



BEA WebLogic Integration™

WebLogic Integration ソリューションの設計

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 Systems, Inc. からの書面による事前の同意なしに、複写、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 E-Business Platform、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、関係各社が著作権を有します。

WebLogic Integration ソリューションの設計

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

目次

このマニュアルの内容

WebLogic Integration の概要についてのマニュアル	vii
このマニュアルの内容	viii
対象読者	viii
このマニュアルの印刷方法	ix
サポート情報	ix
表記規則	x

1. はじめに

統合の主要な設計概念	1-1
ビジネス プロセス	1-2
統合ソリューション	1-2
設計目標	1-3
設計上の考慮事項	1-3
ベスト プラクティスと設計パターン	1-4
統合ソリューション設計のロール	1-4
統合スペシャリスト	1-5
ビジネス アナリスト	1-5
アーキテクト	1-6
エンタープライズ情報システム スペシャリスト	1-6
システム管理者	1-6
主要な設計タスク	1-7

2. 統合ソリューションの要件の決定

統合ソリューションとビジネス プロセスの定義	2-2
ビジネス プロセスのアクタとそのロールの定義	2-3
アクタの解析	2-3
ロールの解析	2-4
タイプによるアクタの分類	2-4
人間ユーザ	2-6
アプリケーション統合 - アプリケーション統合で使用するエンター	

ブライズ情報システム (EIS) アプリケーション	2-6
企業内の他のドメイン	2-8
B2B 統合を実現する外部トレーディング パートナ	2-8
ビジネス イベントの定義	2-9
ビジネス イベントについて	2-10
ビジネス イベントの特性の定義	2-10
ビジネス イベントのメッセージ フォーマットの定義	2-10
ビジネス データ フローの定義	2-11
データ フローの要件の定義	2-11
データ フローの解析	2-12
サービス品質の定義	2-13
パフォーマンス	2-14
可用性と信頼性	2-14
応答時間	2-15
セキュリティ	2-15
スケーラビリティ	2-15
ロギングと否認防止性	2-16
統合ソリューションのトポロジの定義	2-16
サンプル WebLogic Integration アプリケーションの要件の指定	2-17
統合ソリューションとビジネス プロセスの指定	2-17
価格と在庫の照会手順	2-18
発注書の発行手順	2-18
アクタとロールの指定	2-18
アクタのタイプ	2-19
サプライヤの選択におけるアクタとロール	2-19
発注書の発行におけるアクタとロール	2-19
ビジネス イベントの指定	2-20
サプライヤ選択プロセスのイベント	2-20
発注プロセスのイベント	2-20
データ フローの要件の指定	2-20
要件のリスト	2-21
価格と在庫の要求プロセスのデータ フロー	2-22
発注プロセスのデータ フロー	2-22
サービス品質の指定	2-23
パフォーマンス	2-23

可用性と信頼性.....	2-23
応答時間.....	2-24
セキュリティ.....	2-24
スケーラビリティ.....	2-24
ロギングと否認防止性.....	2-24
統合トポロジの指定.....	2-25

3. 統合ソリューションの設計

WebLogic Integration のアーキテクチャについて.....	3-2
統合ソリューション トポロジの WebLogic Integration へのマッピング.....	3-4
ネットワーク トポロジと管理統制のスキームの定義.....	3-5
人間ユーザの会話の定義.....	3-5
Worklist ユーザ.....	3-6
クライアント ユーザ.....	3-7
EIS アプリケーションとの統合.....	3-8
Application Integration アダプタ.....	3-8
他の統合テクニック.....	3-10
カスタム アプリケーション統合ソリューションの開発.....	3-11
他の WebLogic Integration クラスタとの統合.....	3-12
統合オプション.....	3-13
統合サンプル.....	3-13
企業内の他のドメインとの統合.....	3-14
BEA TUXEDO および BEA eLink との統合.....	3-15
外部トレーディング パートナとの統合.....	3-15
B2B 統合のアーキテクチャ.....	3-15
サポート対象のビジネス プロトコル.....	3-18
カスタム アプリケーション開発の統合.....	3-20
BPM プラグイン.....	3-20
カスタム アダプタ.....	3-20
カスタム ロジック プラグイン (非推奨).....	3-20
ビジネス プロセスの実装.....	3-21
ビジネス イベント.....	3-21
条件データ.....	3-22
ビジネス ルール.....	3-22
マッピング.....	3-22

ビジネス トランザクション	3-22
データ トランスフォーメーションの定義	3-23
XML ベースのデータ トランスフォーメーションについて	3-24
バイナリ -XML トランスフォーメーションの定義	3-25
バイナリ -バイナリ トランスフォーメーションの定義	3-25
XML-XML トランスフォーメーションの定義	3-26
Contivo Analyst を使用した XML-XML トランスフォーメーション の定義	3-26
WebLogic Integration リポジトリとの Contivo Analyst の統合	3-26
統合ソリューションでの Contivo Analyst のロール	3-27
詳細な設計の作成	3-29
統合設計ツールの使用	3-30
デプロイメント 環境の設計	3-30
統合設計でのパフォーマンスに関する考慮事項	3-31
メッセージの永続性	3-31
メッセージのサイズ	3-31
XML の解析と検証	3-32
ワークフロー	3-32
Application Integration	3-37
アプリケーションのロギング	3-38
Secure Sockets Layer	3-38
サンプル WebLogic Integration アプリケーションの高度な設計の指定	3-39
WebLogic Integration Studio を使用したビジネス プロセスの自動化	3-39
B2B Integration を使用したサプライ チェーンの統合	3-40
Application Integration アダプタを使用した ERP システムの統合	3-41
Data Integration を使用したデータ トランスフォーメーションの管理	3-41

索引

このマニュアルの内容

このマニュアルでは、BEA WebLogic Integration 環境のための統合ソリューションを設計する方法について説明します。このマニュアルでは、主要な設計概念を定義して、包括的に解析されたビジネスおよび技術的な要件に基づいて統合要件を決定し、高可用性、信頼性、パフォーマンス、スケーラビリティ、およびセキュリティの目標を達成するための統合ソリューションを設計する方法について説明します。

WebLogic Integration の概要についてのマニュアル

このマニュアルは、BEA WebLogic Integration の概要、および WebLogic Integration の機能を統合ソリューションの設計、開発、デプロイメントのさまざまな段階でどのように使用するかについて説明した 4 冊のマニュアルシリーズの 1 つです。まずこれらのマニュアルで WebLogic Integration の機能を包括的に理解してください。他の 3 冊のマニュアルの内容は、以下の通りです。

- 『WebLogic Integration 入門』 – WebLogic Integration の概要について説明したマニュアルです。このマニュアルでは、異種のフラグメント化されたビジネスシステムが集まっている今日の E ビジネスが直面している統合問題について概説します。次に、WebLogic Integration が E ビジネスの統合問題を解決するために提供する Application Integration、B2B Integration、Business Process Management、および Data Integration の各機能について説明します。
- 『WebLogic Integration チュートリアル』 – サンプル統合アプリケーションについて説明したマニュアルです。このサンプル アプリケーションでは、サプライチェーン ハブをデプロイして、ビジネス パートナを接続し、複数のビジネス プロセスを自動化し、バックエンドのエンタープライズ情報システムを統合します。このマニュアルでは、サンプル アプリケーションの設定および実行方法と、WebLogic Integration を使用して統合ソリューションを構築および開発する方法について学習します。

-
- 『*WebLogic Integration ソリューションのデプロイメント*』 – 統合ソリューションを開発からプロダクション環境に移行する方法について説明したマニュアルです。このマニュアルでは、統合アプリケーションのコンフィグレーション、スケーリング、移植、およびパフォーマンス チューニングについて学習します。

これらのマニュアルの内容を理解したら、**WebLogic Integration** の機能に関する詳細マニュアルに進んでください。

このマニュアルの内容

このマニュアルの内容は以下のとおりです。

- 第1章「はじめに」では、主要な設計概念、ロール、および設計タスクについて説明します。
- 第2章「統合ソリューションの要件の決定」では、ビジネスおよび技術的な解析に基づき統合要件を決定する方法について説明します。
- 第3章「統合ソリューションの設計」では、**WebLogic Integration** 環境の統合要件を満たすソリューションのアーキテクチャを設計する方法について説明します。

対象読者

このマニュアルは主に、**WebLogic Integration** を使用して実装される統合の調査および設計を担当する統合スペシャリストを対象としています。

統合スペシャリストは、ビジネスアナリストおよびテクニカルアナリストとともに、各統合プロジェクトの目標、要件、およびスコープを定義します。その後、設計パターンとベスト プラクティスに基づいて **WebLogic Integration** 環境のための統合アーキテクチャを設計します。

したがって、統合スペシャリストは、**WebLogic Integration** のアーキテクチャと機能に精通している必要があります。このマニュアルを使用する前に、まず『*WebLogic Integration 入門*』を読んでください。

このマニュアルでは、**WebLogic Integration** サンプルアプリケーションを頻繁に参照します。このマニュアルを使用する前に、まず『**WebLogic Integration チュートリアル**』の特に第1章「はじめに」と第3章「統合ソリューションの設計」を読み、サンプルアプリケーションに精通することから始めてください。

このマニュアルの印刷方法

Web ブラウザの [ファイル | 印刷] オプションを使用すると、Web ブラウザからこのマニュアルを一度に1ファイルずつ印刷できます>

このマニュアルの PDF 版は、**WebLogic Integration** マニュアル CD にあります。PDF を **Adobe Acrobat Reader** で開くと、マニュアルの全体（または一部分）を書籍の形式で印刷できます。

Adobe Acrobat Reader がない場合は、**Adobe** の Web サイト (<http://www.adobe.co.jp/>) で無料で入手できます。

サポート情報

WebLogic Integration のドキュメントに関するユーザからのフィードバックは弊社にとって非常に重要です。質問や意見などがあれば、電子メールで **docsupport-jp@bea.com** までお送りください。寄せられた意見については、**WebLogic Integration** のドキュメントを作成および改訂する **BEA** の専門の担当者が直に目を通します。

電子メールのメッセージには、**BEA WebLogic Integration** リリース 7.0 のドキュメントご使用の旨をお書き添えください。

本バージョンの **WebLogic Integration** について不明な点がある場合、または **WebLogic Integration** のインストールおよび動作に問題がある場合は、**BEA WebSUPPORT** (<http://websupport.bea.com/custsupp>) を通じて **BEA** カスタマサポートまでお問い合わせください。カスタマサポートへの連絡方法については、製品パッケージに同梱されているカスタマサポートカードにも記載されています。

カスタマ サポートでは以下の情報をお尋ねしますので、お問い合わせの際はあらかじめご用意ください。

- お名前、電子メールアドレス、電話番号、ファクス番号
- 会社の名前と住所
- お使いの機種とコード番号
- 製品の名前とバージョン
- 問題の状況と表示されるエラー メッセージの内容

表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
太字	用語集で定義されている用語を示す。
[Ctrl] + [Tab]	複数のキーを同時に押すことを示す。
<i>斜体</i>	強調または書籍のタイトルを示す。
等幅テキスト	コード サンプル、コマンドとそのオプション、データ構造体とそのメンバー、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。 <i>例</i> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>

表記法	適用
太字の等幅テキスト	コード内の重要な箇所を示す。 <i>例</i> <code>void commit ()</code>
斜体の等幅テキスト	コード内の変数を示す。 <i>例</i> <code>String <i>expr</i></code>
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 <i>例</i> <code>LPT1</code> <code>SIGNON</code> <code>OR</code>
{ }	構文の中で複数の選択肢を示す。実際には、この括弧は入力しない。
[]	構文の中で任意指定の項目を示す。実際には、この括弧は入力しない。 <i>例</i> <code>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</code>
	構文の中で相互に排他的な選択肢を区切る。実際には、この記号は入力しない。
...	コマンドラインで以下のいずれかを示す。 <ul style="list-style-type: none"> ■ 引数を複数回繰り返すことができる ■ 任意指定の引数が省略されている ■ パラメータや値などの情報を追加入力できる 実際には、この省略記号は入力しない。 <i>例</i> <code>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</code>
.	コード サンプルまたは構文で項目が省略されていることを示す。実際には、この省略記号は入力しない。



1 はじめに

以下の節では、統合ソリューションを設計するプロセスを紹介します。

- 統合の主要な設計概念
- 統合ソリューション設計のロール
- 主要な設計タスク

このマニュアルは、統合スペシャリストを対象とした、**WebLogic Integration** に基づくソリューションを設計するための手引き書です。このマニュアルでは、まずビジネスおよび技術的な要件のレベルで統合スペシャリストが確認する必要のある主要な **WebLogic Integration** の概念について説明し、続いてそれらの主要な概念を **WebLogic Integration** を使用して統合ソリューションのアーキテクチャ設計で実装する方法について説明します。

注意： 統合ソリューション全体では、統合ソリューションの手法、プロジェクト管理の手法、ビジネスおよび技術的な指定の手法など、このマニュアルの範囲から外れる多くの手法が関与します。このマニュアルは、統合ソリューション全体の中で、**WebLogic Integration** に基づく統合ソリューションの設計に直接関連する部分を取り扱うことでそれらの手法を補足します。

このマニュアルでは、ビジネス要件を指定する方法や統合を設計する方法の例として **WebLogic Integration** のサンプルアプリケーションを利用します。統合スペシャリストは、『*WebLogic Integration チュートリアル*』を読んでこのサンプルアプリケーションを理解しておく必要があります。

統合の主要な設計概念

以下の節では、主要な設計概念について説明します。

- ビジネス プロセス
- 統合ソリューション

- 設計目標
- 設計上の考慮事項
- ベスト プラクティスと設計パターン

ビジネス プロセス

ビジネス プロセスとは、始まりと終わりがあり、入力と出力が明確に定義されている、相互に関連したビジネス アクティビティのまとまりのことです。ビジネス プロセスには、開始イベント、参加アプリケーション、イベント定義、データ トランスフォーメーション、マッピング、プロセス定義など、特定の統合ソリューションに関連するすべての情報が含まれます。ビジネス プロセスは、販売注文から届け先住所を取得するような単純なものから、特定の品目についての価格と在庫の情報を複数のサプライヤから得るような複雑なものまでさまざまです。

統合ソリューション

統合ソリューションには、1 つまたは複数の関連するビジネス プロセスの統合が伴います。たとえばある企業で、仕入れ処理とサプライヤをよりよく統合する必要がありますとします。その統合ソリューションに含まれるビジネス プロセスは以下の 2 つが考えられます。

- サプライヤからの入札の受け付け
- 発注の処理

ビジネス プロセスは、以下のような要素に分割できます。

- 発注書の作成と承認
- 発注書の更新または編集
- 発注書の変更の処理
- 発注書の取り消し

統合ソリューションは、複数のシステム、地域、および組織をまたがる場合があります。たとえば、**Web**上で複数のサプライヤから入札を受け付ける場合は、バイヤの仕入れシステムとサプライヤの販売および在庫システムを統合する必要があります。

統合ソリューションの詳細については、2-2 ページの「統合ソリューションとビジネスプロセスの定義」および 2-17 ページの「統合ソリューションとビジネスプロセスの指定」を参照してください。

設計目標

設計目標とは、統合ソリューションの正否を判断する大まかな基準のことです。一般的な設計目標には以下のものがあります。

- 可用性—統合ソリューションは、ハードウェアやネットワークで障害が発生したときにはフェイルオーバーが機能し、可用性が十分に確保されていなければなりません。
- 信頼性—統合ソリューションでは、トランザクションの管理、データの処理と検証、およびデータ整合性の確保が確実に行われなければなりません。
- パフォーマンス—統合ソリューションでは、負荷のピーク時でもそれ以外のときでも十分なパフォーマンスが発揮されなければなりません。
- スケーラビリティ—統合ソリューションは、予期される負荷の増加に対応できるだけの十分なスケーラビリティがなければなりません。
- セキュリティ—統合ソリューションでは、無認可アクセスや改ざんからデータが十分に保護されなければなりません。

