



BEAWebLogic Integration™

B2B Integration RosettaNet の実装

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せん。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 Enterprise**、**BEA WebLogic Enterprise Platform**、**BEA WebLogic Express**、**BEA WebLogic Integration**、**BEA WebLogic Personalization Server**、**BEA WebLogic Platform**、**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。

BE2B Integration RosettaNet の実装

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

対象読者	v
e-docs Web サイト	vi
このマニュアルの印刷方法	vi
サポート情報	vii
表記規則	viii

1. はじめに

RosettaNet について	1-2
RosettaNet アーキテクチャ	1-2
WebLogic Integration アーキテクチャと RosettaNet	1-3
RosettaNet プロトコルレイヤ	1-4
PIP テンプレート	1-4
ワークフロー定義	1-5
デジタル署名	1-6
メッセージ検証	1-6
RosettaNet サンプル	1-6
サポートされないアイテム	1-7
RosettaNet の管理	1-8
RosettaNet セキュリティのコンフィグレーション	1-8
SSL およびデジタル署名のコンフィグレーション	1-8
メッセージの暗号化	1-9
環境の設定	1-9
RosettaNet ワークフローの移行	1-10

2. RosettaNet でのワークフローの使用

RosettaNet の理解	2-2
PIP ワークフロー インスタンスの理解	2-3
はじめに	2-4
パブリック ワークフローとプライベート ワークフローの調整	2-5
RosettaNet ワークフロー変数	2-5

ワークフロー変数表の使用	2-6
使い方	2-6
説明	2-7
RNIF 2.0 のワークフロー変数	2-8
RNIF 1.1 のワークフロー変数	2-14
統合された mSystem 変数	2-23
RosettaNet メッセージの受信	2-23
開始ノード	2-23
イベント ノード	2-24
タイムアウトの実装	2-24
順不同の信号受信	2-24
RosettaNet メッセージの送信	2-25
メッセージの検証	2-33
RosettaNet メッセージ検証	2-33
メッセージ検証に関する推奨参考文献	2-34
パフォーマンス チューニングとメッセージ検証	2-34
メッセージ添付ファイル	2-35

3. RosettaNet PIP テンプレート

RosettaNet テンプレートのインポート	3-1
RosettaNet ビジネス処理	3-2
PIP0A1: 障害通知	3-3
PIP3A2: 価格と在庫に関するクエリ	3-3
他の PIP のモデル化	3-4

索引

このマニュアルの内容

このマニュアルでは、BEA WebLogic Integration の B2B アプリケーションに RosettaNet を実装する方法について説明します。

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、WebLogic Integration における RosettaNet のサポートについての概要と、WebLogic Integration の B2B アプリケーションに RosettaNet を実装するために使用されるアーキテクチャについて説明します。
- 第 2 章「RosettaNet でのワークフローの使用」では、RosettaNet アプリケーションで使用されている WebLogic Integration の Business Process Management (BPM) ワークフローのしくみについて説明します。
- 第 3 章「RosettaNet PIP テンプレート」では、WebLogic Integration に付属の 2 つの RosettaNet PIP テンプレートの概要、および新しく PIP を作成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、次のユーザを対象としています。

- WebLogic Integration Studio を使用して、ビジネス メッセージを交換するための RosettaNet 標準に準拠したワークフローを設計するビジネスプロセス設計者
- WebLogic Integration RosettaNet ソリューションを設定および管理するシステム管理者

このマニュアルに進む前に、次のマニュアルの内容について理解されることをお勧めします。

- *WebLogic Integration 入門*

-
- *WebLogic Integration チュートリアル*
 - *WebLogic Integration の起動、停止およびカスタマイズ*
 - *B2B Integration 入門*
 - *B2B Integration 管理ガイド*
 - *B2B Integration セキュリティの実装*

RosettaNet ソリューションの設計を行う前に、次のマニュアルに精通されることをお勧めします。

- *WebLogic Integration ソリューションの設計*
- *WebLogic Integration ソリューションのデプロイメント*

e-docs Web サイト

BEA 製品のドキュメントは、BEA Systems, Inc. の Web サイトで入手できます。BEA のホーム ページで [製品のドキュメント] をクリックするか、または「e-docs」という製品ドキュメント ページ (<http://edocs.beasys.co.jp/e-docs/index.html>) を直接表示してください。

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 の Web サイト (<http://edocs.beasys.co.jp/e-docs/index.html>) で入手できます。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。PDF を表示するには、WebLogic Integration ドキュメントのホームページを開き、[PDF 版] ボタンをクリックして、印刷するマニュアルを選択します。

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

BEA 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、ご使用の WebLogic Integration 製品ドキュメントのリリース番号をお書き添えください。

本バージョンの BEA WebLogic Integration について不明な点がある場合、または BEA 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。
斜体	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>
斜体の等幅テキスト	コード内の変数を示す。 <i>例</i> <pre>String expr</pre>
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> <pre>LPT1 SIGNON OR</pre>

表記法	適用
{ }	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> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
	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> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
.	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

WebLogic Integration B2B は、XOCP、RosettaNet、および cXML メッセージの交換の解決および管理を可能とするルーティング アーキテクチャをサポートしています。このアーキテクチャにより、WebLogic Integration を使用するトレーディング パートナは、これらのプロトコル標準の任意のものを使って、ビジネス メッセージを交換できます。

『*B2B Integration 管理ガイド*』の「**コンフィグレーション要件**」をサポートするために必要な、**B2B** コンフィグレーションの例を記載しています。RosettaNet をサポートするようにコンフィグレーションした **B2B** エンジンを使うと、トレーディング パートナは *RosettaNet Implementation Framework 1.1* (RNIF 1.1) および *RosettaNet Implementation Framework 2.0* (RNIF 2.0) に記載されているように、RosettaNet メッセージの送受信を行えます。加えて、WebLogic Integration の **Business Process Management (BPM)** では、RosettaNet Partner Interface Process (PIP) をモデル化するワークフローの作成と実行をサポートしています。WebLogic Integration は、2 つの PIP テンプレート (0A1 および 3A2) を備えています。これらは、ユーザ独自の PIP 実装を行う際の基準として使用できます。

以下の節では、RosettaNet プロトコルの概要を説明します。

- RosettaNet について
- RosettaNet アーキテクチャ
- RosettaNet の管理
- RosettaNet セキュリティのコンフィグレーション
- 環境の設定

RosettaNet について

ここでは、電子ビジネス トランザクションのための RosettaNet 標準の概要を説明します。RosettaNet Consortium は、主要情報技術、電子部品、および半導体製造に携わる各企業によって構成される、非営利のコンソーシアムです。業界全体にわたる、オープンな E ビジネス プロセスの標準を作成し、実装するために活動しています。これらのプロセスは、参加しているサプライチェーン パートナ間で使用される、電子ビジネス インタフェースを標準化するために設計されます。RosettaNet Implementation Framework 仕様 (<http://www.rosettanet.org> で入手可能) は、RosettaNet Partner Interface Process (PIP) を実装するアプリケーションのガイドラインです。これら PIP は、トレーディング パートナ間で使用される標準化された電子ビジネス プロセスです。

マシン可読 RosettaNet メッセージは、RNIF 1.1 では RosettaNet オブジェクトとして、RNIF 2.0 では RosettaNet ビジネス メッセージとして知られています。RosettaNet メッセージの要素には、すべての RosettaNet メッセージに共通のものと、特定の PIP に固有のものがあります。RosettaNet メッセージが確実に一貫した方法で構造化され、処理されるように、各 PIP はメッセージガイドラインと XML 文書型定義 (DTD) を備えています。

RosettaNet アーキテクチャ

WebLogic Integration による RosettaNet のサポートは、次のコンポーネントで構成されています。

- *RosettaNet* プロトコルレイヤ
B2B エンジンによる RosettaNet プロトコルのサポートを実装するビジネス プロトコル定義。
- *B2B Integration* プラグイン
WebLogic Integration Studio およびプロセス エンジンの機能を拡張して、標準 RosettaNet PIP を実装するワークフローのモデル化と実行をサポートするプラグイン。

- *PIP* テンプレート
標準 RosettaNet PIP をユーザおよびトレーディング パートナ用の実装するようにカスタマイズできるワークフロー テンプレート。

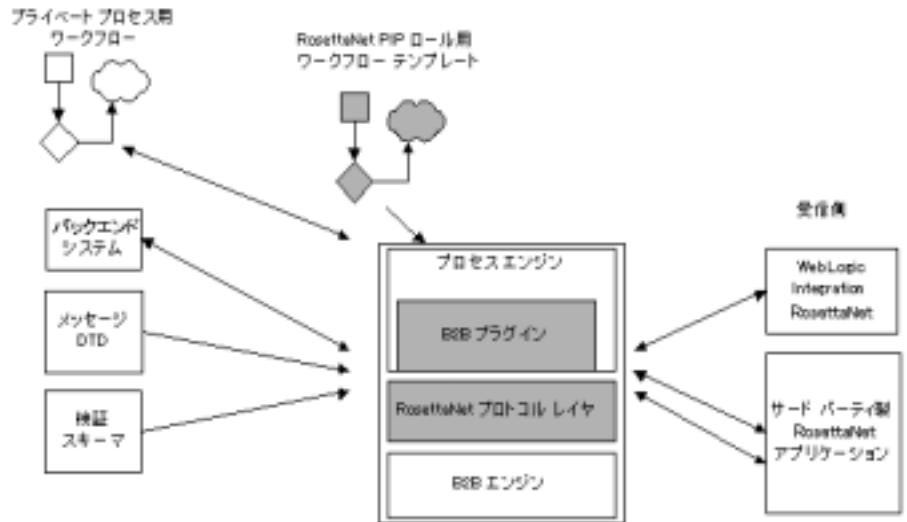
WebLogic Integration アーキテクチャと RosettaNet

