



# BEA WebLogic Integration™

## Implementing RosettaNet for B2B Integration

## Copyright

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

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. 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 Portal, BEA WebLogic Server and How Business Becomes E-Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

## Implementing RosettaNet for B2B Integration

<b>Part Number</b>	<b>Date</b>	<b>Software Version</b>
N/A	January 2002	2.1 Service Pack 1

---

# Contents

## About This Document

What You Need to Know .....	v
e-docs Web Site .....	vi
How to Print the Document .....	vi
Contact Us! .....	vii
Documentation Conventions .....	viii

## 1. Introduction

About RosettaNet .....	1-2
RosettaNet Architecture .....	1-2
WebLogic Integration Architecture and RosettaNet .....	1-3
RosettaNet Protocol Layer .....	1-4
PIP Templates .....	1-4
Workflow Definition .....	1-5
Digital Signatures .....	1-5
Message Validation .....	1-5
RosettaNet Sample .....	1-6
Unsupported Items .....	1-6
RosettaNet Administration .....	1-7
Configuring RosettaNet Security .....	1-7
Configuring SSL and Digital Signatures .....	1-7
Message Encryption .....	1-8
Setting Up the Environment .....	1-8

## 2. Using Workflows with RosettaNet

Understanding RosettaNet .....	2-2
Understanding PIP Workflow Instances .....	2-3

---

Getting Started .....	2-4
Working with Workflows .....	2-5
RosettaNet Workflow Variables .....	2-5
Using the Workflow Variable Tables .....	2-6
Usage .....	2-6
Description .....	2-7
Workflow Variables for RNIF 2.0 .....	2-8
Workflow Variables for RNIF 1.1 .....	2-15
Consolidated rnSystem Variable .....	2-25
Receiving a RosettaNet Message .....	2-25
Start Nodes .....	2-25
Event Nodes .....	2-26
Implementing Timeouts .....	2-26
Out of Sequence Reception of Signals .....	2-26
Sending a RosettaNet Message .....	2-27
Validating a Message .....	2-31
RosettaNet Message Validation .....	2-31
Recommended Reading About Message Validation .....	2-32
Performance Tuning and Message Validation .....	2-32
Message Attachments .....	2-33

### 3. RosettaNet PIP Templates

Importing a RosettaNet Template .....	3-1
RosettaNet Business Operations .....	3-2
PIP0A1: Notification of Failure .....	3-3
PIP3A2: Query Price and Availability .....	3-3
Modeling Other PIPs .....	3-4

## Index

---

# About This Document

This document describes how to implement RosettaNet in a BEA WebLogic Integration™ B2B application.

This document is organized as follows:

- Chapter 1, “Introduction,” provides an introduction to RosettaNet support in WebLogic Integration, and describes the architecture used to implement RosettaNet in WebLogic Integration B2B applications.
- Chapter 2, “Using Workflows with RosettaNet,” describes how WebLogic Integration business process management (BPM) workflows are used in RosettaNet applications.
- Chapter 3, “RosettaNet PIP Templates,” describes the two RosettaNet PIP templates provided with WebLogic Integration, and explains how to create more PIPs of your own.

## What You Need to Know

This document is intended primarily for:

- Business process designers who use the WebLogic Integration Studio to design workflows that conform to the RosettaNet standards for the exchange of business messages.
- System administrators who set up and administer WebLogic Integration RosettaNet solutions.

---

Before reading this document, we recommend that you read the following documents:

- *Introducing BEA WebLogic Integration*
- *Learning to Use BEA WebLogic Integration*
- *Starting, Stopping, and Customizing BEA WebLogic Integration*
- *Introducing B2B Integration*
- *Administering B2B Integration*
- *Implementing Security with B2B Integration*

Before you begin designing your own RosettaNet solutions, we recommend that you read the following documents:

- *Designing BEA WebLogic Integration Solutions*
- *Deploying BEA WebLogic Integration Solutions*

## e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the “e-docs” Product Documentation page at <http://e-docs.bea.com>.

## How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available from the BEA WebLogic Integration documentation Home page, which is available on the documentation CD and on the e-docs Web site at <http://e-docs.bea.com>. You can open the PDF in Adobe Acrobat

---

Reader and print the entire document, or a portion of it, in book format. To access the PDFs, open the BEA WebLogic Integration documentation Home page, click the PDF Files button, and select the document you want to print.

If you do not have the Adobe Acrobat Reader installed, you can download it for free from the Adobe Web site at <http://www.adobe.com/>.

## Contact Us!

Your feedback on the BEA WebLogic Integration documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the WebLogic Integration documentation.

In your e-mail message, please indicate which release of the WebLogic Integration product documentation you are using.

If you have any questions about this version of BEA WebLogic Integration, or if you have problems installing and running BEA WebLogic Integration, contact BEA Customer Support through BEA WebSUPPORT at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

---

# Documentation Conventions

The following documentation conventions are used throughout this document.

<b>Convention</b>	<b>Item</b>
<b>boldface text</b>	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.
monospace text	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard. <i>Examples:</i> <pre>#include &lt;iostream.h&gt; void main ( ) the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> <pre>String expr</pre>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> <pre>LPT1 SIGNON OR</pre>
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.



---

<b>Convention</b>	<b>Item</b>
[ ]	<p>Indicates optional items in a syntax line. The brackets themselves should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...</pre>
	<p>Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.</p>
...	<p>Indicates one of the following in a command line:</p> <ul style="list-style-type: none"> <li>■ That an argument can be repeated several times in a command line</li> <li>■ That the statement omits additional optional arguments</li> <li>■ That you can enter additional parameters, values, or other information</li> </ul> <p>The ellipsis itself should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name ] [-f file-list]... [-l file-list]...</pre>
.	<p>Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.</p>

---



# 1 Introduction

WebLogic Integration B2B supports a routing architecture that allows it to resolve and manage the exchange of XOCP, RosettaNet, and cXML messages. This architecture allows trading partners using WebLogic Integration to exchange business messages using any of these protocol standards.

“[Configuration Requirements](#)” in *Administering B2B Integration* provides examples of the B2B configuration required to support each of these protocols. When configured to support RosettaNet, the B2B engine allows trading partners to send and receive RosettaNet messages as described in *RosettaNet Implementation Framework 1.1* (RNIF 1.1) and *RosettaNet Implementation Framework 2.0* (RNIF 2.0). In addition, WebLogic Integration business process management (BPM) supports the creation and execution of workflows that model RosettaNet Partner Interface Processes (PIPs). WebLogic Integration includes two PIP templates (0A1 and 3A2), which you can use as the basis for your own PIP implementations.

The following sections provide an overview of the RosettaNet protocol:

- About RosettaNet
- RosettaNet Architecture
- RosettaNet Administration
- Configuring RosettaNet Security
- Setting Up the Environment

## About RosettaNet

This section introduces the RosettaNet standard for electronic business transactions. The RosettaNet Consortium is an independent, nonprofit consortium of major information technology, electronic component, and semiconductor manufacturing companies working to create and implement industry-wide, open e-business process standards. These processes are designed to standardize the electronic business interfaces used between participating supply chain partners. The *RosettaNet Implementation Framework* specification (available at <http://www.rosettanet.org>) is a guideline for applications that implement RosettaNet *Partner Interface Processes* (PIPs). These PIPs are standardized electronic business processes used between trading partners.

