コード (server-side code) を参照。

セッション bean (session bean)

- 1) WebLogic Server 環境では、1 つのクライアントを提供する、一時的な EJB インスタンスです。セッション bean は多くの場合、手続き的なロジックを実装します。
- 2) WebLogic Portal 環境では、ビジネス ロジックを実装する、非永続性でサーバサイドのオブジェクト。セッション bean は、サーバ上で動作するクライアントの論理的な拡張機能とみなすことができます。セッション bean は複数のユーザに共有されません。

エンタープライズ JavaBeans (Enterprise JavaBeans: EJB) を参照。

セットベース割引 (set-based discount)

指定した製品セットがショッピング カートにある顧客に基づいた、単一の割引。

形状 (shape)

XML の構造 (XML shape) または Studio の形状 (Studio shape) を参照。

出荷サービス (Shipping service)

顧客の注文に関連した出荷情報を記録し、出荷費用を算出し、注文の出荷ステータスを追跡する、ビジネス ロジックとプレゼンテーション ロジックを実装したもの。

サービス (service) を参照。

ショッピング カート管理サービス (Shopping Cart Management service)

顧客が購入するために選択したアイテムを表示する Commerce Services において、ビジネス ロジックとプレゼンテーション ロジックを実装したもの。顧客はショッピング カートから代金支払い処理を開始できます。

Simple Object Access Protocol (SOAP)

さまざまな Web サービスで解釈できるように XML メッセージのフォーマットを規定するための標準規則のセット。プラットフォームに関係なくインターネットを介してアプリケーションがお互いに通信ができる手段を提供します。OMG の IIOP と違って、サーバのファイアウォールを通過するために SOAP は DOM を HTTP (ポート番号 80) のパケットに添付して送ります。ファイアウォールは通常ポート番号 80 とポート番号

21 (FTP) リクエストを受け入れるようコンフィグレーションされます。**SOAP** では XML によって情報のフォーマットを定義し、必要な HTTP ヘッダーに追加して送信します。

分散オブジェクト モデル (distributed object model: DOM)、Hypertext Transfer Protocol (HTTP)、extensible markup language (XML)、および Internet Interoperability Protocol (IIOP) を参照。

スケルトン (skeleton)

シリアライズされたリクエストをスタブから受け取り、それをデシリアライズしアンパックして、当該オブジェクトの実装に対して呼び出すべきメソッド呼び出しとして送信するリモート オブジェクトのサーバサイド表現。サーバサイド スケルトンは、対応するクライアントサイド スタブから出されたリクエストのデシリアライズとアンパックを担当します。実装 (implementation) およびサーバサイド コード (server-side code) を参照。

SNA

Systems Network Architecture (SNA) を参照。

SOAP

Simple Object Access Protocol (SOAP) を参照。

ソケット (socket)

名前がバインドされる通信の終端。TCP/IP 接続の論理的な終端です。アプリケーションはソケットを介して TCP/IP 接続にアクセスします。

選択 (splitting preference)

商取引サービスで提供される JSP テンプレートで利用可能な WebLogic Portal オプション。このオプションを使用すると、顧客が購入した個々のアイテムが入荷次第、顧客に出荷にするか (複数にわけて出荷が必要な場合でも)、すべてのアイテムが揃うまで出荷を延ばすか決めることができます。それらのうちどちらを顧客が選択したのかを確認した後、パイプラインはデータベースに顧客の選択をコミットします。

パイプライン (pipeline) および JSP テンプレート (JSP template) を参照。

SQL

Structured Query Language (SQL) を参照。

SSL

セキュア ソケット レイヤ (secure sockets layer: SSL) を参照。

SSPI

Security Service Provider Interfaces (SSPI) を参照。

スタック トレース (stack trace)

Java 例外を後でデバッグできるようにログするファイル。

スタンドアロン割引 (stand-alone discount)

特定の顧客に限定したり、顧客の行動をベースにしたり、キャンペーンによって管理するのではなくて、特定の Web サイトで購入する人全員に提供される割引。たとえば、Web サイトのポリシーが「誰でも本が10%引き」である場合、その Web サイトはスタンドアロン割引を実行していることとなります。

スタンドアロン割引は E-Business Control Center のすべてのバージョンで利用できます。

BEA E-Business Control Center™ を参照。

開始 (start)

ワークフローをトリガーするプロパティを指定する、ワークフロー (workflow) の最初のコンポーネント。開始ノードの直後の形状 (shape) は、ワークフローにおいて最初のアクティブなコンポーネントです。

ステートフルオブジェクト (stateful object)

アプリケーションのデータが格納されているオブジェクト。その状態は、あるメソッドの実行によって変化する場合があります。クラスタ化された環境では、ロード バランシングとフェイルオーバーの両面で、ステートフルオブジェクトに対する一定の要件があります。ステートフルサービスを提供するサーバの障害時やロード バランシング時に、オブジェクトが別のホストへ移動しても、それに合わせてサービスの状態が変わらなければならないからです。

ステートフルセッション bean (stateful session bean)

特定のクライアントの代わりに*対話状態*を保持する Java bean。対話状態が bean で維持されているときは、複数のクライアントは bean に別々にアクセスでき、前回までの会話の間に蓄積した情報を使用することができます。

