## Contents

**About This Guide** ........................................................................................................ 7  
  Document Conventions ......................................................................................... 7  
  Additional Information ......................................................................................... 8  

**Chapter 1. Concepts** .............................................................................................. 9  
  What is Transora? .................................................................................................. 10  
  What is the webMethods Transora Module? ......................................................... 10  
  webMethods Transora Module Features ................................................................. 10  
  webMethods Transora Module Packages ............................................................... 12  
  Design-Time Architecture and Components .......................................................... 13  
  Run-Time Architecture and Components ............................................................... 16  
  The Transora Item Synchronization Process ............................................................ 20  
  Item Synchronization with the webMethods Transora Module ......................... 22  
  Processing Transora Item Request/Response Messages ...................................... 22  
  Receiving Item Confirmation Messages ................................................................. 23  

**Chapter 2. Installing webMethods Transora Module 6.1.1** .................................. 25  
  System Requirements ........................................................................................... 26  
    Platform and Operating System Requirements ............................................... 26  
    webMethods Software Requirements ................................................................ 26  
    webMethods Components .................................................................................. 26  
    Hardware Requirements ..................................................................................... 26  
  Install the webMethods Transora Module ............................................................ 27  
  Uninstall the webMethods Transora Module ....................................................... 28  

**Chapter 3. Configuring the webMethods Transora Module** .................................. 29  
  Overview ............................................................................................................... 30  
  Sign up with Transora for Data Synchronization Service .................................... 30  
  Define Trading Networks Profiles ........................................................................ 30  
  Define a Trading Partner Agreement .................................................................... 30  
  Define TN Document Types and/or IS Document Types ..................................... 31  
  Integrate Back-End Applications with Transora .................................................. 31  
  Set Up Business Document Processing ................................................................ 32  
    Using a Process Model ....................................................................................... 32  
    Using Trading Networks Processing Rules ....................................................... 32  
  Update the ACL Settings for the WmEDIINT Package ....................................... 33  
  Run the webMethods Transora Module Sample .................................................. 33
Contents

Items Configured For You When You Install the Transora Module 34

Chapter 4. Defining Trading Networks Partner Profiles 35
  What is a Trading Partner and a Trading Partner Profile? 36
  Information that You Supply in a Profile 36
  External ID Type Field in a Profile 36
  Defining the Profile for Your Corporation 37
  Defining a Profile for Transora 39

Chapter 5. Defining a Trading Partner Agreement 43
  What is a Trading Partner Agreement? 44
  Defining a Trading Partner Agreement 44
  Parameter Settings for the TPA Data 45

Chapter 6. Defining TN Document Types 49
  What are TN Document Types? 50
  TN Document Types for Documents from Your Back-End Systems 50
  TN Document Types for Transora Messages 51
  Defining TN Document Types for Back-End System Documents 51
  TN Document Types Provided with the Transora Module 53
    TDC Item Response TN Document Type 53
      Identification Information 53
      Extraction Information 54
      IS Document Type 54
    TDC Publication Response TN Document Type 54
      Identification Information 54
      Extraction Information 55
      IS Document Type 55
    TDC Accept Reject Response TN Document Type 55
      Identification Information 55
      Extraction Information 56
      IS Document Type 56
    TDC Item Request TN Document Type 56
      Identification Information 56
      Extraction Information 57
    TDC GDSN Item Registry Response TN Document Type 57
      Identification Information 57
      Extraction Information 58
      IS Document Type 58
    TDC GDSN Party Registry Response TN Document Type 58
      Identification Information 58
About This Guide

This guide describes how to implement and use the webMethods Transora Module. It is for users who want to use the webMethods Transora Module to synchronize item information between their back-end systems and their trading partners through TDSN Data Synchronization Service in compliance with Transora XML release 4.2.

To use this guide effectively, you should:

- Be familiar with the webMethods Integration Server, the Server Administrator, and webMethods Developer, and understand the concepts and procedures described in the *webMethods Integration Server Administrator’s Guide* and *webMethods Developer User’s Guide*.