A machine-readable RosettaNet message is known as a *RosettaNet Object* in RNIF 1.1 and a *RosettaNet Business Message* in RNIF 2.0. Some elements of RosettaNet messages are common across all RosettaNet messages, while other elements are unique to specific PIPs. To ensure that RosettaNet messages are structured and processed in a consistent manner, each PIP comes with a message guideline and XML document type definition (DTD).

## RosettaNet Architecture

WebLogic Integration support for RosettaNet consists of the following components:

- *RosettaNet protocol layer*  
The business protocol definitions that implement B2B engine support for the RosettaNet protocol.
- *B2B integration plug-in*  
The plug-in which extends the functionality of the WebLogic Integration Studio and process engine to support the modeling and execution of workflows that implement standard RosettaNet PIPs.
- *PIP templates*  
Workflow templates that can be customized to implement standard RosettaNet PIPs for you and your trading partners.

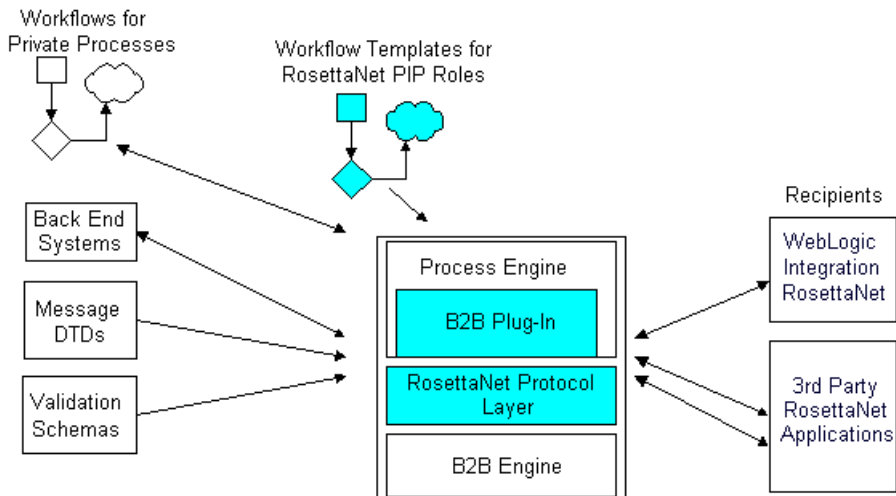
# WebLogic Integration Architecture and RosettaNet

The WebLogic Integration Studio allows you to model and monitor the execution of workflows that implement RosettaNet PIPs. The workflows, process engine, B2B plug-in, and B2B engine work in concert to provide the following capabilities:

- Trading partner delivery channel definitions are bound to the RosettaNet protocol as part of the B2B configuration.
- Conversation definitions and collaboration agreements defined in the B2B configuration associate delivery channels with the appropriate workflows.
- Workflow actions send RosettaNet messages that are routed through the B2B engine to remote trading partner delivery channels.
- Workflow template properties indicate which PIP and role the workflow implements.
- Workflow events wait for RosettaNet messages sent by the B2B engine or by private workflow actions.

The following diagram shows how WebLogic Integration supports the RosettaNet e-business process standards.

**Figure 1-1 WebLogic Integration RosettaNet Architecture**



The following sections provide additional information about WebLogic Integration support for RosettaNet, and the components illustrated.

## RosettaNet Protocol Layer

The RosettaNet protocol layer provides the ability to send and receive messages by way of the Internet according to the RNIF 1.1 and RNIF 2.0 specifications for transport, message packaging, and security.

## PIP Templates

RosettaNet PIPs define the *public processes* in which trading partners participate while performing e-business transactions. For example, PIP 3A2 defines the process that a *Customer* trading partner performs with a *Product Supplier* trading partner to get information about the price and availability of goods that the *Customer* wants to buy and the *Product Supplier* wants to sell. Trading partners participating in PIPs need to implement the public process defined by their role in the PIP, and they need to connect their internal systems, as well as their private processes and workflows, to the public process.

A key feature of the support for RosettaNet provided by WebLogic Integration is the set of workflow PIP templates that trading partners can use to implement PIPs. Template sets are defined for the following PIPs:

- PIP0A1: Notification of Failure
- PIP3A2: Query Price and Availability

Templates for each PIP role and RosettaNet version are found in the following directory:

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

Where, `WLI_HOME` represents the WebLogic Integration installation location.

For further information about these and other PIPs, go to the RosettaNet Web site: <http://www.rosettanet.org>. For more information on the RosettaNet PIPs provided with this release of WebLogic Integration, see Chapter 3, “RosettaNet PIP Templates.”

## Workflow Definition

A workflow that implements a RosettaNet PIP is simply a specialized collaborative workflow. Just as it is used to create any collaborative workflow, the Studio, with the extended functionality provided by the B2B plug-in, is used to develop workflows that implement a RosettaNet PIP. For general information about using the Studio to create collaborative workflows, see *Creating Workflows for B2B Integration*. The information specific to the development of RosettaNet PIPs is provided in Chapter 2, “Using Workflows with RosettaNet.”

## Digital Signatures

WebLogic Integration includes an out-of-the-box implementation of digital signatures, based on the RSA CertJ toolkit. For more information about implementing and configuring digital signatures, see *Implementing Security with B2B Integration*.

WebLogic Integration also provides a sample application that showcases RosettaNet security functionality. For information about this sample, see “[RosettaNet 2.0 Security Sample](#)” in *Running the B2B Integration Samples*.

## Message Validation

The RosettaNet PIP definitions contain detailed validation rules for messages exchanged in the PIP. These rules are significantly more stringent than the validation expressed within an XML Document Type Definition (DTD).

The required validation rules are expressed in XML schema documents (XSD), which are included with the PIP templates provided with WebLogic Integration. For a more detailed description of message validation, see Chapter 2, “Using Workflows with RosettaNet.”

## RosettaNet Sample

The BEA RosettaNet 2.0 security sample demonstrates how WebLogic Integration can be used to implement a RosettaNet 2.0 PIP using workflows. Specifically, it shows two trading partners exchanging business messages that conform to the RosettaNet 2.0 PIP 3A2 standard. For information about running this sample, see [“RosettaNet 2.0 Security Sample”](#) in *Running the B2B Integration Samples*.

As additional samples become available for this release, they will be posted on the BEA Developer Center Web site at the following URL:

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

## Unsupported Items

Several RosettaNet-related features are not supported in this release of WebLogic Integration:

- *SMTP Transport: RNIF 2.0*  
While strongly biased to HTTP transport, the RNIF 2.0 is transport independent and includes documentation showing how SMTP transport might be used. WebLogic Integration supports HTTP and HTTPS transport; it does not support SMTP.
- *WebLogic Collaborate 1.0 Backward Compatibility*  
Due to a number of changes, PIP implementations developed for WebLogic Collaborate 1.0 are not supported in WebLogic Integration. If you have created PIP implementations under WebLogic Collaborate 1.0, you must reimplement them for WebLogic Integration. Workflows created for WebLogic Integration 2.0 must be migrated as described in *Migrating to BEA WebLogic Integration Release 2.1*.
- *Hub-mediated messaging*  
WebLogic Integration 2.1 supports the RosettaNet peer-to-peer messaging model only. The hub-mediated model, where a RosettaNet hub delivery channel is configured to act as an intermediary between RosettaNet trading partner peers, is no longer supported.