WebLogic Integration Studio を使うと、RosettaNet PIP を実装するワークフローのモデル化と、実行のモニタを行えます。ワークフロー、プロセス エンジン、B2B プラグイン、および B2B エンジンが連携することにより、次のことが可能になります。

- トレーディング パートナの配信チャネル定義は、B2B コンフィグレーションの一部として RosettaNet プロトコルにバインドされます。
- B2B コンフィグレーションで定義された対話定義およびコラボレーション アグリーメントは、配信チャネルを適切なワークフローに関連付けます。
- ワークフローのアクションは、B2B エンジン経由でルーティングされる RosettaNet メッセージを、リモートのトレーディング パートナ配信チャネルに送信します。
- ワークフロー テンプレートのプロパティは、ワークフローが実装する PIP とロールを示します。
- ワークフロー イベントは、B2B エンジンまたはプライベートワークフローアクションによって送信される RosettaNet メッセージを待機します。

次の図は、WebLogic Integration がどのように RosettaNet の電子ビジネス プロセス標準をサポートするかを示します。

図 1-1 WebLogic Integration の RosettaNet アーキテクチャ



以下の節では、WebLogic Integration による RosettaNet のサポートと、図示されているコンポーネントに関する追加情報を提供します。

RosettaNet プロトコル レイヤ

RosettaNet プロトコル レイヤにより、転送、メッセージ パッケージング、およびセキュリティに関する RNIF 1.1 および RNIF 2.0 の仕様に従い、インターネットを通じてメッセージを送受信することが可能になります。

PIP テンプレート

RosettaNet PIP は、トレーディング パートナが E ビジネス トランザクションの実行中に参加する、パブリック プロセスを定義します。たとえば PIP 3A2 は、顧客トレーディング パートナが製品サプライヤトレーディング パートナと共に実行して、顧客が購入を希望し、製品サプライヤが販売を希望する商品の価格と在庫に関する情報を取得するプロセスを定義します。PIP に参加しているトレーディング パートナは、PIP 内でそのトレーディング パートナのロールによって

定義されているパブリック プロセスを実装する必要があります。また、内部システムを、プライベート プロセスおよびワークフローと同様に、パブリック プロセスに接続する必要があります。

WebLogic Integration によって提供された **RosettaNet** のサポートの主な特長は、トレーディング パートナが **PIP** の実装に使用できる、ワークフロー **PIP** テンプレートのセットです。テンプレートのセットは、次の **PIP** について定義されています。

- **PIP0A1**: 障害通知
- **PIP3A2**: 価格と在庫に関するクエリ

各 **PIP** ロールと **RosettaNet** バージョン用のテンプレートは、次のディレクトリに格納されています。

```
WLI_HOME\lib\templates\rosettanet
```

この場合、**WLI_HOME** は **WebLogic Integration** をインストールした場所を表します。

これらの **PIP** およびその他の **PIP** に関する詳細については、**RosettaNet** の Web サイト <http://www.rosettanet.org> を参照してください。**WebLogic Integration** のこのリリースで提供される **RosettaNet PIP** の詳細については、第 3 章「**RosettaNet PIP** テンプレート」を参照してください。

ワークフロー定義

RosettaNet PIP を実装するワークフローは、簡単に言えば特殊な協調的ワークフローです。**B2B** プラグインにより提供される拡張機能を備えた **Studio** が、任意の協調的ワークフローの作成に使用されるのと同じように、**RosettaNet PIP** を実装するワークフローの開発に使用されます。協調的ワークフローを作成するための **Studio** の使用に関する一般的な情報については、『*B2B Integration* ワークフローの作成』を参照してください。**RosettaNet PIP** の開発に特有の情報は、第 2 章「**RosettaNet** でのワークフローの使用」に記載されています。

デジタル署名

WebLogic Integration には、RSA CertJ ツールキットに基づく、すぐに使えるデジタル署名の実装が含まれます。デジタル署名の実装およびコンフィグレーションの詳細については、『*B2B Integration セキュリティの実装*』を参照してください。

WebLogic Integration はまた、RosettaNet のセキュリティ機能を例示するサンプルアプリケーションを提供しています。このサンプルについては、『*B2B Integration サンプルの使い方*』の「RosettaNet 2.0 Security サンプル」を参照してください。

メッセージ検証

RosettaNet PIP 定義には、PIP で交換されるメッセージの詳細な検証規則が含まれます。これらの規則は、XML 文書型定義 (DTD) 内で表現される検証と比べると、著しく厳格です。

必要な検証規則は、XML スキーマドキュメント (XSD) で表現されます。これらのドキュメントは、WebLogic Integration に付属の PIP テンプレートに含まれています。メッセージ検証の詳細については、第 2 章「RosettaNet でのワークフローの使用」を参照してください。

WebLogic Integration 2.1 および WebLogic Integration 2.1 SP1 の検証のスキーマは、W3C (World Wide Web Consortium) XSD (2000 XML Schema Definitions) スキーマに基づきます。WebLogic Integration 7.0 の検証のスキーマは、W3C (World Wide Web Consortium) XSD (2001 XML Schema Definitions) スキーマに基づきます。詳細については、『*WebLogic Integration 移行ガイド*』の「移行に関するその他のトピック」にある「RosettaNet スキーマの変更点」を参照してください。

RosettaNet サンプル

BEA RosettaNet 2.0 のセキュリティ サンプルは、ワークフローを使用する RosettaNet 2.0 PIP の実装に、WebLogic Integration をどのように使用できるかを示します。特に、RosettaNet 2.0 PIP 3A2 標準に準拠するビジネス メッセージを

交換する 2 社のトレーディング パートナを示しています。このサンプルの実行については、『*B2B Integration サンプルの使い方*』の「RosettaNet 2.0 Security サンプル」を参照してください。

このリリースで使用できるサンプルが追加された場合は、次の URL にある BEA dev2dev Online サイトにポストされます。

<http://dev2dev.bea.com>

サポートされないアイテム

RosettaNet 関連機能のうちいくつかは、WebLogic Integration のこのリリースではサポートされていません。

- *RNIF 2.0* による *SMTP* 転送の使用
HTTP 転送に著しく偏向しているものの、*RNIF 2.0* は転送に依存してはおらず、*SMTP* 転送の使用例を示すドキュメントが含まれています。WebLogic Integration は HTTP および HTTPS 転送をサポートしていますが、*SMTP* はサポートしていません。
- ハブ仲介によるメッセージング
WebLogic Integration は、RosettaNet ピア ツー ピア メッセージング モデルのみをサポートしています。RosettaNet ハブ配信チャンネルが、RosettaNet トレーディング パートナ ピア間の仲介機能としてコンフィグレーションされている、ハブ仲介モデルはサポートされていません。

RosettaNet の管理

WebLogic Integration では、トレーディング パートナ定義、対話定義、およびコラボレーション アグリーメントが、トレーディング パートナ間におけるメッセージ交換の実装に必要とされるユーザ定義要素となっています。

これらの要素をコンフィグレーションおよびモニタするための B2B Console の使用については、『*B2B Integration 管理ガイド*』および状況依存の B2B Console オンライン ヘルプを参照してください。RosettaNet のサンプル コンフィグレーションは、『*B2B Integration 管理ガイド*』の「RosettaNet アプリケーション」で説明されています。

注意：『*B2B Integration Administration Console* オンライン ヘルプ』も、WebLogic Integration 製品マニュアルに含まれています。

RosettaNet セキュリティのコンフィグレーション

以下の節では、RosettaNet ビジネス プロトコルのためのセキュリティのコンフィグレーション方法について説明します。

- SSL およびデジタル署名のコンフィグレーション
- メッセージの暗号化

SSL およびデジタル署名のコンフィグレーション

RosettaNet の SSL およびデジタル署名は、両方とも B2B Console によってコンフィグレーションされます。SSL およびデジタル署名のコンフィグレーションの詳細については、『*B2B Integration セキュリティの実装*』、『*B2B Integration*

Administration Console オンライン ヘルプ』、および『WebLogic Server 管理者ガイド』を参照してください。『WebLogic Server 管理者ガイド』は、次の URL でダウンロードできます。

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

メッセージの暗号化

RosettaNet Implementation Framework 2.0 は、RosettaNet フレームワーク全体に、新しいセキュリティ オプションを導入します。RNIF 1.1 と違い、RNIF 2.0 では次の 3 つのレベルの 1 つにおいて、メッセージを暗号化できます。

- なし—暗号化は行われません。
- ペイロード—サービス コンテンツとすべての添付ファイルが暗号化されません。
- ペイロード全体—サービス ヘッダ、サービス コンテンツ、および添付ファイルがすべて暗号化されます。

