

Sterling Standards Library

Using RosettaNet[™]

Version 6.2

Sterling Commerce
An IBM Company

Licensed Materials - Property of Sterling Commerce

Sterling Standards Library 6.2

© Copyright Sterling Commerce, an IBM Company 2000, 2010 All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by contract with Sterling Commerce

Additional copyright information is located on the Sterling Integrator 5.0 Documentation
<http://www.sterlingcommerce.com/Documentation/SI50/CopyrightPage.htm>

Contents

Using RosettaNet with the Application	5
Overview	5
Prerequisites	5
About RosettaNet	6
RosettaNet Implementation Framework (RNIF)	6
Partner Interface Processes (PIP)	6
Signal Messages	7
Business Messages	8
Business and Technical Dictionaries	10
Private Processes	10
Interfacing with Non-RosettaNet Processes	10
Interface Between Private and Public PIPs	11
Implementing the Link Between Private PIPs and Internal Business Processes	13
Installing PIPs into the Application	14
Implementation of RosettaNet with the Application	17
Core RosettaNet Components	17
Scope of Implementation	24
Synchronous PIPs Limitations	24
Implementing RosettaNet in the Application	25
Downloading RosettaNet DTDs	25
RosettaNet Trading Partner Objects	26
Filling Logic into the Private Process Business Processes	27
Example (Writing code for PIP3A7_Buyer_Private)	30
Code for Analyzing the PO Update Notification	30
Error Cases in Private PIPs	30
Error Handling and Auditing	31
Business Process Manager and Monitor	33
Checking the Final PIP Status	33
Logging	34
Using RosettaNet in Production	34
GlobalUsageCode	34
Inbound and Outbound Document Flow	35
Working with RosettaNet Trading Partners	36
Before You Begin	36
Setting Up RosettaNet Trading Profiles	37
Creating an Identity Record for Your Organization	37
Creating an Identity Record for Your Trading Partner	38
Creating a Transport Record	39
Creating a Document Exchange Record	42

Creating a Delivery Channel Record	44
Creating a Packaging Record	45
Creating a Profile Record	46
Editing a Profile Record	48
Creating RosettaNet Contracts	48
Validating RosettaNet Service Content Against a DTD or Schema	51
Searching for RosettaNet Correlations	52
RosettaNet Database Tables	54
PIP_DEF Table	54
PIP_INSTANCE_DEF Table	55
SCI_RN_PROFILE Table	56
PIP_STEP Table	57
PIP_INSTANCE_STEP Table	57
PIP_TRACK table	58
PIP_STEP_TRACK	59
Index	60

Using RosettaNet with the Application

Overview

RosettaNet™ is a consortium of computer manufacturers, electronics component suppliers, and distributors focused on automating and improving the efficiency of the supply chain within that industry by using standard electronic business interfaces through Internet transports such as Multipurpose Internet Mail Extensions (MIME), Hypertext Transfer Protocol (HTTP), and Secure Sockets Layer (SSL) to exchange Extensible Markup Language (XML) documents.

RosettaNet defines a format for business processes, which include data formats and flows using RosettaNet-defined transport methods and security. The RosettaNet-defined processes serve as frameworks for business flows throughout trading partner systems. This document explains the concepts and features that explain the application approach to implementing RosettaNet™ standards.

Prerequisites

This manual assumes that you are familiar with the following:

- ◆ UNIX® operating system
- ◆ Windows® operating system
- ◆ Trading partner implementation
- ◆ Extensible Markup Language (XML)
- ◆ Business Process Markup Language (BPML)
- ◆ XML Path Language (XPath)
- ◆ RosettaNet standard and business messages

For complete information about RosettaNet, see the documentation provided by the RosettaNet Consortium.

The following table provides resources to obtain additional information about RosettaNet:

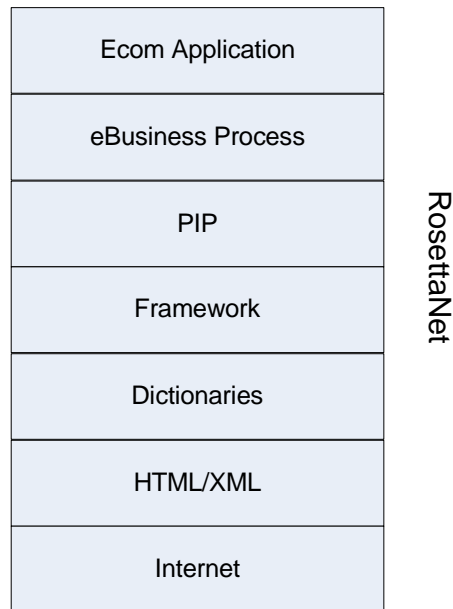
Functionality	Details
RosettaNet	RosettaNet Web site: www.rosettanet.org
RosettaNet PIP	RosettaNet PIP specifications (complete collection). RosettaNet, 1998-ongoing. Source: www.rosettanet.org
RosettaNet DDIC	RosettaNet dictionaries (Business and Technical). RosettaNet, 1998-ongoing. Source: www.rosettanet.org
RosettaNet solution for the application	Sterling Commerce Customer Support Web site at www.sterlingcommerce.com/customer/tech_support.html
RosettaNet concepts and development	See the RosettaNet Web site at www.rosettanet.org

About RosettaNet

A complete RosettaNet Standard implementation consists of the following RosettaNet layers and three core elements:

- ◆ *RosettaNet Implementation Framework (RNIF)* on page 6
- ◆ *Partner Interface Processes (PIP)* on page 6
- ◆ *Business and Technical Dictionaries* on page 10
- ◆ *Private Processes* on page 10

The following figure shows the RosettaNet layers and elements:



System – to – System
eBusiness exchange

RosettaNet Implementation Framework (RNIF)

The RosettaNet Implementation Framework (RNIF) core specification defines the packaging, routing, and transport of all PIP messages and business signals. Furthermore, the use of security mechanisms on message level (authentication, signatures, and encryption) and transport level (SSL) is part of the implementation framework.

Note: Trading partner agreements collect all bilateral agreements between RosettaNet partners.

Partner Interface Processes (PIP)

To classify PIPs, RosettaNet divides the entire supply chain domain into clusters, segments, PIPs, activities, and actions.

The following elements make up a PIP:

- ◆ Trading partner business roles (for example, buyer and seller)
- ◆ Business activities between the roles
- ◆ Type, content, and sequence of business documents
- ◆ Constraints for interactions (time, security, authentication, and performance)
- ◆ Structure and content of business documents, including DTDs that describe message guidelines
- ◆ Message Types:
 - ◆ Signal - Either positive or negative acknowledgments that result from a message
 - ◆ Business - Messages that result from a business action (for example, sending and receiving a purchase order)

The following example represents an order management PIP:

```

1. Clusters 3: Order Management
1.1.1 Segment A: Quote and Order Entry
1.1.1 PIP 3A4: Manage Purchase Order
1.1.1.1 Activity Create Purchase Order
1.1.1.1.1 Action: Purchase Order Request
  
```

The following figure shows PIP interaction:



Signal Messages

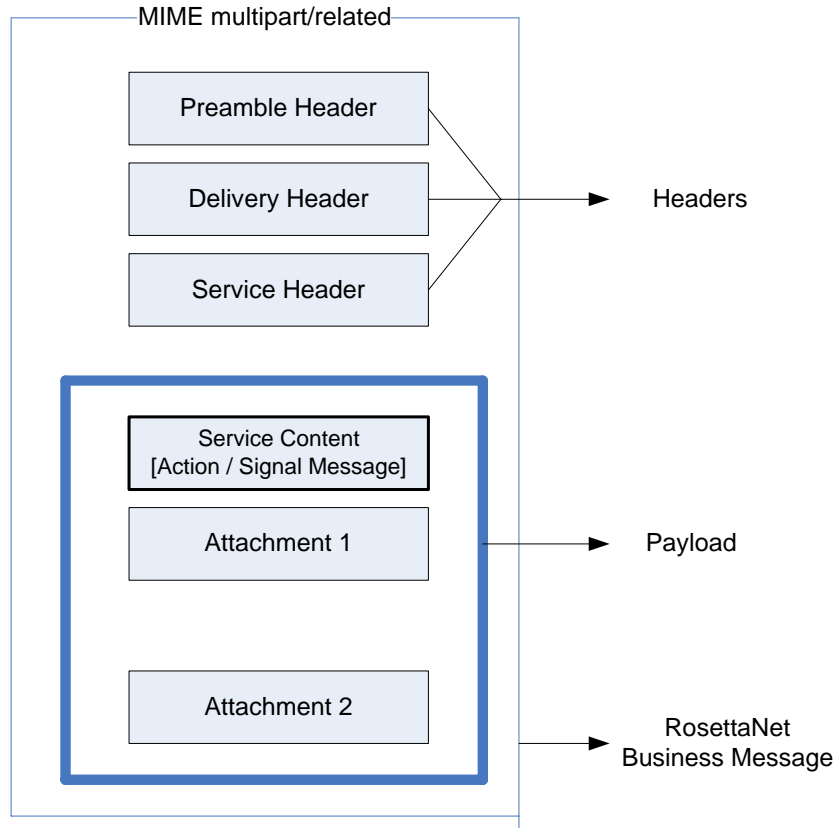
RosettaNet offers two types of message signals:

Signal	Message	Description
Positive	Receipt Acknowledgment	Structurally and syntactically valid RosettaNet action.
Negative	Receipt Acknowledgment Exception	Structurally or syntactically, invalid RosettaNet action.

Signal	Message	Description
	General Exception	Actions performed by processes outside of RosettaNet caused an error

Business Messages

The following figure shows a RosettaNet-compliant business message:

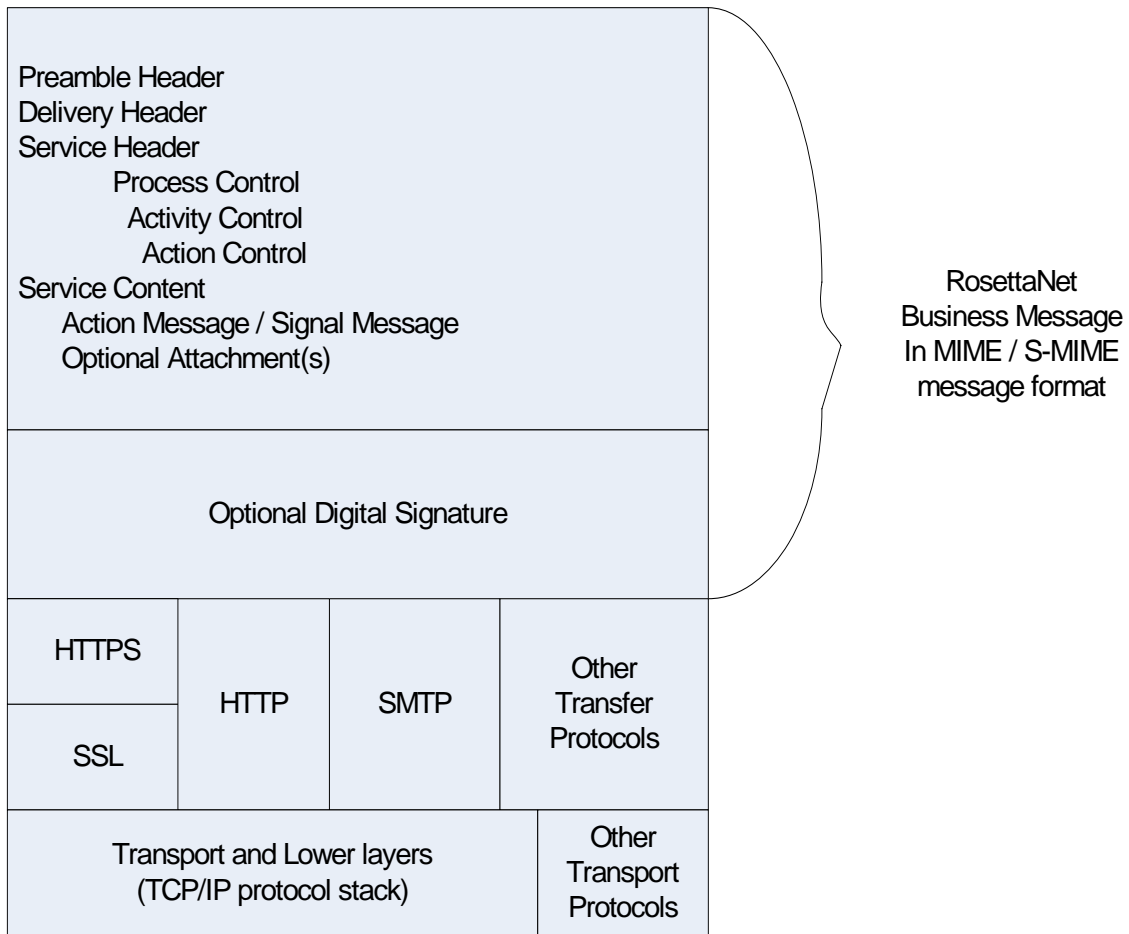


The following table describes the different parts of a RosettaNet message:

Message Part	Description
Preamble Header	<ul style="list-style-type: none"> ◆ XML document (UTF-8/UTF-16 encoding scheme) ◆ Mandatory, exactly one instance ◆ Never encrypted

Message Part	Description
Delivery Header	<ul style="list-style-type: none">◆ Enables routing of business messages through hubs.◆ Contains sending/receiving trading partner identities, tracking ID, date/time stamp.◆ Created by the message sender◆ XML document (UTF-8/UTF-16 encoding scheme)◆ Mandatory, exactly one instance◆ May specify requirement for a secure message transport
Service Header	<ul style="list-style-type: none">◆ XML document (UTF-8/UTF-16 encoding scheme)◆ Mandatory, exactly one instance◆ May be encrypted
Service Content	<ul style="list-style-type: none">◆ XML document (UTF-8/UTF-16 encoding scheme)◆ May be encrypted
Attachments	<ul style="list-style-type: none">◆ Any format◆ Optional, multiple instances allowed

The following figure illustrates the Network Application Protocol Stack of a business message:



Business and Technical Dictionaries

RosettaNet provides a common set of properties for PIPs based on two types of dictionaries:

1. RosettaNet Business – Designates the properties used in basic business activities.
2. RosettaNet Technical – Provides properties for defining products.