# RosettaNet Administration

In WebLogic Integration, trading partner definitions, conversation definitions, and collaboration agreements constitute the user-defined elements required to implement the exchange of messages between trading partners.

For information about using the B2B Console to configure and monitor these elements, see *Administering B2B Integration* and the context-sensitive B2B Console online help. A RosettaNet sample configuration is provided in “[RosettaNet Applications](#)” in *Administering B2B Integration*.

**Note:** The *Online Help for the WebLogic Integration B2B Console* document is also included in the WebLogic Integration product documentation.

## Configuring RosettaNet Security

The following sections describe how to configure security for the RosettaNet business protocol:

- Configuring SSL and Digital Signatures
- Message Encryption

## Configuring SSL and Digital Signatures

Both SSL and digital signature security for RosettaNet are configured through the B2B Console. For more information about configuring SSL and digital signatures, see *Implementing Security with B2B Integration*, the *Online Help for the WebLogic Integration B2B Console*, and *BEA WebLogic Server Administration Guide*. The *BEA WebLogic Server Administration Guide* is available at the following URL:

<http://e-docs.bea.com/wls/docs61/adminguide/index.htm>

## Message Encryption

The RosettaNet Implementation Framework 2.0 introduces a new security option to the overall RosettaNet framework. In contrast to RNIF 1.1, RNIF 2.0 allows encryption of a message at one of three levels:

- None - No encryption occurs.
- Payload - The service content and any attachments are encrypted.
- Entire Payload - The service header, service content, and attachments are all encrypted.

Encryption options can be configured through the WebLogic Integration B2B Console. For more information on the encryption options, see “Configuring Message Encryption” in “[Configuring Security](#)” in *Implementing Security with B2B Integration*.

## Setting Up the Environment

Before you can support RosettaNet messaging in a domain, you must do the following:

- Copy the Document Type Definitions (DTDs) from the `wlintegration2.1/lib/dtd/rosettanet` directory into the domain directory.

For example, for the WebLogic Integration preconfigured domain, `wlidomain`, copy the DTDs from the `WLI_HOME/lib/dtd/rosettanet` directory to the `WLI_HOME/config/wlidomain` directory.

- Create a directory named `schemas` under the domain directory. Copy the files in the `WLI_HOME/lib/xmlschema/rosettanet` directory to the new `domain_name/schemas` directory.

For example, for the WebLogic Integration preconfigured domain, `wlidomain`, create the `WLI_HOME/config/wlidomain/schemas` directory, and then copy the files from the `WLI_HOME/lib/xmlschema/rosettanet` directory to the `WLI_HOME/config/wlidomain/schemas` directory.

# 2 Using Workflows with RosettaNet

The following sections explain how to develop workflows to implement RosettaNet Partner Interface Processes (PIPs).

- Understanding RosettaNet
- Understanding PIP Workflow Instances
- Getting Started
- Working with Workflows
- RosettaNet Workflow Variables
- Receiving a RosettaNet Message
- Sending a RosettaNet Message
- Validating a Message
- Message Attachments

The procedures in this section refer to workflow diagrams as they appear in the WebLogic Integration Studio. To access a workflow diagram, start Studio as described in “Starting the Studio” in [“WebLogic Integration Administration and Design Tools”](#) in *Starting, Stopping, and Customizing BEA WebLogic Integration*. For information about defining collaborative workflows for use in B2B applications, see *Creating Workflows for B2B Integration*.

# Understanding RosettaNet

The following RosettaNet documents are required reading if you want to implement your own PIP using the support for RosettaNet provided by WebLogic Integration, and recommended reading if you want to fully understand the sample RosettaNet PIP implementations:

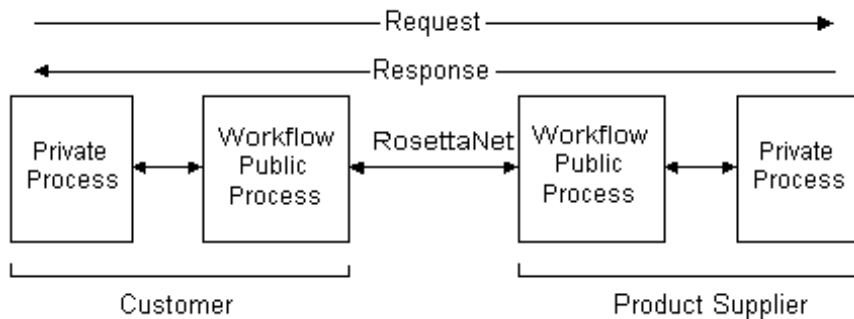
- *Creating Workflows for B2B Integration*
- The following documents are available in the “Standards” section at the RosettaNet Web site (<http://www.rosettanet.org>):
  - RosettaNet Implementation Framework v1.1 (RNIF 1.1) —The RNIF is an open, common networked-application framework designed to allow RosettaNet supply chain and solution partners to collaborate in executing RosettaNet PIPs.
  - RosettaNet Implementation Framework v2.0 (RNIF 2.0)
  - RNIF Technical Advisories—RNIF Technical Advisories are updates and additional information for RNIF 1.1 and RNIF 2.0.
  - RNIF Technical Recommendations—Technical Recommendations describe features or enhancements not yet available in a published version of the RNIF v1.1. Implementation of Technical Recommendations is optional.
  - RNIF Business Signals, Service Header & Preamble—The RNIF business signals, service header & preamble document contains message guidelines and XML document type definitions (DTDs) for the RNIF business signals, service header, and preamble.
  - Understanding a PIP Blueprint—reference for PIP blueprint components and evaluation. Available under Supporting Documents in the Standards Section.
  - PIPs of interest—PIPs are specialized system-to-system, XML-based dialogs that define business processes between supply chain companies. Each PIP includes a technical specification based on the RosettaNet Implementation Framework (RNIF), a Message Guideline document with a PIP-specific version of the Business Dictionary, and XML document type definitions (DTDs) for PIP-specific messages.

# Understanding PIP Workflow Instances

WebLogic Integration implements standard RosettaNet PIPs through public workflows (also known as collaborative workflows). A public workflow provides the interface to other trading partners, while private workflows are used to interface to back-end systems in order to generate and respond to messages.

The following figure shows the process by which PIP workflows pass messages between trading partners.

**Figure 2-1 Message Workflow**



Generally speaking, RosettaNet-oriented workflows process messages as follows:

1. A customer private workflow initiates a RosettaNet message. Data is retrieved and formatted into a RosettaNet message structure, the appropriate PIP is determined, and the message is forwarded to the public workflow that implements the customer role in the PIP.
2. The public workflow process creates the appropriate RosettaNet message. The message is sent to the public workflow implementing the product supplier role in the PIP via the B2B engine.
3. The product supplier public workflow receives the message, processes the header information, and then passes validated customer information and message content to the appropriate private workflow process.

4. The product supplier private process resolves message content and generates a reply. The reply is processed into a RosettaNet message structure and passed back to the product supplier public process.
5. The product supplier public process creates a RosettaNet reply message and sends it to the customer public process via the B2B engine.
6. The customer public process receives the reply message, processes header information, and then passes validated product supplier information and message content to the appropriate private process.
7. The private process resolves content of the reply message.

# Getting Started

While most RosettaNet workflow development tasks are identical to those used for developing standard XOCF-oriented workflows, there are some differences. The standard procedures are documented in *Creating Workflows for B2B Integration*; you should refer to them before creating RosettaNet workflows.