暗号化オプションは、WebLogic Integration B2B Console でコンフィグレーションできます。暗号化オプションの詳細については、『B2B Integration セキュリティの実装』の「セキュリティのコンフィグレーション」に記載の「メッセージ暗号化のコンフィグレーション」を参照してください。

環境の設定

ドメインで RosettaNet メッセージングをサポートするためには、次の作業が必要です。

- `WLI_HOME/lib/dtd/rosettanet` ディレクトリからドメインのディレクトリへ、文書型定義 (DTD) をコピーします。

たとえば、ドメインを `c:\bea\user_projects\mydomain` ディレクトリに作成した場合は、`WLI_HOME/lib/dtd/rosettanet` ディレクトリから `c:\bea\user_projects\mydomain` ディレクトリへ DTD をコピーします。

- `DOMAIN_HOME` ドメイン ディレクトリの下に、「schemas」という名前のディレクトリを作成します (`DOMAIN_HOME` ディレクトリはドメイン名を含む、ドメイン ディレクトリの完全パス名です)。

`WLI_HOME/lib/xmlschema/rosettanet` ディレクトリ内のファイルを、新規作成した `DOMAIN_HOME/schemas` ディレクトリにコピーします。

たとえば、ドメインを `c:\bea\user_projects\mydomain` ディレクトリに作成した場合は、`c:\bea\user_projects\mydomain\schemas` ディレクトリを作成し、次に `WLI_HOME/lib/xmlschema/rosettanet` ディレクトリのファイルを `c:\bea\user_projects\mydomain\schemas` ディレクトリにコピーします。

ドメインのルート ディレクトリを変更すれば、**WebLogic Integration** が **DTD** やスキーマを参照するためのデフォルト ディレクトリを変更することができます。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「ドメインのルート ディレクトリ」を参照してください。

RosettaNet ワークフローの移行

お使いのアプリケーションに **WebLogic Integration 2.1** または **WebLogic Integration ServicePack1** を使用して開発されたワークフローが含まれている場合、どちらも **RosettaNet** プロトコルを実装するため、**WebLogic Integration 7.0** でアプリケーションを実行するにはまずワークフローの変更が必要になります。また、非同期メッセージのサポートにも、これらの変更が必要です。詳細な情報については、『*WebLogic Integration 移行ガイド*』の「**WebLogic Integration 2.1 または WebLogic Integration 2.1 から WebLogic Integration 7.0 への移行**」にある「手順 10. RosettaNet ワークフローの移行」を参照してください。

2 RosettaNet でのワークフローの使用

以下の節では、RosettaNet Partner Interface Process (PIP) を実装するワークフローの開発方法について説明します。

- RosettaNet の理解
- PIP ワークフロー インスタンスの理解
- はじめに
- パブリック ワークフローとプライベート ワークフローの調整
- RosettaNet ワークフロー変数
- RosettaNet メッセージの受信
- RosettaNet メッセージの送信
- メッセージの検証
- メッセージ添付ファイル

この章の手順説明では、WebLogic Integration Studio で表示されるワークフロー図に説明しています。ワークフローダイアグラムにアクセスするには、『*WebLogic Integration の起動、停止およびカスタマイズ*』の「WebLogic Integration 管理ツールと設計ツール」にある「Studio の起動」の説明に従って Studio を起動します。B2B アプリケーションで使用する協調的ワークフローの定義の詳細については、『*WebLogic Integration の起動、停止およびカスタマイズ*』を参照してください。

RosettaNet の理解

次の RosettaNet マニュアルは、WebLogic Integration で提供される RosettaNet のサポートを使用して独自の PIP を実装する場合には、必読です。また、RosettaNet PIP 実装サンプルを完全に理解したい場合にも、一読をお勧めします。

- *WebLogic Integration* の起動、停止およびカスタマイズ
- 次のマニュアルは、RosettaNet Web サイト (<http://www.rosettanet.org>) の「Standards」セクションからダウンロード可能です。
 - 「RosettaNet Implementation Framework v1.1 (RNIF 1.1)」 – RNIF は、RosettaNet サプライチェーンおよびソリューション パートナが RosettaNet PIP 実行時に協調できるように設計された、オープンな共通ネットワーク アプリケーション フレームワークです。
 - 「RosettaNet Implementation Framework v2.0 (RNIF 2.0)」
 - 「RNIF Technical Advisories」 – 「RNIF Technical Advisories」は、RNIF 1.1 および RNIF 2.0 への更新および追加情報です。
 - 「RNIF Technical Recommendations」 – 「Technical Recommendations」は、RNIF v1.1 の公開されているバージョンではまだ利用できない特長や拡張機能を説明しています。「Technical Recommendations」の実装はオプションです。
 - 「RNIF Business Signals, Service Header & Preamble」 – RNIF ビジネス信号、サービス ヘッダおよびプリアンプルに関するドキュメントには、RNIF ビジネス信号、サービス ヘッダ、およびプリアンプルのためのメッセージ ガイドラインと XML 文書型定義 (DTD) が含まれます。
 - 「Understanding a PIP Blueprint」 – PIP Blueprint のコンポーネントと評価に対するリファレンス。「Standards」セクションの「Supporting Documentation」からダウンロードできます。
 - 対象 PIP – PIP は、サプライチェーン企業間のビジネス プロセスを定義する、特殊なシステム間 XML ベース ダイアログです。各 PIP には、RosettaNet Implementation Framework (RNIF) に基づく技術仕様、ビジネス ディクショナリの PIP 専用バージョンを備えたメッセージ ガイドライ

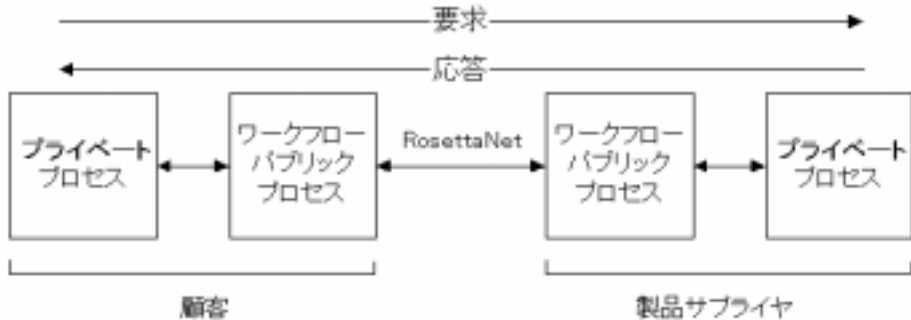
ンのマニュアル、および PIP 専用メッセージのための XML 文書型定義 (DTD) が含まれます。

PIP ワークフロー インスタンスの理解

WebLogic Integration は、標準の RosettaNet PIP をパブリック ワークフロー（協調的ワークフローとも言う）によって実装します。プライベート ワークフローがメッセージの生成や、メッセージへの応答のためのバックエンド システムへのインタフェースに使用されるのに対し、パブリック ワークフローは他のトレーディング パートナにインタフェースを提供します。

次の図は、PIP ワークフローがトレーディング パートナ間でメッセージを渡すプロセスを示します。

図 2-1 メッセージ ワークフロー



一般に、RosettaNet 指向のワークフローは、次のようにメッセージを処理します。

1. 顧客のプライベート ワークフローが、RosettaNet メッセージを開始します。データが取得されて RosettaNet メッセージ構造にフォーマットされ、適切な PIP が決定され、メッセージが PIP 内の顧客のロールを実装するパブリック ワークフローに転送されます。
2. パブリック ワークフローのプロセスにより、適切な RosettaNet メッセージが作成されます。メッセージは B2B エンジンを通じて PIP 内の製品サプライヤのロールを実装しているパブリック ワークフローに送信されます。

3. 製品サプライヤのパブリック ワークフローはメッセージを受信し、ヘッダ情報を処理し、次に検証済みの顧客情報とメッセージ コンテンツを適切なプライベート ワークフロー プロセスに渡します。
4. 製品サプライヤのプライベート プロセスが、メッセージ コンテンツを解決し、応答を生成します。応答は RosettaNet メッセージ構造になるように処理され、製品サプライヤのパブリック プロセスに戻されます。
5. 製品サプライヤのパブリック プロセスは RosettaNet 応答メッセージを作成し、B2B エンジンを介して顧客のパブリック プロセスに送信します。
6. 顧客のパブリック プロセスは応答メッセージを受信し、ヘッダ情報を処理し、次に検証済みの製品サプライヤ情報およびメッセージ コンテンツを適切なプライベート プロセスに渡します。
7. プライベート プロセスが応答メッセージのコンテンツを解決します。

はじめに

RosettaNet ワークフローの開発に必要なタスクは、他のワークフローの開発に使用されるものとはほぼ同じですが、一部で異なります。標準的な手順は、『*WebLogic Integration の起動、停止およびカスタマイズ*』に記載されています。RosettaNet ワークフローを作成する前に参照してください。

WebLogic Integration には、RosettaNet 1.1 および 2.0 用の PIP0A1 と PIP3A2 がフルに実装されています。これらのテンプレートは、追加で PIP ワークフローを作成する場合のテンプレート ベースとして役に立ちます。

この節のこれ以降では、RosettaNet 固有のワークフローの作成に関する問題について説明します。

パブリック ワークフローとプライベート ワークフローの調整

WebLogic Integration では、RosettaNet PIP は B2B エンジンを通じてのメッセージ交換でトレーディング パートナのロールを実装するパブリック ワークフローです。これらのワークフローは、次の処理を行うプライベート ワークフローと協調して動作するように設計されています。