Private Processes

Enterprise business processes are private processes not defined by RosettaNet. Private processes can provide an interface for a RosettaNet implementation—thus, creating a fourth element.

Interfacing with Non-RosettaNet Processes

There are two types of PIP processes:

1. **Public Processes** – Public processes implement the Partner Interface Processes (PIPs) defined by RosettaNet. There are two public processes for each PIP:
 - ◆ One on the initiator side and one on the responder side.
 - ◆ Public processes are not intended to be trading partner specific.

For example, RosettaNet defines two roles (Buyer and Seller) for PIP 3A7. Therefore, there is a public process PIP3A7_V02.00_Seller on the initiator side and a public process PIP3A7_V02.00_Buyer on the responder side.

2. **Private Processes** – Private processes are the link between RosettaNet, where RosettaNet business messages are exchanged between RosettaNet trading partners, and the non-RosettaNet trading partners, which use enterprise internal proprietary business processes and message formats. There are two private PIPs for the public PIPs on the initiator and responder sides. Technically, the private PIPs implement the interface to the public PIPs and the proprietary enterprise processes or business applications. In contrast to the public processes, private processes can be trading partner-specific. For example, PIP3A7_Buyer_Private is the private PIP interfacing to public process PIP3A7_V02.00_Buyer.

Interface Between Private and Public PIPs

Typically, the interfaces between the private and public processes consist of two items:

1. **BPML (Business Process Modeling Language) Messages** – BPML messages are used to communicate between BPML processes that implement the private or public RosettaNet processes. BPML messages contain references to RosettaNet business messages or status information. The communication works bidirectionally: Messages are sent from the private process to the public process and vice versa.
2. **Process Data Fields** – If a private process spawns a public process, additional bookkeeping information is passed through the process data environment to the spawned process. A private process has to implement a certain message protocol that is defined by the public PIP. The message protocol defines:
 - ◆ Message names
 - ◆ Fields and structure of each message
 - ◆ Values ranges or types of each field
 - ◆ Sequence and direction of messages

Some examples for the different interfaces are described in the tables below:

Interface PIP3A7_Seller_Private / PIP3A7_V02.00_Seller:

Message Name in Process Data	Direction	Structure/Fields	Values: Types
RequestMessage	Private process -> Public process	<POUpdateNotification> <PrimaryDocument.../> </POUpdateNotification> <PrivateCaller .../>	◆ PrimaryDocument: document reference (Service Content of a Purchase Order) ◆ PrivateCaller: IntegerProcess ID of private process

Message Name in Process Data	Direction	Structure/Fields	Values: Types
ProcessStatusMessage	Private process -> Public process	<ProcessStatus.../>	ProcessStatus: Boolean"true", if POUpdateNotification could be processed. "false", else
FinalPIPStatusMessage	Public process -> Private process	<FinalPIPStatus.../>	FinalPIPStatus: Boolean"true", if Acknowledgment for POUpdateNotificationMessa ge is received from the Buyer successfully. "false", else
TP ProfileInformationProcess data	Private process -> Public process	<.../>	TP Profile for 3A7

Interface PIP3A7_V02.00_Buyer / PIP3A7_Buyer_Private:

Message Name in Process Data	Direction	Structure/Fields	Values: Types
POUpdateNotificationMessage	Public process -> Private process	<RequestMessage> <PrimaryDocument.../> </RequestMessage><Caller .../>	PrimaryDocument: document reference (Service Content (SC) of a Purchase Order (PO)). Caller:IntegerProcess ID of public process
PrivateResponseMessage	Private process -> Public process	[<POUpdateNotificationMes sage> <PrimaryDocument.../></P OUpdateNotificationMessag e>]	PrimaryDocument: document reference (Service Content of a POUpdateNotificationMessa ge)
FinalPIPStatusMessage	Public process -> Private process	<FinalPIPStatus.../>	FinalPIPStatus: Boolean"true", if Acknowledgment for PO UpdateNotification is sent to the Seller successfully. "false", else

Example (PIP 3A7 Seller-Buyer Dialog)

The Seller sends a PurchaseOrderUpdateNotification Message to the Buyer. The bootstrapped business process of the Buyer system is kicked off when the message is received by the HTTP-Server. The bootstrapped business process initiates the de-enveloping of the business message and sends back an Acknowledgment Signal to inform the Seller that his or her message was received successfully. Then the business process passes the message to the PIP Dispatcher. The PIP Dispatcher decides whether a new PIP has to be started or the message should be passed to an existing public process.

Implementing the Link Between Private PIPs and Internal Business Processes

The private processes shipped with the application core installation serve as business processes for trading partner-specific implementations. The private processes provide the flexibility to adapt the enterprise internal processes to the public RosettaNet processes by writing customized business logic in BPML. This means the process designer is granted full control of the private process content.

Caution: If the private process business processes are modified, it is the responsibility of the process designer to ensure that the new implementation matches the interfaces of the public processes. Disregarding this rule may result in failures or interruptions during execution of the complete chain of processes.

Some examples for the internal processes is described in the following tables:

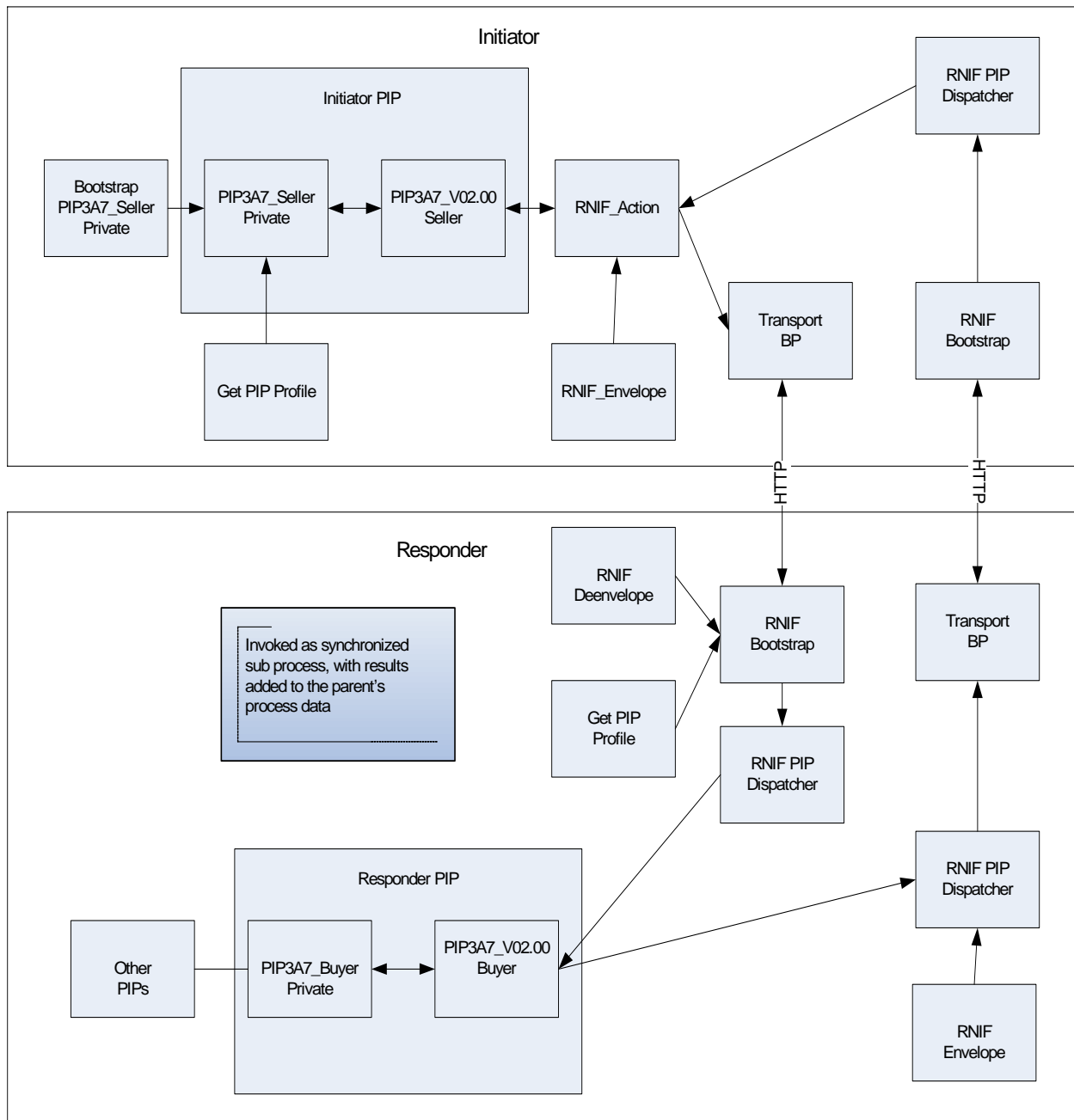
Interface Internal process / PIP3A7_Seller_Private:

Message Name in Process Data	Direction	Structure/Fields	Values: Types
RequestMessage	Internal enterprise process -> Private process	<POUpdateNotification> <PrimaryDocument.../></POUpdateNotification>	PrimaryDocument: document reference (Service Content of a Purchase Order)

Interface Internal / PIP3A7_Buyer_Private:

Message Name in Process Data	Direction	Structure/Fields	Values: Types
PO Update Notification	Private process -> Internal enterprise process	[<POUpdateNotification> <PrimaryDocument.../> </POUpdateNotification>]	PrimaryDocument: document reference (Service Content of a POUpdateNotification message)

The following figure provides an overview of the interdependencies of the different processes for the private, public, and RNIF layers. The figure also shows the private and public processes for PIP3A7 RNIF Version 2.0 only.