WebLogic Integration includes full implementations of PIP0A1 and PIP3A2 for both RosettaNet 1.1 and 2.0. These templates provide a useful template base that you can use when creating additional PIP workflows.

The remainder of this section discusses issues involving the creation of RosettaNet-specific workflows.

# Working with Workflows

In WebLogic Integration, RosettaNet PIPs are public workflows that implement a trading partner's role in the exchange of messages via the B2B engine. These workflows are designed to work with private workflows that:

- Provide raw source and recipient data
- Structure data into standard RosettaNet message body forms
- Process received data
- Initiate public workflows to send data
- Provide an interface to private applications

The procedures you need to configure a private workflow to start a public workflow, or to configure a public workflow to start a private workflow, are provided in [“Starting Collaborative Workflows”](#) in *Creating Workflows for B2B Integration*.

## RosettaNet Workflow Variables

RosettaNet workflows implemented in the WebLogic Integration Studio require a number of workflow variables to operate. These variables are used in three ways:

- Input variables are used to receive data into the workflow. The content of a variable is set by the calling private workflow.
- Output variables are used to send data out of the workflow. The content, created by the PIP when it completes, is passed back to the calling workflow.
- System variables are used to track assorted data within the workflow that are required for a RosettaNet workflow to operate properly.

All RosettaNet workflows, regardless of the PIP role implemented, must include the same basic set of System, Input, and Output variables. The required variables are organized into tables by RNIF version in the following sections:

- Using the Workflow Variable Tables
- Workflow Variables for RNIF 2.0
- Workflow Variables for RNIF 1.1

## Using the Workflow Variable Tables

For each workflow variable, the name and type (boolean, string, integer, xml, or object) are identified. In addition, variable usage and description are provided as described in the following sections.

### Usage

The workflow variables are organized into the following usage categories:

- **Input—Value Mandatory**  
The value for these variables must be set by the calling workflow.
- **Input—Value Optional**  
The value for these variables may be provided by the calling workflow. Although the value is not required, the variable must exist.
- **Output—Value Optional**  
The value for these variables may be set when the workflow ends. Although the value is not required, the variable must exist. A value is not required for any Output variable.
- **System—Value Mandatory**  
These variables are used by the B2B engine to generate the RosettaNet message. The value must be provided by the calling workflow or assigned at the workflow start node.
- **System—Value Optional**  
These variables are used by the B2B engine to generate the RosettaNet message. Although the value is not required, the variable must exist.



An important point to note when using these workflow variables is that all input variables must be initialized when you start a public workflow. The values used to initialize the input variables do not need to be exposed or transmitted.

The required workflow variables include specification of the NOF parties. These parties are the trading partners that will handle the PIP exceptions. An NOF party may be a participant in the original PIP or a 3rd party.

## Description

A brief description of the variable is provided. Where relevant, the variable description includes the element of the RosettaNet Object (RNIF 1.1) or RosettaNet Business Message (RNIF 2.0) to which the variable maps. For example, the Input variable `GlobalUsageCode` maps to:

`ServiceHeader/ProcessControl/GlobalUsageCode`

Here, the slashes indicate position in the hierarchical element structure as follows:

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

## Workflow Variables for RNIF 2.0

Table 2-1 describes the required template variables for RNIF2.0 PIP workflows. Additional, PIP-specific Input and Output variables may also be required. Refer to the message guideline and XML document type definition (DTD) for the specific PIP.

**Table 2-1 Template Variables for RNIF2.0**

Name	Type	Usage and Description
actionCode	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/ActivityControl/MessageControl/Manifest/ServiceContentControl/ActionIdentity/GlobalBusinessActionCode
actionCodeVersion	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/ActivityControl/MessageControl/Manifest/ServiceContentControl/ActionIdentity/VersionIdentifier
attachmentDescriptorInput	xml	Input—value optional  Contains the descriptor of files to be used as attachments.
attachmentDescriptorOutput	xml	Output—value optional  Contains the descriptor of files that may have been received as attachments.

**Table 2-1 Template Variables for RNIF2.0 (Continued)**

Name	Type	Usage and Description
businessActivityID	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/ActivityControl /BusinessActivityIdentifier
DOC_TYPE	string	System—value optional  Contains the doctype string for the XML message. This sting is inserted into the service content if the content does not contain a doctype string.
exceptionError	boolean	System—value optional  Indicates if an exception error was received.
fromDUNS	string	Input—value mandatory  The DUNS of the sender. Must match the business id defined in the repository for the trading partner.  Maps to: DeliveryHeader/messageSenderIdentification/ PartnerIdentification /GlobalBusinessIdentifier  ServiceHeader/ProcessControl/KnownInitiatingPartner /PartnerIdentification/GlobalBusinessIdentifier

## 2 Using Workflows with RosettaNet

---

**Table 2-1 Template Variables for RNIF2.0 (Continued)**

Name	Type	Usage and Description
<code>fromLocation</code>	string	System—value mandatory  Maps to: DeliveryHeader/messageSenderIdentification/ PartnerIdentification /locationID/FreeFormText
<code>fromRole</code>	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/MessageControl /fromRole /GlobalPartnerRoleClassificationCode
<code>fromService</code>	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/MessageControl /fromService /GlobalBusinessServiceCode
<code>globalUsageCode</code>	string	Input—value mandatory  Set to either <i>Test</i> or <i>Production</i> .  Maps to: ServiceHeader/ProcessControl/GlobalUsageCode
<code>gotMessage</code>	string	System—value mandatory  Indicates if a RosettaNet message was received. True if a message was received. False if a timeout had occurred

**Table 2-1 Template Variables for RNIF2.0 (Continued)**

Name	Type	Usage and Description
isSignal	boolean	System—value optional  Indicates if a signal was received.
messageCode	integer	System—value optional  Contains the code returned by the message send.
messageTrackingId	string	System—value optional  Message ID used to populate the message content for a PIP0A1 notification.
NOFParty1	string	Input—value mandatory  The party name of the PIP Failure Notifier role. Used to start the Notification of Failure Error process (PIP0A1).
NOFParty2	string	Input—value mandatory  The party name of the PIP Failure Report Administrator role. Used to start the Notification of Failure Error process (PIP0A1).
PIP	string	System—value mandatory  Must match the conversation name.  Maps to: ServiceHeader/ProcessControl/pipCode/Global ProcessCode
PIPInput	xml	Input—value mandatory  The service content of the message.

**Table 2-1 Template Variables for RNIF2.0 (Continued)**

Name	Type	Usage and Description
PIPOutput	xml	Output—value optional  The service content of the received message.
PIPVersion	string	System—value mandatory  Must match the conversation version.  Maps to: ServiceHeader/ProcessControl/pipVersion/VersionIdentifier
reason	string	System—value optional  Contains the reason for the workflow finishing.
retryCount	integer	System—value mandatory  Counter to store the retry attempts.
rnProcessInstanceId	string	System—value optional  The process id of the PIP instance. Used to populate the message content for OA1 notification.
rnSystem	object	System—value optional  An object used internally to hold application data state. See “Consolidated rnSystem Variable” on page 2-25.
SERVICE_CONTENT_SCHEMA	string	System—value mandatory  Contains the name of the xsd schema to validate the service content against.