使用している bean がステートフルセッション bean (つまりこの bean は対話状態を保持できる bean) の場合、ある日から別の日にかけて同僚との会話を続ける際、*会話を中止したところから再開するのと同じよう*に、クライアントと bean はあるセッションから次のセッションに会話を続行できます。

このためステートフルセッション bean は、複数の対話処理を介してプロセスをマッピングするのに役立ちます。

ステートレスオブジェクト (stateless object)

アプリケーションデータを保存しないオブジェクト。このオブジェクトは等冪です。オブジェクトに関する各処理は、他のすべての処理と独立しています (ステートレスの一般的な概念と、セッション EJB の参照に使用する具体的な定義とを混同しないでください)。

多重呼び出し不変 (idempotent) を参照。

ステートレスセッション bean (stateless session bean)

- 1) WebLogic Server 環境では、呼び出しの間クライアント特有の状態を保持しない Java bean。どのクライアントでも使用できます。セッションの状況に依存しないサービスにアクセスするために使用できます。
- 2) WebLogic Portal 環境では、対話状態を持たないセッション bean。ステートレスセッション bean の全インスタンスは同一です。

静的 (static)

変数をクラス変数として定義する、Java プログラミング言語のキーワード。クラスは、インスタンスがいくつ存在するかに関係なく、クラス変数のコピーを保持します。このキーワードはまた、メソッドをクラスメソッドとして定義するのに使用できます。特定のインスタンスの代わりにクラスがクラスメソッドを起動し、クラス変数でのみ処理されます。

Structured Query Language (SQL)

リレーショナル データベースのテーブルとレコードの管理に使用する標準言語。具体的には **SQL** 文やクエリを使用すると、テーブルやレコードの作成、操作、クエリ、更新、および削除ができます。詳細については <http://www.sql.org> を参照してください。

スタブ (stub)

リモート オブジェクトに実装されたメソッドを起動するために使用する、リモート オブジェクトのクライアント サイドの表示処理。オブジェクトのリモート オブジェクト実装に対するインタフェースを定義します。スタブはクライアントのリクエストをパッケージ化して順番に並び、サーバ側の相手スケルトンに送る責任があります。

スケルトン (skeleton) および実装 (implementation) を参照。

Studio の形状 (Studio shape)

WebLogic Integration Studio の描画範囲における、ワークフロー ノード (workflow node) のビジュアルに表示されるもの。

スタイルシート (stylesheet)

XML ドキュメントで処理するデータ変換 (またはマッピング) を記述する XSLT ドキュメント。スタイルシートでは (XPath を使用して) 操作する対象となる XML ドキュメントのノードと、行う操作が指定されます。

サブジェクト (Subject)

個人などの単一のエンティティに関して関係する情報のためのコンテナ。情報には、サブジェクトの識別子や、それに関連付けられたセキュリティ関連の属性 (パスワードや暗号鍵など) があります。サブジェクトは複数の識別子を持つことがあります。サブジェクトはそれらの識別子を示しません。その代わりにコンテナが示します。各識別子はサブジェクト内部のプリンシパルとして示されます。

WebLogic Server セキュリティプロバイダにおいて、サブジェクトにはユーザ用のプリンシパル (WLSUser プリンシパル) と、ユーザがメンバーとして所属するグループ用のプリンシパル (WLSGroups プリンシパル) があります。カスタム セキュリティプロバイダは識別子を別々に格納する場合があります。

サブスクリプション (subscription)

トレーディング パートナが特定のロール (role) で特定の会話定義 (conversation definition) に関係するためのパーミッション。会話 (conversation) においてロール用にビジネス メッセージ (business message) を送信するためには、トレーディング パートナにそのロールに対応するサブスクリプションが必要です。サブスクリプションを行うには、トレーディング パートナは、自分の名前、対象となる会話の名前とバージョン、およびサブスクリプション対象のロールを指定する必要があります。

サプライ チェーン (supply chain)

製品を製造、出荷、配送するプロセスの順序。

SWIFT

Society for Worldwide Interbank Financial Telecommunications (SWIFT) は、190 ヶ国以上の約 7,000 の金融機関に対して安全なグローバル メッセージ サービスおよびインタフェース ソフトウェアを提供している業界団体です。

<http://www.swift.com> を参照。

Swing

グラフィカル ユーザ インタフェース (graphical user interface: GUI) コンポーネント キットであり、Java 2 platform, Standard Edition (J2SE) に統合されている Java Foundation Classes (JFC) の一部です。Swing はすべてが Java プログラミング言語で記述されたユーザ インタフェース要素の完全なセットを提供することによって、アプリケーションのデプロイメントを簡略化します。Swing コンポーネントでは、特定のウィンドウ システムに依存せずに見た目と操作性をカスタマイズできます。

同期 (synchronous)

値を返すメソッドの属性。この用語は、同期メソッドの呼び出し元は、呼び出されたメソッドが次の文で示す内容を返すのを待つ必要があることに由来しています。呼び出し元はメソッドと同期しています。

非同期 (asynchronous) を参照。

同期メッセージ配信 (synchronous message delivery)

B2B Integration が制御をアプリケーションに一度戻し、メッセージが伝送された結果を知らせるメッセージ配信手法。次のイベントが発生した後に、制御がアプリケーションに戻されます。すなわち、すべての送り先から確認応答を受信した場合、メッセージがタイムアウトした場合、または会話 (conversation) で送られたメッセージが終結した場合が該当します。