Installing PIPs into the Application

The following steps describe how to install PIPs into the application:

1. You can download the information about any PIPs you need to install from www.rosettanet.org. The document that you download contains information about the values for the PIP description, PIP roles, and other information required to add the PIP. From the document you can also determine whether the PIP is a One-Action PIP or Two-Action PIP. A One-Action PIP consists of a document sent by the PIP Initiator, to which the PIP Responder replies with a Receipt Acknowledgment. A Two-Action PIP is more complex, consisting of two or more actions.
2. The `<install_dir>/data/rosettanet` directory contains the `oneactionpipdeftemplate.xml` and `twoactionpipdeftemplate.xml` files. Choose the file that is appropriate for the PIP you are adding and copy the file to a new name. For example, the 3A4 PIP is a two-action PIP, so for this PIP you should use the two-action PIP def template.
3. Use the file downloaded from the RosettaNet site as a guide when filling in values for the various fields. An example for the One-Action PIP 2A10 is the following:

Note: The TODO comments may be misleading, because they do not appear in the correct places. The `TO_ROLE`, for example, is the PIP Responder's role, not the PIP Initiator's Role as the TODO comment implies:

```
<?xml version="1.0" encoding="utf-8"?>
<RosettaNetPIPDefinitions>
<PIP_DEF>
<!-- TODO: Insert PIP code here -->
<PIP_CODE>2A10</PIP_CODE>
<!-- TODO: Insert PIP initiator's role here -->
<TO_ROLE>Design Engineering Information Receiver</TO_ROLE>
<!-- TODO: Insert PIP responder's role here -->
<FROM_ROLE>Design Engineering Information Sender</FROM_ROLE>
<!-- TODO: Insert PIP initiator's service here -->
<TO_SERVICE>Process Design Engineering Information Service</TO_SERVICE>
<!-- TODO: Insert PIP responder's service here -->
<FROM_SERVICE>Design Engineering Message Information Service</FROM_SERVICE>
<!-- TODO: Insert PIP version here -->
<PIP_VERSION>V01.00</PIP_VERSION>
<RNIF_VERSION>All</RNIF_VERSION>
<!-- TODO: Insert PIP description here -->
<PIP_DESCRIPTION>Distribute Design Engineering Information</PIP_DESCRIPTION>
<STATUS>3</STATUS>
<TYPE>1</TYPE>
<RESPONSE_MODE>1</RESPONSE_MODE>
<SENDER_WFD_NAME>PIP2A10_V01.00_Design-Eng-Info-Sender_Private</SENDER_WFD_NAME>
<RESPONDER_WFD_NAME>PIP2A10_V01.00_Design-Eng-Info_Receiver_Private</RESPONDER_WFD_NAME>
<NONREPUD_REQ>1</NONREPUD_REQ>
<TIME_TO_ACK>120</TIME_TO_ACK>
<TIME_TO_ACK_ACCEPT>-1</TIME_TO_ACK_ACCEPT>
<TIME_TO_PERFORM>1440</TIME_TO_PERFORM>
<RETRY_COUNT>3</RETRY_COUNT>
<AUTH_REQ>1</AUTH_REQ>
<NONREPUD_OF_ORIGIN>1</NONREPUD_OF_ORIGIN>
<PIP_STEP>
<STEP_ID>1</STEP_ID>
<!-- TODO: Insert PIP step #1 name here -->
<PIP_BUS_ACTION>Distribute Design Engineering Information Action</PIP_BUS_ACTION>
<TIME_TO_ACK_RCPT>120</TIME_TO_ACK_RCPT>
<TIME_TO_ACK_ACPT>-1</TIME_TO_ACK_ACPT>
```

```

<TIME_TO_RESPOND>1440</TIME_TO_RESPOND>
<IN_TIME_TO_PERFORM>1</IN_TIME_TO_PERFORM>
<AUTH_REQ>1</AUTH_REQ>
<NONREPUD_REQ>1</NONREPUD_REQ>
<SECURE_TRANSP_REQ>1</SECURE_TRANSP_REQ>
<DSIG_REQ>1</DSIG_REQ>
<SSL_REQ>1</SSL_REQ>
</PIP_STEP>
<PIP_STEP>
<STEP_ID>2</STEP_ID>
<PIP_BUS_ACTION>Receipt Acknowledge</PIP_BUS_ACTION>
<TIME_TO_ACK_RCPT>-1</TIME_TO_ACK_RCPT>
<TIME_TO_ACK_ACPT>-1</TIME_TO_ACK_ACPT>
<TIME_TO_RESPOND>1440</TIME_TO_RESPOND>
<IN_TIME_TO_PERFORM>0</IN_TIME_TO_PERFORM>
<AUTH_REQ>1</AUTH_REQ>
<NONREPUD_REQ>1</NONREPUD_REQ>
<SECURE_TRANSP_REQ>1</SECURE_TRANSP_REQ>
<DSIG_REQ>1</DSIG_REQ>
<SSL_REQ>1</SSL_REQ>
</PIP_STEP>
</PIP_DEF>
</RosettaNetPIPDefinitions>

```

4. Once the template has been correctly filled out, save the file as 2A10template.xml and close the file. You can add the new PIP into the application by invoking pipdefloader.sh (or pipdefloader.cmd if you are using the Windows operating system) in the following way: pipdefloader.sh 2A10template.xml nodelete (the nodelete option prevents existing PIP definitions from being overwritten).
5. The zip file downloaded from the RosettaNet site may contain DTD files or schema files. Load the DTD(s) or Schema(s) into the application by choosing **Deployment** > **Schemas** from the application menu and follow the prompts displayed.
6. Just as there are oneactionpipdeftemplate.xml and twoactionpipdeftemplate.xml files, there are also business process template files provided for you in the <install_dir>/data/rosettanet/templates. These are the five files:
 - ◆ Launch_PIP_Private_Template.bpml
 - ◆ OneActionPIP_Initiator_Private_Template.bpml
 - ◆ OneActionPIP_Responder_Private_Template.bpml
 - ◆ TwoActionPIP_Initiator_Private_Template.bpml
 - ◆ TwoActionPIP_Responder_Private_Template.bpml

The launch template is used for testing the PIP. Its functionality is to place the input file into ProcessData as the Primary Document. Choose the template business processes appropriate to the PIP to be used (that is, use the one-action PIP templates for one-action PIPs, and choose the two-action PIP templates if implementing a two-action PIP). Follow the TODO instructions in each business process template file for instructions on how to customize the business process so that it is specific to that particular trading partner and PIP type.

7. It will also be necessary to set up records for Identity, Transport, Document Exchange, Delivery Channel, Packaging, Profile, and Contract information for each partner in the business relationship.

Note: Because the RosettaNet user interface does not have a wizard, you need to set these up manually. In the application Trading Partner interface, use **Trading Partner > Setup > Advanced** to setup the above information.

Implementation of RosettaNet with the Application

The application implementation of RosettaNet provides a framework that implements the RosettaNet Implementation Framework version 1.1 and 2.0, as defined by RosettaNet. The framework consists of a set of business processes (BPML), RosettaNet-specific services, and a trading partner user interface in the application.

The application performs the following tasks:

- ◆ Envelopes and deenvelopes RosettaNet business messages and signals.
- ◆ Sends and receives RosettaNet business messages and signals through HTTP.
- ◆ Defines trading partner agreements for RosettaNet using the Trading Partner menu.
- ◆ Runs asynchronous and synchronous PIP process models (both single-and two-action types).
- ◆ Handles error conditions using Failure of Notification messages (PIP0A1) and exceptions.

Core RosettaNet Components

Using predefined business processes, the application can implement RosettaNet PIP processes according to the RosettaNet framework. Business processes and other components that enable RosettaNet implementation are obtained when installing the application. The following table describes these business processes and other components:

Type	Name	Description
Private layer business processes	<ul style="list-style-type: none"> ◆ PIP0A1_Failure-Report-Administrat or_Private.bpml ◆ PIP0A1_PIP-Failure-Notifier_Private.bpml ◆ PIP2A10_Receiver_Private.bpml ◆ PIP2A10_Sender_Private.bpml ◆ PIP2A12_V01.01.00_Buyer_Private.bpml ◆ PIP2A12_V01.01.00_Supplier_Private.bpml ◆ PIP3A4_Buyer_Private.bpml ◆ PIP3A4_RNIF11_Buyer_Private.bpml ◆ PIP3A4_RNIF11_Seller_Private.bpml ◆ PIP3A4_Seller_Private.bpml 	Predefined business processes for trading partner-specific private processes.

Type	Name	Description
Private layer business processes	◆ PIP3A6_V02.02.00_Buyer_Private.bpml	Predefined business processes for trading partner-specific private processes.
	◆ PIP3A6_V02.02.00_Seller_Private.bpml	
	◆ PIP3A7_Buyer_Private.bpml	
	◆ PIP3A7_Seller_Private.bpml	
	◆ PIP3A8_Buyer_Private.bpml	
	◆ PIP3A8_Seller_Private.bpml	
	◆ PIP3A9_Buyer_Private.bpml	
	◆ PIP3A9_Seller_Private.bpml	
	◆ PIP3B12_Shipper_Private2.bpml	
	◆ PIP3B12_Shipping_Provider_Private2.bpml	
	◆ PIP3B12_V01.01.00_Shipper_Private.bpml	
	◆ PIP3B12_V01.01.00_Shipping-Provider_Private.bpml	
	◆ PIP3B13_Shipper_Private.bpml	
	◆ PIP3B13_Shipping_Provider_Private.bpml	
	◆ PIP3B18_Shipper_Private.bpml	
	◆ PIP3B18_Shipping_Provider_Private.bpml	
	◆ PIP3B18_V01.01.00_Shipper_Private.bpml	
	◆ PIP3B18_V01.01.00_Shipping-Provider_Private.bpml	
	◆ PIP3B2_Receiver_Private.bpml	
	◆ PIP3B2_Shipper_Private.bpml	

Type	Name	Description
Private Layer business processes	◆ PIP3C3_Invoice_Provider_Private.bpml	Predefined business processes for trading partner-specific private processes.
	◆ PIP3C3_Invoice_Receiver_Private.bpml	
	◆ PIP3C4_Invoice_Reject_Provider_Private.bpml	
	◆ PIP3C4_Invoice_Reject_Receiver_Private.bpml	
	◆ PIP3C6_Payee_Private.bpml	
	◆ PIP3C6_Payer_Private.bpml	
	◆ PIP3C7_Invoice_Provider_Private.bpml	
	◆ PIP3C7_Invoice_Receiver_Private.bpml	
	◆ PIP3D9_Solution_Provider_Private.bpml	
	◆ PIP3D9_Solution_Requester_Private.bpml	
	◆ PIP4A1_V02.00_Customer_Private.bpml	
	◆ PIP4A1_V02.00_Supplier_Private.bpml	
	◆ PIP4A3_Forecast_Owner_Private.bpml	
	◆ PIP4A3_Forecast_Recipient_Private.bpml	
	◆ PIP4A5_Forecast_Recipient_Private.bpml	
	◆ PIP4A5_Forecast_Reply_Recipient_Private.bpml	
	◆ PIP4B2_Consignee_Private.bpml	
◆ PIP4B2_User_Private.bpml		

Type	Name	Description
Private Layer business processes	◆ PIP4B3_V01.00.00_Buyer_Private.bpml	Predefined business processes for trading partner-specific private processes.
	◆ PIP4B3_V01.00.00_Seller_Private.bpml	
	◆ PIP4C1_Provider_Private.bpml	
	◆ PIP4C1_User_Private.bpml	
	◆ PIP5C1_Provider_Private.bpml	
	◆ PIP5C1_User_Private.bpml	
	◆ PIP5C2_Demand_Creator_Private.bpml	
	◆ PIP5C2_Product_Provider_Private.bpml	
	◆ PIP5C4_Demand_Creator_Private.bpml	
	◆ PIP5C4_Product_Provider_Private.bpml	
	◆ PIP5D1_Product_Distributor_Private.bpml	
	◆ PIP5D1_Product_Supplier_Private.bpml	
	◆ PIP5D3_Distributor_Private.bpml	
	◆ PIP5D3_Supplier_Private.bpml	
	◆ PIP5D5_Distributor_Private.bpml	
	◆ PIP5D5_Supplier_Private.bpml	
	◆ PIP7B1_Solution_Provider_Private.bpml	
	◆ PIP7B1_Solution_Requester_Private.bpml	
	◆ PIP7B5_Buyer_Private.bpml	
	◆ PIP7B5_Seller_Private.bpml	
◆ PIP7B6_Buyer_Private.bpml		
◆ PIP7B6_Seller_Private.bpml		

Type	Name	Description
Public PIP layer business processes	◆ PIP0A1_V02.00.bpml	Predefined business processes that Implement the RosettaNet PIPs.
	◆ PIP0A1_V02.00_Failure-Report-Administrator.bpml	
	◆ PIP0A1_V02.00_PIP-Failure-Notifier.bpml	
	◆ PIP2A10_V02.00_Receiver.bpml	
	◆ PIP2A10_V02.00_Sender.bpml	
	◆ PIP2A12_V01.01.00_Buyer.bpml	
	◆ PIP2A12_V01.01.00_Supplier.bpml	
	◆ PIP2C3_V01.00.00_Requester.bpml	
	◆ PIP2C3_V01.00.00_Reviewer.bpml	
	◆ PIP3A1_V02.00_Buyer.bpml	
	◆ PIP3A1_V02.00_Seller.bpml	
	◆ PIP3A7_V02.00_Buyer.bpml	
	◆ PIP3A7_V02.00_Seller.bpml	
	◆ PIP3A8_V01.00_Buyer.bpml	
	◆ PIP3A8_V01.00_Seller.bpml	
	◆ PIP3A9_V01.00_Buyer.bpml	
	◆ PIP3A9_V01.00_Seller.bpml	
	◆ PIP3B12_V01.01.00_Shipper.bpml	
	◆ PIP3B12_V01.01.00_Shipping-Provider.bpml	
	◆ PIP3B18_V01.01.00_Shipper.bpml	
	◆ PIP3B18_V01.01.00_Shipping-Provider.bpml	
	◆ PIP4B3_V01.00.00_Buyer.bpml	
	◆ PIP4B3_V01.00.00_Seller.bpml	
	◆ PIP4C1_V02.00_Provider.bpml	
	◆ PIP4C1_V02.00_User.bpml	
	◆ PIP5C1_V02.00_Provider.bpml	
	◆ PIP5C1_V02.00_User.bpml	