**Table 2-1 Template Variables for RNIF2.0 (Continued)**

Name	Type	Usage and Description
timeStamp	string	<p>System—value optional</p> <p>If the retry count is 1 (that is, if this is the first attempt), the timestamp is generated internally.</p> <p>Maps to:            DeliveryHeader/messageDateTime/DateTimeStamp</p>
toDUNS	string	<p>Input—value mandatory</p> <p>The DUNS of the receiver. Must match the business ID defined in the repository for the trading partner.</p> <p>Maps to:            DeliveryHeader/messageReceiverIdentification/PartnerIdentification/GlobalBusinessIdentifier</p>
toLocation	string	<p>Input—value mandatory</p> <p>Maps to:            DeliveryHeader/messageReceiverIdentification/PartnerIdentification/locationID/FreeFormText</p>
toRole	string	<p>System—value mandatory</p> <p>Maps to:            ServiceHeader/ProcessControl/MessageControl/toRole/GlobalPartnerRoleClassificationCode</p>

## 2 Using Workflows with RosettaNet

---

**Table 2-1 Template Variables for RNIF2.0 (Continued)**

Name	Type	Usage and Description
toService	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/MessageControl /toService /GlobalBusinessServiceCode
useDTDValidation	boolean	System—value optional  If set to true, DTD validation is used instead of XSD. If set to false or the variable does not exist, XSD schemas will be used for validation. Whether validation is performed at all is dependent on the values of <code>validateServiceContent</code> and <code>validateServiceHeader</code> .
validateServiceContent	boolean	Input—value mandatory  Flag to indicate if the service content should be validated against the schema. True – validation required False – no validation required
validateServiceHeader	boolean	Input—value mandatory  Flag to indicate if the service header should be validated against the schema. True – validation required False – no validation required
validationError	boolean	System—value optional  Indicates the result of the validation.



## Workflow Variables for RNIF 1.1

Table 2-2 describes the template variables for RNIF1.1 PIP workflows. Additional, PIP-specific Input and Output variables may also be required. Refer to the message guideline and XML document type definition (DTD) for the specific PIP.

**Table 2-2 Template Variables for RNIF1.1**

Name	Type	Usage and Description
actionCode	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/TransactionControl/ActionControl/ActionIdentity/GlobalBusinessActionCode
actionCodeVersion	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/TransactionControl/ActionControl/ActionIdentity/VersionIdentifier
attachmentDescriptorInput	xml	Input—value optional  The descriptor of files to be used as attachments.
attachmentDescriptorOutput	xml	Output—value optional  Contains the descriptor of files that may have been received as attachments.
businessActivityID	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/ProcessIdentity/GlobalProcessCode

## 2 Using Workflows with RosettaNet

---

**Table 2-2 Template Variables for RNIF1.1 (Continued)**

Name	Type	Usage and Description
DOC_TYPE	string	System—value optional  Contains the doctype string for the XML message. This string is inserted into the service content if the content does not contain a doctype string.
docId	string	System—value optional  Used to create the acknowledgment message.  Maps to: ReceiptAcknowledgement/receivedDocumentIdentifier /ProprietaryDocumentIdentifierReceiptAcknowledgmentException /theOffendingDocumentIdentifier/ProprietaryDocumentIdentifierException /theOffendingDocumentIdentifier/ProprietaryDocumentIdentifier
exceptionError	boolean	System—value optional  Indicates if an exception error was received.
fromClass	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/TransactionControl/ActionControl /PartnerRoute/fromPartner/PartnerDescription /GlobalPartnerClassificationCode  ServiceHeader/ProcessControl/TransactionControl/SignalControl /PartnerRoute/fromPartner/PartnerDescription /GlobalPartnerClassificationCode

**Table 2-2 Template Variables for RNIF1.1 (Continued)**

Name	Type	Usage and Description
fromContactName	string	<p>Input—value mandatory</p> <p>Used to construct the XML content for the ‘Notification of Error’ message.</p>
fromDUNS	string	<p>Input—value mandatory</p> <p>The DUNS of the sender. Must match the business ID defined in the repository for the trading partner.</p> <p>Maps to:  ServiceHeader/ProcessControl/TransactionControl/ActionControl  /PartnerRoute/fromPartner/PartnerDescription/BusinessDescription  /GlobalBusinessIdentifierServiceHeader/ProcessControl/TransactionControl  /SignalControl/PartnerRoute/fromPartner/PartnerDescription  /BusinessDescription/GlobalBusinessIdentifier</p>
fromEmailAddress	string	<p>Input—value mandatory</p> <p>Used to construct the XML content for the ‘Notification of Error’ message. The email address of the sender.</p>
fromPhone	string	<p>Input—value mandatory</p> <p>Used to construct the XML content for the ‘Notification of Error’ message. The phone number of the sender.</p>

**Table 2-2 Template Variables for RNIF1.1 (Continued)**

Name	Type	Usage and Description
fromRole	string	<p>System—value mandatory</p> <p>Maps to:            ServiceHeader/ProcessControl/TransactionControl/PartnerRoleRoute/fromRole            /PartnerRoleDescription/GlobalPartnerRoleClassificationCode</p>
fromService	string	<p>System—value mandatory</p> <p>Maps to:            ServiceHeader/ProcessControl/ServiceRoute/fromService            /BusinessServiceDescription/GlobalBusinessServiceCode</p>
fromSupplychain	string	<p>Input—value mandatory</p> <p>Used to create message acknowledgment.</p> <p>Maps to:            ReceiptAcknowledgement/fromRole/PartnerRoleDescription            /PartnerDescription/BusinessDescription            /GlobalSupplyChainCode</p> <p>ReceiptAcknowledgementException/fromRole/PartnerRoleDescription            /PartnerDescription/BusinessDescription/GlobalSupplyChainCode</p> <p>Exception/fromRole/PartnerRoleDescription/PartnerDescription            /BusinessDescription/GlobalSupplyChainCode</p>

Table 2-2 Template Variables for RNIF1.1 (Continued)

Name	Type	Usage and Description
functionCode	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/TransactionControl/ActionControl/GlobalDocumentFunctionCode
globalUsageCode	string	Input—value mandatory  Set to either <i>Test</i> or <i>Production</i> .  Maps to: Preamble/GlobalUsageCode
gotMessage	string	System—value mandatory  Indicates if a RosettaNet message was received. True – message received. False – a timeout occurred
isSignal	boolean	System—value optional  Indicates if a signal was received.
messageCode	integer	System—value optional  Contains the code returned by the message send.
messageTrackingId	string	System—value optional  Message ID used to populate the message content for a PIP0A1 notification.