各統合ソリューションで、設計目標を確認し、優先順位を決定します。

WebLogic Integration は、それらの設計目標を達成するために必要なすべての機能を備えています。

設計上の考慮事項

設計上の考慮事項とは、統合ソリューションの設計段階で評価する必要のある事柄のことです。たとえば、サービス品質（パフォーマンス、可用性、信頼性、スケーラビリティ、応答時間、セキュリティ、ロギング、および監査）を定義およ

び解析し、統合ソリューションの設計で考慮に入れなければなりません。このマニュアルでは、**WebLogic Integration** を使用して構築されるほとんどすべての統合ソリューションに関連する設計上の考慮事項を示します。

ベスト プラクティスと設計パターン

ベスト プラクティスとは、特定の状況で最も効果的に機能することが証明されている手段のことです。たとえば、第3章「統合ソリューションの設計」の説明に従って統合ソリューションを設計する前に、第2章「統合ソリューションの要件の決定」で説明されているようにビジネスおよび技術的な要件を十分に定義しておくことが重要です。このマニュアルの目的は、**WebLogic Integration** のベスト プラクティスを説明し、さまざまな組織で **WebLogic Integration** の実装に成功してきたこれまでの経験を反映する実際的なアイデアで組織の統合ソリューション設計手法をサポートすることです。

設計パターンとは、高いパフォーマンスやデータの整合性といった特定のビジネス要件をサポートできるように統合ソリューションの全体または一部を最適に設計するための推奨事項のことです。たとえば、**BPM** ワークフローで呼び出しサービス アクティビティを使用してビジネス オペレーション（**Java** クラスまたは **EJB**）などの外部サービス呼び出す場合は、1つのサービスのみを呼び出すようにします。複数サービスの呼び出しは、複数のタスク ノードを使用して実装する必要があります（サービスの呼び出しごとに1つのタスク ノード）。タスク ノードの詳細については、『*WebLogic Integration Studio ユーザーズ ガイド*』の「アクションの定義」を参照してください。

統合ソリューション設計のロール

以下の節では、統合ソリューション設計チームのメンバーとして果たさなければならないロールについて説明します。

- 統合スペシャリスト
- ビジネスアナリスト
- アーキテクト
- エンタープライズ情報システム スペシャリスト

- システム管理者

統合ソリューションの設計を成功させるためには、これらすべての参加者からの入力が必要です。1人で複数のロールを受け持つこともできます。すべてのロールがすべての統合ソリューションで必要とされるわけではありません。

統合スペシャリスト

統合スペシャリストの仕事は、**WebLogic Integration** ソリューションの実装を先導し、設計作業を推進することです。統合スペシャリストは、**WebLogic Integration** 製品の特長と機能に精通しています。統合スペシャリストは、ビジネスアナリストおよびアーキテクトと相談して要件を決定し、それらの要件を**WebLogic Integration** の機能に対応付けて、統合ソリューションのアーキテクチャを設計します。統合スペシャリストは、以下の領域の専門知識を備えています。

- ビジネスと技術の解析
- アーキテクチャの設計
- プロジェクトの管理

ビジネスアナリスト

ビジネスアナリストは、組織のビジネスプロセス、手順、方針、ビジネスルール、およびリソースに関する専門知識を提供します。ビジネスアナリストは、事業に精通しており、以下の領域の専門知識を有しています。

- ビジネスの解析
- プロセスの設計とモデル化

アーキテクト

アーキテクトは、組織の情報技術インフラストラクチャ（電気通信、プラットフォーム、アプリケーション、データリポジトリ、将来技術、IT組織など）に関する専門知識を提供する技術のスペシャリストです。アーキテクトは、情報システムに精通しており、以下の領域の専門知識を有しています。

- 技術の解析
- システムアーキテクチャ
- アプリケーションの設計

エンタープライズ情報システムスペシャリスト

EIS スペシャリストは、統合されるエンタープライズ情報システム（EIS）の専門家です。EIS スペシャリストは、EIS を統合ソリューションに接続するために必要な情報を提供します。エンタープライズ情報システムは、通常は WebLogic Integration アダプタを通じて WebLogic Integration に接続されます。EIS スペシャリストは、統合される EIS システムのすべての側面（データフォーマット、動作、外部インタフェースなど）に精通しており、以下の領域の専門知識を有しています。

- 技術の解析
- アプリケーション統合ソリューションの設計

システム管理者

システム管理者は、組織にデプロイされているデータベースとアプリケーションに関する深い技術知識と運用知識を提供します。システム管理者は、デプロイメントトポロジに精通しており、以下の領域の専門知識を有しています。

- 性能と負荷の解析
- パフォーマンスの解析とチューニング
- サポートプランニング

主要な設計タスク

統合ソリューションを設計する場合、統合ソリューション設計チームは以下の基本的なタスクを行う必要があります。

1. ビジネスおよび技術的な要件の徹底的な解析に基づいて統合ソリューションの要件を定義します。詳細については、第2章「統合ソリューションの要件の決定」を参照してください。
2. ビジネスおよび技術的な要件を徹底的に解析し、それらの要件を満たすのに最適な **WebLogic Integration** の機能を理解して、統合ソリューションのアーキテクチャを設計します。詳細については、第3章「統合ソリューションの設計」を参照してください。

統合ソリューションの設計が終わったら、統合スペシャリストはプログラマ、EIS スペシャリスト、およびシステム管理者と共同で **WebLogic Integration** ソリューションを構築します。詳細については、**WebLogic Integration** 製品マニュアルの **Web** サイトの「設計とデプロイメント」を参照してください。

2 統合ソリューションの要件の決定

注意： cXML ビジネス プロトコルは、**WebLogic Integration** の本リリースより非推奨になりました。代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

統合スペシャリストは、統合ソリューションのビジネスおよび技術的な要件を調べる必要があります。統合スペシャリストは、ビジネス アナリストおよびアーキテクトと共同で要件を明確にし、組織のニーズと優先順位について意見の一致を計ります。ビジネス アナリストとアーキテクトは、詳細な要件の定義を提供します。統合スペシャリストは、その定義を使用して **WebLogic Integration** で統合ソリューションのアーキテクチャを設計します。

以下の節では、統合ソリューションの要件を決める際の考慮事項について説明します。最後の節では、それらの事項がサンプルの **WebLogic Integration** アプリケーションとどのように関連するのかを示します。

- 統合ソリューションとビジネス プロセスの定義
- ビジネス プロセスのアクタとそのロールの定義
- ビジネス イベントの定義
- ビジネス データ フローの定義
- サービス品質の定義
- 統合ソリューションのトポロジの定義
- サンプル **WebLogic Integration** アプリケーションの要件の指定

統合ソリューションとビジネス プロセスの定義

統合ソリューションの要件を決めるときには、統合スペシャリストは以下のことを行わなければなりません。

- 統合ソリューションの範囲を明確にし、その作業を小さな別々の単位に分割します。統合ソリューションは、1つまたは複数の関連するビジネスプロセスで構成されます（1-2 ページの「統合ソリューション」を参照）。
- 統合ソリューションを構成するビジネスプロセスを識別します。ビジネスプロセスは、始まりと終わりがあり、入力と出力が明確に定義されている相互に関連したビジネス アクティビティのまとまりです（1-2 ページの「ビジネス プロセス」を参照）。

これらのタスクがサンプルの **WebLogic Integration** アプリケーションでどのように行われるのかを確認するには、2-17 ページの「統合ソリューションとビジネス プロセスの指定」を参照してください。

各ビジネス プロセスは、必要なビジネス アクティビティを記述する単一のプロセス定義で構成されている必要があります。統合スペシャリスト、ビジネスアナリスト、およびアーキテクトは共同で各ビジネス プロセスを定義します。統合スペシャリストは、以下のような事項を確認する必要があります。

- 各ビジネス プロセスでどの作業を行わなければならないのか
- 各ビジネス プロセスの主要なアクティビティは何か
- 各ビジネス プロセスまたはコンポーネントの特性は何か
 - シーケンスまたは順序（他のタスクやアクティビティとの関連で）
 - 頻度
 - 緊急性と適時性
 - 期間（経過時間）
 - ボリューム レンジ（平均、ピーク時、および小康時）

このタスクがサンプルの **WebLogic Integration** アプリケーションでどのように行われるのかを確認するには、2-17 ページの「統合ソリューションとビジネスプロセスの指定」を参照してください。

ビジネス プロセスのアクタとそのロールの定義

アクタとは、ビジネス イベントを生成および消費する利用者およびプロセスのことです。ビジネス プロセスでは、アクタ同士の関係、プロセスにおけるアクタのロール、およびプロセスとアクタによって交換されるビジネス イベントを定義します。統合スペシャリストは、以下の節で説明されているようにアクタとそのロールを識別します。

- アクタの解析
- ロールの解析
- タイプによるアクタの分類

このタスクがサンプルの **WebLogic Integration** アプリケーションでどのように行われるのかを確認するには、2-18 ページの「アクタとロールの指定」を参照してください。

アクタの解析

各ビジネス プロセスについて、統合スペシャリストは以下のような事項を確認します。

- 各ビジネス プロセスのアクタは何か。誰が関与するのか。たとえば、社内の他の部署、ベンダ、顧客、システム、および **Web** ユーザが関与するのか
- 前述のエンティティの地理的なトポロジはどのようになっているのか。すべて同じ場所に存在するのか、それとも分散しているのか。どのエンティティがどの場所に存在するのか
- データがアクタ間でどのように転送されるのか

- 1つの場所で（LAN またはドメイン内で）転送されるのか
- WAN 経由で同じ組織内の他の場所に転送されるのか
- WAN の範囲外で、ファイアウォールを介してアクタに転送されるのか
- 各ビジネスプロセスの各エンティティを出入りするイベントはどのようなフォーマットなのか

ロールの解析

ビジネスプロセスの各エンティティのロールを定義するには、統合スペシャリストは以下のような事項を確認する必要があります。

- どのようなロールがビジネスプロセスに関与するのか
- 各ロールの作業はどのようなシーケンスで行われるのか
- 各ロールの作業はビジネスプロセスにどのように組み込まれるのか。具体的には、特定のロールの遂行者は以下のことを行うのか
 - プロセスを開始するのか
 - プロセスにサービスを提供するのか
 - 応答を返すことなくプロセスで生成されたビジネス イベントを消費するのか
 - 途中で決定を検討、承認、または拒否するのか、あるいは決断を行うのか
- 各ロールに関わる人数とプロセス数はどれくらいか

タイプによるアクタの分類

統合スペシャリストは、ビジネスプロセスに関与するさまざまなタイプのアクタを識別および記述する必要があります。ビジネスプロセスのアクタには、**WebLogic Integration Worklist** などのクライアント アプリケーションを使用する人間とさまざまなソフトウェア エンティティの両方が含まれます。

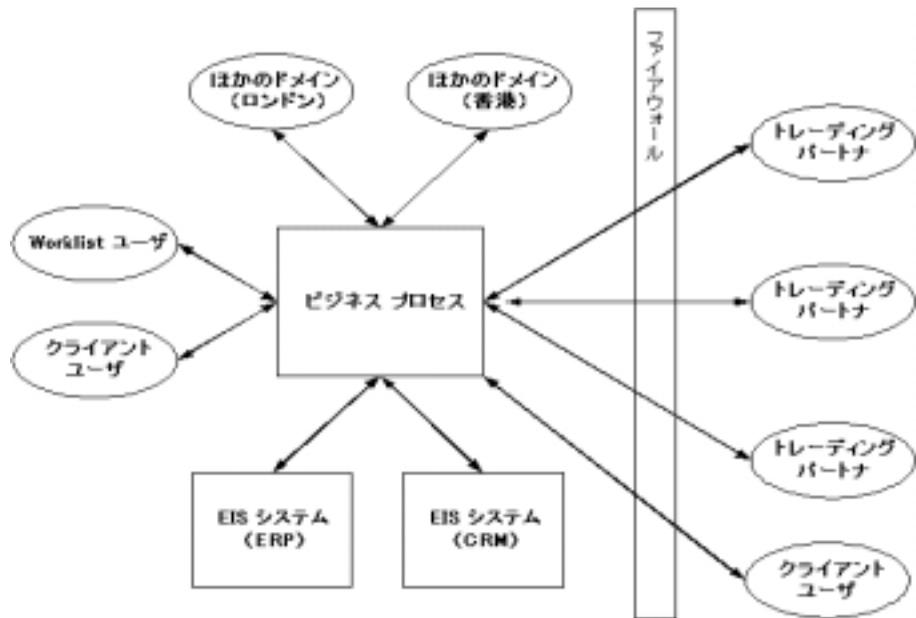
注意： Worklist クライアントは WebLogic Integration リリース 7.0 より非推奨となりました。代替機能に関する詳細については、『WebLogic Integration リリース ノート』を参照してください。

以下の節では、さまざまなタイプのアクタについて説明します。

- 人間ユーザ
- アプリケーション統合—アプリケーション統合で使用するエンタープライズ情報システム (EIS) アプリケーション
- 企業内の他のドメイン
- B2B 統合を実現する外部トレーディング パートナ

次の図は、これらのアクタを示しています。

図 2-1 ビジネス プロセスのアクタ



人間ユーザ

人間ユーザには以下の2つのカテゴリがあります。

- **Worklist ユーザ**には、ビジネスプロセスによって直接作業が割り当てられます。たとえば、**Worklist**を使用するマネージャには、100,000ドルを超えるすべての注文を承認するタスクが割り当てられます。ユーザは、ルールに基づいてグループ化できます。タスクは、個々のユーザまたはルールに関連付けられているユーザに割り当てることができます。このようにタスクの割り当てられたユーザは、ビジネスプロセスと密接に関わります。

注意： **Worklist** クライアントは **WebLogic Integration** リリース 7.0 より非推奨となりました。代替機能に関する詳細については、『**WebLogic Integration** リリースノート』を参照してください。

- **クライアント ユーザ**は、プロセスの一部であるアプリケーションを使用してビジネスプロセスと間接的に会話をします。たとえば、このタイプのユーザは **Web** ベースの **GUI** から **注文作成** プロセスを開始し、電子メールで注文確認を受信します。

統合スペシャリストは、各ユーザが必要とするクライアントのタイプ（**Swing GUI**、**HTML**、電子メールなど）を識別する必要があります。

アプリケーション統合—アプリケーション統合で使用するエンタープライズ情報システム（EIS）アプリケーション

組織の技術インフラストラクチャは、従来のメインフレームシステム、クライアント/サーバアプリケーション、パッケージアプリケーション（**ERP** や **CRM**）など、さまざまなエンタープライズ情報システム（**EIS**）で構成されている場合があります。

統合スペシャリストは、統合ソリューションに直接関わる **EIS** アプリケーションについて、以下のような事項を確認します。

- 統合ソリューションは **EIS** と直接接続されるのか
- 統合をサポートするように **EIS** をカスタマイズできるのか
- **EIS** とビジネスプロセスはイベントを交換するのか

統合スペシャリストは、ビジネスプロセスに関与し、かつ上記の質問が「イエス」になる **EIS** アプリケーションを識別する必要があります。

さらに、統合スペシャリストは以下の事項を決める必要があります。

- 各 EIS アプリケーションについて、それが提供するビジネス機能（発注管理、在庫管理など）
- ビジネス プロセスに参加する各 EIS アプリケーションについて、アプリケーションの細部とインタフェース技術

EIS アプリケーションの例

EIS アプリケーションとしては以下のものがあります。

- パッケージアプリケーション（電子メールなど）
- カスタム アプリケーション（WebLogic Server アプリケーションやそれ以外のアプリケーション）
- ミドルウェア技術を使用するアプリケーション（CORBA や MQSeries など）
- キュー ベースのインタフェースを使用するアプリケーション（MQSeries など）
- 技術アダプタ（ファイルまたはデータベースのアダプタ）

EIS 統合要件の解析

統合スペシャリストは、以下のような事項を確認してアプリケーションの要件を評価します。

- EIS の名前とバージョン
- EIS が動作するプラットフォーム
- EIS の提供するインタフェース技術は何か。EIS は API を提供するのか。EIS にファイルベースのインタフェースはあるのか。EIS でミドルウェアベースのインタフェースが使用されるのか
- 統合に関与する EIS 内部のビジネス プロセスで特定の EIS インタフェースの使用が要求されるのか
- インタフェースによって渡されるデータのフォーマットは何か（バイナリ、XML など）。メタデータ（データを説明するデータ）は利用可能か、あるいはメタデータを手動で作成する必要があるのか