Type	Name	Description
RNIF layer business processes	◆ RNHTTPAsyncSend.bpml	Complete message enveloping and transport tasks
	◆ RNIF_Action.bpml	
	◆ RNIF_Bootstrap.bpml	
	◆ RNIF_DeEnvelope.bpml	
	◆ RNIF_Envelope.bpml	
	◆ RNIF_PIP_Dispatcher.bpml	
	◆ RNIF_SendAck.bpml	
	◆ RNIF_SendException.bpml	
	◆ RNIF_TwoAction_Async_Responder.bpml	
	◆ RNPIPTracking.bpml	
	◆ HTTP_Sync_Response.bpml	
	◆ GetPipProfileData.bpml	
	◆ RNIF_Attachment.bpml	
	◆ RNPIPInitiator.bpml	
◆ RNPIPResponder.bpml		
Schemas and DTDs	◆ RNIF20Attachments.dtd	Required for services on RNIF layer (for example, message parsing). Note: RNIF header DTDs, Acknowledgment, and Exception DTDs for both RNIF 1.1 and RNIF 2.0 should be obtained from the RosettaNet Web site.
	◆ RNIF20MIMEHeaders.dtd	
	◆ RNIF20OutboundMIC.dtd	
	◆ RNIF20ReceivedContentMIC.dtd	
	◆ RNIF20ReceivedMessageInfo.dtd	
Services	RNIF Message Builder Service	Envelopes RosettaNet messages
	RNIF Message Parser Service	Parses RosettaNet messages
	RosettaNet Profile Service	Loads RosettaNet contracts and trading profile data
	RosettaNet PIP Tracking Service	Used to track the correlation details of the PIP

Type	Name	Description
Additional business processes	◆ Launch_PIP_Private_Template.bpml	Business processes used in the RosettaNet implementation
	◆ OneActionPIP_Initiator_Private_Template.bpml	
	◆ OneActionPIP_Responder_Private_Template.bpml	
	◆ TwoActionPIP_Initiator_Private_Template.bpml	
	◆ TwoActionPIP_Responder_Private_Template.bpml	

Scope of Implementation

The application supports the following RosettaNet 1.1 and 2.0 core specifications:

- ◆ Single-action and two-action processes.
- ◆ Asynchronous messaging.
- ◆ Shipped PIPs: 0A1, 2A10, 2A12, 2C3, 3A1, 3A4, 3A6, 3A7, 3A8, 3A9, 3B12, 3B13, 3B18, 3B2, 3C3, 3C4, 3C6, 3C7, 3D9, 4A1, 4A3, 4A5, 4B2, 4B3, 4C1, 5C1, 5C2, 5C4, 5D1, 5D3, 5D5, 7B1, 7B5, 7B6.
- ◆ Error / Exception handling: All PIP 0A1 calling types for single-action / two-action (synchronous and asynchronous); retries and time outs.
- ◆ Character sets UTF-8.
- ◆ Encryption algorithms:
 - ◆ Digital signature
 - ◆ HTTP transport

Synchronous PIPs Limitations

This section describes limitations for some synchronous PIPs, according to RosettaNet specifications.

Handling the x-RN-Response-Type Header

RosettaNet messages include header information to indicate whether a synchronous or asynchronous response is expected. The parsing service in the application obtains the value included in the message response and uses it to determine whether to close the connection (for asynchronous) or leave it open (for synchronous).

Limitation: Because the application uses the value during a lookup, the application overrides the header information. Therefore, if the response-type differs from the profile value, the remaining message exchange will be incorrect.

Validating the x-RN-Response-Type Header

According to the specification, if the header is present and the value does not match (case-insensitive) either async or sync, then the request could be rejected with a code of 400 (Bad Request).

Note: The RNIF20 Parser service typically detects any invalid response-type headers.

Limitation: For this case, rejecting the message with a code of 400 is not implemented.

Bad Response Codes

Response code 202 results for messages sent to the application that are correct for asynchronous communication and incorrect for synchronous communication.

Empty Message as Acknowledgment for Synchronous Single-action PIPs

According to the specification, a response with a code of 200 and an empty entity-body message indicates an acknowledgment. Closing the connection with response code 200 not sending a message for acknowledgment causes the RosettaNet framework to stop processing.

Limitation: The following actions occur in the current implementation:

For asynchronous messages, the application returns response code 202 (appropriate action)

If the message is not asynchronous (async is case insensitive), the application returns response code 501.

For any errors during processing a message (for example, authentication errors), the application returns response code 500.

Authentication and Authorization Failures

According to the specification, if the application cannot authenticate a message (for example, non-repudiation of origin or no corresponding contract), the application returns 403-error code or closes the connection without returning a code. While this specification applies to synchronous responses, it can also apply to HTTP asynchronous responses.

Limitation: Not implemented in the current version.

Implementing RosettaNet in the Application

The following process is an overview of the tasks you must complete to implement the RosettaNet in the application:

1. Ensure that the application is installed. For information, see the *Installation Guide*.
2. Obtain a license for the RosettaNet Standard from the Sterling Commerce Self-Service Licensing Web site at www.productupdates.stercomm.com.
3. Download specific document type definitions (DTDs) from the RosettaNet Web site. For information, see *Downloading RosettaNet DTDs* on page 25.
4. You must have an EDIINT certificate to import into the application.

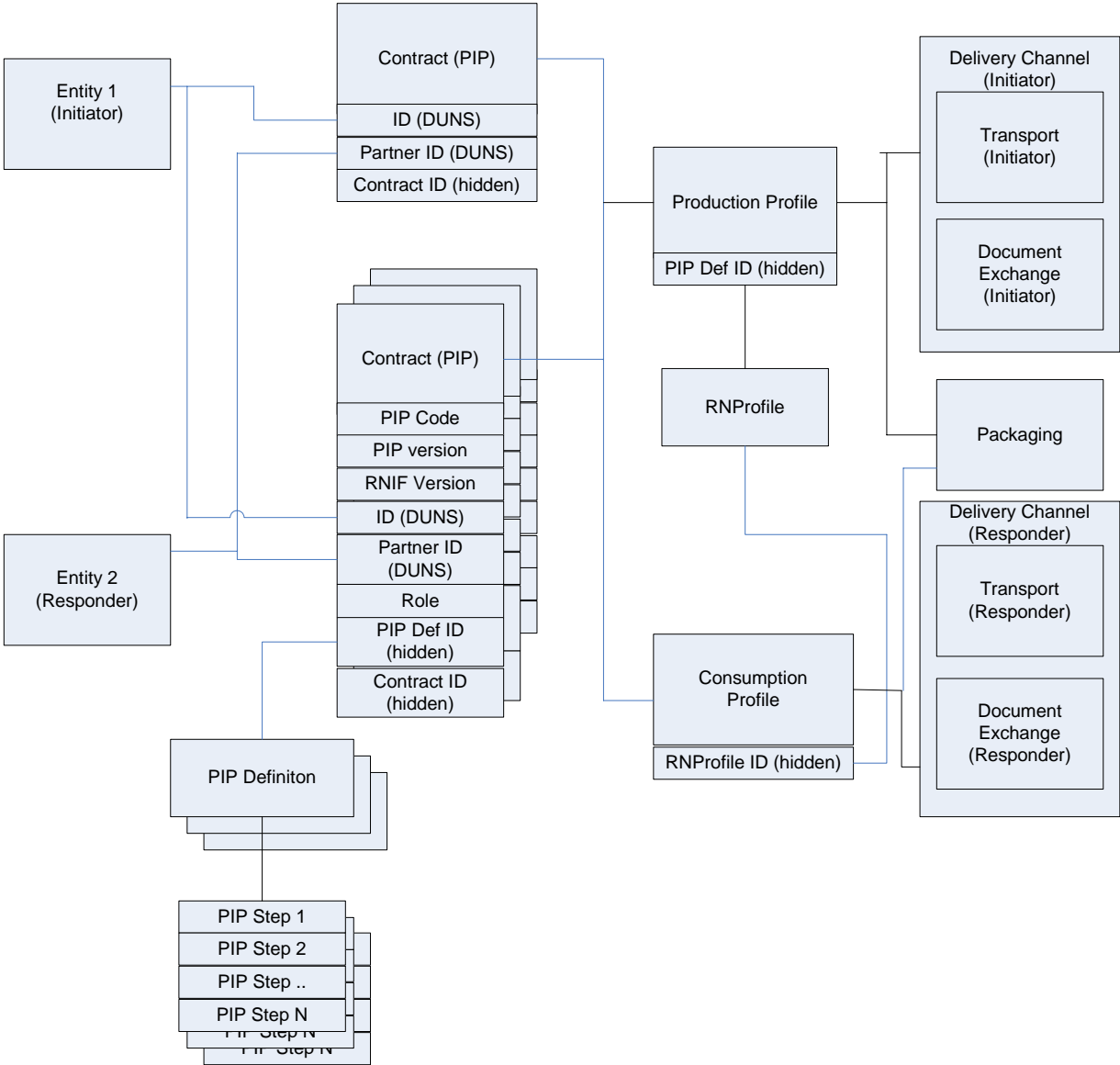
Downloading RosettaNet DTDs

The application requires the specific DTDs to ensure an accurate RosettaNet implementation. You can obtain the DTDs from the RosettaNet Web site at www.rosettanet.org. RosettaNet provides the DTDs in ZIP files that you download. After downloading the ZIP files, transfer them to the machine where the application is installed.

RosettaNet Trading Partner Objects

The trading partner agreements consist of several substructures. The following figure provides an overview of the different substructures that make up trading partner agreements (objects) on the initiator side and responder side:

RosettaNet Trading Partner Objects



Filling Logic into the Private Process Business Processes

To fill logic into the private process business processes you can use any text editor or the text editor in the application. The following procedure describes the steps to use the text editor in the application. If you use another text editor, you still need to copy the text and paste it into the text editor in the application for check in and usage.

To fill logic into the private process business processes:

1. From the Admin Console, select **Business Process > Manager**.
2. Under Create, next to Process Definition, click **Go!**
3. Type a unique name for the business process.
4. Under **Select an input mode for defining the new process**, select **Business Process Text Editor** and click **Next**.
5. Type a description for the business process that you are creating.
6. In the **Business Process** field, either type or copy and paste BPML code.
7. Click **Validate** to validate the BPML syntax and if valid click **Next**.

Note: You cannot enable a business process until it is validated.

8. In the Process Editor, complete the following configuration parameters for the business process you are creating and click **Next**:

Parameter	Description
Document Tracking	When document tracking is enabled for a business process, tracking information is carried with the message throughout the process, and the tracking information persists with the message regardless of the persistence level you configured globally for the application.
Set onfault processing	Onfault processing allows the process to immediately execute the onfault activity specified in the process, even if the process has not yet reached that step in the process. For example, if a process fails at step 3, but the onfault activity is specified in step 7, if onfault processing is enabled, the process proceeds to the step 7 onfault rather than halting at step 3.
Set Queue	The application enables you to set performance optimizations by queue, defining queue levels to allocate resources. This number indicates the previously allocated queue level that you want for this business process model. Default is 4.
Enable Async Start Mode	Asynchronous initiation is selected by default. Starting business processes asynchronously is recommended. Asynchronous mode is standard application processing, wherein the business process is placed in queue and processed. Note: <i>Always</i> use asynchronous mode, with one possible exception. If you are running this process in Sync mode, you must use synchronous start. You may be using Sync mode if, for example, you are using the MSMQ adapter, set to Embedded mode. Clear this check box to use synchronous mode. Sync mode functionality is limited in that your Persistence Level is Zero by default, and you cannot, therefore, manually restart or resume the process.
Use BP Queuing (Recommended)	Whether to queue business processes (this may improve system performance).