システム規約 (system contract)

接続リクエストが複数のエンティティの間で渡されるメカニズム。WebLogic Server とエンタープライズ情報システム (enterprise information systems: EIS) など、アプリケーションサーバ間での標準のシステムレベルプラグイン可能性を実現するために、コネクタアーキテクチャではアプリケーションサーバと EIS 間の標準のシステムレベル規約を定義しています。システムレベルのコントラクトで EIS 側は、リソースアダプタに実装されています。

システム エラー ハンドラ (system error handler)

すべてのワークフロー テンプレート定義 (workflow template definition) で使用できる Business Process Management (BPM) エラー ハンドラ。ワークフロー テンプレート定義でオーバーライドされない限り、例外が発生すると必ず BPM はシステム エラー ハンドラを起動します。システム エラー ハンドラが呼び出されると、指定されたトランザクションをロールバックし、例外をクライアント アプリケーションに送出します。

システム ログ メッセージ (system log message)

WebLogic Integration アクティビティを記録するローカル ログにあるメッセージ。例としては、B2B Integration が終了したためトリガ商品 (trigger item) を配信できないことを示すメッセージがあります。

システムティック コラボレーション (systematic collaboration)

すべての参加コンポーネントがビジネス リレーションシップを確立したときから他のすべての参加コンポーネントを認識している場合の、複数トレーディング パートナ間における協同作業の種類。このような状況においては、ビジネス リレーションシップは継続しプロセスは頻繁には変更されません。

Systems Network Architecture (SNA)

IBM が開発したネットワーク プロトコルで、ピアツーピア ネットワーク、ワークステーション、およびメインフレームを接続するために使用されます。

T

タグ ライブラリ (tag library)

JSP オーサーが利用できるカスタム タグの集合体。

ターゲット (target)

呼び出しチェーン内で、呼び出しの起点でない唯一の受信側。

対象アイテム (target items)

割引が適用されるアイテム。

ターゲット オブジェクト (target object)

ビジネス リクエスト メッセージの受信側。

ターゲット プログラム モデル (targeted programming model)

クラスタ規模の JNDI ツリーで示されるファイル システムなど、単一のステートフル オブジェクト用スタブの取得を目的にルックアップを行うために使用するモデル。このようなスタブは、フェイルオーバーもロード バランシングもできません。サービスそのものが一意であるためです。

間接プログラミング モデル (indirect programming model) および直接プログラミング モデル (direct programming model) を参照。

タスク (task)

ワークフロー (workflow) を構成する一連の独立したアクティビティの 1 つ。タスクはワークフロー ユーザによって、または WebLogic Server で自動的に実行されます。タスクの動作はさまざまなタスク プロパティおよび、あるタスク イベント (task event) が発生したときに実行する処理を定義するタスク アクション (action) によって決まります。

タスク イベント (task event)

ワークフロー (workflow) アナリストによってプログラムされたアクション (action) がタスク (task) のライフサイクルに実行されること。B2B Integration では次の 4 つのイベントがサポートされています。Created、Activated、Executed、および Marked Done です。

課税サービス (Taxation service)

TAXWARE とのインタフェースにより、顧客の注文に対して税額を算出するビジネスロジックとプレゼンテーションロジックを実装したものです。

サービス (service) を参照。

TCP

トランザクション マネージャ (transaction manager) を参照。

Tengah

Tengah は、現在 *BEA WebLogic Server* と総称されるさまざまな製品および API の旧名です。Tengah Server など、以前 *Tengah* に含まれていたすべてのエンティティは、現在 *WebLogic Server* と呼ばれています。*tengah* という語が含まれているメソッド名とクラス名は下位互換性のために変更されずに残されています。

スレッド (thread)

プログラムの他の部分と独立して実行可能な、プログラムの一部。マルチスレッド処理 (multithreading) を参照。

有効期限 (time to live: TTL)

アイテムがキャッシュされている時間の長さ。

タイムアウト (Timeout)

メッセージがタイムアウトになるまで、送信側がメッセージの確認応答を待つ時間の長さを指定するサービス品質 (Quality of Service: QoS) オプション。タイムアウトが発生すると、メッセージ、確認応答処理、および再試行は廃棄されます。このオプションを利用できるかどうかは、使用するビジネスプロトコル (business protocol) によって決まります。

トークン (token)

メッセージ トークン (message token) を参照。

トレーディング パートナ (trading partner)

会話 (conversation) においてビジネス メッセージ (business message) を送受信できる認可を受けたビジネス エンティティ。トレーディング パートナ用のコンフィグレーションには、複数の配信チャネル (delivery channel) があります。

トランザクション (transaction)

Weblogic Server 環境では、以下を示します。

- アプリケーションが共有リソース (たとえばデータベースなど) に対する処理を実行するための論理構成。トランザクションに代わって実行される処理は、原子性、一貫性、隔離性、持続性という 4 つの ACID プロパティに従っています。
- データベースをある持続的な状態から別の状態へ変換するひとまとまりの処理単位。分散トランザクション処理では、トランザクションには 1 つまたは複数のシステム上で実行される複数の処理が入っていることがあります。
- WebLogic Integration 環境において、データを変更する処理の最小単位トランザクションのコードにはプログラム文が含まれており、そのすべてが処理を完了するかロールバックします。トランザクションによって複数のユーザが同じデータに同時にアクセスできません。