2 統合ソリューションの要件の決定

- ビジネスプロセスと EIS アプリケーションの間で交換される各ビジネス イベントについて、使用される EIS インタフェースは何か（API、イベント、データベース テーブルなど）
- 人間とアプリケーションのインタフェースは何か
- インタフェース間のマッピングは何か

追加のカスタマイズが必要かどうかの判断

EIS アプリケーションには、統合ソリューションのサポートに必要なすべての機能を実装していないものもあり、したがって内部のカスタマイズが必要な場合があります。統合スペシャリストは、カスタマイズが必要かどうかを判断し、必要な場合は技術的な要件を定義する必要があります。

企業内の他のドメイン

統合ソリューションでは、リモート オフィスなど、企業内の他のローカル管理ドメインにアクセスする場合があります。その場合、他のドメインの内部構造に統合ソリューションの管理は及びません。具体的に言うと、その他方のドメインでは **WebLogic Integration** が統合ソリューションとして使用される場合も、使用されない場合もあります。

ビジネスプロセスと他のドメインのインタフェースは、それらの間で交換されるビジネス イベントのセットです。このインタフェースは定義する必要があります。

1 つの方法は、**B2B 統合機能**を使用して情報を交換できるようにし、独立したドメインを企業内部のトレーディング パートナとして表すことです。たとえば、ある企業がマレーシアに工場を持ち、その工場から日本にあるその企業の全額出資の組み立て工場に部品が供給されているとします。**B2B 統合**を利用すると、マレーシアのオフィスは日本のオフィス（バイヤ）に対する部品のサプライヤ（セラー）として機能できます。

B2B 統合を実現する外部トレーディング パートナ

統合ソリューションには、ビジネスプロセスが通信する必要のある外部トレーディング パートナが関与する場合があります。そのようなプロセスは企業のファイアウォールを経由します。トレーディング パートナの統合では、統合スペシャリストは以下の事項を定義する必要があります。

- B2B 統合の性質
 - バリューチェーン統合かどうか。バリューチェーン統合の場合は、サプライチェーン統合（トレーディング パートナがサプライヤ）、デマンドチェーン統合（トレーディング パートナが顧客）、またはそれらの組み合わせのどれであるのか
 - B2B トランザクションにトレーディング パートナを参加させることができる B2B 交換かどうか
 - 統合ソリューションに含まれなければならない必須のプロトコルがあるかどうか（XOCP、RosettaNet、cXML、または EDI）
- 注意：** XOCP および cXML ビジネス プロトコルは、WebLogic Integration の本リリースより非推奨になりました。代替機能に関する情報については、『*BEA WebLogic Integration リリース ノート*』を参照してください。

ビジネス イベントの定義

以下の節では、ビジネス イベントの定義方法について説明します。

- ビジネス イベントについて
- ビジネス イベントの特性の定義
- ビジネス イベントのメッセージフォーマットの定義
- ビジネス イベントの指定

このタスクがサンプルの WebLogic Integration アプリケーションでどのように行われるのかを確認するには、2-20 ページの「ビジネス イベントの指定」を参照してください。

ビジネス イベントについて

ビジネス イベントとは、ビジネス プロセスにおいてアクタ間で発生するメッセージまたはタスクの交換のことです。ビジネス イベントは、ビジネス アクティビティが発生し、それを遂行する必要があることを示します。たとえば、EIS では新規顧客作成 イベントをパブリッシュでき、注文管理アプリケーションでは新規注文 イベントをサブスクライブして新規注文を処理できます。各アクタは、ビジネス プロセスとビジネス イベントの交換を行えます。各ビジネス イベントには、ビジネス プロセスでそのイベントをユニークに識別する説明的な名前を付ける必要があります。

ビジネス イベントには、ビジネス データが格納されます。たとえば、新規顧客作成 イベントが顧客管理アプリケーションの実行時に発生したとします。新規顧客を作成する過程で、アプリケーションはビジネス プロセスの別のアクタから顧客の名前とアドレスを受信し、顧客の番号をそのアクタに返すことができます。

ビジネス イベントの特性の定義

アクタとプロセスの各会話について、統合スペシャリストは以下の事項を定義する必要があります。

- 送信および受信できるビジネス イベント
- ビジネス イベントの要素のユニークな特性

ビジネス イベントのメッセージ フォーマットの定義

WebLogic Integration で、ビジネス イベントは XML メッセージまたはバイナリメッセージとして送信されます。統合スペシャリストは、以下の事項を定義する必要があります。

- ビジネス イベントのフォーマット (XML またはバイナリ) とメタデータ。
できる限り、一般的なフォーマット (XML など) を使用します。EIS との

アプリケーション統合では、専用の EIS プロトコルまたはメッセージフォーマットがアダプタによって一般的なフォーマットに変換されます。

- 条件処理、ルーティング、トリガなどの決定で使用するイベントの情報。

ビジネス データ フローの定義

ビジネスプロセスでは、アクタ間のデータフローを定義します。このデータフローに基づいて、統合スペシャリストはビジネスデータをどのように操作する必要があるのかを判断できます。たとえば、場合によっては、ビジネスデータは複数のイベントに分割するか、複数のイベントからつなぎ合わせるか、またはフォーマットを変換する必要があります。

このデータフローでは、処理上の決定を行うためにビジネスプロセスで使用するビジネスデータも定義する必要があります。たとえば、5,000ドルを超える発注は副社長の承認を得なければならないというアルゴリズムがビジネスプロセスに含まれている場合は、各発注の金額を必須のビジネスデータとして定義しなければなりません。

以下の節では、データフローの要件を作成する作業について説明します。

- データフローの要件の定義
- データフローの解析
- データフローの要件の指定

このタスクがサンプルの **WebLogic Integration** アプリケーションでどのように行われるのかを確認するには、2-20 ページの「データフローの要件の指定」を参照してください。

データフローの要件の定義

各データフローについて、統合スペシャリストは以下の要件を定義する必要があります。

- 条件データ処理上の決定を行うときに必要なデータです。このデータはビジネスイベントから抽出します。

- ビジネスルール（アプリケーションルール）ープロセスの実行時の実行パスを決めるために条件データに適用される処理ルールです。
- マッピングー入力として使用されるビジネス イベントと出力として使用されるビジネス イベントの間のデータ トランスフォーメーションです。
- ビジネス トランザクションープロセスのトランザクションの境界です。1つのプロセスには、多くのビジネス トランザクションが含まれている場合があります。統合スペシャリストは、各ビジネス トランザクションについて、トランザクションをロールバックする必要がある場合に実行しなければならない調整アクションも定義する必要があります。
- エラー処理ーどのような例外が発生し、それらの例外をどのように処理するのかということです。

データ フローの解析

統合スペシャリストは、以下のような事項を確認してデータ フローの技術的な側面を解析する必要があります。

- 各データ要素の特性
- メッセージの特性
 - メッセージのサイズ（最小、最大、および平均）
 - メッセージの量（ピーク時、小康時、および平均のメッセージ数と周期的なパターン）
 - 単一メッセージか、それとも一括（集約）メッセージか。複数のメッセージが集約されている場合は、それらを適切に分割してルーティングする必要があるのか。その場合はどのようなルーティングの基準または条件があるのか
- ソース データと対象のデータの間でどのようなデータ トランスフォーメーションが必要なのか

たとえば、注文管理アプリケーションから新規注文イベントが処理を目的として出荷アプリケーションに送信されるとします。また、出荷アプリケーションは3つの別々の地域のオフィス（東部、中部、および西部）で別々のインスタンスとして動作しているとします。この場合、注文管理アプリケーションでは、注文に

ある出荷先住所に基づいて適切なアプリケーション インスタンスに通知する必要があります。さらに、注文管理アプリケーションでは請求処理アプリケーションに新規注文を通知することも必要です。

この場合、データ フローの要件は次のようになります。

- 3つのビジネス イベント
 - 注文管理アプリケーションからの *新規注文 イベント*
 - 出荷アプリケーションへの *商品出荷 イベント*
 - 請求処理アプリケーションへの *インボイス送付 イベント*
- 条件データー注文の商品が出荷される州です。この情報は、新規注文イベントの請求先住所から抽出されます。
- ビジネス ルールー *商品出荷 イベント*を受信する出荷アプリケーション インスタンスの識別に使用するルールです。注文にある出荷先の州 / 地域と各アプリケーション インスタンスに関連付けられている州 / 地域のリストに基づきます。
- マッピングー以下のようなデータ トランスフォーメーション マッピングです。
 - *新規注文 イベント*から *商品出荷 イベント*
 - *新規注文 イベント*から *インボイス送付 イベント*
- ビジネス トランザクションー出荷アプリケーションと請求処理アプリケーションの両方が正常に更新されるか、またはどちらも更新されません。(他方が失敗した場合に) 各アプリケーションをロールバックするための調整アクションを定義する必要があります。
- エラー処理ーデータ エラーが発生した場合 (受信した注文の州 / 地域が無効である場合など) の処理です。

サービス品質の定義

統合スペシャリストは、以下のサービス品質特性をビジネスおよび技術的な観点で定義する必要があります。以下の節では、それらの特性について説明します。

- パフォーマンス

- 可用性と信頼性
- 応答時間
- セキュリティ
- スケーラビリティ
- ロギングと否認防止性

パフォーマンス

統合スペシャリストは、以下のような事項を確認する必要があります。

- ビジネスの観点で、ビジネスプロセスをどのくらい速く実行しなければならないか
- ピーク時およびピーク以外のときのパフォーマンスの要件

たとえば、統合スペシャリストは最大の注文負荷を **5,000** としてシステムで **1** 営業日（9時から17時）に **20,000** の注文を処理できるように指定できます。

WebLogic Integration のパフォーマンスの詳細については、3-31 ページの「統合設計でのパフォーマンスに関する考慮事項」を参照してください。大規模な統合ソリューションでは、パフォーマンスの要件を満たすためにクラスタ構成が必要な場合もあります。クラスタの詳細については、『**WebLogic Integration** ソリューションのデプロイメント』の「**WebLogic Integration** クラスタのコンフィグレーション」を参照してください。

可用性と信頼性

統合スペシャリストは、以下のような事項を確認してビジネスプロセスの可用性の要件を定義する必要があります。

- システムが利用できなければならない時間帯。毎日 24 時間必要とされるのか
- スケジューリングしたシステムの中断時間の予定の期間とスケジュールとは関係ない中断時間の予測される期間
- 許容できる最大の中断時間

- ハードウェアやネットワークで障害が発生した場合にどのようなフェイルオーバーおよび回復機能が必要なのか
- ビジネス メッセージを永続化し、システムの再起動時に回復する必要があるのか

一部のシステムではフェイルオーバーが即時に機能して毎日 24 時間の可用性が必要とされ、その他のシステムでは可用性は営業時間内のみ必要とされ、営業終了後や週末にメンテナンスがスケジューリングされます。高可用性の統合ソリューションでは、信頼性の要件を満たすためにクラスタ構成が必要となります。クラスタ構成の詳細については、『*WebLogic Integration ソリューションのデプロイメント*』の「クラスタ デプロイメントのコンフィグレーション」を参照してください。

応答時間

統合スペシャリストは、ビジネス プロセスの応答時間の要件を定義する必要があります。たとえばサプライヤは、24 時間以内に入札要求に応答する必要がある場合や、60 分以内に落札の通知を受ける必要がある場合があります。応答時間の限界に達すると、イベントはタイムアウトになります。

セキュリティ

統合スペシャリストは、以下のような事項を確認してビジネス プロセスのセキュリティの要件を定義する必要があります。

- ビジネス プロセスの機密レベル
- 各ロールで必要とされるプライバシー
- 現時点で備わっているセキュリティ セーフガード

スケーラビリティ

統合スペシャリストは、現在の仕事量と将来予想される仕事量に基づいてビジネス プロセスのスケーラビリティの要件を定義する必要があります。たとえば注文処理の統合ソリューションでは、サービスや追加のアプリケーション開発を妨

げることなく、2ヶ月以内に能力の3倍の注文数を処理できるようにしなければならぬ場合があります。スケーラビリティの要件を満たすために、統合ソリューションではクラスタ構成を利用できます。クラスタ構成の詳細については、『*WebLogic Integration ソリューションのデプロイメント*』の「WebLogic Integration クラスタについて」を参照してください。

ロギングと否認防止性

統合スペシャリストは、以下のような事項を確認してビジネスプロセスのシステム ロギングの要件を定義する必要があります。

- どのような問題が生じる恐れがあるのか
- どの情報でロギングおよびモニタを行う必要があるのか
- B2B 統合を利用する統合ソリューションについて、監査または否認防止を目的としてどの情報でロギングと管理を行う必要があるのか

否認防止性とは、トレーディング パートナ間で特定のビジネス メッセージの送受信が行われたこと、または行われなかったことを証明する能力のことです。重要なビジネス メッセージについては、送信および受信の否認防止性を備えることが法律によって義務付けられている場合があります。

統合ソリューションのトポロジの定義

統合スペシャリストは、統合環境のさまざまなエンティティの物理的な位置を指定する統合トポロジを定義する必要があります。このトポロジでは、エンティティ間のネットワーク接続のタイプ（LAN、WAN、インターネット、ダイアルアップなど）についての情報も必要です。

統合スペシャリストは、統合ソリューションのすべてのエンティティが以下の形態で配置されるのかを確認する必要があります。

- 物理的に1つの場所（単ドメインの LAN）
- ファイアウォールの内側の、WAN で接続された複数の場所
- ファイアウォールが介在する、さまざまなトレーディング パートナによる複数の場所

サンプル WebLogic Integration アプリケーションの要件の指定

以下の節では、サンプルの WebLogic Integration アプリケーションで要件がどのように定義されるのかを説明します。

- 統合ソリューションとビジネス プロセスの指定
- アクタとロールの指定
- ビジネス イベントの指定
- データ フローの要件の指定
- サービス品質の指定
- 統合トポロジの指定

統合ソリューションとビジネス プロセスの指定

サンプルの WebLogic Integration アプリケーションでは、General Control Systems (GCS) がサプライヤとのバリュー チェーンを管理しやすくするために単一の統合ソリューション（仕入れの統合）を定義します。同社の解析によると、以下の 2 つのビジネス プロセスで統合要件を定義する必要があります。

- サプライヤの入札を募集し（つまり価格と在庫をサプライヤに照会する）、入札額が最高のサプライヤに注文を与える
- 選ばれたサプライヤに発注書を発行し、販売注文の確認書を受け取る

これら 2 つのプロセスにはつながりがあります（GCS がサプライヤを選んだ後、そのサプライヤに対して発注書が発行される）。これらのプロセスは、今後見込まれる販売または実際の販売によって **EnergyMiser 76** 製品の増産が必要になったときに開始されます。増産が要求されれば、さらに多くの部品を調達する必要性が高まります。

これらのビジネス プロセスの特性はサンプルのシナリオでは定義されていませんが、以下のような要素を解析することが望まれます。

- 部品を注文する頻度（個々の部品とすべての部品について）
- サプライヤの選択に要する時間
- 発行された発注書の総数
- 注文数の季節変動

価格と在庫の照会手順

サプライヤを選ぶビジネス プロセスは次の手順で行われます。

1. バイヤが特定品目の価格と在庫についての情報をサプライヤに要求します。
2. サプライヤが価格と在庫の情報をバイヤに提供します。
3. バイヤがサプライヤの応答を取りまとめて、発注マネージャに提出します。
4. 発注マネージャが応答を検討し、サプライヤを選びます。

発注書の発行手順

選ばれたサプライヤに発注書を発行するビジネス プロセスは次の手順で行われます。

1. 発注マネージャが発注書を発行し、選ばれたサプライヤに通知します。
2. 発注書が社内のエンタープライズ リソース プランニング（ERP）システムに自動的に入力されます。
3. 選ばれたサプライヤがバイヤに発注確認書を返します。
4. 発注確認書の情報（サプライヤの販売注文番号）で発注記録が自動的に更新されます。