- 未加工のソースおよび受信者データの提供
- 標準 RosettaNet メッセージ本文の形式へのデータの構造化
- 受信データの処理
- データを送信するパブリック ワークフローの開始
- プライベート アプリケーションへのインタフェースの提供

パブリック ワークフローを開始するためのプライベート ワークフロー、またはプライベート ワークフローを開始するためのパブリック ワークフローのコンフィグレーションに必要な手順は、『*WebLogic Integration の起動、停止およびカスタマイズ*』の「協調的ワークフローの開始」で説明しています。

RosettaNet ワークフロー変数

WebLogic Integration Studio で実装される RosettaNet ワークフローが動作するには、いくつかのワークフロー変数が必要です。これらの変数の使われ方は、次の 3 通りです。

- 入力変数は、ワークフローにデータを受信するために使用されます。変数の内容は、呼び出し元プライベート ワークフローによって設定されます。
- 出力変数は、ワークフローからデータを送信するために使用されます。完了時に PIP によって作成される内容は、呼び出し元ワークフローに戻されません。

- システム変数は、RosettaNet ワークフローが正しく動作するために必要な、ワークフロー内の類別されたデータをトラッキングするのに使用されます。

実装されている PIP ロールとは関係なく、RosettaNet ワークフローはすべて、システム変数、入力変数、および出力変数からなる同じ基本セットを含んでいる必要があります。以下の節で、必要な変数を RNIF のバージョン別に表にまとめています。

- ワークフロー変数表の使用
- RNIF 2.0 のワークフロー変数
- RNIF 1.1 のワークフロー変数

ワークフロー変数表の使用

各ワークフロー変数について、名前と型 (`boolean`、`string`、`integer`、`xml`、または `object`) が識別されます。加えて、以下の節に示されるように、変数の使い方と説明が記載されています。

使い方

ワークフロー変数は、次のように使い方別に分類されます。

- 入力値は必須
これらの変数の値は、呼び出し元ワークフローによって設定する必要があります。
- 入力値は省略可能
これらの変数の値は、呼び出し元ワークフローによって指定できます。この値は必須ではありませんが、変数は存在している必要があります。
- 出力値は省略可能
これらの変数の値は、ワークフローの終了時に設定できます。この値は必須ではありませんが、変数は存在している必要があります。出力変数に対する値は、必須ではありません。
- システム値は必須
これらの値は、B2B エンジンによって RosettaNet メッセージの生成に使用されます。値は、呼び出し元ワークフローによって指定されるか、ワークフローの開始ノードで割り当てられる必要があります。

- システム値は省略可能

これらの値は、**B2B** エンジンによって **RosettaNet** メッセージの生成に使用されます。この値は必須ではありませんが、変数は存在している必要があります。

これらのワークフロー変数の使用時に注意すべき重要な点は、パブリックワークフローを開始するときには、すべての入力変数が初期化されなくてはならないということです。入力変数の初期化に使用される値は、公開や送信の必要はありません。

必須ワークフロー変数には、**NOF** パーティの指定が含まれます。これらのパーティが、**PIP** 例外を処理するトレーディング パートナです。**NOF** パーティは、元の **PIP** の参加者であっても、第三者であってもかまいません。

説明

変数について簡単に説明します。関連のある場合、変数の説明には変数のマップ先となる **RosettaNet** オブジェクト (**RNIF 1.1**) または **RosettaNet** ビジネス メッセージ (**RNIF 2.0**) の要素が含まれます。たとえば、入力変数 `GlobalUsageCode` のマップ先は、次のとおりです。

`ServiceHeader/ProcessControl/GlobalUsageCode`

この場合、スラッシュは次のように階層的な要素構造における位置を示します。

```
<ServiceHeader>
  <ProcessControl>
    <GlobalUsageCode>Value</GlobalUsageCode>
  </ProcessControl>
</ServiceHeader>
```

RNIF 2.0 のワークフロー変数

表 2-1 では、RNIF 2.0 PIP ワークフローで必要とされるテンプレート変数について説明しています。付加的な PIP 固有の入力変数および出力変数も必要な場合があります。特定の PIP については、メッセージガイドラインと XML 文書型定義 (DTD) を参照してください。

表 2-1 RNIF 2.0 のテンプレート変数

名前	型	使い方と説明
actionCode	string	システム一値は必須 マップ先: ServiceHeader/ProcessControl /ActivityControl/MessageControl /Manifest/ServiceContentControl /ActionIdentity/GlobalBusinessActionCode
actionCodeVersion	string	システム一値は必須 マップ先: ServiceHeader/ProcessControl /ActivityControl/MessageControl /Manifest/ServiceContentControl /ActionIdentity/VersionIdentifier
attachmentDescriptorInput	xml	入力一値は省略可能 添付ファイルとして使用されるファイルの記述子を含む。
attachmentDescriptorOutput	xml	出力一値は省略可能 添付ファイルとして受信された可能性のあるファイルの記述子を含む。
businessActivityID	string	システム一値は必須 マップ先: ServiceHeader/ProcessControl /ActivityControl /BusinessActivityIdentifier

表 2-1 RNIF 2.0 のテンプレート変数 (続き)

名前	型	使い方と説明
DOC_TYPE	string	システム値は省略可能 XML メッセージの doctype 文字列を含む。この文字列は、サービス コンテンツに doctype 文字列が含まれていなかった場合に、そのコンテンツに挿入される。
exceptionError	boolean	システム値は省略可能 例外エラーが発生したかどうかを示す。
fromDUNS	string	入力値は必須 送信者の DUNS。トレーディング パートナのリポジット内で定義されたビジネス ID と一致している必要がある。 マップ先： DeliveryHeader /messageSenderIdentification /PartnerIdentification /GlobalBusinessIdentifier ServiceHeader/ProcessControl /KnownInitiatingPartner /PartnerIdentification /GlobalBusinessIdentifier
fromLocation	string	システム値は必須 マップ先： DeliveryHeader /messageSenderIdentification /PartnerIdentification /locationID/FreeFormText
fromRole	string	システム値は必須 マップ先： ServiceHeader/ProcessControl /MessageControl/fromRole /GlobalPartnerRoleClassificationCode

2 RosettaNet でのワークフローの使用

表 2-1 RNIF 2.0 のテンプレート変数 (続き)

名前	型	使い方と説明
fromService	string	システム値は必須 マップ先: ServiceHeader /ProcessControl/MessageControl /fromService/GlobalBusinessServiceCode
globalUsageCode	string	入力値は必須 <i>Test</i> または <i>Production</i> のいずれかに設定される。 マップ先: ServiceHeader/ProcessControl /GlobalUsageCode
gotMessage	string	システム値は必須 RosettaNet メッセージが受信されたかどうかを示す。 メッセージが受信された場合は True 。 タイムアウトが発生した場合は False 。
isSignal	boolean	システム値は省略可能 信号が受信されたかどうかを示す。
messageCode	integer	システム値は省略可能 メッセージ送信によって返されたコードを含む。
messageTrackingId	string	システム値は省略可能 PIPOA1 通知のためのメッセージ コンテンツにデータを入れるために使用されるメッセージ ID。
NOFParty1	string	入力値は必須 PIP Failure Notifier ロールのパーティ名。Notification of Failure Error プロセス (PIPOA1) を開始するために使用される。
NOFParty2	string	入力値は必須 PIP Failure Report Administrator ロールのパーティ名。Notification of Failure Error プロセス (PIPOA1) を開始するために使用される。

表 2-1 RNIF 2.0 のテンプレート変数 (続き)

名前	型	使い方と説明
PIP	string	システム値は必須 対話名に一致している必要がある。 マップ先： ServiceHeader/ProcessControl /pipCode/GlobalProcessCode
PIPInput	xml	入力値は必須 メッセージのサービス コンテンツ。
PIPOutput	xml	出力値は省略可能 受信したメッセージのサービス コンテンツ。
PIPVersion	string	システム値は必須 対話のバージョンに一致している必要がある。 マップ先： ServiceHeader/ProcessControl /pipVersion/VersionIdentifier
reason	string	システム値は省略可能 ワークフローの終了理由を含む。
retryCount	integer	システム値は必須 再試行回数を保存するカウンタ。
rnProcessInstanceId	string	システム値は省略可能 PIP インスタンスのプロセス ID。0A1 通知のための メッセージ コンテンツにデータを入れるために使用さ れる。
rnSystem	object	システム値は省略可能 アプリケーションのデータ状態を保持するために内部 で使用されるオブジェクト。2-23 ページの「統合され た rnSystem 変数」を参照。
SERVICE_CONTENT_SCHEMA	string	システム値は必須 照らし合わせてサービス コンテンツを検証するための XSD スキーマの名前を含む。

2 RosettaNet でのワークフローの使用

表 2-1 RNIF 2.0 のテンプレート変数 (続き)