Parameter	Description
Enable Transaction	<p>Select this option to instruct the application to treat the entire process as a single transaction, so that either all of the steps complete, or, in the event of an error, none of them do. When an error occurs, no data is committed; data returns to its pre-process state.</p> <p>Note: Enabling transactions applies only to services that support transaction mode. See the reference information for specific services to determine whether this option is supported.</p> <p>By default, this transaction mode is not enabled.</p> <p>Note: If transaction management is already built into the process model (the model includes Start Transaction Services and End Transaction Services), do not select this option or the process fails.</p>
Commit All Steps when there is error	Choose this option if you do not need tracking data for your process when it is successful and your recovery does not depend on process step data being saved.
Category	You can optionally enter a category name to which this process model belongs, creating the category. The category does not affect processing; categories are for future product enhancements.
Persistence Level	<p>The level of data to retain for generating a status report that describes each step that the business process completes. System default indicates that the configuration is already defined in the application to retain data. The different levels are:</p> <ul style="list-style-type: none"> ◆ Full ◆ Step Status - Engine May Override ◆ BP Start Stop - Engine May Override ◆ System Default ◆ Step Status Only ◆ BP Start Stop Only ◆ Zero ◆ Error Only ◆ BP Start Stop Only (No Errors) <p>The default is Full.</p>
Event Reporting Level	<p>The level of event reporting that is retrieved for this business process when it runs. Full specifies to generate events for the business process, including the business process start and end time, start and end times for all services or services running as a result of this business processes, and any resulting errors and exceptions. The different levels are:</p> <ul style="list-style-type: none"> ◆ Full ◆ Minimal ◆ None. <p>The default is None.</p>

Parameter	Description
Recovery Level	<p>The level of recovery for this business process if the business process should halt during execution. Manual requires you to resume or restart the business process manually. The different levels are:</p> <ul style="list-style-type: none"> ◆ Auto Resume ◆ Auto Restart ◆ Terminate ◆ Manual <p>The default is Manual.</p>
Document Storage Type	<p>The level of document storage for messages that process when the business process runs. System Default specifies to store messages in the file system or database, according to how you configured archiving and purging in the application. The default types are:</p> <ul style="list-style-type: none"> ◆ Database ◆ File System ◆ System Default ◆ Inherited <p>The default is System Default.</p>
Deadline Settings	<p>Complete by – The deadline time, in hours and minutes, by which the business process must complete after start.</p> <ul style="list-style-type: none"> ◆ First Notification: Hours and Minutes – Whether to receive notification before a business process deadline. ◆ Second Notification: Hours and Minutes – Whether to receive another notification before a business process deadline. <p>Default is Do not set deadline.</p> <p>Note: To set a deadline you must change it in the business process.</p>
Life Span	<p>The length of time, in days and hours, to retain the data in the application, along with the life span type and removal method. The options are:</p> <ul style="list-style-type: none"> ◆ Process Specific: Enables you to define the length of time, in days and hours, to retain the data in Sterling Integrator. You must indicate numeric values, even when you do not want to configure archive information—for example, the value 0 (zero) indicates that you do not want to configure archive information. <p>Note: Indicate whether you want the expired business process to be archived or purged.</p> <ul style="list-style-type: none"> ◆ System Default: Assume that for the data, archiving configuration is already defined in the application. <p>Default is System Default.</p>

9. In the Confirm page, ensure that **Enable Business Process** is selected, and click **Finish**.

Note: If you choose not to enable this business process, clear the **Enable Business Process** check box. When you are ready to enable the business process, from the Admin Console, select **Business Process > Manager**. In the List section, select **ALL** and locate the business process. Next to the

business process, click **Execution Manager**. Under **Enable**, select the check box next to the business process.

Example (Writing code for PIP3A7_Buyer_Private)

The following steps provide an example of how the application adds or modifies the code:

1. Header comment with a process description.
2. Declaration of process name: <process name = "PIP3A7_Buyer_Private">
3. Code for consuming the POUpdateNotificationMessage.
4. According to the RosettaNet specification, the Purchase Order Update Notification has to be processed.

Note: RosettaNet does not make any assumptions about the infrastructure of your business and, therefore, cannot start transactions for you. If a business process fails before completion, you must manually start a new transaction.

Code for Analyzing the PO Update Notification

Typically, the following steps are performed to analyze a PO Update Notification:

1. Translate PO Update Notification into the application format of the ERP system.

Note: The ERP system mentioned here is an external system that interfaces with the application usually at the customer end.

2. Pass translated PO Update Notification to the ERP system.
3. Transform response of ERP system into RosettaNet document(s).
4. Pass the translated response document(s) back to private PIP 3A7.
5. Write the logic to read POChangeRequest or POCancellationRequest from the file system. If you do not want to read the messages from the file system, you can change it. You must be sure your code matches the interface defined above.
6. Send POChangeRequest or POCancellationRequest to public PIP 3A7.
7. Consume the FinalPIPStatus from the public 3A7 Seller process. If the value is true, an Acknowledgment for the POChangeRequest or POCancellationRequest has been received and the PIP3A7 process finishes with a success status. If the value is false, it might be necessary to roll back the transaction in step 8.
8. If you have opened a transaction in step 4, you need to insert the code to close the transaction for PIP3A7 or roll back the transaction. Instead of an automated rollback, sending an e-mail notification to the responsible administrator is an alternative “manual” approach.
9. If pending items exist, private process 3A7 is started and the POUpdateNotification service content is passed through the process data.

Error Cases in Private PIPs

These are some RosettaNet cases in which a private process causes an error condition, for example:

- ◆ On the Initiator side in a two-action process: the private process is unable to process the response message successfully. In this case, the flag `ProcessStatus` must be set to `false`. After passing the `ProcessStatus` back to the public PIP, a `PIPOA1` will be sent to the responder side automatically.
- ◆ On the initiator side, a private process is started without `PrimaryDocument` (that is, no `PO Update Notification`): in this case public PIP is not started and the `FinalPIPStatus` is set to `Error: Process started without PO Update Notification document`.
- ◆ On the responder side in a single- or two-action process: the private process is unable to process the request document (for example, no `PrimaryDocuments` are read from file system). In this case, a `PIPOA1` is sent back to the initiator automatically.

Error Handling and Auditing

PIP 0A1 (Notification of Failure or NoF) is a process control PIP that is used to communicate an exception condition that occurs outside the normal message choreography of the subject PIP.

Typically, PIP 0A1 is invoked when exceptions happen under a specific condition (namely, when the process is in “execution” state at one partner's system and may have possibly reached a “completed” state in the other partner's system) during the execution of any other process.

Execution of PIP 0A1 is shown in the asynchronous and synchronous interaction diagrams (chapter 2.6.8 of the RNIF Core spec) in both single-action and two-action activities.

A failure to complete an interface business activity in compliance with the RNIF 2.0 or RNIF 1.1 core specification can be detected by either of the two parties involved in the interface activity.

In seven different contexts (Calling Types)- the PIP 0A1 is started. The following table describes each Calling Type in detail:

Calling Type	Initiating PIP or RNIF Process	Failure Reason
1	RNIF_Action (Two or Single Action)	The action process sent a request message on initiator side and is waiting for an acknowledgment. Maximum numbers of retries exceeded or time to perform expired.
2	PIP3A7_Buyer_Private (Single Action)	During processing, the business message on the responder side of the private process indicates that it cannot process the business message.
3	RNIF_Action (Two Action)	Time-out during waiting for response message on initiator side.
4	RNIF_Action (Two Action)	During processing of the response message in the private process on the initiator side, the private process indicates that it cannot process the business process.
5	RNIF_Two_Action_A sync_Responder (Two Action)	The action process sent a response message and is waiting for an acknowledgment. Maximum numbers of retries is exceeded.
6	RNIF_Action (Two Action)	Communications error occurred during sending a business message from the initiator to the responder
7	RNIF_Two_Action_A sync_Responder (Two Action)	Communications error occurred during sending a business message from the responder to the initiator.

Similar to other RosettaNet business processes, PIP0A1 has an initiator and a responder process. The initiator process is implemented in the BPML module PIP0A1_V02.00. In the current implementation, there is no private process for PIP0A1 on the initiator side. The responder (which has a public and a private part) is implemented in the modules PIP0A1_V02.00_Failure-Report-Administrator and PIP0A1_Failure-Report-Administrator_Private. The RNIF 1.1 equivalent for the modules specified are PIP0A1_1.0.bpml, PIP0A1_1.0_Failure-Report-Administrator.bpml, and PIP0A1_Failure-Report-Administrator_Private.bpml. The initiator creates a PIP0A1 message and sends it. On the responder side, the framework starts the public responder process, which spawns the private process. Within the private process, customized code can be used to handle the error condition adequately.

Business Process Manager and Monitor

The RosettaNet processing in the application uses a number of public processes to ensure that the document handling is RNIF 1.1 or 2.0 compliant. As stated before, the interface between the public and private processes is done using the PIP<PIPnumber>_<role>_private processes. The following table shows the business processes in the application after starting PIP 3A7 with PIP3A7_Seller_Private.

Here, the roles are Initiator (Seller) and Responder (Buyer), as specified the RosettaNet PIP 3A7 documentation.

Role	Business Process	Action
Initiator	PIP3A7_Seller_Private	Message is received from the private process, partner lookup, pass message to public process
	PIP3A7_V02.00_Seller	Public Process of PIP3A7
	RNIF_Action	Initiates enveloping and sending
	RNIF_Envelope	MIME enveloping takes place
	RNHTTPAsyncSend	Message is being sent to the partner
Responder	RNIF_Bootstrap	Message is received by the responder and the initial context is created
	RNIF_DeEnvelope	Message de-enveloping
	GetPipProfileData	Partner lookup
	HTTP_Sync_Response	Send HTTP success response to the client
	RNIF_SendAck	Generate the Signal message for the initiator
	RNHTTPAsyncSend	Send the Signal message to the initiator
Initiator	RNIF_Bootstrap	Message is received by the Initiator and the initial context is created
	RNIF_DeEnvelope	Action message de-enveloping
	GetPipProfileData	Partner lookup
	HTTP_Sync_Response	Send HTTP success response to the client
	RNIF_PIP_Dispatcher	Check for existing process. If "no": Start new process
	PIP3A7_V02.00_Buyer	Message is received by the Responder
	PIP3A7_Buyer_Private	Hand over to private process

Checking the Final PIP Status

To check the final PIP status on the Seller side:

1. In Sterling Integrator, select **Business Process > Monitor**.
2. Select the PIP3A7_Seller_Private ID.

3. Select the Instance Data of the last process step.
4. Open the Process Data.
5. In the second last line, you can see the FinalPIPStatus. This should be `<FinalPIPStatus>true</FinalPIPStatus>`.

Logging

The RosettaNet solution uses the application standard mechanism for logging. The RosettaNet log file `rnif.log` is located in the folder `<install>/logs`. As a default, configuration errors are logged only. Refer to the application documentation for details about changing logging properties.

For testing and debugging, increase the log level as necessary. For example, set log level to ALL and the default values in `<install>/properties/Log.properties`:

```
rnlogger.logfilename = <install>/logs/rnif.log
rnlogger.logkey = rnlogger
rnlogger.rotatelogs = true
rnlogger.maxlogsize = 100000
rnlogger.loglevel = ERROR
rnlogger.displayname = Log.WorkFlow
```

Using RosettaNet in Production

We recommend that you have two separate instances of the application. The first instance serves as a Test system and the second as a production system. Before you switch the GlobalUsageCode on the test system to production, ensure that the tests cover all production scenarios and have been finished successfully.

GlobalUsageCode

RosettaNet messages contain a GlobalUsageCode flag in the Service Header that indicates the intended use of the message. These codes are:

- ◆ Test
- ◆ Production

Ensure that your RosettaNet trading partner is using the same GlobalUsageCode settings as you. The GlobalUsageCode is configured in the trading partner contract.

To configure the GlobalUsageCode:

1. From the application **Trading Partner** menu, select **Partners > Contracts**.
2. In the List alphabetically section, click **Go!**
3. Select the contract that you want to configure and click **edit**.
4. Click **Next** three times to proceed to the RosettaNet Extensions dialog.
5. Select the GlobalUsageCode option and then select either Test for a Test system or Production for a production system.
6. Click **Save** to update the application with your changes.

Inbound and Outbound Document Flow

The inbound (responder) document flow for a RosettaNet document through the application is as follows:

Inbound Sequence	Description
1	A RosettaNet message is sent using HTTP.
2	The RN HTTP Server adapter is configured to invoke the Bootstrap business process when a message arrives.
3	The RNIF Develope business process invokes the RosettaNet Message Parser service to construct the header elements into process data and extracts the service content and attachments.
4	The RosettaNet Message Parser service parses and validates the entire message.
5	Based on the result of the parsing, the RNIF Parsing service decides whether to send an Acknowledgment or an exception.
6	The Transaction Responder Private business process is invoked to determine if the message is a two-action PIP.
7	If the transaction is a two-action PIP, the responder sends the response Action Message to the initiator.

The outbound (initiator) document flow for a RosettaNet document through the application is as follows:

Outbound Sequence	Description
1	Using a compiled map, the translator translates the document in the application format to the RosettaNet document and creates a Primary Document.
2	The initiator invokes a business process that assigns any Attachment Files.
3	The Transaction Initiator Private business process is invoked by the initiator business process to set the PIP/Transaction code and any other Contract information.
4	The RNIF Envelope business process is invoked to construct the Message Header elements.
5	The RNIF Attachment business process is invoked to loop through the attachment files and construct the necessary attachment elements.
6	The RosettaNet Message Builder service is invoked to create the actual message using the header elements that were created, the primary document, and the attachment file information.
7	The RNIF HTTP Send business process is invoked to send the message to the responder using the RN HTTP Server adapter.

Working with RosettaNet Trading Partners

To effectively set up trading partners that implement RosettaNet, you must configure the application with specific RosettaNet message data settings. This section provides an overview of the steps to set up trading partner information, including the different types of contracts necessary, along with the specific configuration settings that enable you to meet your RosettaNet data exchange requirements. After setting up your trading partner information, you can then create and implement business processes that comply with RosettaNet standards.

The application provides you with attachment support, which enables you to specify files (of any format) to be attached, by using the attach file element, and specifying a path to the file.