アクタとロールの指定

サンプルの **WebLogic Integration** アプリケーションでは、両方のビジネス プロセスでバイヤおよびサプライヤ（セラー）のロールを果たすトレーディング パートナが関与します。さらに、発注ビジネス プロセスでは **EIS** との統合が伴います。

アクタのタイプ

GCS のシナリオには、人間のアクタとソフトウェアのアクタが存在します。

- 人間のアクタ
 - 価格と在庫の要求をサプライヤに発行する GCS の仕入れ担当者
 - 価格と在庫の情報で要求に応答するサプライヤの販売担当者
 - サプライヤを選び、プロセスを終わらせる GCS の発注マネージャ
- ソフトウェアのアクタ : ERP システム（発注情報を管理するエンタープライズ情報システム）

サプライヤの選択におけるアクタとロール

GCS は、以下のタスクを行うバイヤのロールを果たします。

- 価格と在庫の情報を要求する
- サプライヤからの応答を検討する
- サプライヤを選んでプロセスを終了する

各サプライヤは、以下のタスクを行うセラー候補のロールを果たします。

- バイヤから要求を受け取る
- 要求された品目について価格を設定し、在庫を確認する
- その情報をバイヤに送信する

発注書の発行におけるアクタとロール

GCS は、以下のタスクを行うバイヤのロールを果たします。

- 選ばれたサプライヤに通知する
- 発注書を発行する
- 発注書を ERP システムの発注モジュールに入力する
- サプライヤから確認を受け取る

GCS の ERP システムは、発注の情報マネージャとしてのロールを果たします。その ERP システムでは、発注書の作成とサプライヤからの確認書の受信が処理されます。

選ばれたサプライヤは、以下のタスクを行うセラーのロールを果たします。

- バイヤから発注確認書の要求を受け取る
- 発注確認と補足情報（サプライヤの販売注文番号など）をバイヤに提供する

ビジネス イベントの指定

サンプルの **WebLogic Integration** アプリケーションでは、ビジネス イベントが両方のビジネスプロセスに関連付けられています。

サプライヤ選択プロセスのイベント

サプライヤの選択プロセスには、以下のイベントがあります。

- バイヤが価格と在庫の情報をセラーに要求する
- セラーが価格と在庫の応答をバイヤに送信する
- バイヤがセラーの応答を取りまとめ、セラーを選ぶ

発注プロセスのイベント

発注プロセスには、以下のイベントがあります。

- バイヤが新しい発注を要求する
- バイヤが発注確認書を選ばれたセラーに要求する
- セラーが確認書をバイヤに送信する

データ フローの要件の指定

サンプルの **WebLogic Integration** アプリケーションには、以下の主要なデータ フローがあります。

- 価格と在庫の要求
- 価格と在庫の応答
- 集約された応答
- 発注の要求
- 発注
- 発注確認

これらのデータフローについての特定の情報（予想されるデータ量、ピーク時と小康時のデータ量、周期的なパターンなど）は、サンプルの WebLogic Integration アプリケーションでは定義されていません。

要件のリスト

上記のデータフローには、以下の要件があります。

- ビジネス イベント – 2-20 ページの「ビジネス イベントの指定」を参照。
- 条件データ
 - バイヤーが価格と在庫の要求を送信するサプライヤのリスト
 - バイヤーが選択し、発注書を発行するサプライヤ
- ビジネス ルール – すべてのサプライヤ候補から受信した応答の比較に基づいてサプライヤを選択するためのルール
- マッピング – データを変換するプロセス
 - 発注要求を発注マネージャから ERP システムにマッピングする必要があります。
 - 発注確認を選ばれたセラーから ERP システムにマッピングする必要があります。
- ビジネス トランザクション
 - 指定の時間内にサプライヤから応答がない場合は、価格と在庫の要求トランザクション全体をロールバックできます。
 - 選ばれたセラーから発注の確認が届かない場合は、発注トランザクション全体をロールバックできます。

注意： これらの条件でのトランザクションのロールバックは、サンプルの WebLogic Integration アプリケーションでは実装されていません。

価格と在庫の要求プロセスのデータ フロー

価格と在庫の要求のデータ フローには、以下の情報が含まれます。

- ユニークな要求識別子、および要求の日付と時刻
- 品目情報（ユニークな部品 ID、要求量、納期限の日付と時刻、および希望単価）
- 出荷元の希望所在地
- 備考

価格と在庫の応答データ フローには、以下の情報が含まれます。

- ユニークな応答識別子、および応答の日付と時刻
- 関連する要求の識別子
- サプライヤの情報
- 品目の情報（ユニークな部品 ID、在庫量、在庫の日付、単価）
- 出荷元の所在地
- 備考

集約された応答のデータ フローには価格と在庫の応答のすべての情報が含まれますが、その情報は共通の要求識別子と要求の日付に基づいて収集されます。

発注プロセスのデータ フロー

発注要求のデータ フローには、以下の情報が含まれます。

- サプライヤの名前
- 見積もり識別子
- 品目の情報（部品番号、量、単価、出荷量、出荷日）

発注のデータ フローには、以下の情報が含まれます。

- 発注ヘッダ（発注番号、発行日、ステータス、バイヤの連絡先、サプライヤの情報、請求情報、出荷情報、金融機関情報、および総量）
- 発注の細目（ライン番号、部品番号、部品名、量、単価、出荷日、および備考）

発注確認のデータフローには、発注のすべての情報に加えて販売注文の番号と日付が含まれます。

サービス品質の指定

サンプルの WebLogic Integration アプリケーションには、以下の項目のサービス品質（QoS）要件があります。

- パフォーマンス
- 可用性と信頼性
- 応答時間
- セキュリティ
- スケーラビリティ
- ロギングと否認防止性

パフォーマンス

サンプル WebLogic Integration アプリケーションのシナリオにはパフォーマンスの要件、トランザクションの量、またはピーク時の負荷が含まれていませんが、送信する価格と在庫の照会の数や受信するサプライヤの応答の数に関係なく、常に許容レベルのパフォーマンスを提供するための統合ソリューションが GCS で必要とされることは十分に推測できます。

可用性と信頼性

サンプルの WebLogic Integration アプリケーションに可用性と信頼性の要件はありませんが、統合ソリューションで以下のことを実現しなければならないことは明らかです。

- 営業時間（午前 8 時から午後 6 時まで）の間動作を続ける

- 営業時間後もオフライン処理のために利用することができる
- 価格と在庫の要求および発注を永続化し、システムが停止しても再起動後に処理を続ける
- 最大の中断時間（2時間など）に達するまでに再起動する

応答時間

サンプルのシナリオでは応答時間の要件が示されていませんが、ERPアプリケーションで新しい発注が作成されたときには、発注確認の要求を直ちに（たとえば10秒以内に）送信しなければなりません。

セキュリティ

GCSには、以下のセキュリティ要件があります。

- 価格と在庫の要求は承認済みのサプライヤのみに送信する
- サプライヤからの価格と在庫の応答は機密情報であり、サプライヤが競争相手の入札について何かを知ることにはあってはならない
- ERPシステムは、データの整合性を確保し、無認可のアクセスを防止するために独自のセキュリティメカニズムを備えている。統合には、ERPシステムで発注を作成および更新するための適切なアクセス権がなければならない

スケーラビリティ

クラスタ環境でWebLogic Integrationを使用すると、GCSではシステムを拡張し、多数のサプライヤおよび大幅に増加した入札募集と発注を処理することができます。

ロギングと否認防止性

発注と発注確認はバイヤとセラーの適法契約を示すものなので、それらは否認防止性の基準に従っていなければなりません。

統合トポロジの指定

サンプル WebLogic Integration アプリケーションの両方のビジネスプロセスでは、GCS とそのトレーディング パートナの間で GCS のファイアウォールを介して通信が行われます。また、GCS のファイアウォールの内側では、ERP システムとの統合のために、統合ソリューションは ERP システムにネットワーク アクセスできなければなりません。

3 統合ソリューションの設計

統合スペシャリストは、要件定義の段階で収集したデータに基づいて **WebLogic Integration** 環境の統合ソリューションを設計します。統合スペシャリストは、ビジネスおよび技術的な要件を **WebLogic Integration** のアーキテクチャにマッピングする方法と、利用可能な **WebLogic Integration** のアーキテクチャと機能を最大限に利用する統合ソリューションの設計方法を知らなければなりません。

以下の節では、統合プロジェクトのソリューションを設計するときに考慮する事項について説明します。最後の節では、それらの事項がどのようにサンプル **WebLogic Integration** アプリケーションの高度な設計の指定と関連するのかが示します。

- **WebLogic Integration** のアーキテクチャについて
- 統合ソリューション トポロジの **WebLogic Integration** へのマッピング
- ビジネス プロセスの実装
- データ トランスフォーメーションの定義
- 詳細な設計の作成
- サンプル **WebLogic Integration** アプリケーションの高度な設計の指定

WebLogic Integration のアーキテクチャについて

統合ソリューションを設計する前に、統合スペシャリストは WebLogic Integration のアーキテクチャを理解する必要があります。WebLogic Integration アーキテクチャの概要については、『*WebLogic Integration 入門*』の「E ビジネス統合への道」を参照してください。

特に、以下の WebLogic Integration コンポーネントを理解することが重要です。

表 3-1 WebLogic Integration アーキテクチャの主要なコンポーネント

コンポーネント	説明
Application Integration	<p>以下のタイプのアダプタを使用して WebLogic Integration と EIS アプリケーションを統合する。</p> <ul style="list-style-type: none"> ■ サービス アダプタは、WebLogic Integration から EIS アプリケーションへの要求と応答の同期統合を実現する ■ イベント アダプタは、EIS アプリケーションから WebLogic Integration への一方向の非同期統合を実現する <p>Application Integration アダプタは、Web ブラウザを使用してコンフィグレーションする。Application Integration アダプタを経由するビジネス イベントはすべて、WebLogic Integration リポジトリの XML スキーマ定義 (XSD) として定義する。Application Integration の機能は BPM プラグインを通じて BPM と直接統合されるので、Application Integration サービス アダプタにイベントを送信し、Application Integration イベント アダプタで生成されるイベントを消費するプロセスを簡単に定義することができる。詳細については、『<i>WebLogic Integration 入門</i>』の「Application Integration」を参照してください。</p>
Business Process Management (BPM)	<p>ビジネス プロセスの定義と実行をサポートする。WebLogic Integration Studio を使用すると、ユーザはグラフィカルにプロセスを定義し、実行時に実行をモニタできる。詳細については、『<i>WebLogic Integration 入門</i>』の「Business Process Management」を参照してください。</p>

表 3-1 WebLogic Integration アーキテクチャの主要なコンポーネント (続き)

コンポーネント	説明
B2B Integration	<p>WebLogic Integration で、外部のトレーディング パートナとインターネット経由で通信できるようにする。B2B Integration の機能は、XOCP、RosettaNet (1.1 と 2.0)、および Ariba cXML プロトコルでサポートされる (すべて XML を使用してイベントを表す)。</p> <p>注意： XOCP および cXML ビジネス プロトコルは、WebLogic Integration の本リリースより非推奨になりました。代替機能に関する詳細については、『<i>WebLogic Integration リリース ノート</i>』を参照してください。</p> <p>B2B は BPM プラグインを通じて BPM と直接統合されるので、外部のトレーディング パートナとメッセージを送受信するプロセスを簡単に定義することができる。詳細については、『<i>WebLogic Integration 入門</i>』の「B2B Integration」を参照してください。</p>
Data Integration	<p>XML ドキュメントのトランスフォーメーションに関して、Contivo Analyst で提供されるデータ マッピング機能だけでなく、XML データとバイナリデータの実行時の変換を可能にする。WebLogic Integration では、バイナリ-XML、XML-XML、およびバイナリ-バイナリのトランスフォーメーションがサポートされる。詳細については、『<i>WebLogic Integration 入門</i>』の「Data Integration」を参照してください。</p>
EDI 統合	<p>WebLogic Integration で、外部のトレーディング パートナと EDI メッセージを使用して通信できるようにする。EDI Integration は、Application Integration アダプタを通じて WebLogic Integration と結合される。Application Integration アダプタは、BPM ワークフローが EDI を使用して通信できるようにする。サービス アダプタを使用すると、WebLogic Integration プロセス エンジンで XML イベントを EDI Integration に送信できる。EDI Integration では、その受信したイベントが適切な EDI メッセージにマッピングされる。イベント アダプタを使用すると、EDI コンポーネントでは EDI メッセージを受信して、そのメッセージを XML イベントにマッピングし、その XML イベントを WebLogic Integration プロセス エンジンに送信できる。EDI Integration の詳細については、『<i>WebLogic Integration EDI ユーザーズ ガイド</i>』を参照。</p>

表 3-1 WebLogic Integration アーキテクチャの主要なコンポーネント (続き)

コンポーネント	説明
WebServices	UDDI、WebServices Description Language (WSDL)、Simple Object Access Protocol (SOAP) などの WebServices 技術を使用して WebServices 統合をサポートするためのサンプルコードを提供する。WebLogic Integration では、BPM ワークフローから WebService を呼び出したり、BPM ワークフローを WebService として使用したり、アプリケーションビューを WebService として使用したりすることができる。WebLogic WebServices の概要については、次の URL の dev2dev Online で Technology Tracks の「WebServices と XML」を参照。 http://dev2dev.bea.com/index.jsp

統合ソリューション トポロジの WebLogic Integration へのマッピング

WebLogic Integration のアーキテクチャを理解することで、統合スペシャリストは要件で指定した統合トポロジを適切な WebLogic Integration の機能にマッピングできます。統合スペシャリストは、以下の事項がどのように実装されるのかを確認する必要があります。

- ネットワーク トポロジと管理統制のスキームの定義
- 人間ユーザの会話の定義
- EIS アプリケーションとの統合
- 他の WebLogic Integration クラスタとの統合
- 企業内の他のドメインとの統合
- BEA TUXEDO および BEA eLink との統合
- 外部トレーディング パートナとの統合
- カスタム アプリケーション開発の統合

ネットワーク トポロジと管理統制のスコープの定義

地理的に分散した WAN を使用している企業では、別々の場所には別々のドメインがある場合があります。たとえば、本社がニューヨークにあり、出張所がロンドンと東京にある企業では、ドメインが3つに分かれている場合があります。

設計方法はドメイン管理のタイプによって決まるので、WebLogic Integration ソリューションを設計するときには、それらの場所がどのように管理されるのかを理解することが重要です。

ドメインは、以下の方法で管理できます。

- 一元的な管理—全社的な IT サポート組織によってすべてのドメインが一元的に管理されます。この環境では、WebLogic Integration クラスタを各ドメインにインストールでき、WebLogic Integration ソリューションでドメイン間の通信を調整できます。詳細については、3-12 ページの「他の WebLogic Integration クラスタとの統合」を参照してください。
- ローカル管理—独立した IT サポート組織によってドメインの一部またはすべてがローカルで管理されます。この環境では、WebLogic Integration クラスタを各ドメインにインストールでき、WebLogic Integration ソリューションでは B2B Integration を利用してドメイン間の通信を管理します。この場合、各サイトはトレーディング パートナとして扱われます。詳細については、3-14 ページの「企業内の他のドメインとの統合」を参照してください。

人間ユーザの会話の定義

WebLogic Integration では、以下の 2 種類の人間ユーザの会話がサポートされます。

- Worklist ユーザ
- クライアント ユーザ

人間ユーザに関する統合ソリューション要件の定義の詳細については、2-6 ページの「人間ユーザ」を参照してください。

Worklist ユーザ

注意： Worklist クライアントは WebLogic Integration リリース 7.0 より非推奨となりました。代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

Worklist ユーザには、ビジネスプロセスによって直接作業が割り当てられます。このようにタスクの割り当てられたユーザは、ビジネスプロセスと密接に関わります。WebLogic Integration では、Worklist というクライアントアプリケーションを使用してユーザにタスクを割り当てることができます。Worklist を使用して、タスクはユーザまたはロールに割り当てることができます。ロールとは指定されたユーザのグループのエリアスのことであり、そのロールのユーザはすべて、ロールに割り当てられたタスクを実行できます。Worklist では、タスクの割り当てられたユーザがそのタスクを実行してからプロセスが継続されるように指定することもできます。Worklist ユーザは、以下のタイプの Worklist クライアントのいずれかを使用して WebLogic Integration と会話しします。