名前	型	使い方と説明
timeStamp	string	システム値は省略可能 再試行のカウン트가 1 であった場合 (すなわち、初めて行われた再試行の場合)、タイムスタンプは内部で生成される。 マップ先: DeliveryHeader/messageDateTime /DateTimeStamp
toDUNS	string	入力値は必須 受信者の DUNS。トレーディング パートナのリポジット内で定義されたビジネス ID と一致している必要がある。 マップ先: DeliveryHeader /messageReceiverIdentification /PartnerIdentification /GlobalBusinessIdentifier
toLocation	string	入力値は必須 マップ先: DeliveryHeader /messageReceiverIdentification /PartnerIdentification /locationID/FreeFormText
toRole	string	システム値は必須 マップ先: ServiceHeader/ProcessControl /MessageControl/toRole /GlobalPartnerRoleClassificationCode
toService	string	システム値は必須 マップ先: ServiceHeader/ProcessControl /MessageControl/toService /GlobalBusinessServiceCode

表 2-1 RNIF 2.0 のテンプレート変数 (続き)

名前	型	使い方と説明
useDTDValidation	boolean	システム一値は省略可能 True に設定した場合、XSD ではなく DTD 検証が使用される。False に設定した場合または変数が存在しない場合、検証には XSD スキーマが使用される。そもそも検証を実行するかどうかは、validateServiceContent および validateServiceHeader の値に応じて決定される。
validateServiceContent	boolean	入力一値は必須 サービス コンテンツをスキーマに照らし合わせて検証する必要があるかどうかを示すフラグ。 True - 検証は必要 False - 検証は不要
validateServiceHeader	boolean	入力一値は必須 サービス ヘッダをスキーマに照らし合わせて検証する必要があるかどうかを示すフラグ。 True - 検証は必要 False - 検証は不要
validationError	boolean	システム一値は省略可能 検証結果を示す。

RNIF 1.1 のワークフロー変数

表 2-2 では、RNIF 1.1 PIP ワークフローのテンプレート変数について説明しています。付加的な PIP 固有の入力変数および出力変数も必要な場合があります。特定の PIP については、メッセージガイドラインと XML 文書型定義 (DTD) を参照してください。

表 2-2 RNIF 1.1 のテンプレート変数

名前	型	使い方と説明
actionCode	string	システム値は必須 マップ先： ServiceHeader/ProcessControl /TransactionControl/ActionControl /ActionIdentity/GlobalBusinessActionCode
actionCodeVersion	string	システム値は必須 マップ先： ServiceHeader/ProcessControl /TransactionControl /ActionControl/ActionIdentity /VersionIdentifier
attachmentDescriptorInput	xml	入力値は省略可能 添付ファイルとして使用されるファイルの記述子。
attachmentDescriptorOutput	xml	出力値は省略可能 添付ファイルとして受信された可能性のあるファイルの記述子を含む。
businessActivityID	string	システム値は必須 マップ先： ServiceHeader/ProcessControl /ProcessIdentity /GlobalProcessCode

表 2-2 RNIF 1.1 のテンプレート変数 (続き)

名前	型	使い方と説明
DOC_TYPE	string	システム値は省略可能 XML メッセージの doctype 文字列を含む。この文字列は、サービス コンテンツに doctype 文字列が含まれていなかった場合に、そのコンテンツに挿入される。
docId	string	システム値は省略可能 確認応答メッセージの作成に使用される。 マップ先： ReceiptAcknowledgement/ receivedDocumentIdentifier /ProprietaryDocumentIdentifierReceiptAcknowledg mentException /theOffendingDocumentIdentifier /ProprietaryDocumentIdentifierException /theOffendingDocumentIdentifier /ProprietaryDocumentIdentifier
exceptionError	boolean	システム値は省略可能 例外エラーが発生したかどうかを示す。
fromClass	string	システム値は必須 マップ先： ServiceHeader/ProcessControl /TransactionControl/ActionControl /PartnerRoute/fromPartner/PartnerDescription /GlobalPartnerClassificationCode ServiceHeader/ProcessControl /TransactionControl/SignalControl /PartnerRoute/fromPartner/PartnerDescription /GlobalPartnerClassificationCode
fromContactName	string	入力値は必須 「Notification of Error」メッセージのための XML コンテンツを構成するために使用される。

表 2-2 RNIF 1.1 のテンプレート変数 (続き)

名前	型	使い方と説明
fromDUNS	string	<p>入力値は必須</p> <p>送信者の DUNS。トレーディング パートナのリポジトリ内で定義されたビジネス ID と一致している必要がある。</p> <p>マップ先:</p> <pre>ServiceHeader/ProcessControl /TransactionControl/ActionControl /PartnerRoute/fromPartner /PartnerDescription/BusinessDescription /GlobalBusinessIdentifierServiceHeader /ProcessControl/TransactionControl /SignalControl/PartnerRoute /fromPartner/PartnerDescription /BusinessDescription/GlobalBusinessIdentifier</pre>
fromEmailAddress	string	<p>入力値は必須</p> <p>「Notification of Error」メッセージのための XML コンテンツを構成するために使用される。送信者の電子メール アドレス。</p>
fromPhone	string	<p>入力値は必須</p> <p>「Notification of Error」メッセージのための XML コンテンツを構成するために使用される。送信者の電話番号。</p>
fromRole	string	<p>システム値は必須</p> <p>マップ先:</p> <pre>ServiceHeader/ProcessControl /TransactionControl/PartnerRoleRoute /fromRole/PartnerRoleDescription /GlobalPartnerRoleClassificationCode</pre>
fromService	string	<p>システム値は必須</p> <p>マップ先:</p> <pre>ServiceHeader/ProcessControl /ServiceRoute/fromService /BusinessServiceDescription /GlobalBusinessServiceCode</pre>

表 2-2 RNIF 1.1 のテンプレート変数 (続き)

名前	型	使い方と説明
fromSupplychain	string	<p>入力値は必須 メッセージ確認応答の作成に使用される。 マップ先: ReceiptAcknowledgement /fromRole/PartnerRoleDescription /PartnerDescription/BusinessDescription /GlobalSupplyChainCode ReceiptAcknowledgmentException /fromRole/PartnerRoleDescription /PartnerDescription/BusinessDescription /GlobalSupplyChainCode Exception/fromRole /PartnerRoleDescription/PartnerDescription /BusinessDescription/GlobalSupplyChainCode</p>
functionCode	string	<p>システム値は必須 マップ先: ServiceHeader/ProcessControl /TransactionControl/ActionControl /GlobalDocumentFunctionCode</p>
globalUsageCode	string	<p>入力値は必須 <i>Test</i> または <i>Production</i> のいずれかに設定される。 マップ先: Preamble/GlobalUsageCode</p>
gotMessage	string	<p>システム値は必須 RosettaNet メッセージが受信されたかどうかを示す。 True - メッセージは受信された。 False - タイムアウトが発生した。</p>
isSignal	boolean	<p>システム値は省略可能 信号が受信されたかどうかを示す。</p>
messageCode	integer	<p>システム値は省略可能 メッセージ送信によって返されたコードを含む。</p>

表 2-2 RNIF 1.1 のテンプレート変数 (続き)

名前	型	使い方と説明
messageTrackingId	string	システム値は省略可能 PIPOA1 通知のためのメッセージ コンテンツにデータを入れるために使用されるメッセージ ID。
NOFParty1	string	入力値は必須 PIP Failure Notifier ロールのパーティ名。Notification of Failure Error (PIPOA1) を開始するために使用される。
NOFParty2	string	入力値は必須 PIP Failure Report Administrator ロールのパーティ名。Notification of Failure Error (PIPOA1) を開始するために使用される。
PIP	string	システム値は必須 対話名に一致している必要がある。 マップ先： ServiceHeader/ProcessControl /ProcessIdentity/GlobalProcessIndicatorCode
PIPInput	xml	入力値は必須 メッセージのサービス コンテンツ。
PIPOutput	xml	出力値は省略可能 受信したメッセージのサービス コンテンツ。
PIPVersion	string	システム値は必須 対話のバージョンに一致している必要がある。 マップ先： ServiceHeader/ProcessControl /ProcessIdentity/VersionIdentifier
reason	string	システム値は省略可能 ワークフローの終了理由を含む。
retryCount	integer	システム値は必須 再試行回数を保存するカウンタ。

表 2-2 RNIF 1.1 のテンプレート変数 (続き)

名前	型	使い方と説明
rnProcessInstanceId	string	システム値は省略可能 PIP インスタンスのプロセス ID。0A1 通知のためのメッセージコンテンツにデータを入れるために使用される。
rnSystem	object	システム値は省略可能 アプリケーションのデータ状態を保持するために内部で使用されるオブジェクト。「2-23 ページの「統合された rnSystem 変数」」を参照。
SERVICE_CONTENT_SCHEMA	string	システム値は必須 照らし合わせてサービスコンテンツを検証するための XSD スキーマの名前を含む。
timeStamp	string	システム値は省略可能 マップ先: ReceiptAcknowledgement /receivedDocumentDateTime/DateTimeStamp ReceiptAcknowledgmentException /theOffendingDocumentDateTime /DateTimeStamp Exception/theOffendingDocumentDateTime /DateTimeStamp
toClass	string	システム値は必須 マップ先: ServiceHeader/ProcessControl /TransactionControl/ActionControl /PartnerRoute/toPartner/PartnerDescription /GlobalPartnerClassificationCode ServiceHeader/ProcessControl /TransactionControl/SignalControl /PartnerRoute/toPartner/PartnerDescription /GlobalPartnerClassificationCode

表 2-2 RNIF 1.1 のテンプレート変数 (続き)