**Table 2-2 Template Variables for RNIF1.1 (Continued)**

Name	Type	Usage and Description
NOFParty1	string	Input—value mandatory  The party name of the PIP Failure Notifier role. Used to start the Notification of Failure Error (PIP0A1)
NOFParty2	string	Input—value mandatory  The party name of the PIP Failure Report Administrator role. Used to start the Notification of Failure Error (PIP0A1)
PIP	string	System—value mandatory  Must match the conversation name.  Maps to: ServiceHeader/ProcessControl/ProcessIdentity/GlobalProcessIndicatorCode
PIPInput	xml	Input—value mandatory  The service content of the message.
PIPOutput	xml	Output—value optional  The service content of the received message.
PIPVersion	string	System—value mandatory  Must match the conversation version.  Maps to: ServiceHeader/ProcessControl/ProcessIdentity/VersionIdentifier

**Table 2-2 Template Variables for RNIF1.1 (Continued)**

Name	Type	Usage and Description
reason	string	System—value optional  Contain the reason for the workflow finishing.
retryCount	integer	System—value mandatory  Counter to store the retry attempts.
rnProcessInstanceId	string	System—value optional  The process id of the PIP instance. Used to populate the message content for OA1 notification.
rnSystem	object	System—value optional  An object used internally to hold application data state. See “Consolidated rnSystem Variable” on page 2-25.
SERVICE_CONTENT_SCHEMA	string	System—value mandatory  Contains the name of the xsd schema to validate the service content against.
timeStamp	string	System—value optional  Maps to:  ReceiptAcknowledgement/receivedDocumentDateTime/DateTimeStamp  ReceiptAcknowledgmentException/theOffendingDocumentDateTime/DateTimeStamp  Exception/theOffendingDocumentDateTime/DateTimeStamp

## 2 Using Workflows with RosettaNet

---

**Table 2-2 Template Variables for RNIF1.1 (Continued)**

Name	Type	Usage and Description
toClass	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/TransactionControl/ActionControl /PartnerRoute/toPartner/PartnerDescription /GlobalPartnerClassificationCode  ServiceHeader/ProcessControl/TransactionControl/SignalControl /PartnerRoute/toPartner/PartnerDescription /GlobalPartnerClassificationCode
toDUNS	string	Input—value mandatory  The DUNS of the receiver. Must match the business id defined in the repository for the trading partner.  Maps to: ServiceHeader/ProcessControl/TransactionControl/ActionControl /PartnerRoute/toPartner/PartnerDescription/BusinessDescription /GlobalBusinessIdentifier  ServiceHeader/ProcessControl/TransactionControl/SignalControl /PartnerRoute/toPartner/PartnerDescription/BusinessDescription /GlobalBusinessIdentifier

---



**Table 2-2 Template Variables for RNIF1.1 (Continued)**

Name	Type	Usage and Description
toRole	string	<p>System—value mandatory</p> <p>Maps to:            ServiceHeader/ProcessControl/TransactionControl/PartnerRoleRoute            /toRole/PartnerRoleDescription/GlobalPartnerRoleClassificationCode</p>
toService	string	<p>System—value mandatory</p> <p>Maps to:            ServiceHeader/ProcessControl/ServiceRoute/toService            /BusinessServiceDescription/GlobalBusinessServiceCode</p>
toSupplyChain	string	<p>Input—value mandatory</p> <p>Used to create message acknowledgment.</p> <p>Maps to:            ReceiptAcknowledgement/toRole/PartnerRoleDescription/PartnerDescription            /BusinessDescription/GlobalSupplyChainCode</p> <p>ReceiptAcknowledgmentException/toRole/PartnerRoleDescription            /PartnerDescription/BusinessDescription/GlobalSupplyChainCode</p> <p>Exception/toRole/PartnerRoleDescription/PartnerDescription            /BusinessDescription/GlobalSupplyChainCode</p>

## 2 Using Workflows with RosettaNet

---

**Table 2-2 Template Variables for RNIF1.1 (Continued)**

Name	Type	Usage and Description
transactionCode	string	System—value mandatory  Maps to: ServiceHeader/ProcessControl/TransactionControl/TransactionIdentity/GlobalTransactionCode
useDTDValidation	boolean	System—value optional  If set to true, DTD validation is used instead of XSD. If set to false or the variable does not exist, XSD schemas will be used for validation. Whether validation is performed at all is dependent on the values of <code>validateServiceContent</code> and <code>validateServiceHeader</code> .
validateServiceContent	boolean	Input—value mandatory  Flag to indicate if the service content should be validated against the schema. True – validation required False – no validation required
validateServiceHeader	boolean	Input—value mandatory  Flag to indicate if the service header should be validated against the schema. True – validation required False – no validation required
validationError	boolean	System—value optional  Indicates the result of the validation.

# Consolidated rnSystem Variable

In WebLogic Integration 2.0, the following variables were also required.

```
signalCode  
signalCodeVersion  
inReplyToActionCode  
inReplyToMessageId  
initiatingPartnerDUNS  
inReplyToActionCodeVersion (RNIF2.0 only)
```

These variables have been replaced by the consolidated `rnSystem` variable (workflow variable type: Java Object), to reduce the number of system variables exposed in a workflow template.

If you have PIP workflows that were developed for WebLogic Integration 2.0, you can remove invocations or definitions of these variables from existing workflow templates and replace them with `rnSystem`.

## Receiving a RosettaNet Message

WebLogic Integration supports two different methods of receiving RosettaNet messages—Start nodes and Event nodes. These two different node types are used depending on the circumstances under which the message is received.

### Start Nodes

You can configure a workflow so that WebLogic Integration starts it automatically when WebLogic Integration receives the first message for a PIP instance. To configure this action, declare received PIPs as Start node events. Your receiving workflow starts and processes the incoming PIP. For an example, see the Participant Workflow discussion in [“Starting Collaborative Workflows”](#) in *Creating Workflows for B2B Integration*.

The PIP3A2 Supplier workflow template serves as an example of how to configure a workflow template definition to be started by an incoming message. In this example, the starting event is set as a Business Message Event. The workflow is automatically started and the output and system variables are set when a RosettaNet message has been received.

## Event Nodes

Workflows can contain events that are triggered when a message is received for the PIP instance associated with the workflow. For an example, see the Initiator Workflow discussion in “Starting Collaborative Workflows” in *Creating Workflows for B2B Integration*.

## Implementing Timeouts

Workflows can use an optional timeout path to wait for an incoming RosettaNet message. If the workflow waits for a response to a sent message (for example, the 3A2 Customer workflow), you must create a separate timeout path to wait for the response. This path, illustrated in the 3A2 Customer workflow template, consists of a timer, set for the appropriate timeout period, and a stop node.

## Out of Sequence Reception of Signals

RNIF 1.1 and RNIF 2.0 define different standards for reception of signals and replies. These differing standards affect how your PIP workflow resolves these messages.

RNIF 1.1 specifies that replies must always follow signals. Thus, you might see a signal/response pattern as follows:

1. Initiator→(request)→Recipient
2. Recipient→(receipt acknowledgment)→Initiator
3. Recipient→(response)→Initiator
4. Initiator→(handling acknowledgment)→Recipient

RNIF 2.0 allows out-of-sequence receipt of signals. For example, a workflow might receive a reply before the receipt acknowledgment. Thus, RNIF 2.0 allows a signal/response pattern such as the following:

1. Initiator→(request)→Recipient
2. Recipient→(response)→Initiator
3. Recipient→(receipt acknowledgment)→Initiator
4. Initiator→(receipt acknowledgment)→Recipient

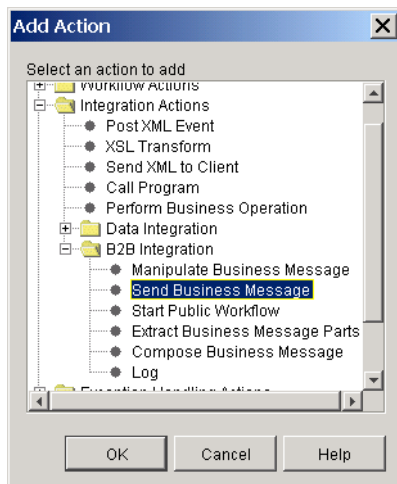
This pattern must be processed accurately in your workflow design. The PIP3A2 Customer workflow template provides a useful example of how to handle this. The RNIF 1.1 version of the PIP template processes the signal, followed by the response. The RNIF 2.0 version provides logic to process the signal and the response separately.