表 3-2 Worklist クライアントのタイプ

Worklist タイプ	説明
Swing GUI	Swing GUI クライアントは、WebLogic Integration に付属の Worklist アプリケーションを使用して BPM の機能を使用する。
JSP ベース	JSP ベースのクライアントは、BPM API にアクセスする Java Server Pages (JSP) で記述された Worklist クライアントを通じて BPM の機能を使用する。
コマンドライン	Worklist クライアントは、CLWorklist コマンドを実行して BPM の機能を使用する。

JSP ベースのクライアントとコマンドライン クライアントについては、サンプルコードが `WLIHome/samples/bpm_api` ディレクトリに用意されています。

カスタム Worklist クライアントの開発については、『*BPM クライアント アプリケーションプログラミングガイド*』を参照してください。

クライアント ユーザ

クライアント ユーザは、プロセスの一部であるアプリケーションを使用してビジネス プロセスと間接的に会話をします。たとえば、販売担当者は JSP ベースのアプリケーションにログインし、販売注文を作成して（注文を処理するワークフローが開始される）、電子メールの確認を受信できます。

クライアント ユーザは、3 種類のメッセージ（以下の節を参照）のいずれかを使用してメッセージ ベースの非同期交換を通じて BPM と会話をします。

- メールボックス ベースのメッセージ
- 電子メール メッセージ

メールボックス ベースのメッセージ

WebLogic Integration では、B2B Integration で、データベース ベースのメールボックスを通じて BPM プロセスと会話する HTML ベースの **Zeroweight Client** が提供されます。ブラウザ クライアントによってメールボックスに配置された XML メッセージは、WebLogic Integration プロセス エンジンに送信されます。WebLogic Integration プロセス エンジンによってメールボックスに配置された XML メッセージは、ブラウザ クライアントで読むことができます。ブラウザ クライアントがメールボックスと会話できるようにする JSP タグのライブラリも用意されています。ブラウザ クライアントでは、XML メッセージを処理しなければなりません。詳細については、次の URL で「Zeroweight Client の使用」を参照してください。

<http://edocs.beasys.co.jp/e-docs/wli/docs70/interm/b2bhome.htm>

注意： WebLogic Integration リリース 7.0 より、このセクションで説明する、Trading Partner Zeroweight Client、B2B Mail Box、B2B JSP ブラウザ タグ ライブラリ機能は非推奨になりました。代替機能の詳細については、『WebLogic Integration リリース ノート』を参照してください。

電子メール メッセージ

WebLogic Integration プロセス エンジンでは、フリー テキストまたは XML データの含まれる電子メール メッセージを送信できます。詳細については、『WebLogic Integration Studio ユーザーズ ガイド』の「アクションの定義」の「電子メール メッセージの送信」を参照してください。電子メールで XML メッセージ

ジを受信し、それを BPM プロセス（WebLogic Integration Studio の開始 / イベント ノードおよびイベント ノードで表される BPM プロセスなど）に転送するアダプタを作成することもできます。

EIS アプリケーションとの統合

WebLogic Integration では、EIS アプリケーションと統合するためのメカニズムが複数用意されています。以下の節では、それらのメカニズムについて説明します。

- Application Integration アダプタ
- 他の統合テクニック
- カスタム アプリケーション統合ソリューションの開発

方法は、各アプリケーションで個別に選ぶ必要があります。できる限り、EIS アプリケーションは LAN 経由で WebLogic Integration に接続するべきです。EIS アプリケーション統合の統合要件の定義については、2-6 ページの「アプリケーション統合—アプリケーション統合で使用するエンタープライズ情報システム (EIS) アプリケーション」を参照してください。

Application Integration アダプタ

WebLogic Integration と外部の EIS アプリケーションを統合する最も一般的な方法は、Application Integration アダプタを使用することです。統合する EIS によっては、既製のアダプタが販売されています。カスタム アダプタを開発することも可能です。以下の節では、Application Integration アダプタの利点について説明します。

- 実際的な機能
- 汎用性
- ビジネスレベル インタフェース
- BPM との統合
- Application Development Kit

Application Integration アダプタの詳細については、次の URL にある『*Enterprise Application Integration (EAI): Providing Stability in the Whirlwind of E-Commerce*』を参照してください。

http://www.bea.com/products/elink/EAI_business_wp.shtml

実際のな機能

Application Integration アダプタは J2EE コネクタ アーキテクチャ (J2EE-CA) に基づいているので、接続管理、トランザクション管理、セキュリティ管理といった実際のな機能を備えています。WebLogic Server によって提供される基盤の J2EE CA エンジンでは、それらの機能がサポートされます。

主な利点は接続のプールです。EIS アプリケーションへの接続は、場合によっては高コストのリソースとなります。WebLogic Integration の接続プール機能を利用すると、そのリソースを多数の WebLogic Integration 要求で共有できます。

汎用性

Application Integration アダプタは、統合要件に合わせてコンフィグレーションされる汎用的なアダプタです。アダプタは、ブラウザを通じてコンフィグレーションします。したがって、開発者でなくてもアダプタをコンフィグレーションできます。コンフィグレーションは、開発者ではなく、統合するアプリケーションの実行に精通している人に委ねるのが適切です。この汎用的なアダプタの設計では、動的な統合環境もサポートされます。

ビジネスレベル インタフェース

サービス アダプタおよびイベント アダプタによって WebLogic Integration プロセス エンジンに提示されるインタフェースは、EIS アプリケーション固有の API 呼び出しではなくビジネスレベル インタフェースです (*create_new_customer* など)。このため、BPM プロセスの設計者は API を介してプログラマ的にアプリケーションと統合する仕組みではなくプロセスの機能に集中することができます。

アプリケーション統合のすべてのサービス メッセージとイベント メッセージは、WebLogic Integration リポジトリの XSD として定義します。

BPM との統合

BPM と Application Integration の機能は、設計時と実行時の両方で直接統合されます。ワークフローを構築するワークフロー設計者は、アプリケーション統合のイベントとサービスを WebLogic Integration Studio で表示できます。

Application Development Kit

WebLogic Integration には、アプリケーション統合のサービスアダプタとイベントアダプタを開発するための Application Development Kit (ADK) があります。詳細については、『アダプタの開発』を参照してください。

他の統合テクニック

EIS アプリケーションの他の統合テクニックには以下のものがあります。

- BPM ビジネス オペレーション
- JMS ラッパー

BPM ビジネス オペレーション

WebLogic Integration ソリューションでは、WebLogic Integration プロセス エンジンから始まった要求 / 応答モデルを BPM ビジネス オペレーションを使用して実装できます。ビジネス オペレーションを使用して、開発者は統合ソリューションで使用される EIS アプリケーション インタフェースをラップする EJB クラスまたは Java クラスを作成します。ビジネス オペレーションを利用すると、この EJB クラスまたは Java クラスは WebLogic Integration プロセス エンジンと統合し、ワークフローから呼び出すことができます。統合スペシャリストは、ビジネス オペレーションを使用して WebLogic Integration プロセス エンジンから EIS を同期的に呼び出します。

JMS ラッパー

WebLogic Integration ソリューションでは、JMS ラッパーを使用して非同期統合モデルを実装できます。開発者は、EIS アプリケーションから WebLogic Integration に（またはその逆方向で）メッセージを送信できる JMS ラッパーを作成できます。それらのメッセージは、以下のいずれかのフォーマットで作成できます。

- XML –適切な DTD または XSD を WebLogic Integration リポジトリで定義する必要があります。
- バイナリ –Data Integration プラグイン機能を使用して、バイナリ フォーマットと XML フォーマットの間で変換を実行できます。

統合スペシャリストは、JMS ラッパーを使用して WebLogic Integration プロセス エンジンから EIS を非同期で呼び出します。

カスタム アプリケーション統合ソリューションの開発

EIS アプリケーションとの統合を実現する最も簡単な方法は、既製の Application Integration アダプタを購入することです。ただし、カスタム アプリケーションで使用できるそのようなアダプタがない場合は、カスタム統合ソリューションを開発する必要があります。統合スペシャリストは、Application Integration アダプタ、ビジネス オペレーション、または JMS ラッパーのどれを開発するのかを決める必要があります。

既製の Application Integration アダプタを使用する欠点は、カスタムの統合よりも多くの開発作業を要する場合があることです。Application Integration アダプタの設計モデルは、開発者が一度アダプタをビルドし、それを多くの統合環境で利用できるようにするというものです。これは、商用の既製アダプタを開発するのに最適なモデルと言えます。

ただし、EIS アプリケーションの統合要件が単純で変化しない場合（たとえば、変更されることのない 2 つの API メソッドのみを呼び出すようにアプリケーションが設計されている場合）は、ビジネス オペレーションまたは JMS ラッパーを使用する方が開発作業が少なくて済みます。

データベースの統合 – サンプルの Application Integration データベース アダプタ

WebLogic Integration では、統合ソリューションで使用できるサンプルの Application Integration データベース アダプタのソース コードが提供されます。このアダプタではサービスとイベントの両方がサポートされます。

- サービス アダプタは、コンフィグレーションされた SQL 文をデータベースに対して実行します。
- イベント アダプタは、データベース テーブルの更新をモニタします。

必要に応じてソースコードを拡張することもできます。詳細については、『アダプタの開発』の「DBMS アダプタ」を参照してください。

ファイルと FTP の統合—サンプルのアプリケーション統合バイナリ ファイル アダプタ

サンプルのアプリケーション統合バイナリ ファイルアダプタのソースコードは、次の URL の BEA dev2dev Online で提供されています。

<http://dev2dev.bea.com/index.jsp>

このサンプルアダプタではサービスとイベントの両方がサポートされます。

- サービス アダプタは、XML メッセージを受け取り、WebLogic Integration で提供される Data Integration の機能を使用してそれをバイナリ フォーマットに変換して、そのバイナリ データをファイルに書き込みます。
- イベント アダプタは、ディレクトリでバイナリ ファイルをモニタし、WebLogic Integration で提供される Data Integration の機能を使用してそのバイナリ データを XML に変換して、その XML を JMS メッセージとして WebLogic Integration プロセス エンジンに送信します。

他の WebLogic Integration クラスタとの統合

3-5 ページの「ネットワークトポロジと管理統制のスキームの定義」で説明されているように、企業内では他の管理ドメイン（外国のオフィスなど）との統合が必要な場合があります。それらのドメインが一元的に管理されており、WAN 経由で統合ソリューションから認識できる場合は、WebLogic Integration クラスタを各サイトにインストールし、WebLogic Integration でそれらの通信を調整できます。

この場合、各 WebLogic Integration クラスタは個別に扱う必要があります（ビジネスプロセスをローカルの WebLogic Integration クラスタの要件に合わせてカスタマイズする）。WebLogic Integration クラスタのビジネスプロセスは、交換される XML メッセージを使用して統合します。それらの XML メッセージを定義する DTD または XSD は、各 WebLogic Integration クラスタの WebLogic Integration リポジトリに格納します。

WebLogic Integration クラスタの詳細については、『*WebLogic Integration ソリューションのデプロイメント*』の「WebLogic Integration クラスタについて」を参照してください。

統合オプション

WebLogic Integration クラスタは、WebLogic Integration の以下の機能を使用して統合できます。

- XOCP プロトコルを使用する B2B Integration 機能。WebLogic Integration クラスタは XOCP コンフィグレーションのトレーディング パートナです。詳細については、『*B2B Integration 入門*』を参照してください。

注意： XOCP ビジネス プロトコルは、WebLogic Integration の本リリースより非推奨になりました。代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

- WebLogic Integration WebServices のサンプル コード。詳細については、次の URL にある BEA dev2dev Online で、Technology Tracks の「WebServices と XML」を参照してください。

<http://dev2dev.bea.com/index.jsp>

統合サンプル

ある企業で、営業本部がニューヨークにあり、地域別の倉庫が世界中にあるグローバルな流通システムが運営されているとします。ニューヨークのオフィスでは注文管理システムが使用され、各倉庫では流通アプリケーションが実行されます。それぞれの場所では、WebLogic Integration クラスタが実行されています。

この場合、注文管理アプリケーションは注文を受け、どの地域でその注文に対応すべきかを判断し、その地域に通知します。各地域では、ワークフローで要求を処理し、応答を注文管理システムに送り返すことができます。

企業内の他のドメインとの統合

企業内では、統合ソリューションの管理が及ばない、または統合ソリューションから見ることでできない他の管理ドメイン（外国のオフィスなど）との統合が必要な場合もあります。そのような他のドメインでは、それら独自の統合要件として **WebLogic Integration** が使用される場合も、使用されない場合もあります。

WebLogic Integration ドメインは、**WebLogic Integration** の以下の機能を使用して統合できます。

- **XOCP** プロトコルを使用する **B2B Integration** 機能。**WebLogic Integration** クラスタは **XOCP** コンフィグレーションのトレーディング パートナです。詳細については、『*B2B Integration 入門*』を参照してください。このような場合、他のドメインは企業内部にある（かつファイアウォールの内側にある）トレーディング パートナとして考える必要があります。他のドメインとやり取りする必要があるビジネス イベントを定義します。それらのビジネス イベントは **XML** を使用して表されるので、関連する **DTD** または **XSD** を定義する必要があります。

注意： **XOCP** ビジネスプロトコルは、**WebLogic Integration** の本リリースより非推奨になりました。代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

- **WebLogic Integration WebServices** のサンプルコード。詳細については、次の URL にある **BEA dev2dev Online** で、**Technology Tracks** の「**WebServices と XML**」を参照してください。

`http://dev2dev.bea.com/index.jsp`

- 組織内の他のドメインで **WebLogic Integration** が使用されている場合は、3-12 ページの「他の **WebLogic Integration** クラスタとの統合」で説明されている技術を使用して **WebLogic Integration** クラスタを統合できます。
- 他のドメインで別の統合ソリューションが使用されている場合は、カスタムの統合を開発しないと **XML** メッセージをそのソリューションと交換できません。

BEA TUXEDO および BEA eLink との統合

WebLogic Integration ソリューションでは、WebLogic Server に既にデプロイされている以下のアプリケーションを統合できます。

- BEA WebLogic Tuxedo Connector は、WebLogic Server アプリケーションと Tuxedo サービスの統合を実現します。詳細については、次の URL にある WebLogic Tuxedo Connector のマニュアルを参照してください。

<http://edocs.beasys.co.jp/e-docs/wls/docs70/wtc.html>

- BEA eLink は、WebLogic Server アプリケーションと従来のアプリケーションの統合を実現するエンタープライズアプリケーション統合 (EAI) 製品のファミリーです。詳細については、次の URL にある eLink のマニュアルを参照してください。

<http://edocs.bea.com/mlink/index.html>

外部トレーディング パートナとの統合

外部のトレーディング パートナと統合する最も一般的な方法は、B2B 統合ソリューションを開発することです。以下の節では、WebLogic Integration が提供する B2B Integration の機能の基本事項について説明します。

- B2B 統合のアーキテクチャ
- サポート対象のビジネスプロトコル

B2B 統合のアーキテクチャ

B2B 統合ソリューションのアーキテクチャは、それがバリューチェーン統合 (ピアツーピア コンフィグレーション) であるのか、それとも B2B 交換 (ハブアンドスポーク コンフィグレーション) であるのかによって異なります。それらのアーキテクチャを比較する詳細な説明については、『*B2B Integration 入門*』の「B2B Integration の基礎」の「コンフィグレーション モデル」を参照してください。

注意： このセクションで説明した2つのビジネスプロトコル **cXML** と **XOCP** は **WebLogic Integration** リリース 7.0 より非推奨になっています。代替機能に関する詳細については、『*WebLogic Integration* リリースノート』を参照してください。