名前	型	使い方と説明
toDUNS	string	<p>入力値は必須</p> <p>受信者の DUNS。トレーディング パートナのリポジトリ内で定義されたビジネス ID と一致している必要がある。</p> <p>マップ先:</p> <pre>ServiceHeader/ProcessControl /TransactionControl/ActionControl /PartnerRoute/toPartner/PartnerDescription /BusinessDescription /GlobalBusinessIdentifier</pre> <p>ServiceHeader/ProcessControl /TransactionControl/SignalControl /PartnerRoute/toPartner /PartnerDescription/BusinessDescription /GlobalBusinessIdentifier</p>
toRole	string	<p>システム値は必須</p> <p>マップ先:</p> <pre>ServiceHeader/ProcessControl /TransactionControl/PartnerRoleRoute /toRole/PartnerRoleDescription /GlobalPartnerRoleClassificationCode</pre>
toService	string	<p>システム値は必須</p> <p>マップ先:</p> <pre>ServiceHeader/ProcessControl /ServiceRoute/toService /BusinessServiceDescription /GlobalBusinessServiceCode</pre>

表 2-2 RNIF 1.1 のテンプレート変数 (続き)

名前	型	使い方と説明
toSupplyChain	string	<p>入力値は必須 メッセージ確認応答の作成に使用される。 マップ先: ReceiptAcknowledgement/toRole /PartnerRoleDescription/PartnerDescription /BusinessDescription/GlobalSupplyChainCode ReceiptAcknowledgmentException/toRole /PartnerRoleDescription /PartnerDescription/BusinessDescription /GlobalSupplyChainCode Exception/toRole/PartnerRoleDescription /PartnerDescription /BusinessDescription/GlobalSupplyChainCode</p>
transactionCode	string	<p>システム値は必須 マップ先: ServiceHeader/ProcessControl /TransactionControl/TransactionIdentity /GlobalTransactionCode</p>
useDTDValidation	boolean	<p>システム値は省略可能 True に設定した場合、XSD ではなく DTD 検証が使用される。False に設定した場合または変数が存在しない場合、検証には XSD スキーマが使用される。そもそも検証を実行するかどうかは、validateServiceContent および validateServiceHeader の値に応じて決定される。</p>
validateServiceContent	boolean	<p>入力値は必須 サービス コンテンツをスキーマに照らし合わせて検証する必要があるかどうかを示すフラグ。 True - 検証は必要 False - 検証は不要</p>

表 2-2 RNIF 1.1 のテンプレート変数 (続き)

名前	型	使い方と説明
<code>validateServiceHeader</code>	<code>boolean</code>	入力値は必須 サービス ヘッダをスキーマに照らし合わせて検証する必要があるかどうかを示すフラグ。 True - 検証は必要 False - 検証は不要
<code>validationError</code>	<code>boolean</code>	システム値は省略可能 検証結果を示す。

統合された rnSystem 変数

WebLogic Integration 2.0 では、次の変数も必要とされていました。

```
signalCode  
signalCodeVersion  
inReplyToActionCode  
inReplyToMessageId  
initiatingPartnerDUNS  
inReplyToActionCodeVersion (RNIF 2.0 のみ)
```

以降のリリースでは、これらの変数は、ワークフロー テンプレート内で公開されているシステム変数の数を低減するために、rnSystem という変数（ワークフロー変数型: Java Object）に統合されて置き換えられます。

WebLogic Integration 2.0 用に開発された PIP ワークフローがあれば、既存のワークフロー テンプレートからこれらの変数の呼び出しまたは定義を削除して、rnSystem に置き換えることができます。

RosettaNet メッセージの受信

WebLogic Integration がサポートする RosettaNet メッセージの受信方法には、開始ノードによるものと、イベント ノードによるものの、2通りがあります。どちらを使用するかは、メッセージが受信される状況によります。

開始ノード

WebLogic Collaborate が PIP インスタンスに対するメッセージを初めて受信したときに、WebLogic Collaborate が自動的にワークフローを開始するように、ワークフローをコンフィグレーションできます。このアクションをコンフィグレーションするには、受信した PIP を開始ノードのイベントとして宣言します。受信するワークフローが、受信される PIP を開始して、処理します。例としては、『*WebLogic Integration の起動、停止およびカスタマイズ*』の「協調的ワークフローの開始」にある参加者ワークフローの説明を参照してください。

PIP3A2 サプライヤ ワークフロー テンプレートは、受信されるメッセージによって開始されるワークフロー テンプレート 定義のコンフィグレーション方法の一例です。この例では、開始イベントは **Business Message Event** として設定されています。RosettaNet メッセージが受信されると、ワークフローが自動的に開始され、出力変数とシステム変数が設定されます。

イベント ノード

ワークフローには、ワークフローと関連付けられた PIP インスタンスに対するメッセージが受信されるとトリガされるイベントを入れることができます。例としては、『*WebLogic Integration の起動、停止およびカスタマイズ*』の「協調的ワークフローの開始」にある開始者ワークフローの説明を参照してください。

タイムアウトの実装

ワークフローでは、受信される RosettaNet メッセージを待機するオプションのタイムアウト パスを使用できます。ワークフローが送信されたメッセージ（たとえば 3A2 カスタマ ワークフロー）に対する応答を待機する場合は、応答を待機するための別個のタイムアウト パスを作成する必要があります。3A2 カスタマ ワークフロー テンプレートに示されるこのパスは、適切なタイムアウト期間に合わせて設定されたタイマと、停止ノードで構成されます。

順不同の信号受信

RNIF 1.1 と RNIF 2.0 では、信号受信と応答について、異なった標準を定義しています。これらの異なった標準は、PIP ワークフローによるメッセージの解決方法に影響を与えます。

RNIF 1.1 では、応答が常に信号の後に来るように指定しています。したがって、信号/応答パターンは、次のようになります。

1. 開始者 → (要求) → 受信者
2. 受信者 → (受信応答確認) → 開始者

3. 受信者→（応答）→開始者
4. 開始者→（応答確認処理）→受信者

RNIF 2.0 により、順不同の信号受信が可能になります。たとえば、ワークフローが、受信応答確認より前に応答を受信する場合があります。したがって、RNIF 2.0 では、次のような信号 / 応答パターンが可能です。

1. 開始者→（要求）→受信者
2. 受信者→（応答）→開始者
3. 受信者→（受信応答確認）→開始者
4. 開始者→（受信応答確認）→受信者

このパターンは、ワークフロー設計の中で正確に処理される必要があります。PIP3A2 顧客ワークフロー テンプレートは、これを処理する方法の一例として有用です。PIP テンプレートの RNIF 1.1 バージョンは、信号と、それに続く応答を処理します。RNIF 2.0 バージョンは、信号と応答を個別に処理するロジックを提供します。

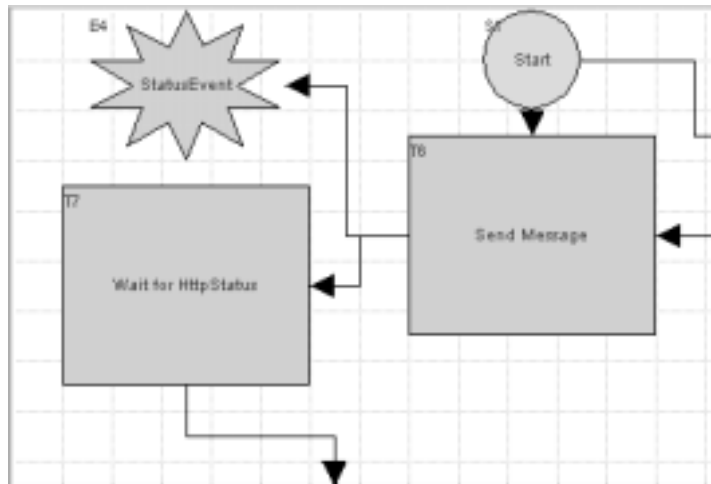
RosettaNet メッセージの送信

ワークフロー タスクは次のノードを使用して RosettaNet メッセージを送信します。

- Send Business Message タスク
- RosettaNet Status Event 型イベント ノード
- Wait for HTTP Status タスク

PIP3A2_Customer_RN2 ワークフローの一部を次の図に示します。

図 2-2 RosettaNet メッセージの送信に使用されるワークフロー ノード



RosettaNet メッセージの伝送は、**Send Business Message** タスクがメッセージを送信すると開始されます。イベント ノード (RosettaNet Status Event) はメッセージの **http** ステータス通知を待機します。この種のステータス通知を受信すると、イベント ノードは **Wait for HTTP Status** タスク ノードに「完了」マークを付け、ワークフローは次のノードへと進みます。

Send Business Message タスクは同期で実行され、**http** ステータス応答を待ってからワークフローの次のノードへ進むということをしません。

RosettaNet メッセージは次の方法で送信します。

1. **Send Message** タスクを右クリックし、次にショートカット メニューから [プロパティ] を選択します。
[タスクのプロパティ] ダイアログ ボックスが表示されます。
2. [追加] ボタンをクリックして [アクションを追加] ダイアログ ボックスを表示します。
3. 次の図のように、[統合アクション | B2B Integration | Send Business Message] を選択します。