- Be familiar with the webMethods Trading Networks Console and understand the concepts and procedures described in the various webMethods Trading Networks guides.

- Be familiar with webMethods EDI Module and understand the concepts and procedures for EDIINT that are described in the webMethods EDI Module documentation.

- Be familiar with webMethods Modeler and understand the concepts and procedures described in the *webMethods Modeler User’s Guide*.

- Have a basic knowledge of Transora and Transora terminology. For more information, see [http://www.transora.com/](http://www.transora.com/).

- Be familiar with the EDIINT AS2 transport protocols.

- Have installed the Integration Server, webMethods Developer, Trading Networks (server side and console side) software, webMethods Modeler (server side and client side) software, and the webMethods Transora Module software.

Document Conventions

<table>
<thead>
<tr>
<th>Convention</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bold</strong></td>
<td>Identifies elements on a screen.</td>
</tr>
<tr>
<td><em>Italic</em></td>
<td>Identifies variable information that you must supply or change based on your specific situation or environment. Identifies terms the first time they are defined in text. Also identifies service input and output variables.</td>
</tr>
<tr>
<td>Narrow font</td>
<td>Identifies storage locations for services on the webMethods Integration Server using the convention <code>folder.subfolder.service</code>.</td>
</tr>
</tbody>
</table>
Additional Information

The webMethods Advantage Web site at http://advantage.webmethods.com provides you with important sources of information about webMethods components:

- **Troubleshooting Information.** webMethods provides troubleshooting information for many webMethods components in the webMethods Knowledge Base.

- **Documentation Feedback.** To provide documentation feedback to webMethods, go to the Documentation Feedback Form on the webMethods Bookshelf.

- **Additional Documentation.** All webMethods documentation is available on the webMethods Bookshelf.
Concepts

- What is Transora? ................................................................. 10
- What is the webMethods Transora Module? .......................... 10
- webMethods Transora Module Features ................................. 10
- webMethods Transora Module Packages ............................... 12
- Design-Time Architecture and Components .......................... 13
- Run-Time Architecture and Components .............................. 16
- The Transora Item Synchronization Process ........................... 20
- Item Synchronization with the webMethods Transora Module ...... 22
What is Transora?

Transora is a global trading exchange that provides item synchronization and registration services to trading partners within the Consumer Goods and Retail industry. Trading partners can be manufacturers, suppliers, distributors, or retailers.

Transora uses a centralized database and a hub-and-spoke model to allow trading partners to create business documents and publish those documents to other trading partners via Transora.

What is the webMethods Transora Module?

The webMethods Transora Module allows suppliers to synchronize item information between their back-end systems and their trading partners through the Transora Data Synchronization Network (TDSN) Data Synchronization Service.

The Transora Module is built on top of the Integration Server and other webMethods components. The Transora Module specifically leverages the following webMethods components:

- webMethods Trading Networks for trading partner management
- webMethods EDIINT Module for EDIINT AS2 support
- webMethods Modeler for business process integration

The webMethods Transora Module version 6.1.1 supports Transora release 4.2. Transora release 4.2 is backward compatible with Transora releases 4.0.x and 4.1.x.

webMethods Transora Module Features

The Transora Module provides the following features:

- **Out-of-the-box support for Transora and Transora Messages.** You can quickly automate Publisher-side functionality with Transora. Typically the Publisher is the seller, such as a manufacturer, wholesaler, or distributor, or their agent. Detailed samples for Item Add, Item Modify, Item Link, and Item Publication are provided to help you implement item synchronization.

  Transora XML does not support retailers (subscribers) to Transora interfaces.

- **Management and maintenance of Transora business processes through a graphical interface.** You can use webMethods Modeler to create process models that define the business processes for communicating with Transora and integrating with your back-end systems. The Transora Module provides a sample process model that you can customize.
Support for specific Transora outbound message actions. The Transora Module supports the following outbound message actions that you can send using the Transora Module:

- Item Add
- Item Link
- Item Modify
- Item Publication
- Item Append

Note: Transora XML release 4.2 does not support Item Unlink and Item Delete messages. Item Unlink must be performed manually through the Transora UI service and Item Delete must be done through Transora Customer Support.