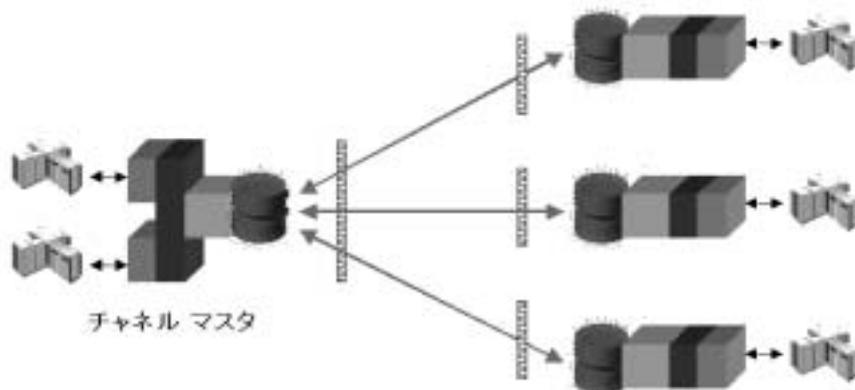
バリューチェーン統合ソリューションのピアツーピアコンフィグレーション

バリューチェーン統合ソリューションには、以下のタイプの統合のいずれかまたは両方が関わります。

- サプライチェーン。企業とサプライヤを統合します。
- デマンドチェーン。企業と顧客を統合します。

ピアツーピアコンフィグレーションは、次の図のようにバリューチェーン統合ソリューションで使用します。

図 3-1 バリュチェーン統合ソリューションのピア ツー ピア コンフィグレーション

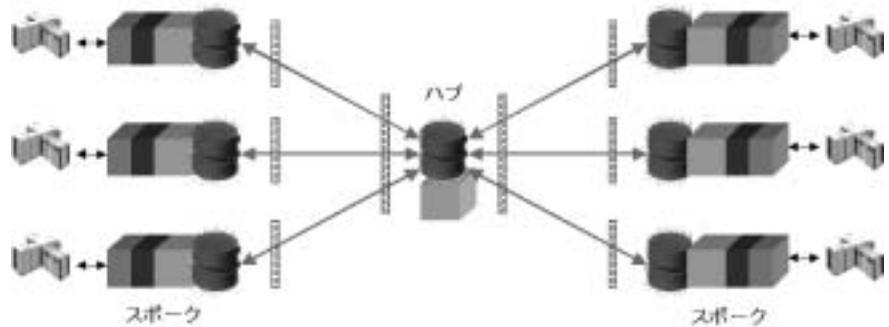


ピア ツー ピア コンフィグレーションでは、1つのトレーディング パートナ（企業）がチャンネル マスタ（CM）になります。チャンネル マスタとは、他のすべてのトレーディング パートナ（サプライヤ、顧客、または両方）が接続される管理エンティティのことです。このコンフィグレーション モデルは、トレーディング パートナが RosettaNet、cXML、または EDI プロトコルを使用して通信する必要がある場合にも使用します。

B2B 交換ソリューションのハブ アンド スポーク コンフィグレーション（非推奨）

B2B 交換ソリューションは、トレーディング パートナを B2B トランザクションに参加させます。B2B ハブ アンド スポーク アーキテクチャは、次の図のように B2B 交換ソリューションで使用します。

図 3-2 B2B 交換ソリューションのハブ アンド スポーク コンフィグレーション



ハブ アンド スポーク コンフィグレーションでは、1つの企業が参加トレーディング パートナ（スポーク）の間の B2B メッセージを調整するハブとして機能します。B2B アプリケーションは、XOCP プロトコルを使用する配信チャネルを通じて通信します。このコンフィグレーション モデルでは、メッセージのルーティングとフィルタ処理、サービス品質、会話中のトレーディング パートナに対するサービスといった付加価値サービスを実行するメッセージフローの仲介機能が提供されます。

ハイブリッド アーキテクチャ

WebLogic Integration ソリューションでは、ピア ツー ピア コンフィグレーションとハブ アンド スポーク コンフィグレーションを組み合わせる使用もあります。このコンフィグレーション モデルでは、一部のトレーディング パートナでは XOCP を使用し、その他のトレーディング パートナではピア ツー ピア プロトコル（RosettaNet、cXML、または EDI）を使用してトレーディング パートナ間でメッセージを交換できます。

サポート対象のビジネス プロトコル

ビジネス プロトコルは、トレーディング パートナ間で交換されるビジネス メッセージの構造、ビジネス メッセージを処理する方法、および適切な相手にメッセージをルーティングする方法を指定します。

WebLogic Integration は以下のビジネス プロトコルをサポートします。

- ebXML – WebLogic Integration をデプロイしているトレーディング パートナが使用し、WebLogic Integration のトレーディング パートナ軽量クライアント

ントである **Business Connect** をデプロイしているトレーディング パートナとやり取りします。**WebLogic Integration** は、HTTP などの通信プロトコルで **ebXML** メッセージの転送に使用されるメッセージエンベロープとヘッダドキュメント スキーマを定義する **ebXML** メッセージ サービス仕様 **v1.0** をサポートしています。

- **XOCP** (eXtensible Open Collaboration Protocol) – ハブ アンド スポーク コンフィグレーションおよび拡張機能を提供するために使用します。拡張機能には、メッセージのマルチキャスト、メッセージのペイロード定義の独立、会話ライフサイクルの管理、サービス品質 (QoS) 機能 (メッセージ持続性、タイムアウト、再試行、関連 ID など) があります。

注意： XOCP ビジネス プロトコルは、**WebLogic Integration** の本リリースより非推奨になりました。代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

- **RosettaNet** – トレーディング パートナで **RosettaNet** プロトコルを使用した接続が必要な場合、または統合ソリューション要件で標準的なプロトコルの使用が要求される場合に使用します。**B2B Integration** における **RosettaNet** のサポートについては、『*B2B Integration RosettaNet の実装*』を参照してください。

- **cXML** (Commerce eXtensible Markup Language) – **Ariba** 接続が必要なときに使用します。**B2B Integration** における **cXML** の詳細については、『*B2B Integration cXML の実装*』を参照してください。

注意： cXML ビジネス プロトコルは、**WebLogic Integration** の本リリースより非推奨になりました。代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

- **EDI** (電子データ交換) – トレーディング パートナで **EDI** を使用した接続が必要な場合に使用します。**EDI** の詳細については、『*WebLogic Integration EDI ユーザーズ ガイド*』を参照してください。

これらの異なるビジネス プロトコルを比較する詳細な説明については、『*B2B Integration 入門*』の「概要」の「E ビジネスの要件を満たす」の「ビジネス プロトコルのサポート」を参照してください。

カスタム アプリケーション開発の統合

WebLogic Integration ソリューションの全体的なアーキテクチャを設計するときには、統合スペシャリストはカスタム コードの開発を要するギャップを識別する必要があります。アプリケーション カスタマイズ オプションには、BPM プラグイン、カスタム アダプタ、およびロジック プラグインがあります。

BPM プラグイン

WebLogic Integration ソリューションでは、BPM プラグイン フレームワークを実装するカスタム BPM プラグインを作成しなければならない場合があります。BPM プラグインの設計と開発の詳細については、『*WebLogic Integration BPM プラグイン プログラミング ガイド*』を参照してください。

カスタム アダプタ

Application Integration の機能に基づく WebLogic Integration ソリューションでは、カスタム Application Integration アダプタを作成しなければならない場合があります。

カスタム ロジック プラグイン（非推奨）

B2B Integration の機能に基づく WebLogic Integration ソリューションでは、カスタム ロジック プラグインを作成しなければならない場合があります。カスタム ロジック プラグインは、ビジネス メッセージの特殊な処理を実行する Java クラスです。具体的には、ロジック プラグインはビジネス メッセージがノードを経由する場合にルールとビジネス ロジックを必要な箇所に挿入します。詳細については、『*B2B Integration ロジック プラグイン プログラミング ガイド*』を参照してください。

注意： カスタム ロジック プラグインは、本リリースの WebLogic Integration で非推奨となった XOCP プロトコルに基づいています。XOCP に代わる機能の詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

ビジネス プロセスの実装

統合ソリューション トポロジを定義したら、統合スペシャリストはビジネス プロセスの実装を始めることができます。統合ソリューション要件で定義された各ビジネス プロセスについて、以下の節で説明されているいくつかの項目を実装する必要があります。

- ビジネス イベント
- 条件データ
- ビジネス ルール
- マッピング
- ビジネス トランザクション

ビジネス イベント

ビジネス イベントとは、統合ソリューションのエンド ポイント間を行き来するイベントのことです。一部のイベントは入力としてプロセスに送信され、他のイベントはプロセス内で生成する必要があります。**WebLogic Integration** で、ビジネス メッセージは **XML** フォーマットまたはバイナリ フォーマットで送信されます。

各ビジネス イベントについて、統合スペシャリストは以下の事項を定義する必要があります。

- ユニークなイベント名
- イベントの構造とデータ内容を記述するメタデータ
 - XML データの場合、フォーマットは **DTD** または **XSD** のいずれかです。
 - バイナリ データの場合、フォーマットは **WebLogic Integration** の **Data Integration** の機能で生成されたメッセージ フォーマット言語 (MFL) でなければなりません。

これらの定義は、**WebLogic Integration** リポジトリに格納します。できる限り、メタデータを保持するリポジトリのエンティティの名前としてはイベント名を使用してください。**WebLogic Integration** では、**Application Integration**

アダプタがコンフィグレーションされるとこの情報が自動的にリポジトリに格納されます。

条件データ

処理上の決定を行う時に必要な、メッセージをルーティングする基準や条件に応じてコードを実行する基準などの、条件データ情報です。通常、このようなデータはビジネス イベントから抽出します。しかし、ビジネス プロセス時に、他のデータから動的に構築されます。条件データはそれぞれ、ワークフロー変数で表現されます。

ビジネス ルール

処理ルールは、プロセスの実行時の実行パスを決めるために条件データに適用されます。それらのルールは、**BPM** ワークフローの分岐ノードとして表されます。場合によっては、多数の複合的な条件をデータ セットに適用しなければならない場合もあります。それらの条件の結果によって、ワークフローの内容が決まります。条件を適用して結果を返すためには、**Java** クラスを開発する必要があります。結果は、**BPM** ワークフローの分岐ノードで使用できます。結果を取得するために作成した **Java** クラスは、**WebLogic Integration** プロセス エンジンからビジネス オペレーションとして呼び出すことができます。

マッピング

マッピングは、入力ビジネス メッセージと出力ビジネス メッセージのデータ トランスフォーメーションを定義します。

ビジネス トランザクション

プロセスには、1つまたは多数のビジネス トランザクションが含まれます。ビジネス トランザクションと調整アクションはビジネス トランザクションのロールバックが必要になった時に実行されるので、両方を定義する必要があります。

データ トランスフォーメーションの定義

以下の節では、**WebLogic Integration** ソリューションのデータ トランスフォーメーションについて説明します。

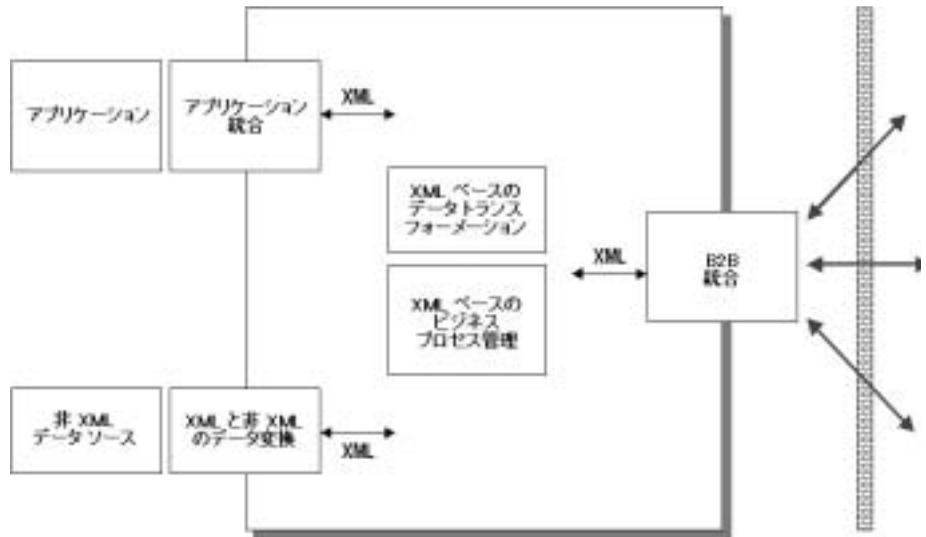
- XML ベースのデータ トランスフォーメーションについて
- バイナリ -XML トランスフォーメーションの定義
- バイナリ - バイナリ トランスフォーメーションの定義
- XML-XML トランスフォーメーションの定義

データ トランスフォーメーションの詳細については、『*WebLogic Integration 入門*』の「Data Integration」および『*WebLogic Integration データ変換*』を参照してください。

XML ベースのデータ トランスフォーメーションについて

XML は、WebLogic Integration で一番よく使用されるメッセージフォーマットです。次の図は、XML ベースのデータ トランスフォーメーションが WebLogic Integration ソリューションのアーキテクチャにどのように組み込まれるのかを示しています。

図 3-3 XML ベースのデータ トランスフォーメーション



XML データの最も一般的なソースには以下のものがあります。

- トレーディング パートナから受信したメッセージ
- バックエンド EIS システム (J2EE-CA アダプタなど) からのイベント
- 非 XML データのソース (フラット ファイルや EDI など)

XML は、多くの固有言語がある共通言語です。XML データの各ソースでは、異なる固有言語が使用されます。つまり、異なる固有言語の変換を処理するメカニズムが必要となります。XSLT トランスフォーメーションは、このメカニズムとして使用します。

たとえば、バイヤが発注書を RosettaNet フォーマットでセラーに送信し、セラーがそのデータを（SAP IDOC フォーマットのデータのみを受け付ける）バックエンド SAP ERP システムにインポートする必要があるとします。XSLT トランスフォーメーションを使用すると、RosettaNet の発注データを IDOC に変換できます。

バイナリ -XML トランスフォーメーションの定義

XML は WebLogic Integration で一番よく使用されるメッセージフォーマットですが、WebLogic Integration はバイナリ メッセージ インタフェースが使用される環境と統合しなければならない場合もあります。WebLogic Integration の Data Integration の機能は、XML フォーマットとバイナリ フォーマットの双方向の変換を実行します。この種の変換では、メッセージのデータ構造と内容は変更されません。メッセージフォーマット（XML またはバイナリ）のみが変更されます。

バイナリ データのフォーマットは、Format Builder アプリケーションを使用してグラフィカルに作成されるメッセージフォーマット言語（MFL）ドキュメントで記述します。MFL ドキュメントは、バイナリ データを解析して XML ドキュメントに変換するためにデータ統合実行時クラスで使用されます。

バイナリ - バイナリ トランスフォーメーションの定義

バイナリ - バイナリ トランスフォーメーションおよび任意の組み合わせ（バイナリ -XML や XML - バイナリ など）は、以下の 2 つのツールを組み合わせ使用してサポートします。

- WebLogic Integration（XML - バイナリ変換）
- XSLT（XML - XML トランスフォーメーション）

XML-XML トランスフォーメーションの定義

データ トランスフォーメーションプロセスは、ソース メッセージのデータ構造と内容を対象のメッセージのデータ構造と内容にマッピングするプロセスです。**WebLogic Integration** では、XML メッセージでのみデータ トランスフォーメーションがサポートされています。データ トランスフォーメーションのメカニズムには XSLT を使用します。

ユーザは、(**WebLogic Integration** にバンドルされている) **Contivo Analyst** などのグラフィカル マッピング ツールを使用して XSL スタイルシートを作成するか、または手動でスタイルシートを作成できます。XSL スタイルシートは、**WebLogic Integration** リポジトリに格納します。

実行時に、トランスフォーメーションはすべて **WebLogic Integration** プロセス エンジンによって実行されます。入力メッセージは、ワークフロー変数で提供されます。変換を実行する際、**WebLogic Integration** プロセス エンジンは **WebLogic Integration** リポジトリの XSL スタイルシートを参照します。出力メッセージは、別のワークフロー変数に格納されます。