To configure trading partner information to implement RosettaNet, you must:

1. Create a trading profile for your organization and each of your trading partners. The trading profile enables you to define how you want to:
 - ◆ Build and parse RosettaNet message data.
 - ◆ Define message security and transport protocols.
 - ◆ Determine whether you want to attach files.
2. Depending on contractual requirements agreed upon by you and your trading partner, you then create contracts for the following:
 - ◆ At least one Partner Interface Process™ (PIP). A PIP is a RosettaNet-defined business process designed to be a framework for business flows between RosettaNet trading partners. For each RosettaNet PIP you plan to exchange, you must create a contract. For example, if the company is initiating and responding to PIP 3A4, you must create a contract to initiate PIP 3A4 and one to respond to PIP 3A4.
 - ◆ RosettaNet-specific services to build and parse messages, promote security, and send outbound RosettaNet messages to the appropriate URLs.

For more information about:

- ◆ RosettaNet solution for the application, see the Sterling Commerce Customer Support Web site at www.sterlingcommerce.com/customer/tech_support.html.
The Web site provides access to Support on Demand, which enables you to download the application RosettaNet Implementation white paper. This white paper also describes the framework, business processes (BPML), and RosettaNet-specific services used to implement RosettaNet in the application.
- ◆ RosettaNet concepts and development, see the RosettaNet Web site at www.rosettanet.org.

Before You Begin

Before you set up a RosettaNet trading profile and to simplify the tasks you must perform, collect and prepare the following information about your organization and trading partners:

- ◆ Trading partner name and address information, including:
 - ◆ DUNS numbers

- ◆ Contact information, such as addresses, telephone numbers, e-mail addresses
- ◆ PIP information, including:
 - ◆ PIP code and PIP version to exchange
 - ◆ RNIF (RosettaNet Implementation Framework) version (either v1.1 or v02.00)
 - ◆ Roles of trading partners using each PIP (that is, the trading partners who is initiating and the trading partner who is responding)
- ◆ RNIF version information
- ◆ Transport information, including:
 - ◆ HTTP URLs
 - ◆ CA certificates for trading partner if using SSL
- ◆ Security information, digital signatures, and encryption and certificates (for example, certificates used for encryption and decryption)
- ◆ If you are using digital certificates, check in the appropriate certificate

Setting Up RosettaNet Trading Profiles

When setting up trading profiles in the application, you must perform the following tasks:

1. Create an identity record for your organization, indicating your organization as the base identity.
2. Create an identity record for each of your trading partners.
3. Create the following records in order to complete the trading profiles for your organization and each of your trading partners:
 - ◆ Transport: defines the URI where the PIP will be sent, certificates, SSL information, and so forth.
 - ◆ Document exchange: defines number of retries, and the Signing Certificate (if any) to be used.
 - ◆ Delivery channel: defines the Delivery Mode (Asynchronous HTTP for RosettaNet) and the Receipt Timeout (the necessary value is 86400 seconds).
 - ◆ Packaging (only one is necessary for each trading relationship regardless of the number of trading partners): defines how the data is to be sent (for example, Plain Text, Signed Detached, or Signed Detached Encrypted).
 - ◆ Profile: defines the Role (Initiator or Responder), Location Code, Global Partner Classification Code, and Supply Chain Code.
 - ◆ Contract: defines PIP Code, PIP Version, RNIF Version, and the Content Transfer Encoding to be used.

Creating an Identity Record for Your Organization

When setting up your trading profile, consider that your organization is the base identity.

To create an identity record:

1. From the **Trading Partner** menu, select **Setup > Advanced > Identities**.
2. Next to Create New Identity, click **Go!**
3. In the Base Identity page, for Select Base Identity, verify that **No Base Identities Available** is selected, and click **Next**.
4. In the Information page, complete the following fields and click **Next**:

Field	Description
Name	Name of your organization. Required.
Reference URL	Internet address of your organization. Optional.
DUNS	Unique 9-13 character identification number of your organization, as issued by Dun & Bradstreet Corporation. Required.
Identifier	Primary ID of your organization. Use DUNS number. Required.
Address Line 1	Conventional mailing address of your organization. Required.
Address Line 2	Additional mailing address information of your organization. Optional.
City	City where your organization is located. Required.
State or Province	State or province where your organization is located. Required.
Zip or Postal Code	ZIP Code or postal code where your organization is located. Required.
E-mail Address	E-mail address of your organization. Required.
Telephone Number	Telephone number of your organization. Required.
Time Zone	Time zone where your organization is located. Optional.
Country	Country where your organizations located. Optional.
Federation Peers	Discovery point URL for the trading partner you want to associate this identity with. Optional.
VAT ID	The VAT transaction tax identifier. Optional. Note: When you add VAT IDs to an identity, an associated permission ID is created, and the user is assigned that permission ID.
Internet Notification E-mail	The internal e-mail address to which you want error notifications to be sent. Optional.
Lookup ID	The identifier used to look up the e-Invoice agreement. Optional.
Lookup ID Comments	Any comments regarding the identifier used to look up the e-Invoice agreement. Optional.

5. Click **Finish** to add the identity.

Creating an Identity Record for Your Trading Partner

For each RosettaNet trading partner, you must create an identity record.

To set up an identity record for your trading partner:

1. From the **Trading Partner** menu, select **Setup > Advanced > Identities**.
2. Next to Create New Identity, click **Go!**
3. In the Base Identity page, for Select Base Identity, select the name of the identity record that you created for your organization, and click **Next**.
4. In the Information page, complete the following fields and click **Next**:

Field	Description
Name	Name of your RosettaNet trading partner. Required.
Reference URL	Internet address of your RosettaNet trading partner. Optional.
DUNS	Unique 9-13 character identification your RosettaNet trading partner number, as issued by Dun & Bradstreet Corporation. Required.
Identifier	Primary ID of your RosettaNet trading partner. Use DUNS number. Required.
Address Line 1	Conventional mailing address of your RosettaNet trading partner. Required.
Address Line 2	Additional mailing address information of your RosettaNet trading partner. Optional.
City	City where your RosettaNet trading partner is located. Required.
State or Province	State or province where your RosettaNet trading partner is located. Required.
Zip or Postal Code	ZIP Code or postal code where your RosettaNet trading partner is located. Required.
E-mail Address	E-mail address of your RosettaNet trading partner. Required.
Telephone Number	Telephone number of your RosettaNet trading partner. Required.
Time Zone	Time zone where your RosettaNet trading partner is located. Optional.
Country	Country where your RosettaNet trading partner is located. Optional.
Federation Peers	Discovery point URL for the trading partner you want to associate this identity with. Optional.
VAT ID	The value-added tax identifier.
Internet Notification E-mail	The Internet E-mail address used for notification.
Lookup ID	The identifier used lookup the Internet Notification E-mail.
Lookup ID Comments	Comments regarding the specified lookup identifier.

5. Click **Finish** to add the identity of this trading partner.

You are now ready to create a transport record for this RosettaNet trading partner.

Creating a Transport Record

To define how to send a RosettaNet message using HTTP, you must create a transport record.

To create a transport record:

1. From the **Trading Partner** menu, select **Setup > Advanced > Transports**.
2. Next to Create New Transport, click **Go!**
3. In the Information page, complete the following fields and click **Next**:

Field	Description
Name	Name for the transport record to create for this RosettaNet trading partner. Required. Note: You should use part of a company name or DUNS in this field and an indicator that references RosettaNet.
Identity	Unique ID used when you created an identity record for this RosettaNet trading partner. Required.

4. In the Communications page, complete the following fields and click **Next**:

Field	Description
Receiving Protocol	Communications protocol supported by this transport record when receiving data. Select HTTP. Required.
Sending Protocol	Communications protocol supported by this transport when sending data. Select HTTP. Required.
End Point IP	Leave this field blank.
End Point Port	Leave this field blank.

5. In the End Point Settings page, complete the following fields and click **Next**:

Field	Description
End Point	Destination URL for this trading partner. Required. Note: If using SSL, verify use of the following URL format: http://host:port/continued URL .
End Point Type	Purpose of this end point. Select All Purpose . Required.

6. In the Protocol Settings page, complete the following fields and click **Next**:

Field	Description
Response Timeout (seconds)	Number of seconds to wait for remote response to an outbound queue. Type the value 600 in this field. Required for RosettaNet.
Firewall Proxy	IP address, port number, login ID, and password of the remote firewall represented in URL format where commas separate values (for example, <host>, <port>). If used, you may need to specify both a login ID and a password. Used for outbound sessions. Optional.
Firewall Connect Count (seconds)	Number of seconds that the application attempts to contact port address of the transport before timing out. Optional.

Field	Description
Socket Timeout (seconds)	Number of seconds to wait for remote response to a command. Type the value 600 in this field. Required.

7. In the Security page, complete the following fields and click **Next**:

Field	Description
SMIME Encryption User Certificate	Certificate used to encrypt RosettaNet messages that are sent outbound through this transport record. This certificate has no impact in RosettaNet messages being received through this transport. Therefore, if this transport is for the company running the application, leave this field blank . If this transport is for your RosettaNet trading partner, choose the Trusted encryption certificate from the trading partner that you previously checked in. Click on the Configure Certificates link to select the appropriate certificates. Required for RosettaNet.
SMIME Decryption Certificate (System Store)	Certificate used to decrypt RosettaNet messages that are sent outbound through this transport record. This certificate has no impact in RosettaNet messages being received through this transport. Therefore, if this transport is for the company running the application, leave this field blank. If this transport is for your RosettaNet trading partner, choose the encryption certificate from the trading partner that you previously checked in. Click on the Configure Certificates link to select the appropriate certificates. Valid values are blank (recommended), OpsKey, OpsDrv, UIKey, and B2BHTTP. Required for RosettaNet.
SSL	Whether Secure Sockets Layer (SSL) must be active. Select Must to enable this transport to use this protocol with a key certificate. This field is used in conjunction with the key certificate field. If not using a key certificate, select None . Required for RosettaNet.
Key Certificate Passphrase	Password that protects the Web server key certificate. Required if SSL option is set to Must.
Cipher Strength	Strength of the algorithms used to encrypt messages. Select Strong (default) to indicate strength of algorithms with a key certificate. This field is used in conjunction with the key certificate field. If not using a key certificate, select All or Weak . Required.
Key Certificate (System Store)	Certificate used to decrypt an inbound encrypted RNIF v2.0 message sent to the identity associated with this transport (RNIF 1.1 does not support encryption). The trading partner encrypts the message with the public key and the receiver decrypts the message with this private key certificate. Click on the Configure Certificates link to select the appropriate certificates. If using a key certificate, select the appropriate key certificate from the list (valid values are blank, OpsKey, OpsDrv, UIKey, and B2BHTTP). Otherwise, leave this field blank.
CA Certificates	Certificate used to verify the Web server certificate of a trading partner. These certificates must come from your RosettaNet trading partners and you must have previously checked them in. Click on the Configure Certificates link to select the appropriate certificates. Required if the SSL option is Must or Optional.

8. In the Protocol Mode page, complete the following fields and click **Next**:

Field	Description
Protocol Mode	Use the default value PUT . Required.
Document	Leave this field blank.

9. In the Transport Account page, complete the following fields and click **Next**:

Field	Description
User ID	Unique ID that associates this transport with the remote system of your RosettaNet trading partner. Optional.
Password	Password that authenticates the user ID to the remote system of your RosettaNet trading partner. Optional.
Confirm Password	Password typed in the Password field. Optional.

10. Click **Finish** to update the transport record.

You are now ready to create a document exchange record.

Creating a Document Exchange Record

To describe properties of the documents and messages passed between you and your RosettaNet trading partner, you must create a document exchange record.

To create a document exchange record:

1. From the **Trading Partner** menu, select **Setup > Advanced > Document Exchange**.
2. Next to Create New Document Exchange, click **Go!**
3. In the New Document Exchange: Information page, complete the following fields and click **Next**:

Field	Description
Name	Name for the document exchange to create for this RosettaNet trading partner. Required. Note: You should use part of the company name or DUNS number in this field.
Identity	Unique ID used when you created an identity record for this RosettaNet trading partner. Required.

4. In the Retry Settings page, complete the following fields and click **Next**:

Field	Description
Number of Retries	Number of times that the application tries to contact this RosettaNet trading partner before timing out. Type the value 3 . Required.

Field	Description
Retry Interval (sec)	Number of seconds that the application waits before trying again to contact this RosettaNet trading partner after the previous attempt timed out. Type the value 3000 . Required.
Persist Duration (sec)	Number of seconds that a message received is kept before being passed to the integration engine. Type the value 10000 . Required.

5. In the Enveloping Properties page, complete the following field and click **Next**:

Field	Description
Protocol	The security protocol to be used. If using digital signatures to exchange RosettaNet messages, select SMIME from this field (if not exchanging RosettaNet messages, you would select None). Required for RosettaNet.