ACID プロパティ (ACID properties) を参照。

トランザクション ブランチ (transaction branch)

分散トランザクションをサポートする、リソース マネージャ内部における作業単位。トランザクション マネージャがリソース マネージャに提供する各トランザクション識別子によって、グローバル トランザクションと個々のブランチの両方が識別されます。

リソース マネージャ (resource manager)、トランザクション マネージャ (transaction manager)、および分散トランザクション (distributed transaction) を参照。

トランザクション マネージャ (transaction manager)

アプリケーションプログラムに代わってグローバルトランザクションを管理するシステム ソフトウェア コンポーネント。トランザクション マネージャは、グローバルトランザクションの開始と終了を行うアプリケーションプログラムからのコマンドを、それらのトランザクションに関わるすべてのリソース マネージャと通信することで調整します。リソース マネージャがグローバルトランザクション中に失敗した場合、トランザクションマネージャはリソース マネージャが保留中のグローバルトランザクションをコミットするかロールバックするかどうかの決定を支援します。

リソース マネージャ (resource manager) を参照。

一時オブジェクト (transient object)

そのオブジェクトを作成したプロセスの生存期間中のみ存在するオブジェクト。

永続オブジェクト (persistent object) を参照。

transmission control protocol (TCP)

インターネットでの通信方式を規定するプロトコルファミリ。このプロトコルに基づいて、2つのホストが接続を確立し、データストリームを交換することができます。TCP ではデータが確実に配信されることと、パケットが送信時と同じ順序で配信されることが保証されます。

インターネット (Internet) を参照。

転送 (transport)

配信チャネル (delivery channel) の転送レベルのプロパティ。

Transport Layer Security (TLS)

SSL に基づくプロトコルで、暗号化通信および認証を用いた通信のためのインターネット標準として使用されます。

トリガ商品 (trigger item)

割引を提供するきっかけとなる商品。

トリプル A (triple A)

製品における認証、認可、監査の各サービスの組み合わせ。

信頼モデル (trust model)

システムがセキュアな状態を保つとした場合に、システムのどのコンポーネントおよびシステム外部のどのエンティティが信頼される必要があるか、またどのような対象に対して信頼される必要があるかについて記述したもの。

信頼性のあるコンピューティング基盤 (trusted computing base: TCB)

システムがセキュアであり続けるために正しく機能しなければならないシステムの部位。TCB が変更に対する耐性を備えていることと、そのポリシーの施行が回避可能であることが理想的です。

TTL

有効期限 (time to live: TTL) を参照。

トンネリング (tunneling)

あるネットワークから、別のネットワークの接続を通してデータを転送するための手段。

Hypertext Transfer Protocol (HTTP) を参照。

2 フェーズ コミット (two-phase commit: 2PC)

複数のリソース マネージャ間で単一のトランザクションを調整する方法。2PC により、トランザクションによる更新を関連データベースのすべてにコミットするか、またはすべてのデータベースから完全にロールバックし、トランザクションによる状態の前の状態に戻すことで、データの完全性が保証されます。

2 層 (two-tier)

ユーザ インタフェースがクライアント上で動作し、データベースがサーバ上に格納されるようなクライアント / サーバ間の関係。実際のアプリケーション ロジックは、クライアントまたはサーバのどちらでも動作させることができます。

クライアント / サーバ (client/server) を参照。

U

UDDI

Universal Description, Discovery, and Integration (UDDI) を参照。

Unicode

世界中の大部分の言語で書かれたテキストの表示、処理、および交換に用いられる 16 ビットの文字エンコード体系。Java では、文字列にデフォルトで Unicode を使用します。

Unicode Consortium を参照。

統合プロフィール タイプ (unified profile type)

WebLogic Personalization Server データベースに格納されていないデータに明示的にアクセスできるように、基本ユーザ オブジェクトを拡張したもの。

統合ユーザプロフィール (unified user profile: UUP)

WebLogic Portal データベース テーブルおよびその他のデータ ソースからユーザ管理サービスが収集したユーザ プロパティの集合体。たとえばユーザ管理サービスは、WebLogic Portal データベースのユーザ プロパティと LDAP サーバのユーザ プロパティを統合ユーザプロフィールにまとめることができます。統合ユーザプロフィールによってユーザ情報を検索する場所が 1 つになるので、開発者やシステム ユーザはさまざまな基本データ ソースの違いを無視することができます。

LDAP を参照。

Uniform Resource Locator (URL)

インターネット上でリソースを識別し、その場所を探すための文字列。最新の URL ガイドラインは W3C で入手できます。

Hypertext Transfer Protocol (HTTP) を参照。

Universal Description, Discovery, and Integration (UDDI)

World Wide Web Consortium (W3C) によってメンテナンスされている、分散 Web ベース情報レジストリの仕様。この仕様では、Web サービスに関する情報を調査し公開する手法が定義されています。UDDI は主として Web Services Description Language (WSDL) をサポートするために定義されています。UDDI の主要コンポーネントとしては、UDDI ビジネス登録 (ビジネス エンティティとその Web サービスを規定する XML ファイル) があります。プログラムとプログラマは UDDI ビジネス登録を使用してサービスに関する情報がある場所を突き止めます。

<http://www.uddi.org> を参照。

URL

Uniform Resource Locator (URL) を参照。

ユーザ (user)