図 2-3 B2B Integration アクション



4. [OK] をクリックして、次の図に示す [Send Business Message] ダイアログボックスを表示します。

図 2-4 RosettaNet 2.0 の [Send Business Message] ダイアログ ボックス



要求される情報は、選択したメッセージタイプ ([アクション]、[例外]、または[受信確認]) によって異なります。次の表で、入力を求められる情報をメッセージタイプ別にまとめます。

表 2-3 メッセージタイプおよびオプション

メッセージタイプ	フィールド名	説明
Action	[メッセージタイプ] – [アクション]	RosettaNet Action Business Message が送信されることを示す。
	[入力内容変数]	サービスコンテンツを表現する XML データを含むワークフロー XML 変数。表 2-1 (RNIF 2.0) または表 2-2 (RNIF 1.1) の PIPInput 変数を参照。
	[入力添付ファイル記述子変数]	RosettaNet メッセージの一部として送信される添付ファイルを説明する XML データを含む、ワークフロー XML 変数 (2-35 ページの「メッセージ添付ファイル」および表 2-1 (RNIF 2.0) または表 2-2 (RNIF 1.1) の attachmentDescriptorInput 変数を参照)。
Receipt Acknowledgment	[メッセージタイプ] – [受信確認]	RosettaNet Receipt Acknowledgment 信号が送信されることを示す。これ以外のデータ値は不要。
例外	[メッセージタイプ] – [例外]	RosettaNet Receipt Exception Acknowledgment 信号が送信されることを示す。
	[エラーコード]	(RosettaNet によって定義された) 送信されるエラーコード。
	[エラーの説明]	エラーの簡単な説明。

5. **Status Event** ノードを右クリックし、次にショートカットメニューから [プロパティ] を選択します。

次の図に示す [イベントのプロパティ] ダイアログボックスが表示されます。

図 2-5 RosettaNet Status Event の [イベントのプロパティ] ダイアログ ボックス



必要な情報の種類は、ユーザが選択した Event Type によって異なります。次の表は、RosettaNet ステータス イベントを定義するときに通知される情報をまとめたものです。

表 2-4 RosettaNet Status Event オプション

フィールド	説明
[説明]	イベントの説明。説明は、ワークフローのイベントノードのラベルとして表示されます。
[タイプ]	設定されるイベントのタイプです。メニューから RosettaNet Status Event を選択します。
[Output Status Variable]	メッセージの送信ステータスが保存される変数です。

6. [アクション] タブを選択し、[追加] ボタンをクリックして [アクションを追加] ダイアログ ボックスを表示します。
7. [タスク アクション | タスクに完了マークを付ける] を選択します。次に、[OK] をクリックして [タスクに完了マークを付ける] ダイアログ ボックスを表示します。
8. [Wait for Http Status] タスクを選択して完了マークを付け、[OK] をクリックします。
9. [Wait for Http Status] タスクを右クリックし、次にショートカット メニューから [プロパティ] を選択します。
[タスクのプロパティ] ダイアログ ボックスが表示されます。
10. [追加] ボタンをクリックして [アクションを追加] ダイアログ ボックスを表示します。
11. [タスク アクション | タスク期日を設定] を選択します。次の図に示すように、[タスク期日を設定] ダイアログ ボックスが表示されます。

図 2-6 [タスク期日を設定] ダイアログ ボックス



12. [式に設定] フィールドに、タイムアウトまでにアプリケーションを待機させる時間数を定義する式を入力します。次に [ビジネス カレンダー] メニューから適切なビジネス カレンダーを選択します。[OK] をクリックします。[タスクのプロパティ] ダイアログ ボックスが表示されます。
13. [タスクのプロパティ] ダイアログ ボックスで [OK] をクリックします。ワークフローの中で RosettaNet メッセージの送信に使用される部分はこれで完了です。

メッセージの検証

メッセージ検証プロセスは、XML スキーマ仕様のアルファ版実装をサポートする Xerces 1.3 DOM パーサを使用します。Xerces 1.3 DOM パーサは、WebLogic Server 6.1 ソフトウェアに同梱されています。

以下の節では、WebLogic Collaborate がどのように RosettaNet メッセージを検証するかを説明し、メッセージ検証に関するさらなる参考文献を提示します。

- RosettaNet メッセージ検証
- メッセージ検証に関する推奨参考文献

RosettaNet メッセージ検証

WebLogic Integration は、RNIF 1.1 のメッセージと RNIF 2.0 メッセージの両方に対して、メッセージ検証サービスを提供します。実行される検証は `validateServiceContent`、`validateServiceHeader`、および `useDTDValidation` の各ワークフロー変数に指定された値によって異なります。検証オプションは、以下のとおりです。

- なし - `validateServiceHeader` および `validateServiceContent` が `false` に設定されている場合は、検証は実行されません。
- サービス ヘッダ検証 - `validateServiceHeader` が `true` に設定されている場合は、あるテンプレートについて送受信されるすべてのメッセージのサービス ヘッダが検証されます。実行される検証の種類は、`useDTDValidation` 変数の設定によって異なります。
- サービス コンテンツ検証 - `validateServiceContent` が `true` に設定されている場合は、あるテンプレートについて送受信されるすべてのメッセージのサービス コンテンツが検証されます。実行される検証の種類は、`useDTDValidation` 変数の設定によって異なります。
- DTD - `useDTDValidation` が `true` に設定されている場合は、DTD の検証が実行されます。DTD ファイルは PIP 仕様の一部として提供されており、www.rosettanet.org からダウンロード可能です。

- XSD スキーマー useDTDValidation が false に設定されているかワークフロー テンプレートに存在しない場合は、XSD スキーマの検証が実行されません。XSD 検証を使用するには、PIP 用の DTD から、独自のスキーマファイルを作成する必要があります。

例外処理プロセスの説明については、次の URL にある RNIF 仕様を参照してください。

<http://www.rosettanet.org>

サンプルの XML スキーマ ファイルおよび文書型定義 (DTD) ファイルは、WebLogic Integration のインストール先の、次の場所に格納されています。

`WLI_HOME/lib/xmlschema/rosettanet`

メッセージ検証に関する推奨参考文献

以下は、XML スキーマ例を十分に理解するために推奨される参考文献の情報です。独自の XML スキーマを実装する場合は、必ずお読みください。

- XML スキーマのツール、使用法、仕様、および開発に関する情報については、次の URL を参照してください。

<http://www.w3.org/XML/Schema>

「*XML Schema Part 0: Primer*」には、XML スキーマの特長と機能についての詳しい説明があります。

- XML スキーマ仕様の Xerces 実装に関する情報については、次の URL を参照してください。

<http://xml.apache.org/xerces-j/schema.html>

パフォーマンス チューニングとメッセージ検証

メッセージ検証は、主にシステムをパートナーと共に設定およびコンフィグレーションしているときに使用されます。無効なメッセージ生成に関する問題をシステムから取り除いたと確信している場合は、オプションとしてメッセージ検証をオフにし、パフォーマンスを向上させることができます。前の節で述べたように、メッセージ処理中にメッセージ検証の実行が要求されることはありません。

メッセージ添付ファイル

RNIF 1.1 と RNIF 2.0 は双方とも、RosettaNet アクション メッセージ内のオプションのメッセージ添付ファイルをサポートしています。添付ファイルは、ファイルタイプを限定しておらず、バイナリ データを含んでいてもかまいません。添付できるファイルの例としては、Word 文書、GIF 画像、PDF ファイルなどがあります。各添付ファイルの情報は、メッセージのサービス ヘッダに含まれます。

WebLogic Integration は、ユーザ アプリケーション（たとえばプライベート ワークフロー）が、BEA 固有の構造化された XML ファイルに含まれる添付ファイルの説明を PIP ワークフローへの入力として提供できるようにすることにより、添付ファイルをサポートしています。XML ファイルは、添付ファイルの説明です。実際の添付ファイルではありません。WebLogic Integration によって添付されるべきファイルを指定するだけのものです。

以下は、添付ファイルを説明する DTD 情報です。

コード リスト 2-1 RosettaNet 用 WebLogic Integration 添付ファイル DTD 情報

```
<!ELEMENT WLCRosettaNet ( Attachment+ )>
<!ELEMENT Attachment (
    description?,
    Type,
    Id,
    LocalLocation
)>
<!ELEMENT description ( FreeFormText ) >
<!ELEMENT FreeFormText ( #PCDATA ) >
<!ATTLIST FreeFormText xml:lang CDATA #IMPLIED >
<!ELEMENT Type ( #PCDATA ) >
<!ELEMENT Id ( #PCDATA ) >
<!ELEMENT LocalLocation ( #PCDATA ) >
```

次の表で、DTD において使用される要素について説明します。

表 2-5 RosettaNet 添付ファイル要素

要素	説明
説明	添付ファイルの説明（省略可能）。
Type	添付ファイルの MIME 型修飾子コード。
ID	添付ファイルの Universal Resource Identifier。
LocalLocation	送信時： 添付ファイルとして送信するファイルの（ローカル システムにおける）絶対パスを格納する。 受信時： 受信した添付ファイルの（ローカル システムにおける）絶対パスを格納する。

メッセージが受信されると、添付ファイルはすべてローカルの `WLI_HOME/config/domain_name/rnattachments` ディレクトリに保存されます。保存された添付ファイルには、先頭がタイムスタンプになったファイル名が付きます。たとえば、添付ファイルに次の XML 記述を使用したとします。