6. Skip this step if you are not using digital signatures to exchange RosettaNet messages. Otherwise, in the Security page, complete the following fields and click **Next**:

Field	Description
Encryption Algorithm	<p>Algorithm used to encrypt outbound RNIF v02.00 messages. Not used for inbound and RNIF v1.1. Required if this document exchange is for the trading partner running the application and sending encrypted RNIF v02.00 messages to a trading partner. Optional.</p> <p>Supported algorithms are:</p> <ul style="list-style-type: none"> ◆ Triple DES 168 CBC with PKCS5 padding ◆ 56-bit DES CBC with PKCS5 padding ◆ 128-bit RC2 CBC with PKCS5 padding ◆ 40-bit RC2 CBC with PKCS5 padding
Signing Algorithm	<p>Algorithm used to sign outgoing RosettaNet messages. Required if this document exchange is for the trading partner running the application . Otherwise, use the default value. Optional.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ MD5 – Message digest algorithm that verifies the integrity of data through a 128-bit message digest. ◆ SHA1 – Secure hash algorithm that verifies the integrity of data through a 160-bit (20 byte) message digest. Transmission is a bit slower than MD5; however, the larger message digest size offers a more secure data transference.
Signing Key Certificate	Key certificate used to sign outgoing RosettaNet messages. This certificate is a system certificate created for the trading partner running the application and it contains a private key. Select the appropriate key certificate from this field. If this document exchange is used for a trading partner other than the one running the application , use the default value. Required.

Field	Description
Signing User Certificate	Trusted certificate that contains the public key used to verify incoming digital signatures from a trading partner. If using this document exchange for a trading partner other than the one running the application, verify that None Available is selected from this field. Otherwise, select the appropriate trusted certificate from this field. Required.

7. Click **Finish** to add the document exchange record.

You are now ready to create a delivery channel record.

Creating a Delivery Channel Record

You can define several delivery channels in one trading profile.

To create a delivery channel record:

1. From the **Trading Partner** menu, select **Setup > Advanced > Delivery Channels**.
2. Next to Create New Delivery Channel, click **Go!**
3. In the Information page, complete the following fields and click **Next**:

Field	Description
Name	Name for the delivery channel record to create for this RosettaNet trading partner. Required. Note: You should use part of a company name or DUNS in this field or use an indicator that references RosettaNet.
Identity	Unique ID used when you created an identity record for this RosettaNet trading partner. Required.

4. In the Channel Definition page, complete the following fields and click **Next**:

Field	Description
Transport	Name of the transport record you created for this RosettaNet trading partner. This name must match the Name field of the transport record. Required.
Document Exchange	Name of the document exchange you created for this RosettaNet trading partner. This name must match the Name field of the document exchange record. Required.

5. The fields in the Security page are not used to implement RosettaNet. Therefore, click **Next** to skip this page.
6. In the Delivery Settings page, complete the following fields and click **Next**:

Field	Description
Receipt Type	Not used to implement RosettaNet. Use the default value NONE . Required.

Field	Description
Receipt Signature Type	Not used to implement RosettaNet. Use the default value NONE . Required.
Receipt Timeout	Time-out value for receipt of expected MDNs, in seconds. Type the value 86400 in this field. Required. Default value is 3600 seconds.
Delivery Mode	Select Asynchronous HTTP . Required.
Receipt to Address	Not used to implement RosettaNet. Leave this field blank.

7. Click **Finish** to add the document channel record.

You are now ready to create a packaging record.

Creating a Packaging Record

To define how to build and package outbound messages to your RosettaNet trading partner, you must create a packaging record.

To create a packaging record:

1. From the **Trading Partner** menu, select **Setup > Advanced > Packaging**.
2. Next to Create New Packaging, click **Go!**
3. In the Information page, complete the following fields and click **Next**:

Field	Description
Name	Name for the packaging record to create for this RosettaNet trading partner. Required. Note: You should use part of a company name or DUNS in this field or use an indicator that references RosettaNet.
Message Parsable	Whether the message can be parsed. Ensure that No is selected from this field. Required for RosettaNet.

4. In the Packaging Settings page, complete the following fields and click **Next**:

Field	Description
Payload Type	<p>Payload type that describes the message format for transporting documents. Required for RosettaNet.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ Plain Text – Payload is not signed and it is not encrypted. ◆ Signed Detached – Payload is signed with a detached signature for RNIF v1.1 and v02.00 messages. ◆ Encrypted – Payload is encrypted according to the EDIINT specifications. ◆ Signed Detached Encrypted – Payload is signed with a detached signature and then encrypted for RNIF v02.00 messages. <p>Default value is Signed Detached Encrypted.</p>
MIME Type	Select Application from this field. Required for RosettaNet.
MIME Sub Type	Select XML from this field. Required for RosettaNet. Default value is EDI X12.
Compress Data	<p>Whether to compress inbound and outbound messages, according to the ability of each trading partner to decompress and extract messages. Required for RosettaNet.</p> <p>Valid values (specific to RosettaNet v02.00 and higher) are:</p> <ul style="list-style-type: none"> ◆ None – Do not compress messages ◆ Low – Low message compression ◆ Medium – Medium message compression ◆ High – Compress all messages ◆ Default – default value <p>Note: When specifying a value in this field, considered that the value applies to all inbound and outbound messages, regardless of the message sizes. If you choose to specify the value None and you encounter a large inbound or outbound message that you want to compress, you can return to this packaging record and edit the record, specifying High in this field.</p>

5. Click **Finish** to add the packaging record.

You are now ready to create a profile record.

Creating a Profile Record

To define the roles (that is, initiator or responder of data exchange) of each trading partner exchanging RosettaNet data and create business processes that comply with RosettaNet standards, you must create a profile.

To create a profile record:

1. From the **Trading Partner** menu, select **Setup > Advanced > Profiles**.
2. Next to Create New Profile, click **Go!**

3. On the Information page, complete the following fields and click **Next**:

Field	Description
Name	Name for the profile record to create for this RosettaNet trading partner. Required. Note: You should use part of a company name or DUNS in this field or use an indicator that references RosettaNet.
Identity	Unique ID used when you created an identity record for this RosettaNet trading partner. Required.

4. On the Profile Definition page, complete the following fields and click **Next**:

Field	Description
Profile Type	Type of profile that must be used for data exchanged for this RosettaNet trading profile. Ensure that RN is selected. Required for RosettaNet. Default value is RN.
GLN	Not used to implement RosettaNet. Leave this field blank.
Delivery Channel	Name of the delivery channel you created for this RosettaNet trading partner. This name must match the Name field of the delivery channel record. Required.
Packaging	Name of the packaging you created for this RosettaNet trading partner. This name must match the Name field of the packaging record. Required.
Provider	Third-party service provider name. Type rosettanet (using lowercase letters). Required.

5. On the RN Data page, complete the following fields and click **Next**:

Field	Description
Role	Role of this RosettaNet trading partner. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Responder – Receiving RosettaNet data ◆ Initiator – Sending RosettaNet data
RNIF Version	Version of RosettaNet Implementation Framework (RNIF) being used by this RosettaNet trading partner. Select all from this field. Required. Default value is all .
Location Code	Location code for this RosettaNet trading partner. If this profile is associated with the partner receiving a RosettaNet message being built and delivered by the application, the location code is used to fill in the locationID element under the messageReceiverIdentification element (inside the outbound Delivery Header). Depending on which partner initiated the PIP and the specific RosettaNet message that is built by the application, the location code is also used to fill in the KnownInitiatingPartner/PartnerIdentification/locationID element inside the outbound Service Header, to indicate which partner initiated the PIP. Required for RNIF v2.00. Not used in RNIF v1.1.

Field	Description
Partner Location Code	The partner location code for this RosettaNet trading partner. If this profile is associated with the partner receiving a RosettaNet message being built and delivered by the application, the partner location code is to fill in the locationID element under the messageSenderIdentification element (inside the outbound Delivery Header). Depending on which partner initiated the PIP and the specific RosettaNet message that is built by the application, the partner location code is also used to fill in the KnownInitiatingPartner/PartnerIdentification/locationID element inside the outbound Service Header to indicate which partner initiated the PIP. Required for RNIF v2.00. Not used in RNIF v1.1.
Classification Code	RosettaNet classification code. Required.
Supply Chain Code	RosettaNet supply chain code. Required.
Contact	RosettaNet trading partner contact information. Required.

- The fields in the Business Processes page are not used to implement RosettaNet. Therefore, click **Next** to skip this page.
- Click **Finish** to add the profile record.

You have now finished creating trading profiles and can create the appropriate contracts.

Editing a Profile Record

To edit a profile record:

- From the Administration menu, select **Trading Partner > Setup > Advanced > Profiles**.
- In the Profiles page, under the **List** option, next to the drop down lists for the parameter **Alphabetically** and for the parameter **and by identity**, click **Go!** By default in the drop down list **ALL** is auto-populated. In the Profiles page, a list of profiles is displayed.
- In the list, locate the profile you want to edit and click **edit**.
- Update your profile information as necessary and click **Next**.
- Click **Finish** to update the organization.

Creating RosettaNet Contracts

Contracts enable you to tie RosettaNet trading partners together to create an effective trading relationship. To enable the application to comply with RosettaNet PIP specifications when running business processes, and to enable RosettaNet-specific services to build and parse message, promote security, and send and receive outbound RosettaNet messages to the appropriate URLs (for example, RosettaNet Message Builder and RosettaNet Message Parser services), you must create a contract that distinguishes *each* PIP interaction involved in a business process. You can create the contracts in any order.

The Contract Wizard enables you specify information about your enterprise and your partners' enterprises, to ensure that the system can accurately process PIP messages, signals, and OA1 messages.

To create a contract:

1. From the Administration menu, select **Trading Partner > Contracts**.
2. Under Create, next to New Contract, click **Go!**
3. On the Information page, complete the following fields and click **Next**:

Field	Description
Name	<p>Descriptive name for the contract. Required.</p> <p>You should use part of a company name, an indicator that references whether this trading partner is the initiator or responder, or the type of PIP being used. For example:</p> <ul style="list-style-type: none"> ◆ If this contract is for your organization and considered the initiator sending data, type FromCompanyToTradingPartner_PIP3A4. ◆ If this contract is for your RosettaNet trading partner and considered responding to data sent, type ToTradingPartnerFromCompany_PIP3A4.
Start Date	<p>Start date and time of the contract. Valid value is any valid date in the format <i>yyyy-mm-dd</i>. Must be less than or equal to End Date. Default value is the current date. Required.</p>
End Date	<p>Expiration date and time of the contract. Valid value is any valid date in the format <i>yyyy-mm-dd</i>. Must be greater than or equal to Start Date. Default value is the current date. Required.</p>
Status	<p>State of the composition/negotiation process that creates the contract. Default value is SIGNED. Required.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ PROPOSED – Contract being negotiated by the parties. The first stage of negotiation. ◆ AGREED – Contract agreed upon by the parties. The next stage of negotiation. ◆ SIGNED – Contract signed by the parties. The final stage of negotiation. The contract is usable.

4. On the Contract Definition page, complete the following fields and click **Next**:

Field	Description
Production Profile	<p>Unique ID of the profile for the RosettaNet trading partner producing (or sending) the data. This name must match the Name field used when creating the identity record. Required.</p>
Consumption Profile	<p>Unique ID of the profile for the RosettaNet trading partner consuming (or receiving) the data. This name must match the Name field used when creating the identity record. Required.</p>
Business Process	<p>Not used to implement RosettaNet. Not Applicable should be selected.</p>

5. On the Extensions page, select **RN** and click **Next**.

6. You must specify RN extensions if you are creating two PIP contracts. For example, if the company is initiating and responding to PIP 3A4, you must create a contract to initiate PIP 3A4 and one to respond to PIP 3A4.

On the RN Extensions page, complete the following fields and click **Next**:

Field	Description
PIP Code	RosettaNet PIP settings used by the RosettaNet business processes. Select the appropriate code from the list. Required.
PIP Version	Version of PIP to trade. Valid values are listed, based on the PIP code selected. Required.
RNIF Version	RNIF version number to trade with your RosettaNet trading partner. Required. Valid values are: <ul style="list-style-type: none"> ◆ 1.1 ◆ V02.00
PIP Role	PIP role for company that uses the application and is trading this PIP. Valid values are listed, based on the PIP code selected. Required.
Global Usage Code	Use this contract in either test or production. Required.

7. On the RosettaNet Message Building Configuration page, complete the following fields and click **Next**:

Note: Message building options are only available if you select RNIF Version V02.00 on the RN Extensions page. These options are not available for RNIF version 1.1.

Field	Description
Content Transfer Encoding	Encoding method for the message. Required. Valid values are: <ul style="list-style-type: none"> ◆ None ◆ Base64 ◆ Quoted - Printable
Encode binary message parts only	Select to encode binary message parts only. If not selected, all message parts are encoded. Optional.
Include MIME version header	Select to include the MIME version in the HTTP header. If not selected, no MIME version information is included. Optional.

8. On the PIP Definition page, complete the following fields and click **Next**:

Caution: You should select a Sender Business Process and Responder Business Process only on this page and accept all other default values, which are based on RosettaNet PIP standards. No additional changes need to be made to these fields unless you require customization.