個々のプリンシパルのこと。個人のパーミッション (肯定的パーミッションと否定的パーミッションのどちらも) は、そのユーザが所属しているグループに与えられているあらゆるパーミッションに優先します。

パーミッション (permission)、グループ (group)、プリンシパル (principal)、およびレルム (realm) を参照。

ユーザ管理コンポーネント (User Management component)

ユーザおよびグループ プロファイルのプロパティの作成、維持を容易にする JSP タグ、EJB、およびツールのセット。これによって **WebLogic Portal** にあるユーザ プロファイル情報にアクセスできます。ユーザ管理コンポーネントは、ユーザ認証機構およびユーザとグループとの関連付け機能も提供します。

ユーザ プロセッシング サービス (User Processing service)

登録およびログイン サービス (Registration and Login service) を参照。

ユーザ プロファイル (user profile)

ユーザまたはグループと関連付けられている、名前 / 値属性ペアのセット。ユーザ管理コンポーネント (User Management component) を参照。

UUP

統合ユーザ プロファイル (unified user profile: UUP) を参照。

V

付加価値サービス (value-added service)

トレーディング パートナによって、**WebLogic Integration** が提供するサービスに追加される論理。例としては信用情報のチェックや出荷があります。

ベンダ クライアント ライブラリ (vendor client library)

クライアントからの関数呼び出しを、データベースで使われるプロトコルに変換するネイティブレイヤ。

仮想マシン (virtual machine: VM)

独立したコンピュータのように機能する、自己包含形式のオペレーティング環境。通常、Java 仮想マシン (Java Virtual Machine: JVM) のことを示します。

Java 仮想マシン (Java virtual machine: JVM) を参照。

W

W3C

World Wide Web Consortium (W3C) を参照。

WAN

ワイド エリア ネットワーク (wide area network: WAN) を参照。

WAP

Wireless Application Protocol (WAP) を参照。

WAP ゲートウェイ (WAP gateway)

モバイルクライアントを含むモバイルネットワークとアプリケーションサーバを含むコンピュータ ネットワークとの間のブリッジ。WAP ゲートウェイには、一般に WAP プロトコル スタックから WWW プロトコル スタック (HTTP と TCP/IP) へのリクエストを変換するプロトコル ゲートウェイと、Web コンテンツをコンパクトなエンコード済みフォーマットに変換するコンテンツ エンコーダおよびデコーダが含まれています。

Wireless Application Protocol (WAP) を参照。

Web アプリケーション (Web application: Webapp)

対話型オンライン アプリケーションを構成する、サーバ側の Web リソース グループ。Web リソースには、Web ブラウザにデプロイできるアプレット、Java サブレット、JavaServer Pages (JSP)、静的なドキュメント (HTML ドキュメント) などがあります。Web アプリケーションは、Web アプリケーション サーバのコンテキストで実行する必要があります。WebLogic Portal は Web アプリケーションとしてパッケージされています。

Java サーブレット (Java servlet)、アプリケーションプログラミング インタフェース (application programming interface: API)、Web アプリケーション サーバ (Web application server)、およびアプレット (applet) を参照。

Web アプリケーション サーバ (Web application server)

e ビジネスで使用するデータベースおよびバックエンド システムと、Web ベースのアプリケーションでデータを交換できるようにするソフトウェア。BEA の Web アプリケーション サーバは WebLogic Server です。Web サービス (Web service) および Web アプリケーション (Web application: Webapp) を参照。

Web サービス (Web service)

ある企業が、オンライン ビジネス機能を一般にインターネットなどのネットワークを介して他の企業やソフトウェア プログラムに提供するコード モジュール。Web サービスを提供するコード モジュールは、特定の言語やプラットフォームに制約されず、提供するサービスをコード自体で説明するしくみをとっています。

Web サービスの場所は、サービスを使用するために設計されたアプリケーション内部にコードとして埋め込まれている場合も、UDDI (Universal Description, Discovery, and Integration) を使って識別される場合もあります。Web サービスはコード自体で内容を説明するしくみをとっているため、利用するアプリケーション側で、利用可能な機能や呼び出し方法を特定できます。

Web Services Description Language (WSDL)

Web サービスを記述するための XML ベース仕様のマークアップ言語。これによってさまざまなクライアントでサービスを呼び出すことができます。2 つの別々のオンライン システムで人手を介さずに通信する必要がある場合は、WSDL が必要になります。

web.xml ファイル (web.xml file)

デプロイメント記述子 (deployment descriptor) を参照。

Webflow

Web サイトのフローを制御するメカニズム。WebLogic Portal ではさまざまなサンプルアプリケーションで Webflow を使用することにより、Web ブラウザで表示するページや実行するビジネス ロジックを決定します。

Webflow ネームスペース ファイルを使用して、Webflow を特定の Web アプリケーション用にコンフィグレーションできます。

Webflow ネームスペース ファイル (Webflow namespace file) およびパイプライン (pipeline) を参照。

Webflow および Pipeline エディタ

Web アプリケーション用に、Webflow を視覚的に作成、変更、および検証するためのグラフィック ツール。Webflow および Pipeline エディタで、基盤をなす Webflow ネームスペースおよびパイプライン ネームスペース XML ファイルが生成されます。手作業で編集する必要はありません。

これらのエディタは E-Business Control Center で入手できます。

Webflow ネームスペース ファイル (Webflow namespace file)