## Sending a RosettaNet Message

A workflow task can send a RosettaNet message by invoking an integration action called Send Business Message. To add this action to a task:

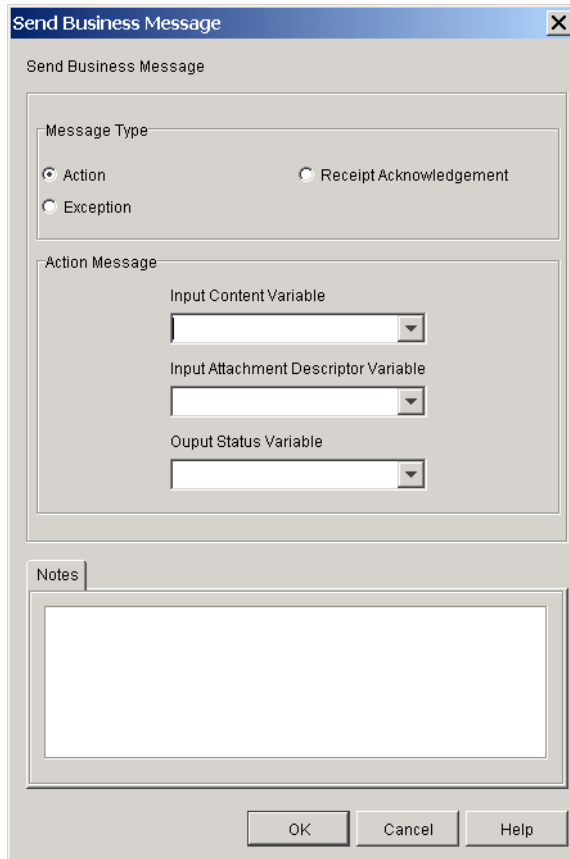
1. Right-click the task, and then select Properties from the shortcut menu.  
The Task Properties dialog box is displayed.
2. Click the Add button to display the Add Action dialog box.
3. Choose Integration Actions→B2B Integration→Send Business Message, as shown in the following figure:

**Figure 2-2 B2B Integration Actions**



4. Click OK to display the Send Business Message dialog box as shown in the following figure.

**Figure 2-3 RosettaNet 2.0 Send Business Message Dialog**



The information required is dependent on the Message Type (Action, Exception, or Receipt Acknowledgment) you select. The following table summarizes the information you are prompted for by message type.

**Table 2-3 Message Types and Options**

<b>Message Type</b>	<b>Field Name</b>	<b>Description</b>
Action	Message Type—Action	Indicates that a RosettaNet Action Business Message is to be sent.
	Input Content Value	A workflow XML variable containing the XML data to represent the service content. See the <code>PIPInput</code> variable in Table 2-1 (RNIF 2.0) or Table 2-2 (RNIF 1.1).
	Attachment Descriptor Value	A workflow XML variable containing the XML data for describing the attachment(s) to be sent as part of the RosettaNet message. (See “Message Attachments” on page 2-33 and the <code>attachmentDescriptorInput</code> variable in Table 2-1 (RNIF 2.0) or Table 2-2 (RNIF 1.1).
	Output Status Variable	A workflow Integer variable to contain the send status of the message.
Receipt Acknowledgment	Message Type—Receipt Acknowledgment	Indicates that a RosettaNet Receipt Acknowledgment signal is to be sent. No other data values are required.
Exception	Message Type—Exception	Indicates that a RosettaNet Receipt Exception Acknowledgment signal is to be sent.
	Error Code	The Error Code (as defined by RosettaNet) to be sent.
	Error Description	A brief description of the error.
	Output Status Variable	The variable to contain the send status of the message.



# Validating a Message

The message validation process uses the Xerces 1.3 DOM parser, which supports an alpha implementation of the XML Schema specification. The Xerces 1.3 DOM parser is packaged with the WebLogic Server 6.1 software.

The following sections describe how WebLogic Integration validates RosettaNet messages and provide a bibliography for further reading about message validation:

- RosettaNet Message Validation
- Recommended Reading About Message Validation

## RosettaNet Message Validation

WebLogic Integration provides message validation services for both RNIF 1.1 and RNIF 2.0 messages. The validation performed is dependent on the values specified for the `validateServiceContent`, `validateServiceHeader`, and `useDTDValidation` workflow variables. You have the following validation options:

- None—No validation is performed if `validateServiceHeader` and `validateServiceContent` are set to false.
- Validate Service Header—If `validateServiceHeader` set to true, the service header of all messages sent and received for a template are validated. The type of validation performed is dependent on the setting for the `useDTDValidation` variable.
- Validate Service Content—If `validateServiceContent` set to true, all service content of all messages sent and received for a template are validated. The type of validation performed is dependent on the setting for the `useDTDValidation` variable.
- DTD—DTD validation is performed if `useDTDValidation` is set to true. The DTD files are supplied as part of the PIP specification and can be obtained from [www.rosettanet.org](http://www.rosettanet.org).

- XSD Schemas—XSD schema validation is performed if `useDTDValidation` is set to false, or is absent from the workflow template. In order to use XSD validation, you must create your own schema files from the DTD for the PIP.

For an explanation of the exception handling process, see the RNIF specification at the following URL:

<http://www.rosettanet.org>

Example XML schema files and document type definition (DTD) files are located on your WebLogic Integration installation at:

`WLI_Home/lib/xmlschema/rosettanet`

## Recommended Reading About Message Validation

The following information is recommended reading to fully understand the example XML schemas; it is required reading if you are planning to implement your own XML schemas:

- Information about XML schema tools, usage, specifications, and development is available at the following URL:

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

*XML Schema Part 0: Primer* provides good descriptions of the features and capabilities of XML schema.

- Information about the Xerces implementation of the XML Schema specification is available at the following URL:

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

## Performance Tuning and Message Validation

Message validation is used primarily while setting up and configuring your system with a partner. Once you are satisfied that invalid message generation issues have been removed from the system, you may optionally turn off message validation to boost performance. As noted in the preceding section, there is no requirement for message validation to be performed during message processing.

# Message Attachments

Both RNIF 1.1 and RNIF 2.0 include support for optional message attachments in the RosettaNet action message. Attachments are not file-type-specific, and may contain binary data. Examples of possible attachments include Word documents, GIF images, PDF files, and so on. The information for each attachment is contained in the service header of the message.

WebLogic Integration supports attachments by allowing a user application (for example a private workflow) to provide a description of the attachment, contained in a BEA-specific structured XML file, as an input to the PIP workflow. The XML file is a description of the file attachment. It is not the actual attachment. It only specifies which files should be attached by WebLogic Integration.

The following is the DTD information describing the attachments:

## Listing 2-1 WebLogic Integration Attachment DTD Information for RosettaNet

```
<!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 ) >
```

The following table describes the elements used in the DTD.

**Table 2-4 RosettaNet Attachment Elements**

Element	Description
Description	An optional description of the attachment.
Type	The MIME type qualifier code of the attachment.
ID	The Universal Resource Identifier for the attachment.
LocalLocation	On sending: Contains the full path (on the local system) of the file for sending as an attachment. On receiving: Contains the full path (on the local system) of the file attachment that was received.

When the message is received, any attachments are stored locally in the *WLI\_Home/config/domain\_name/rnattachments* directory. The filename of the stored attachment is prefixed with a timestamp. For example, you might use the following XML description for an attachment.