コード リスト 2-2 サンプル XML 添付ファイル

```
<?xml version="1.0"?>
<!DOCTYPE WLCRosettaNet SYSTEM "WLCRosettaNet.dtd">
<WLCRosettaNet>
  <Attachment>
    <description>
      <FreeFormText>
        Product user guide in PDF
      </FreeFormText>
    </description>
    <Type>application/pdf</Type>
    <Id>"001801236324xyz@xyz.test.com"</Id>
    <LocalLocation>c:\pdf\myfile.pdf</LocalLocation>
  </Attachment>
  <Attachment>
    . . .
  </Attachment>
</WLCRosettaNet>
```

このサンプルは、サービス ヘッダおよび MIME ヘッダ内に次のデータを生成します。

コード リスト 2-3 サンプル サービス出力

```
<Attachment>
  <description>
    <FreeFormText>
      Product user guide in PDF
    </FreeFormText>
  </description>
  <GlobalMimeTypeQualifierCode>
    PDF
  </GlobalMimeTypeQualifierCode>
  <UniversalResourceIdentifier>
    cid:Attachment.001801236324xyz@xyz.test.com
  </UniversalResourceIdentifier>
</Attachment>
```

コード リスト 2-4 サンプル MIME ヘッダ

```
Content-Type: application/pdf; name="myfile.pdf"
Content-ID: <Attachment.001801236324xyz@xyz.test.com>
Content-Description: Product user guide in PDF
```

3 RosettaNet PIP テンプレート

WebLogic Integration は、2つの PIP テンプレートを備えています。これらを使うと、RosettaNet ベースのソリューションを即座に実装できます。この章では、各 PIP テンプレートの内容と、PIP を実装するために必要な情報の双方について説明します。どちらの PIP についても RNIF 1.1 用のバージョンと RNIF 2.0 用のバージョンが提供されています。

RosettaNet テンプレートのインポート

次の XML テンプレート ファイルは、`WLI_HOME/lib/templates/rosettanet` ディレクトリに格納されています。

- PIP0A1_Admin_RN1.xml
- PIP0A1_Admin_RN2.xml
- PIP0A1_Notifier_RN1.xml
- PIP0A1_Notifier_RN2.xml
- PIP3A2_Customer.xml
- PIP3A2_Customer_RN2.xml
- PIP3A2_Customer_RN2_R02_00.xml
- PIP3A2_Supplier.xml
- PIP3A2_Supplier_RN2.xml
- PIP3A2_Supplier_RN2_R02_00.xml

提供されているテンプレートの 1 つを使用する手順は、次のとおりです。

このパッケージには、Get RN timestamp ビジネス処理が含まれています。これは、RN 固有フォーマットでタイムスタンプを返す処理です。PIP0A1 メッセージのサービス コンテンツにデータを入れるために使用されます。詳細については、顧客ワークフローの Start Notification of Failure タスクを参照してください。

PIP0A1: 障害通知

この PIP は、すべての実装に必要なシステム レベルの RosettaNet PIP です。プロセス障害の発生時に情報を送信するために使用されます。通常は、大幅なカスタマイズをする必要はありません。むしろ、この PIP に呼び出し元の PIP ワークフローからデータが渡されます。このため、PIP0A1 ワークフローの呼び出し前に、呼び出し元ワークフローで適切なワークフロー変数が設定されている必要があります。特に、2-5 ページの「RosettaNet ワークフロー変数」で説明した変数を設定して、PIP0A1_Admin テンプレートにインポートする必要があります。入力変数はすべて、ワークフロー呼び出し時に初期化する必要があります。

PIP3A2: 価格と在庫に関するクエリ

PIP3A2 は、PIP の実装方法の例としてテンプレートの形式で提供されています。この PIP には RNIF 1.1 用に実装されたものと RNIF 2.0 用に実装されたものの、2つのバージョンがあり、各バージョンは送信者用にも受信者用にもモデル化されています。ご使用の RosettaNet のバージョンに適したバージョンの PIP テンプレートを使用してください。

PIP3A2 を実装する場合は、必須の入力変数を設定する必要があります。これらの変数はすべて 2-5 ページの「RosettaNet ワークフロー変数」で説明しています。これらの変数は、PIP ワークフロー内から設定することも、呼び出し元ワークフローによって設定することも可能です。入力変数はすべて、ワークフロー呼び出し時に初期化する必要があります。

他の PIP のモデル化

PIP3A2 テンプレートは、WebLogic Integration において RosettaNet PIP がどのようにサポートされるかを例示するために提供されています。PIP は複雑なので、BEA は新しい PIP を作成する場合には PIP3A2 のソース ファイルをコピーし、このコピーを修正して新しい PIP を作成することをお勧めしています。

他の PIP をモデル化する際には、次の点に注意してください。

- この PIP とその他のプロセスの間でデータを転送するには、入力ワークフロー変数と出力ワークフロー変数を使用します。PIP の入力変数は別のワークフローからデータを受信するために、出力変数は別のワークフローへデータを送信するために使用します。PIP ワークフロー開始時にはすべての必須入力変数を必ず初期化してください。
- 他のワークフローで、PIP ワークフローを開始できます。このタイプの開始のコンフィグレーションを行うには、[アクション | B2B Integration | パブリック ワークフローを開始] を選択します。この方法を使用できるのは、PIP トランザクションを開始するトレーディング パートナであるユーザーです。この方法を使用する場合は、必ず 2-5 ページの「RosettaNet ワークフロー変数」に示した PIP ワークフロー変数をすべて、手動で入力してください。
- PIP ワークフローで、プライベート ワークフローなど、他のワークフローを開始できます。このタイプの開始のコンフィグレーションを行うには、[アクション | ワークフローを開始] を選択します。この方法を使用できるのは、PIP トランザクションを受信するトレーディング パートナであり、かつそのデータをプライベート ワークフローに渡そうとしているユーザーです。プライベート ワークフローは、たとえばバックエンド ERP システムへの接続に使用します。

索引

D

- DTD 1-2
 - WebLogic Integration 内 1-9

P

- PIP
 - PIP0A1 テンプレート 3-3
 - PIP3A2 テンプレート 3-3
 - マニュアル 2-2
 - PIP3A2 テンプレート 3-3
 - PIP テンプレート 1-4
 - インポート 3-1
 - PIP ワークフロー テンプレートのインポート 3-1
 - PIP ワークフロー、メッセージ交換 2-3
 - PIP0A1 テンプレート 3-3

R

- RNIF マニュアル 2-2
- RosettaNet 2-2
 - RosettaNet メッセージ 1-2
 - セキュリティ 1-6
 - メッセージ検証の概要 1-6
 - メッセージの検証 2-33
 - メッセージの受信 2-23
 - メッセージの送信 2-25
 - ワークフロー変数 2-5
- RosettaNet 2.0 セキュリティ サンプル 1-6
- RosettaNet オブジェクト 1-2
 - 要素 2-5
- RosettaNet サポートの設定 1-9
- RosettaNet メッセージの検証 2-33
- RosettaNet メッセージの受信 2-23
- RosettaNet メッセージの送信 2-25

W

- WebLogic Integration における RosettaNet 1-2

X

- Xerces DOM パーサ 2-33

あ

- アーキテクチャ 1-4
 - RosettaNet サポート 1-3

い

- イベント ノード 2-23
- 印刷、製品のマニュアル vi

か

- 開始ノード 2-23
- カスタマ サポート 情報 vii
- 環境 1-9
- 管理 1-8

き

- 協調的ワークフロー 2-3

け

- 検証マニュアル 2-34

さ

- サポート
 - テクニカル vii

サンプル アプリケーション 1-6

し

システム変数 2-5

出力変数 2-5

す

推薦参考文献

RosettaNet Implementation Framework
(RNIF) 2-2

技術勧告 2-2

メッセージ検証 2-34

せ

セキュリティ 1-6

た

タスク プロパティ 2-25

て

定義

PIP ワークフロー 1-5

会話定義 1-8

コラボレーション アグリーメント 1-8

トレーディング パートナ 1-8

デジタル署名 1-6

添付ファイル 2-35

に

入力変数 2-5

の

ノード

イベント 2-23

開始 2-23

は

パーサ、Xerces DOM 2-33

パフォーマンス 2-34

パブリック ワークフロー 2-3

ひ

ビジネス メッセージ

受信 2-23

送信 2-25

ビジネス メッセージ送信アクション 2-25

ふ

プライベート ワークフロー 2-3

へ

変数

RosettaNet、必須 2-5

システム 2-5

出力 2-5

入力 2-5

ワークフロー 2-5

ま

マニュアル

PIP 2-2

RosettaNet Implementation Framework
(RNIF) 2-2

メッセージ検証 2-34

マニュアル入手先 vi

め

メッセージ

RosettaNet メッセージの検証 2-33

RosettaNet メッセージの受信 2-23

RosettaNet メッセージの送信 2-25

検証マニュアル 2-34

メッセージ検証 2-34

メッセージ添付ファイル 2-35

メッセージのガイドライン 1-2

わ

ワークフロー

開始 2-23

協動的 2-3

パブリック 2-3

プライベート 2-3

変数 2-5

変数テーブル 2-5

メッセージ交換 2-3

ワークフロー テンプレート 1-4

PIP0A1 3-3

PIP3A2 3-3

PIP テンプレートのインポート 3-1

ワークフローの開始 2-23