Webflow を Web アプリケーション用にコンフィグレーションする XML ファイル。Web ページの表示順番を制御し、Web ページと関連付けられたビジネス ロジックを実行させます。

Webflow ネームスペース ファイルを編集するには、Webflow および Pipeline エディタを使用します。これらのエディタは E-Business Control Center で入手できます。

E-Business Control CenterTM、アプリケーションプログラミング インタフェース (application programming interface: API)、Webflow、Webflow および Pipeline エディタ、およびビジネス ロジック (business logic) を参照。

WebLogic Express

Java でデータベース アプリケーションを開発するための BEA 製品コンフィグレーション。WebLogic Express には、好みの 2 層 WebLogic jDriver、セキュリティ機能、HTTP サーブレット サポート、Name and Directory Services のほかに、JDBC 仕様の多層型実装である WebLogic Server JDBC が含まれています。

WebLogic Server jDriver、JDBC、および WebLogic Server jDriver を参照。

WebLogic ホーム (WebLogic Home)

WebLogic Server ソフトウェアのルート ディレクトリ。個々のサーバ ディレクトリが入っているディレクトリです。

WebLogic Integration

アプリケーションサーバ、アプリケーション統合、Business Process Management、企業間 (B2B) 統合機能を企業に提供する、標準に準拠したオープンなソフトウェアプラットフォーム。WebLogic Integration は EJB プラットフォーム WebLogic Server 上で動作し、Java でその全体が実装されています。WebLogic Integration の中心は XML であり、緩やかに対をなした参加コンポーネントの間でのオープンなデータ相互交換フォーマットを提供します。WebLogic Integration はまた、J2EE および ワークフロー (workflow) 技術 (Business Process Management) を利用して、RosettaNet、ebXML MS などの先進の電子商取引ビジネスプロトコル (business protocol) をサポートします。

WebLogic Integration B2B Console

管理者が WebLogic Integration の B2B Integration 機能をコンフィギュレーションしモニタするための HTML インタフェース。

WebLogic Integration プロセス エンジン (WebLogic Integration process engine)

Business Process Management (BPM) ワークフロー (workflow) をモニタし制御する、WebLogic Integration の実行時コンポーネント。

WebLogic Integration リポジトリ (WebLogic Integration repository)

リポジトリ (repository) を参照。

WebLogic Integration Studio

ワークフロー (workflow) を定義しモニタする、Business Process Management クライアント アプリケーション。

WebLogic JDBC

JDBC 仕様の pure-Java 多層実装で、これは WebLogic Express にも WebLogic Server の製品パッケージにも含まれています。

Java Database Connectivity (JDBC) および WebLogic Server ファイル サービス (WebLogic Server file services) を参照。

WebLogic JHTML

Java コードを含む標準 HTML ページからサーブレットを自動的に生成する機能を、以前サポートしていた **WebLogic Server** の旧サービス。

JHTML は **JavaServer Pages (JSP)** に置き換えられました。

Java Server Pages (JSP) を参照。

WebLogic Personalization Server

ユーザ プロファイルおよびビジネス ルールに基づいてカスタマイズしたコンテンツを提供する、**Web** アプリケーション用の開発キット。

WebLogic Portal とともに提供されます。

ユーザ プロファイル (user profile)、**Web** アプリケーション (Web application: Webapp)、および **WebLogic Portal** を参照。

WebLogic Portal

ポータル、動的な販売キャンペーン、電子商取引ソリューション、および **Web** アプリケーション用にパーソナライズしたコンテンツをすばやく構築する、コンポーネントおよびテンプレート。以前 **WebLogic Personalization Server** で別々にライセンスされていた機能が含まれます。

WebLogic Server Remote を参照。

WebLogic Server

Java 言語および **J2EE** プラットフォーム (**Sun Microsystems, Inc.** の) を使用して、電子商取引アプリケーションを構築し実行するためのサービスを提供する、**BEA** の **Web** アプリケーション サーバ。

WebLogic Server は標準に準拠して **Pure Java** で記述されたアプリケーション サーバで、分散 **Java** アプリケーションの作成、デプロイメント、管理を行うことができます。**WebLogic Server** はアプリケーション コンポーネントと **DBMS** 接続を管理して、セキュリティ、スケーラビリティ、パフォーマンス、およびトランザクションの完全性を確保します。また、**Enterprise JavaBeans (EJB)**、**RMI**、分散 **JavaBeans**、および **JDBC** など、分散コンポーネント サービスや企業内データベースへのアクセスに対するサポートも提供します。

WebLogic Integration、**WebLogic Portal**、および **WebLogic Workshop** など、**BEA WebLogic Platform** の全コンポーネントは、**WebLogic Server** 上で動作します。

Java Enterprise API、Remote Method Invocation (RMI)、Java Database Connectivity (JDBC)、および Web アプリケーション サーバ (Web application server) を参照。

WebLogic Server Administration Console

管理者がブラウザから WebLogic Server をコンフィグレーションしモニタするための HTML ベースの GUI。

WebLogic Server Beans

サーバ側で動作する、WebLogic Server の分散 beans。JavaSoft JavaBean コンポーネント アーキテクチャを拡張したもので、JavaBeans の分散化と、分散アプリケーションの構築プロセスの劇的な簡略化に使われます。