### Listing 2-2 Sample XML Attachment

---

```
<?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>
```

---

This sample would generate the following data in the service header and in the MIME header.

### Listing 2-3 Sample Service Output

---

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

---

### Listing 2-4 Sample MIME Header

---

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

---



# 3 RosettaNet PIP Templates

WebLogic Integration includes two PIP templates with which you can immediately implement RosettaNet-based solutions. This section includes descriptions of both the contents of each PIP template and the information required to implement the PIP. Versions of both PIPs are provided for both RNIF 1.1 and RNIF 2.0.

## Importing a RosettaNet Template

The following XML template files are located in the `WLI_HOME/lib/templates/rosettanet` directory.

- 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

To use one of the templates provided:

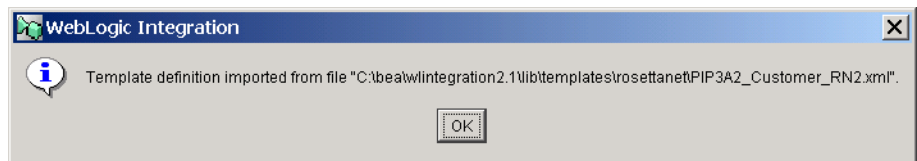
1. Start Studio as described in “Starting the Studio” in [“WebLogic Integration Administration and Design Tools”](#) in *Starting, Stopping, and Customizing BEA WebLogic Integration*.
2. In the right pane, right-click the Templates node.
3. Select Create Template from the shortcut menu.
4. Enter the name of the template. (This is typically the same name as the template definition to be imported.)
5. Click OK.

An entry is added to the tree for the new template.

6. Right-click the new entry, and then select Import Template Definition.
7. Locate the required RosettaNet template, then click Open to import the definition.

A message such as the following is displayed when the template has been successfully imported.

**Figure 3-1 Import Successful Message**



## RosettaNet Business Operations

In addition to the templates, WebLogic Integration includes the `RNBusinessOP.jar` file. This file is a workflow package that contains business operations used in RosettaNet workflows. You should import this package as described in [“Importing and Exporting Workflow Packages”](#) in *Using the WebLogic Integration Studio*.



This package contains the `Get RN timestamp` business operation, an operation that returns a timestamp in a RN specific format. It is used to populate the service content for the PIP0A1 message. See the Start Notification of Failure task of the Customer workflow for more details.

## PIP0A1: Notification of Failure

This PIP is a system-level RosettaNet PIP required for all implementations. It is used to transmit information in the event a process failure occurs. Normally, it does not require significant customization. Rather, data is passed to it from calling PIP workflows. For this reason, the appropriate workflow variables must be set in the calling workflow before the PIP0A1 workflow is called. Specifically, the variables described in “RosettaNet Workflow Variables” on page 2-5 must be set and imported to the PIP0A1\_Admin template. All input variables must be initialized when the workflow is called.

## PIP3A2: Query Price and Availability

PIP3A2 is provided in template form as an example of how to implement a PIP. There are two versions of this PIP (one implemented for RNIF 1.1, and one for RNIF 2.0), and each version is modeled for both the sender and receiver roles. Use the PIP template version appropriate for the version of RosettaNet that you are using.

When implementing PIP3A2, you must set the mandatory input variables, all of which are listed “RosettaNet Workflow Variables” on page 2-5. These variables can be set from within the PIP workflow, or they can be set by a calling workflow. All input variables must be initialized when the workflow is called.

## Modeling Other PIPs

The PIP3A2 templates are provided to demonstrate how RosettaNet PIPs are supported in WebLogic Integration. Because of the complexity of the PIP, BEA recommends that when creating a new PIP, you copy the PIP3A2 source files and modify the copies to create your new PIP.

You should note the following points when modeling other PIPs:

- Use input and output workflow variables to transport data between this PIP and other processes. Use input variables for the PIP to receive data from another workflow, and use output variables to send data to another workflow. You must remember to initialize all mandatory input variables when you start a PIP workflow.
- Other workflows may initiate a PIP workflow. To configure this type of initiation, choose Actions→B2B Integration→Start Public Workflow. Use this method when you are the trading partner initiating a PIP transaction. When you use this method, remember that you must manually enter all the PIP workflow variables listed in “RosettaNet Workflow Variables” on page 2-5.
- PIP workflows may initiate other workflows, such as private workflows. To configure this type of initiation, choose Actions→Start Workflow. Use this method when you are the trading partner receiving a PIP transaction, and you want to pass that data to a private workflow. Use the private workflow to connect, for example, to a back-end ERP system.

---

# Index

## A

- administration 1-7
- architecture 1-3
  - RosettaNet support 1-3
- attachments 2-33

## B

- business message
  - receiving 2-25
  - sending 2-27

## C

- collaborative workflows 2-3
- customer support contact information vii

## D

- defining
  - collaboration agreements 1-7
  - conversation definitions 1-7
  - PIP workflows 1-5
  - trading partners 1-7
- digital signatures 1-5
- documentation
  - message validation 2-32
  - PIPs 2-2
  - RosettaNet Implementation Framework (RNIF) 2-2
- documentation, where to find it vi

- DTDs 1-2

  - included in WebLogic Integration 1-8

## E

- Environment 1-8
- event nodes 2-25

## I

- importing PIP workflow templates 3-1
- input variables 2-5

## M

- message attachments 2-33
- message guidelines 1-2
- message validation 2-32
- messages
  - receiving RosettaNet messages 2-25
  - sending RosettaNet messages 2-27
  - validating RosettaNet messages 2-31
  - validation documentation 2-32

## N

- nodes
  - event 2-25
  - start 2-25

---

## O

output variables 2-5

## P

parsers, Xerces DOM 2-31

performance 2-32

PIP templates 1-4

importing 3-1

PIP workflows, message exchange between  
2-3

PIP0A1 template 3-3

PIP3A2 template 3-3

PIPs

documentation 2-2

PIP0A1 template 3-3

PIP3A2 template 3-3

printing product documentation vi

private workflows 2-3

public workflows 2-3

## R

receiving RosettaNet messages 2-25

recommended reading

message validation 2-32

RosettaNet Implementation Framework  
(RNIF) 2-2

technical advisories 2-2

RNIF documentation 2-2

RosettaNet 2-2

message validation overview 1-5

receiving messages 2-25

RosettaNet messages 1-2

security 1-5

sending messages 2-27

validating messages 2-31

workflow variables 2-5

RosettaNet 2.0 security sample 1-6

RosettaNet in WebLogic Integration 1-2

RosettaNet Object 1-2

elements of 2-5

## S

sample application 1-6

security 1-5

send business message action 2-27

sending RosettaNet messages 2-27

setting up RosettaNet support 1-8

start nodes 2-25

starting a workflow 2-25

support

technical vii

system variables 2-5

## T

task properties 2-27

## V

validating RosettaNet messages 2-31

validation documentation 2-32

variables

input 2-5

output 2-5

RosettaNet required 2-5

system 2-5

workflow 2-5

## W

workflow

variable tables 2-5

variables 2-5

workflow templates 1-4

importing PIP templates 3-1

PIP0A1 3-3

PIP3A2 3-3

---

workflows

- collaborative 2-3

- message exchange between 2-3

- private 2-3

- public 2-3

- starting 2-25

## **X**

Xerces DOM parsers 2-31