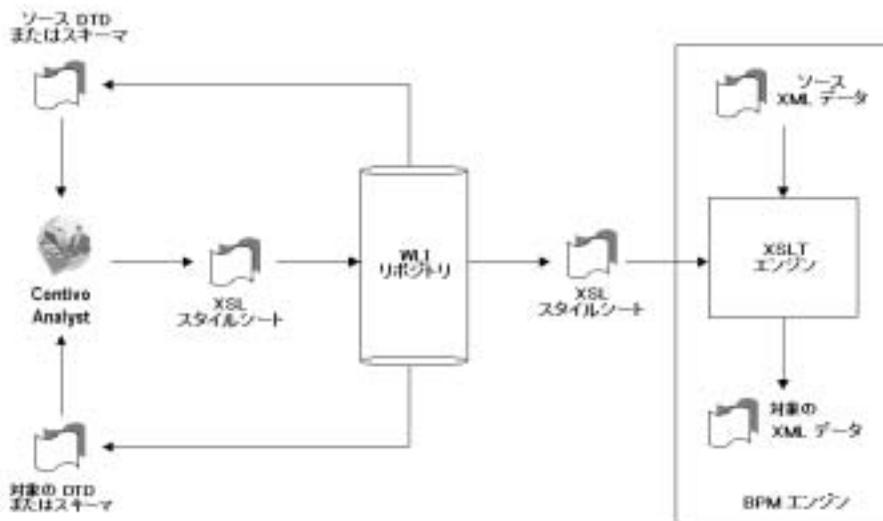
Contivo Analyst を使用した XML-XML トランスフォーメーションの定義

WebLogic Integration には、**Contivo Analyst** があります。**Contivo Analyst** は、XSL スタイルシートの生成に使用できる設計時のグラフィカル マッピング ツールです。**WebLogic Integration** プロセス エンジンは、実行時に XSL スタイルシートを実行してソース メッセージを対象のメッセージに変換できます。ソース メッセージと対象のメッセージは DTD または XSD を使用して定義し、それらの定義は **WebLogic Integration** リポジトリに格納します。**Contivo Analyst** のインストール、コンフィグレーション、および使用の詳細については、**Contivo** の製品マニュアルを参照してください。

WebLogic Integration リポジトリとの Contivo Analyst の統合

Contivo Analyst は、次の図のように **WebLogic Integration** リポジトリと直接統合されます。

図 3-4 WebLogic Integration リポジトリとの Contivo の統合

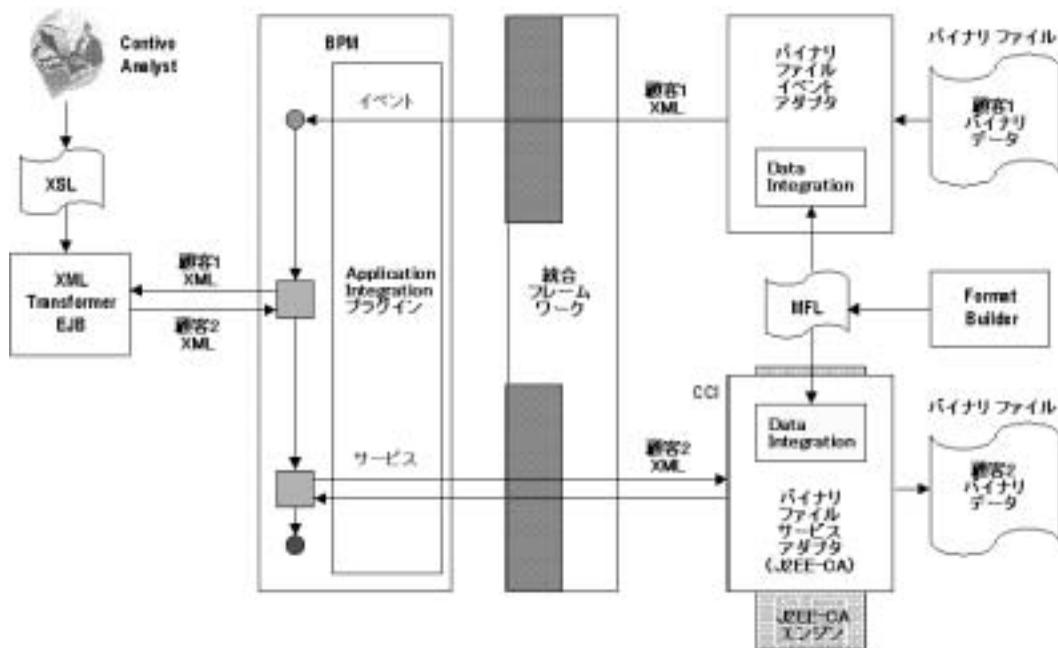


Contivo Analyst は、リポジトリを参照してソースと対象のメッセージ定義を選択します。選択された定義が Contivo Analyst に読み込まれると、ソースから対象へのマッピングがグラフィカルに指定されます。マッピングが完了すると、Contivo Analyst は XSLT を生成し、それを直接 WebLogic Integration リポジトリに保存します。WebLogic Integration プロセス エンジンでは、XSL Transform ワークフロー アクションを使用して XSLT を実行します。

統合ソリューションでの Contivo Analyst のロール

次の図は、Contivo Analyst がどのように WebLogic Integration ソリューションに組み込まれるのかを示しています。

図 3-5 WebLogic Integration ソリューションでの Contivo Analyst のロール



この図で、Contivo Analyst は XML Transformer EJB が顧客間で XML データを変換するために使用する XSL スタイルシートを生成します。

注意： XSL ベースの変換コンポーネントだけでなく、この図では WebLogic Integration ソリューションの他の多くの要素が示されています。それらの要素には、BPM ワークフロー、Application Integration アダプタ、および Data Integration を使用し、Format Builder アプリケーションで作成されたメッセージフォーマット言語 (MFL) ドキュメントを使用してバイナリ -XML トランスフォーメーションを実行するアダプタがあります。

3-24 ページの「XML ベースのデータ トランスフォーメーションについて」で説明されている RosetteNet-IDOC トランスフォーメーションのシナリオを使用して、次の手順は Contivo Analyst がどのように統合ソリューションに組み込まれるのかを示しています。

1. 開発者は、顧客とのビジネス会話向けに B2B エンジン をコンフィグレーションします。

2. 開発者は、受信した発注を処理するビジネス プロセスを定義します（SAP システムのアダプタがコンフィグレーションされて準備ができていると仮定する）。
3. 開発者は、ソース（RosettaNet）と対象（IDOC）のフォーマットを Contivo Analyst にロードします。
フォーマットは DTD または XSD のいずれかです。それらは WebLogic Integration リポジトリまたはローカルのファイルシステムに配置できます。
4. 開発者は、ソースのフィールドから対象のフィールドのマッピング ルールを関連付けて Contivo Analyst ツールでマップを作成します。
5. マップが完了したら、開発者は XSLT トランスフォーメーション ファイルを生成し、それを WebLogic Integration リポジトリに保存します。
XSLT トランスフォーメーション ファイルは、入力ドキュメント（発注書）と一緒に、実行時に XSLT エンジンに渡される XML ファイル（拡張子 .XSL）です。
6. WebLogic Integration Studio で、開発者は発注を受け取り、それを RosettaNet から IDOC に変換して、それを ERP システムに入力するビジネスプロセスを設計します。

詳細な設計の作成

統合ソリューションでどの WebLogic Integration コンポーネントを使用するのが決まったら、統合スペシャリストはビジネスアナリスト、開発者、およびシステム管理者と共同で詳細な設計を作成し、統合ソリューションの構築を開始します。

以下の節では、WebLogic Integration ソリューションの詳細な設計の作成に関連する情報を提供します。

- 統合設計ツールの使用
- デプロイメント環境の設計
- 統合設計でのパフォーマンスに関する考慮事項

統合設計ツールの使用

以下の WebLogic Integration ツールを使用すると、詳細な設計を作成しやすくなります。

- **WebLogic Integration Studio** を使用すると、ビジネスプロセスをモデル化し、BPM ワークフローを設計および生成できます。詳細については、『*WebLogic Integration Studio ユーザーズガイド*』を参照してください。
- **Contivo Analyst** を使用すると、XML-XML トランスフォーメーションを処理する XSL スタイルシートを作成できます。詳細については、3-26 ページの「XML-XML トランスフォーメーションの定義」を参照してください。
- **Format Builder** を使用すると、バイナリ -XML トランスフォーメーションを処理する MFL ドキュメントを作成できます。詳細については、『*WebLogic Integration データ変換*』の「フォーマット定義のビルド」を参照してください。

デプロイメント環境の設計

通常、統合スペシャリストは統合ソリューションの構築を始める前にデプロイメント環境を設計します。統合スペシャリストは、デプロイメントのリソースとコストを計画するためにこの情報を必要とします。**WebLogic Integration** ソリューションのデプロイメントについては、『*WebLogic Integration ソリューションのデプロイメント*』を参照してください。

たとえば、より多くの作業負荷に対応し、スケーラビリティを高めるためには、統合ソリューションは通常 **WebLogic Integration** クラスタにデプロイします。クラスタとは、1 つの単位として管理できる複数のサーバのグループのことです。クラスタの設計については、『*WebLogic Integration ソリューションのデプロイメント*』の「**WebLogic Integration** クラスタについて」の「クラスタデプロイメントの設計」を参照してください。

統合設計でのパフォーマンスに関する考慮事項

以下の節では、WebLogic Integration デプロイメントのパフォーマンスに影響する要素について説明します。統合スペシャリストは、統合ソリューションの設計段階でこれらの要素を考慮する必要があります。

- メッセージの永続性
- メッセージのサイズ
- XML の解析と検証
- ワークフロー
- Application Integration
- アプリケーションのロギング
- Secure Sockets Layer

WebLogic Integration デプロイメントの実行時パフォーマンスのチューニングについては、『WebLogic Integration ソリューションのデプロイメント』の「パフォーマンスのチューニング」を参照してください。

メッセージの永続性

メッセージの永続性は、ハードウェアやネットワークの障害が発生したときのデータの回復を向上させる WebLogic Integration の機能です。ただし、メッセージの永続性を有効にすると、メッセージを永続データストア（データベースまたはファイル）に保存またはパージするために必要な処理オーバーヘッドが発生するので、システム全体のパフォーマンスが低下します。詳細については、『B2B Integration 管理ガイド』の「永続性と回復のコンフィグレーション」を参照してください。

メッセージのサイズ

大規模メッセージは、特に大容量で交換されるときには、小規模メッセージよりも多くのシステムリソースを消費します。このため、大規模メッセージはパフォーマンスを低下させる場合があります。大規模メッセージは、以下の2つの形態で作成されます。

- 大規模な XML メッセージ
- バイナリ ペイロード

WebLogic Integration B2B Console を使用すると、B2B Integration の機能での大規模メッセージのサポートをコンフィグレーションできます。詳細については、『*B2B Integration Administration Console* オンライン ヘルプ』の「B2B Integration のコンフィグレーション」の「B2B Integration パラメータの定義」を参照してください。

注意： 大規模メッセージを有効にすると、より小さなメッセージの処理が遅くなる場合があります。

XML の解析と検証

XML の解析は高コストなプロセスなので、できる限り避ける必要があります。XML の検証はさらに高コストなので、高いパフォーマンスが要求されるすべてのデプロイメントで避ける必要があります。XML の解析と検証を管理する際は、以下の事項を考慮してください。

- アプリケーションを設計するときには、メッセージングプロセスでできる限りペイロード ベースの解析を回避するようにしてください。
- WebLogic Integration では、以下の 2 種類のキューが提供されます。
 - イベント検証キュー (com.bea.wlpi.ValidatingEventQueue)
 - イベント非検証キュー (com.bea.wlpi.EventQueue)

イベント検証キューを使用するには、`-validate` パラメータを MDBGenerator ユーティリティに渡します。詳細については、『*WebLogic Integration の起動、停止およびカスタマイズ*』の「WebLogic Integration のカスタマイズ」にある「NULL 変数をサポートする BPM のコンフィグレーション」を参照してください。

ワークフロー

この節では、WebLogic Integration ワークフローのパフォーマンス向上のヒントを紹介します。

同期処理

WebLogic Integration では、**JMS** イベントを通じて呼び出されるワークフローの代わりに、呼び出されるサブワークフローを使用できます。通常、呼び出されるワークフローは高速ですが、より多くのリソースが消費されます。より多くのリソースが消費されるのは、たとえば、呼び出されるワークフローが実行時に呼び出し元ワークフローをブロックすることが原因です。

高いパフォーマンスが要求されるデプロイメントでは、次のようにしてください。

- ワークフローがイベントを待たずに動作する場合は、呼び出されるワークフローを使用します。
- ワークフローがイベントを待つ場合は、**JMS** イベントを使用してワークフローを呼び出します。
- 最高レベルのパフォーマンスが必要な場合は、サブワークフローを呼び出すことなくすべての処理を単一テンプレートで実行します。

ビジネス メッセージの同期的送信

WebLogic Integration の高可用性機能を利用するワークフローを設計できます。**WebLogic Integration** では **RosettaNet** プロトコルを使用するビジネス メッセージの送信は非同期が要求されます（ビジネス メッセージの同期送信はサポートしていません）。

ビジネス メッセージの同期的送信には次の利点があります。ビジネス メッセージを送信するタスク ノードが実行を完了したら、必要に応じて即時にワークフローのプロセスを次のノードへ進めることができます。ビジネス メッセージの返信ステータスを待っている間、ワークフローの実行を待機する必要がありません。このセクションでは、**RosettaNet** ベースのワークフローにおけるメッセージの非同期送信設計の方法について説明しました。

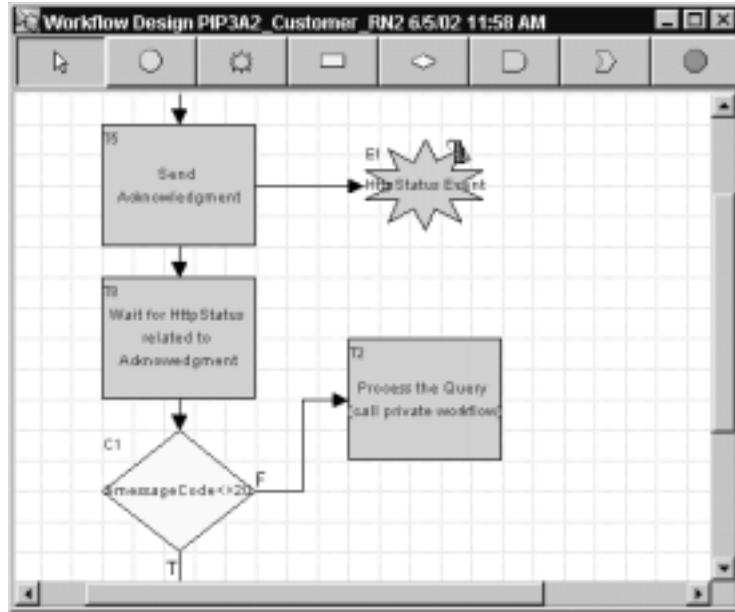
ビジネス メッセージを同期的に送信するワークフローを設計する場合は、次の定義を行います。

- ビジネス メッセージを（以前のように）送信するタスク ノード
- **HTTP** ステータス イベントのタイムアウト値、期日を指定するタスク ノード
- **HTTP** ステータス イベントを受け取るイベント ノード

注意： 7.0 以前のバージョンの **WebLogic Integration** を使用して作成された **RosettaNet** をベースとするワークフローをご使用の場合は、ビジネスメッセージの送信方法を変更する必要があります。既存の **RosettaNet** ベースのワークフローから現在の **WebLogic Integration** リリースへ移行する詳しい方法については、『*WebLogic Integration 移行ガイド*』を参照してください。

次の図では、4つのノード（T5、T8、C1、E1）がどのように **RosettaNet** ベースのワークフローで非同期のビジネスメッセージの送信タスクと関連性をもっているかを説明します。

図 3-6 ビジネス メッセージの送信 ワークフロー



ビジネス メッセージの非同期送信を指定する方法など、RosettaNet プロトコルをベースとするワークフローを作成する手順については、次を参照してください。

- *B2B Integration RosettaNet の実装*
- 『*B2B Integration ワークフローの作成*』の「ビジネス メッセージの送受信」の「RosettaNet ビジネス メッセージの送信」

メッセージ タイムアウト

[ビジネス メッセージの送信] タスクに関連付けられているノードを定義し、接続したら、タイムアウト値を指定するタスク ノードから分岐ノードに移動します。分岐ノードでは HTTP ステータスが評価され、実行フローが適切であるか判断されます。