クラスタ (cluster) および JavaBeans を参照。

WebLogic Server クラスタ (WebLogic Server cluster)

相互に依存した形で動作する複数の WebLogic Server のインスタンス間で共有される複製サービスで、クライアントにとってフェイルセーフで可用性の高い運用をサポートします。WebLogic Server クラスタには、クライアントがサービス オブジェクトにアクセスするための、クラスタ規模の単一 JNDI ツリーがあります。

Java Naming and Directory Interface (JNDI) を参照。

WebLogic Server COM

WebLogic Server RMI 内部から Microsoft COM オブジェクトにアクセスするための WebLogic Server ブリッジ。

Component Object Model (COM) および WebLogic Server RMI を参照。

WebLogic Server EJB

Java で分散オブジェクト指向ビジネス アプリケーションを構築するためのコンポーネント アーキテクチャ。EJB アーキテクチャは、エンタープライズ アプリケーションのライフサイクルのうち、開発、デプロイメント、および実行時の側面に対処するものです。

エンタープライズ JavaBeans (Enterprise JavaBeans: EJB) は、セキュリティ、トランザクション、および状態管理の詳細を管理するコンポーネント フレームワークの内部にビジネス ロジックをカプセル化します。

WebLogic Server EJB は、トランザクション コンポーネントの JavaSoft 仕様を実装したものです。

エンタープライズ JavaBeans (Enterprise JavaBeans: EJB) および Java Enterprise API を参照。

WebLogic Server イベント (WebLogic Server events)

イベント管理と WebLogic Server 環境で機能する通知サービス。

WebLogic Server ファイル サービス (WebLogic Server file services)

WebLogic Server との間でファイルの読み書きを行うための、WebLogic のファイル管理機能。WebLogic File サービスにより、WebLogic クライアントは標準の `java.io` メソッドを使用して、サーバ側で読み込み / 書き出しアクセスを行うことができます。

WebLogic Server HTTP サーブレット (WebLogic Server HTTP servlets)

JavaSoft Java Servlet API に準拠する HTTP サーブレットをサポートする機能。WebLogic Server により提供されます。WebLogic Server で提供されるビルトイン Web サーバに含まれています。

サーブレット (servlet) を参照。

WebLogic Server jDriver

ネイティブのタイプ 2 WebLogic jDriver for Oracle および all-Java、タイプ 4 WebLogic jDriver for Informix および WebLogic jDriver for Microsoft SQL Server からなる 2 層ドライバ。

Java Database Connectivity (JDBC) を参照。

WebLogic Server JNDI

JavaSoft JNDI 標準を実装する WebLogic サービスで、WebLogic Server または Cluster 内のオブジェクトの透過的なバインディングとロックアップを提供するもの。

Java Naming and Directory Interface (JNDI) を参照。

WebLogic Server Remote

T3Servlets をサポートする機能。これは、RPC に似た単純で効率的な分散コンピューティング モデルです。この機能により、計算量の膨大なタスクやプライバシー上慎重な取り扱いが必要なタスクを、クライアントから WebLogic Server に移すことができます。

remote procedure call (RPC) を参照。

WebLogic Server RMI

WebLogic において JavaSoft RMI 仕様を実装したもので、WebLogic Server フレームワーク内での標準に準拠した分散オブジェクト コンピューティングを提供します。

Remote Method Invocation (RMI) を参照。

WebLogic Server セキュリティ プロバイダ (WebLogic Server security providers)

WebLogic Server に Security Service Provider Interfaces (SSPI) を実装したものです。セキュリティプロバイダによって、監査、認証、認可、裁決、ロール マッピング、資格マッピング、およびキーストアの使用の各機能が WebLogic Server に追加されます。

WebLogic Server service advertisement protocol (WSAP)

クラスタ規模の複製された JNDI ツリーで、オブジェクトのバインド、アンバインドを行うためのプロトコル。WSAP を使用することにより、WebLogic Server インスタンスはクラスタを動的に結合したり切り離したりできるようになります。クラスタで利用できるサービスをトラッキングし、クラスタ規模の JNDI ツリー内の同じノードで提供されるサービスの競合の調停も行います。

バインド (bind)、クラスタ (cluster)、および JNDI を参照。

weblogic-ra.xml

WebLogic Server に特有なデプロイメント情報を ra.xml ファイルに追加する XML ファイル。

整形形式 (well-formed)

XML 構文規則に準拠する XML ドキュメントの属性。使用するドキュメントでは、構文ルールは一般的に XML パーサがチェックします。

ワイド エリア ネットワーク (**wide area network: WAN**)

通常、地理的に離れた場所に広がる、相互接続された多くのネットワークの集まり。通常非公開で他のネットワークへのアクセスが許されない LAN とは違い、WAN は通常公開されており誰でもアクセスできます。インターネットは最大の WAN です。

ローカル エリア ネットワーク (Local Area Network: LAN) を参照。

Wireless Application Environment (**WAE**)

ナローバンド デバイス用のワイヤレス アプリケーションのための、ネットワーク中立なフレームワーク。

WMLScript (WMLS) および Wireless Markup Language (WML) を参照。

Wireless Application Protocol (**WAP**)

WAP Forum によって開発されたプロトコルセットで、携帯電話やその他のモバイル デバイス向けにインターネットの開発や Web ベースのサービスを提供します。

Wireless Markup Language (**WML**)