Field	Description
Sender Business Process	PIP private initiating process for the sender business process. Select the appropriate business process from the list. Required.
Responder Business Process	PIP private responding process for the responder business process. Select the appropriate business process from the list. Required.
Response Mode	Mode in which to respond to a PIP. Required. Valid values are: <ul style="list-style-type: none"> ◆ Synch – Synchronously ◆ Asynch – Asynchronously
Non-Repudiation Required?	Receiving trading partner cannot deny having received a message. To disable this feature, clear the check box. Optional.
Time to Acknowledge	Amount of time, in minutes, within which you must receive an Acknowledgment. Optional.
Time to Acknowledge Acceptance	Amount of time, in minutes, within which you must accept an Acknowledgment. Optional.
Time to Perform	Time in which the PIP private responding process should respond to the business process. Optional.
Retry Count	The application continues to retry contacting the trading partner for the specified number of times. Optional.
Is Authorization Required?	Instructs the application to verify the presence of appropriate contracts for this transaction, to ensure that your trading partner is authorized to send the message to you. This feature is automatically enabled and cannot be disabled.
Non-Repudiation of Origin and Content	Originating trading partner can not deny having originated and sent a message. To disable this feature, clear the check box. Optional.

9. On the PIP Steps page, review the steps you have selected to run using the PIP and click **Next**.

10. Click **Finish** to add the contract. Repeat steps 1 - 10 to create a contract for the trading partner who is either initiating and responding to a PIP, if necessary.

Validating RosettaNet Service Content Against a DTD or Schema

The application now supports validating RosettaNet service content against either a DTD or an XML schema, since many new PIPs are XML schema-based.

Note: You must check in the schemas that you will use for validation into the application schema repository. See *Checking In Schemas* for more details.

Searching for RosettaNet Correlations

To search for RosettaNet correlations:

1. From the **Administration** menu, select **Business Processes > Advanced Search > Correlation**.
2. Select the start and end dates and times within which the correlated data you are searching for can be located. Using the following formats, type the date and time range and select **A.M.** or **P.M.**:
 - ◆ Date – MM/DD/YYYY
 - ◆ Time – HR:MN:SC

These fields default to a range of the last 24 hours.
3. In the **Type** field, select **ANY**, **Document**, or **Business process**.
ANY is the default value. Selecting Document or Business process further targets your returned results.
4. In the **Location** field, select one of the following options:
 - ◆ **Live Tables** – Display correlations for live (active) instances.
 - ◆ **Restored Tables** – Display correlations for instances restored from an offline location.
5. To refine your search to correlations you specify (rather than all generated correlations), select **Standard** from the Name list.
6. In the Value fields, type **RN** as the value that corresponds with the Standard name, and click **Go!**

Note: Alternatively, you can search by standard RosettaNet correlation keys such as the following:

- ◆ IdentityID
- ◆ PartnerID
- ◆ PIPName
- ◆ PIPCode
- ◆ PIPVersion
- ◆ RNIFVersion
- ◆ PIPInstanceID
- ◆ IsInitiator
- ◆ PIPStepNumber
- ◆ PIPStepName
- ◆ PIPStepDateTime
- ◆ PIPStepDocumentID

- ◆ PIPStatus

7. To access your search results, select the numbered link that indicates the number of document and/or business process matches that were found. The document page opens to display the documents that match the search criteria you selected. However, archived and purged documents and business process IDs are not displayed.

RosettaNet Database Tables

This section provides information about database tables that enable the application to support RosettaNet standards.

PIP_DEF Table

This is the master table that defines properties of each PIP: for example, the PIP codes, all of the supported PIP and RNIF versions, initiator, and responder entities used in the PIP. It also provides other additional PIP-specific information such as Time to Acknowledge, Retry Count, and whether authorization and non-repudiation of message is required.

Column Name	Data Type	Column Size	Whether the default value is Null
OBJECT_ID	INTEGER	11	NOT NULL
PIP_CODE	VARCHAR	32	NOT NULL
TO_ROLE	VARCHAR	255	NOT NULL
FROM_ROLE	VARCHAR	255	NOT NULL
TO_SERVICE	VARCHAR	255	NOT NULL
FROM_SERVICE	VARCHAR	255	NOT NULL
PIP_VERSION	VARCHAR	10	NOT NULL
RNIF_VERSION	VARCHAR	50	NULL
PIP_DESCRIPTION	VARCHAR	255	NULL
EDITED_BY	VARCHAR	32	NULL
STATUS	INTEGER	11	NULL
TYPE	INTEGER	11	NULL
LAST_MOD_DATE	DATETIME		NULL
RESPONSE_MODE	INTEGER	11	NULL
SENDER_WFD_NAME	VARCHAR	255	NULL
RESPONDER_WFD_NAME	VARCHAR	255	NULL
NON_REPUDIATION_REQ	INTEGER	11	NULL
TIME_TO_ACK	INTEGER	11	NULL
TIME_TO_ACK_ACCEPTANCE	INTEGER	11	NULL
TIME_TO_PERFORM	INTEGER	11	NULL
RETRY_COUNT	INTEGER	11	NULL

Column Name	Data Type	Column Size	Whether the default value is Null
AUTH_REQ	INTEGER	11	NULL
NON_REPUDIATION_OF_ORIGIN	INTEGER	11	NULL

PIP_INSTANCE_DEF Table

This table stores the same information as the PIP_DEF table, but is used for any customizations of the PIP. When you create a Rosettanet contract, the related PIP definition is stored in this table. Trading partners can customize their PIP definition based on their contract/agreement and the modified PIP definition is stored in this table. The original PIP_DEF table is not modified.

Column Name	Data Type	Column Size	Whether the default value is Null
OBJECT_ID	VARCHAR	255	NOT NULL
PIP_CODE	VARCHAR	32	NOT NULL
TO_ROLE	VARCHAR	255	NOT NULL
FROM_ROLE	VARCHAR	255	NOT NULL
TO_SERVICE	VARCHAR	255	NOT NULL
FROM_SERVICE	VARCHAR	255	NOT NULL
PIP_VERSION	VARCHAR	10	NOT NULL
RNIF_VERSION	VARCHAR	50	NULL
PIP_DESCRIPTION	VARCHAR	255	NULL
EDITED_BY	VARCHAR	32	NULL
STATUS	INTEGER	11	NULL
TYPE	INTEGER	11	NULL
LAST_MOD_DATE	DATETIME		NULL
RESPONSE_MODE	INTEGER	11	NULL
SENDER_WFD_NAME	VARCHAR	255	NULL
RESPONDER_WFD_NAME	VARCHAR	255	NULL
NON_REPUDIATION_REQ	INTEGER	11	NULL
TIME_TO_ACK	INTEGER	11	NULL
TIME_TO_ACK_ACCEPTANCE	INTEGER	11	NULL
TIME_TO_PERFORM	INTEGER	11	NULL
RETRY_COUNT	INTEGER	11	NULL

Column Name	Data Type	Column Size	Whether the default value is Null
AUTH_REQ	INTEGER	11	NULL
NON_REPUDIATION_OF_ORIGIN	INTEGER	11	NULL

SCI_RN_PROFILE Table

This table provides additional information that is used to build RosettaNet message headers. This table is used when creating a RosettaNet profile during the trading partner configuration process.

Column Name	Data Type	Column Size	Whether the default value is Null
OBJECT_ID	VARCHAR	200	NOT NULL
OBJECT_VERSION	VARCHAR	200	NULL
OBJECT_NAME	VARCHAR	200	NOT NULL
RN_ROLE	VARCHAR	255	NULL
RNIF_VERSION	VARCHAR	10	NULL
CONTACT	VARCHAR	255	NULL
LOCATION_CODE	VARCHAR	255	NULL
SUPPLY_CHAIN_CODE	VARCHAR	255	NULL
CLASSIFICATION_CODE	VARCHAR	255	NOT NULL
PARTNER_LOCATION_CODE	VARCHAR	255	NULL
RN_ROLE_INH	VARCHAR	255	NOT NULL
RNIF_VERSION_INH	VARCHAR	255	NULL
CONTACT_INH	INTEGER	11	NOT NULL
LOCATION_CODE_INH	INTEGER	11	NOT NULL
SUPPLY_CHAIN_CODE_INH	INTEGER	11	NOT NULL
CLASSIFICATION_CODE_INH	INTEGER	11	NOT NULL
PARTNER_LOCATION_CODE_INH	INTEGER	11	NOT NULL
EXTENDS_OBJECT_ID	INTEGER	11	NOT NULL
EXTENDS_OBJECT_VERSION	INTEGER	11	NOT NULL
AUTH_REQ	VARCHAR	255	NULL
NON_REPUDIATION_OF_ORIGIN	INTEGER	11	NULL

PIP_STEP Table

This is a master table that provides all the steps that are applicable to a PIP, such as: PIP Code, PIP Version, and the Step ID, which provide a unique combination key for this table. For each PIP step, information is stored such as Time to Acknowledge, Time to Respond with action message, and whether authorization and non-repudiation of message is required.

Column Name	Data Type	Column Size	Whether the default value is Null
PIP_CODE	VARCHAR	32	NOT NULL
PIP_VERSION	VARCHAR	10	NOT NULL
STEP_ID	INTEGER	11	NOT NULL
PIP_BUSINESS_ACTION	VARCHAR	255	NULL
TIME_TO_ACKNOWLEDGE_RECEIPT	INTEGER	11	NULL
TIME_TO_ACKNOWLEDGE_ACCEPT	INTEGER	11	NULL
TIME_TO_RESPOND	INTEGER	11	NULL
INCLUDED_IN_TIME_TO_PERFORM	INTEGER	11	NULL
AUTH_REQ	INTEGER	11	NULL
NON_REPUDIATION_REQ	INTEGER	11	NULL
SECURE_TRANSPORT_REQ	INTEGER	11	NULL
DSIG_REQ	INTEGER	11	NULL
SSL_REQ	INTEGER	11	NULL

PIP_INSTANCE_STEP Table

This table stores the same information as the PIP_STEP table, but is used for any customizations of the PIP Steps. When you create a RosettaNet contract, the related PIP Step definition is stored in this table. Trading partners can customize their PIP Steps based on their contract/agreement and the modified PIP Step definition is stored in this table. The original PIP_STEP table is not modified.

Column Name	Data Type	Column Size	Whether the default value is Null
OBJECT_ID	VARCHAR	255	NOT NULL
PIP_CODE	VARCHAR	32	NOT NULL
PIP_VERSION	VARCHAR	10	NOT NULL
STEP_ID	INTEGER	11	NOT NULL
PIP_BUSINESS_ACTION	VARCHAR	255	NULL

Column Name	Data Type	Column Size	Whether the default value is Null
TIME_TO_ACKNOWLEDGE_RECEIPT	INTEGER	11	NULL
TIME_TO_ACKNOWLEDGE_ACCEPT	INTEGER	11	NULL
TIME_TO_RESPOND	INTEGER	11	NULL
INCLUDED_IN_TIME_TO_PERFORM	INTEGER	11	NULL
AUTH_REQ	INTEGER	11	NULL
NON_REPUDIATION_REQ	INTEGER	11	NULL
SECURE_TRANSPORT_REQ	INTEGER	11	NULL
LAST_MOD_DATE	DATETIME		NULL
EDITED_BY	VARCHAR	32	NULL
DSIG_REQ	INTEGER	11	NULL
SSL_REQ	INTEGER	11	NULL

PIP_TRACK table

This table stores the overall status of the PIP, both from initiator and responder side. For example, the status of the PIP could be Success/InProgress/Failed. The PIP ID is the primary key for this table and is used as a key to obtain correlation information about this PIP.

Column Name	Data Type	Column Size	Whether the default value is Null
PIP_ID	VARCHAR	255	NOT NULL
PIP_CODE	VARCHAR	32	NOT NULL
PIP_VERSION	VARCHAR	10	NOT NULL
RNIF_VERSION	VARCHAR	50	NOT NULL
ISINITIATOR	VARCHAR	10	NOT NULL
STATUS	VARCHAR	20	NOT NULL

PIP_STEP_TRACK

This table tracks the steps of a PIP as it is executed. The step number is stored against each PIP ID, both from the initiator and the responder side.

Column Name	Data Type	Column Size	Whether the default value is Null
PIP_ID	VARCHAR	255	NOT NULL
ISINITIATOR	VARCHAR	10	NOT NULL
STEP_NUM	INTEGER	11	NOT NULL

B

business process correlations 52

C

configuring
 RosettaNet trading profile 37
consumption profile
 description 49
contract
 RosettaNet 48
correlation
 standard search for document or business process
 correlations 52

D

database tables 54
delivery channel 44
document correlations
 basic search 52
document exchange record 42

I

identity record
 RosettaNet trading partner 37

P

packaging record 45
production profile
 description 49

R

RosettaNet 36

S

SMIME 41, 43

T

trading partner
 RosettaNet 37
trading partner configuration
 RosettaNet 37
trading profile
 before setting up 36
 creating 46
 RosettaNet 37
transport record
 creating 39
 RosettaNet trading partner 39