タイムアウト値は、メッセージ遅延問題回避するのに十分な大きさになるように指定する必要があります。この問題は、ワークフローが **HTTP** ステータス イベントを受信側トレーディング パートナから受け取る前にタイムアウト値に到達すると発生します。タイムアウトに到達してからステータス イベントを受け取った場合は、ワークフローはこのステータス イベントを無視します。

一般に、特にリモートのトレーディング パートナとの間では、タイムアウト値は2時間以上に設定する必要があります。テストを行って、適切なタイムアウト値になるよう調整してください。

メッセージの再試行

ワークフローのベースとなるビジネス プロトコルごとに、**WebLogic Integration** で利用可能な、以前に送ったビジネス メッセージが正常に送信されなかった場合に送り直すように指定するメカニズムが異なります。

RosettaNet プロトコルをベースにしているワークフローには、**Studio** を使用して、メッセージの再送信機能を **RosettaNet** ベースのワークフローに埋め込むロジックを設計することが可能です。

ebXML ベースのワークフローには、ワークフローではなく、**B2B Console** でメッセージ再送信を指定します。この値は、このワークフローに関連する **ebXML** 会話を実行しているトレーディング パートナに対してドキュメント交換をコンフィグレーションする時に指定します。メッセージ再送信値を指定し、**[Delivery Semantics]** 選択ボックスで **[OnceAndOnlyOnce]** を選択すると、**B2B** エンジンがこの値を使用して、前の試行が失敗した後ビジネス メッセージを何度再送信するかを決定します。ビジネス メッセージが正常に送信された場合は、再送信は実行されません。これにより、メッセージは確実に一度だけ送信されることとなります (**[Delivery Semantics]** 選択ボックスを選択した場合は、**B2B** エンジンでは再送信値を無視します)。

条件

パフォーマンスが重要な場合は、**BPM** の開始ノードまたはイベント ノードでできる限りイベント条件を使用しないでください。完了ノードの設定に関する詳細については、『*WebLogic Integration Studio ユーザーズ ガイド*』の「ワークフローテンプレートの定義」を参照してください。

XPath の使用

XPath を使用するときには、以下のガイドラインを考慮してください。

- XPath を使用して XML イベントのデータから変数を設定するときには、XML イベントを受信したノードの変数テーブルで最初にすべての変数を設定する方が効率的です。
- XML 変数を XML イベントのペイロードに設定し、それからワークフロー変数を設定するアクションで XPath を使用することは効率的ではありません。
- ワークフロー変数を設定するアクションで XPath を使用するときには、XML ソース変数が小さいほどパフォーマンスが向上します。XPath が XML ソース変数のサブセットに対して繰り返し使用される場合は、サブセットの含まれる一時 XML 変数を作成し、それからその一時変数に対して XPath を実行する方が適切です。

一時変数の使用

BPM のパフォーマンスを向上させるには、静止状態の前に一時変数を NULL に設定します。このように設定することで、(イベントのように) トランザクションの境界でディスクに書き込まれることがなくなります。

統計収集の無効化

BPM のパフォーマンスを向上させるには、次のフラグを設定して統計の収集を無効にします。

```
-wli.bpm.server.nostatistics=true
```

統計の収集が無効になると、WebLogic Integration Studio に統計が表示されなくなります。

Application Integration

デフォルトでは、Application Integration の機能は非同期の要求および応答でのトランザクション対応で永続的なメッセージ配信の使用に依存します。この種の配信では、不要なメッセージングのオーバーヘッドが生じることがあります。システムの全体的なパフォーマンスを向上させるには、以下のソリューションを使用してトランザクション対応で永続的な配信を無効にします。

- トランザクションを無効にするには、wlai.properties ファイル (config/domain/wlai リポジトリ) を開き、そのファイルに次の行が含まれているようにします。

```
wlai.jms.asyncServiceTransFlag=false
```

- 通常、イベント配信のパフォーマンスはイベントの生成に使用されるアダプタに依存します。パフォーマンスを強化する場合、たいていは EIS インスタンスおよび EIS のイベント アダプタをチューニングします。WebLogic Server の Administration Console を使用して、JMS 接続ファクトリのデフォルトの配信モードを非永続に変更します。
- 非同期要求は、サーバごとにプーリングされたスレッドで処理されます。このプールのスレッド数が同時に保留されている非同期要求に割り当てられたスレッド数よりも少ない場合、余った要求はスレッドプールから空きスレッドを取得できるようになるまで処理されません。

この問題を解決するには、`wlai.properties` ファイルの `wlai.numAsyncServiceRequestProcessors` プロパティを大きな値に設定して非同期プロセッサ スレッドの数を増やします。このプロパティは、`wlai.properties` ファイルに追加しなければならない場合もあります。このプロパティのデフォルト値は 2 です。つまり、デフォルトでは、特定のサーバインスタンスの特定の時点で非同期サービス要求は 2 つしか処理されません。

アプリケーションのロギング

プロダクション環境で WebLogic Integration の全体的なパフォーマンスを向上させるには、アプリケーションで実行するロギングを最低限に抑える必要があります。出力のロギングを無効にする方法については、次の URL にある『WebLogic Server 管理者ガイド』の「ログ メッセージを使用した WebLogic Server の管理」を参照してください。

<http://edocs.beasys.co.jp/e-docs/wls/docs70/adminguide/logging.html>

Secure Sockets Layer

B2B Integration の機能が使用される WebLogic Integration デプロイメントでパフォーマンスを向上させるには、Secure Sockets Layer (SSL) の使用をセキュアなメッセージングが必要な状況のみに制限します。SSL では、ビジネスメッセージの暗号化と解読などのタスクを実行することによって余計な処理オーバーヘッドが生じ、その結果としてシステム全体のパフォーマンスが低下します。

サンプル WebLogic Integration アプリケーションの高度な設計の指定

以下の節では、これまでの節で説明された設計事項と関連するサンプル WebLogic Integration アプリケーションの高度な設計について説明します。

- WebLogic Integration Studio を使用したビジネス プロセスの自動化
- B2B Integration を使用したサプライ チェーンの統合
- Application Integration アダプタを使用した ERP システムの統合
- Data Integration を使用したデータ トランスフォーメーションの管理

これらの節では、第 2 章「統合ソリューションの要件の決定」で定義された要件がどのように WebLogic Integration アーキテクチャのコンポーネントにマッピングされるのかについて説明します。サンプル WebLogic Integration アプリケーションの設計の詳細については、『WebLogic Integration チュートリアル』を参照してください。

WebLogic Integration Studio を使用したビジネス プロセスの自動化

2-17 ページの「統合ソリューションとビジネス プロセスの指定」で説明されているビジネス プロセスを自動化するために、サンプルの WebLogic Integration アプリケーションでは WebLogic Integration Studio を使用して以下のビジネス プロセスをモデル化および管理します。

- Query Price and Availability (QPA)
- Purchase Order

これらのビジネス プロセスは、トレーディング パートナの間でプライベート ワークフローと協調的ワークフローを使用します。それらのワークフローでは、2-20 ページの「ビジネス イベントの指定」で説明されているビジネス イベントと 2-20 ページの「データ フローの要件の指定」で説明されているデータ フローがカプセル化されます。

ワークフロー設計の詳細については、『*WebLogic Integration チュートリアル*』の「サンプルについて」の「ビジネスプロセスおよびワークフローのモデル化」を参照してください。

B2B Integration を使用したサプライチェーンの統合

サプライチェーンの外部エンティティ（サプライヤ）とのビジネス トランザクションを管理するために、サンプルの *WebLogic Integration* アプリケーションでは *WebLogic Integration* の **B2B Integration** の機能を使用します。具体的に言うと、**XOCP** プロトコルを通じてトレーディング パートナの間で仲介メッセージングを実現するハブ アンド スポーク コンフィグレーション（3-15 ページの「外部トレーディング パートナとの統合」で説明）に基づいてビルドされます。**XOCP** を使用することで、サンプルの *WebLogic Integration* アプリケーションは 2-23 ページの「サービス品質の指定」で説明されているサービス品質（QoS）要件を満たします。

注意： **XOCP** ビジネス プロトコルは、*WebLogic Integration* の本リリースより非推奨になりました。代替機能に関する詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

サンプルの *WebLogic Integration* アプリケーションは、ハブ、パイヤ、トレーディング パートナである 2 つのサプライヤからなる 4 つのビジネス エンティティをデプロイします。ただし、このシナリオではサプライヤが **GCS** 外の別々のマシンで独自のソフトウェアを実行するトポロジが求められますが、サンプルを使いやすくするためにサンプルの *WebLogic Integration* アプリケーションではすべてのエンティティが同じマシンに配置されます。クラスタ環境でこのコードを実行する方法の詳細については、『*WebLogic Integration ソリューションのデプロイメント*』を参照してください。

独自の設計に **B2B Integration** 機能を組み込む方法の詳細については、『*WebLogic Integration チュートリアル*』の「サンプルについて」の「**B2B Integration**」を参照してください。

Application Integration アダプタを使用した ERP システムの統合

ERP 発注システムと統合するために、サンプルの WebLogic Integration アプリケーションでは Application Integration アダプタを使用します。バイヤ側では、イベント アダプタを使用して発注を ERP システムに追加します。また、サービス アダプタを使用して、選択したサプライヤから受信した発注確認書に基づいて発注を更新します。

Application Integration アダプタの詳細については、『WebLogic Integration チュートリアル』の「サンプルについて」の「Application Integration と Data Integration」を参照してください。

Data Integration を使用したデータ トランスフォーメーションの管理

XML- バイナリおよびバイナリ -XML のデータ トランスフォーメーションを処理するために、サンプルの WebLogic Integration アプリケーションでは Data Integration の機能を使用します。WLIS_SupplierPOPrivate ワークフローでは、以下の変換が実行されます。

- 発注の場合：XML データからバイナリ データ
- 発注確認書の場合：バイナリ データから XML

Data Integration の詳細については、『WebLogic Integration チュートリアル』の「サンプルについて」の「Application Integration と Data Integration」を参照してください。

索引

A

Application Development Kit (ADK)、
Application Integration 3-10

B

B2B Integration

永続性 3-31

Secure Sockets Layer (SSL) 3-38

外部トレーディング パートナ 2-8

大規模メッセージ、サポート 3-31

ハブ アンド スポーク コンフィグレーション 3-17

ピア ツー ピア コンフィグレーション
3-16

B2B 交換 3-17

C

Contivo Analyst

WebLogic Integration でのロール 3-27

WebLogic Integration リポジトリ、統
合 3-26

説明 3-26

cXML プロトコル 3-19

E

EDI プロトコル 3-19

eLink 免 3-15

J

J2EE コネクタ アーキテクチャ (J2EE-
CA) 3-9

Java Message Service (JMS)
ラッパー 3-10

R

RosettaNet プロトコル 3-19

RosettaNet メッセージ、送信 3-33

S

Secure Sockets Layer (SSL) 3-38

T

TUXEDO Connector 免 3-15

W

WebLogic Integration リポジトリ 3-26

WebServices 3-13, 3-14

Worklist ユーザ

タイプ 3-6

説明 2-6

X

XML-XML トランスフォーメーション
3-26

XML の解析 3-32

XML の検証 3-32

XOCP プロトコル 3-19

XPath、ワークフロー 3-36

Z

Zeroweight Client 3-7

あ

アーキテクト、説明 1-6

アクタ

解析 2-3
タイプによる分類 2-4
アダプタ 3-8
アプリケーション統合
Application Development Kit (ADK)
3-10
BPM の統合 3-10
アダプタ 3-8
カスタム ソリューション 3-11
データベース アダプタ 3-11
バイナリ ファイル アダプタ 3-12
パフォーマンス向上のヒント 3-37
アプリケーションのロギング 3-38

い

一時変数 3-37
イベント検証キュー 3-32
イベント条件、ワークフロー 3-36
印刷、製品のマニュアル ix

え

永続性 3-31
エラー処理 2-12
エンタープライズ情報システム 2-6
スペシャリスト、説明 1-6
統合 3-8

お

応答時間の要件 2-15

か

概念、主要 1-1
外部トレーディング パートナ 2-8
カスタマ サポート ix
可用性の要件 2-14
監査の要件 2-16

く

クライアント ユーザ
説明 2-6
タイプ 3-7
クラスタ
統合、企業内 3-12

さ

サービス品質 (QoS)、定義 2-13
サプライチェーン統合 3-16
サポート ix
サンプル WebLogic Integration アプリケーション
B2B Integration 3-40
Data Integration 3-41
アプリケーション統合 3-41
サービス品質 (QoS)、定義 2-23
設計、高度 3-39
データフローの要件、定義 2-20
統合ソリューション、定義 2-17
トポロジ、定義 2-25
ビジネス イベント、定義 2-20
ビジネスプロセス、定義 2-17
要件 2-17
ワークフロー 3-39
アクタ、定義 2-18
ロール、定義 2-18

し

システム管理者、説明 1-6
条件データ 2-11, 3-22
信頼性の要件 2-14

す

スキルセット 1-4
スケーラビリティの要件 2-15

せ

製品サポート ix

セキュリティの要件 2-15
設計上の考慮事項、説明 1-3
設計段階 1-7
設計パターン、説明 1-4
設計目標、説明 1-3

た

大規模メッセージ、B2B Integration 3-31
対象読者 viii
段階、主要 1-7

て

テクニカル サポート ix
データ フロー
 解析 2-12
 要件、定義 2-11
データベース アダプタ 3-11
デプロイメント、設計 3-30
デマンド チェーン 統合 3-16
電子メール アダプタ 3-7

と

同期処理、ワークフロー 3-33
統計収集 3-37
統合スペシャリスト、説明 1-5
統合ソリューション、説明 1-2
ドキュメンテーション
 表記規則 x
ドキュメント
 概要についてのマニュアル vii
 このマニュアルの内容 viii
トポロジ
 管理統制 3-5
 定義 2-16
ドメイン
 一元管理 3-5
 企業内 2-8
 統合 3-14
 ローカル管理 3-5
トランザクション 3-22

トランスフォーメーション
 XML-XML 3-26
 バイナリ -XML 3-25
 バイナリ -バイナリ 3-25
トレーディング パートナ
 外部 2-8, 3-15
 企業内 3-14

に

人間ユーザ 2-6

は

バイナリ -XML トランスフォーメーション 3-25
バイナリ -バイナリ トランスフォーメーション 3-25
バイナリ ファイル アダプタ 3-12
ハブ アンド スポーク コンフィグレーション、B2B 統合 3-17
パフォーマンスに関する考慮事項 3-31
パフォーマンスの要件 2-14
バリュー チェーン 統合 3-16

ひ

ピア ツー ピア コンフィグレーション、B2B 統合 3-16
ビジネス メッセージ、RosettaNet 送信 3-33
ビジネス アナリスト、説明 1-5
ビジネス イベント
 説明 2-10, 3-21
 特性 2-10
 メッセージ フォーマット 2-10
ビジネス オペレーション 3-10
ビジネス トランザクション 2-12, 3-22
ビジネス プロセス
 サンプル WebLogic Integration アプリケーション 3-39
説明 1-2
定義 2-2

ビジネスプロセス管理

統計収集 3-37

パフォーマンス向上のヒント 3-32

ビジネス オペレーション 3-10

プラグイン 3-20

ビジネス ルール 2-12, 3-22

ビジネスレベル インタフェース 3-9

ふ

否認防止性の要件 2-16

プラグイン

ビジネス プロセス管理プラグイン
3-20

ロジック プラグイン 3-20

へ

ベスト プラクティス、説明 1-4

変数、一時 3-37

ま

マッピング 2-12, 3-22

マニュアルで使用する表記規則 x

め

メッセージ

RosettaNet 送信 3-33

永続性 3-31

再試行 3-36

サイズ 3-31

メッセージ タイムアウト 3-35

メッセージの同期的送信 3-33

も

目標、説明 1-3

り

リポジトリ 3-26

れ

例外処理 2-12

ろ

ロギング (logging) 3-38

ロギングの要件 2-16

ロジック プラグイン 3-20

ルール

解析 2-4

統合の設計 1-4