WAP 対応デバイスのマイクロブラウザとのインタフェース用として設計された、XML ベース言語。WML ドキュメントの構造およびタグは、Wireless Markup Language 仕様で定義されています。

extensible markup language (XML) および Wireless Application Protocol (WAP) を参照。

WLC

WebLogic Integration リポジトリ (repository) でのルート要素の名前。

WLPS

WebLogic Personalization Server を参照。

WLS

WebLogic Server を参照。

WMLScript (WMLS)

ナローバンド通信およびモバイル クライアントの制限を克服するために開発した、WAP アーキテクチャ用のスクリプト ツール。

Wireless Application Protocol (WAP) を参照。

ワークフロー (workflow)

受注処理など、全体または一部が自動化されたビジネス プロセス。ワークフローの実行時には、高度なビジネス ルールに従って、特定の時刻に特定の参加コンポーネントに情報が渡されます。このため、ほとんどの作業はコンピュータによって処理され、人間は例外が発生した場合のみ対処します。

ワークフロー式 (workflow expression)

実行時に **WebLogic Integration** によって行われる計算または評価。ワークフロー (workflow) 式の構文は、**SQL** の構文に相当します。

ワークフロー ノード (workflow node)

開始 (start)、タスク (task)、イベント、結合 (join)、決定 (decision)、および完了 (done) で構成される、**WebLogic Integration Studio** ワークフロー (workflow) のコンポーネント。

ワークフロー テンプレート (workflow template)

ワークフロー (workflow) を示すために、ワークフローの実装に関するさまざまな定義 (ワークフロー テンプレート定義 (workflow template definition)) が組み合わされた **WebLogic Integration Studio** の青写真またはプロトタイプ。このテンプレートでは、定義を使用する組織に対して制御が維持されます。

各ワークフロー テンプレートには、注文処理や請求などの意味のある名前が付けられています。

ワークフロー テンプレート定義 (workflow template definition)

有効期間によって区別されるワークフロー (workflow) テンプレートの特定バージョン。実行時は、最も有効な (つまり現在アクティブな) 定義を選択することにより、**WebLogic Integration** はワークフロー テンプレート定義のインスタンス (またはセッション) を開始します。

World Wide Web Consortium (W3C)

インターネット ベースの通信と商取引を容易にするために、クライアントとサーバプロトコルの標準の作成する国際的な組織。

WSDL

Web Services Description Language (WSDL) を参照。

X

X.509

証明書のフォーマットを仕様化した標準。デジタル証明書 (digital certificate) で幅広く使用されています。標準フォーマットを使用することにより、名前と公開鍵とをセキュアに関連付けることができ、認証度を強化することができます。

証明書 (certificate) を参照。

XA インタフェース (XA interface)

トランザクションマネージャとリソースマネージャ間における双方向のインタフェース。JTA によって定義されています。このインタフェースにより、トランザクションマネージャは、2 フェーズコミット X/Open XA プロトコルを使用して複数のリソースマネージャによって実行される処理のトランザクションの境界を制御できます。

リソースマネージャ (resource manager)、トランザクションマネージャ (transaction manager)、および Java Transaction API (JTA) を参照。

XML

extensible markup language (XML) を参照。

XML マップ (XML map)

XML メッセージのデータをパラメータに相互に関連づけ、Java メソッドの値を Web サービスに戻す XML の構造。XML マップを使用することにより、特定の XML 構造に拘束されるのを回避できます。この結果、Web サービスにコードを実装する方法の選択肢が増えます。

XML スキーマ (XML schema)

XML ドキュメントの構造、内容、およびセマンティックスを指定したファイル。XML DTD を置き換えます。

スキーマ (schema) を参照。

XML サービス (XML service)

WebLogic Integration に提供される Apache XML サービス。サービスには、Apache XML パーサ (Apache XML Parser)、DOM 実装、XSLT スタイルシート プロセッサなどがあります。

XML の構造 (XML shape)

階層と要素の順序など、XML ドキュメントの構造および構成。

XML ボキャブラリ (XML vocabulary)

DTD に含まれている要素を定義する XML タグのセット。XML ボキャブラリは、特定の産業用や業務用に開発できます。

XPath

XML パス言語 XPath はメッセージ コンテキストの XML ドキュメントをノードのツリーとしてモデル化し、その後 XML ドキュメントのノードを処理します。XPath 言語には、ノードがパターンに一致するかどうかを確認するスタンドアロンのサブセットが含まれます。

ビジネス サービス (business service) および対話仕様 (interaction specification) を参照。

XSL

eXtensible Stylesheet Language (XSL) を参照。

XSL Transformations (XSLT)

World Wide Web Consortium (W3C) による、XML ドキュメントをあるドキュメント定義 (document definition) フォーマットから別のフォーマットへに変換するための仕様。www.w3c.org で入手できます。

XSLT

eXtensible Stylesheet Language Transformations (XSLT) を参照。

Y

(Y で始まる用語はありません)

Z

ZAC

Zero Administration Client (ZAC) を参照。

Zero Administration Client (ZAC)

World Wide Web Consortium (W3C) が発行している HTTP Distribution and Replication Protocol (HTTP による分散およびレプリケーションプロトコル) 仕様を BEA が実装したものです。この仕様では、アプリケーション ソフトウェアの分散および保守を自動化する方法が定義されています。