Support for specific Transora inbound message actions. The Transora Module supports the following inbound Transora message actions that you can receive using the Transora Module:

- Item Add Response
- Item Modify Response
- Item Correction Response
- Item Append Response
- Item Link Response
- Item Publication Response
- Item Accept/Reject Response
- GDSN Item Registry Response
- GDSN Party Registry Response

Note: An Item Modify message results in an Item Modify Response or an Item Correction response, depending on the attribute modified (for example, changes to Brand name, or changes to Height that is greater than 20% of the previous value. For more information on modification of item attributes that can result in item correction, contact Transora Customer Support.

Support for EDIINT. The Transora Module leverages the webMethods EDI Module to allow communication with Transora over HTTP or HTTPS through open standards (EDIINT AS2) for more secure communication.

Persistent message storage. All inbound and outbound messages processed with the Transora Module are maintained in persistent storage using webMethods Trading Networks.
webMethods Transora Module Packages

The webMethods Transora Module contains several packages (sets of webMethods services and related files) that the webMethods Installer installs on the Integration Server. The following table describes the contents of each package.

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WmIPRoot</td>
<td>Contains the shared framework components, including implementations of common utilities and abstractions and base classes for frameworks. For example, this package includes shared logging facilities as an abstract transport framework. The other packages contain concrete- and application-specific implementations. These services are for internal use only.</td>
</tr>
<tr>
<td>WmTransora</td>
<td>Contains the core components for:</td>
</tr>
<tr>
<td></td>
<td>- Enveloping and packaging Transora messages</td>
</tr>
<tr>
<td></td>
<td>- Records based on the Transora 4.2 DTDs for validation</td>
</tr>
<tr>
<td></td>
<td>- DSP pages for the Transora Module user interface</td>
</tr>
<tr>
<td></td>
<td>- Initialization services for the Transora Module</td>
</tr>
<tr>
<td></td>
<td>For detailed information about the contents of this package, see Chapter 8, “webMethods Transora Module Services”.</td>
</tr>
<tr>
<td>WmTransoraDocuments</td>
<td>Contains the records and schemas for each of the BizDocTypes available for Transora. For detailed information about the contents of this package, see Chapter 8, “webMethods Transora Module Services”.</td>
</tr>
</tbody>
</table>
Design-Time Architecture and Components

The following figure illustrates the webMethods Transora Module design-time architecture, components, and the relationships between the design-time components. For further explanation, see the table that follows the figure.

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WmTransoraSamples</td>
<td>Provides a sample that demonstrates sending messages to and receiving responses from Transora, specifically from a publishers perspective. This package includes one process model to:</td>
</tr>
<tr>
<td></td>
<td>Add items using the Catalog Request Message</td>
</tr>
<tr>
<td></td>
<td>Modify items using the Catalog Request Message</td>
</tr>
<tr>
<td></td>
<td>Publish items using the Catalog Request Message</td>
</tr>
<tr>
<td></td>
<td>Link items using Catalog Request Message</td>
</tr>
<tr>
<td></td>
<td>All of the samples process the response message received from Transora.</td>
</tr>
<tr>
<td></td>
<td>For detailed information about the contents of this package, see, Appendix A, “webMethods Transora Module Sample”.</td>
</tr>
</tbody>
</table>

Design-Time Architecture and Components

The following figure illustrates the webMethods Transora Module design-time architecture, components, and the relationships between the design-time components. For further explanation, see the table that follows the figure.
## Chapter 1 Concepts

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webMethods Integration Server</td>
<td>The webMethods Integration Server contains the documents, services, and IS document types that you will want to access when creating your process models. For more information about the Integration Server, see the webMethods Integration Server Administrator’s Guide.</td>
</tr>
<tr>
<td>webMethods Trading Networks</td>
<td>Trading Networks enables your enterprise to link to the trading partners with whom you want to exchange Transora messages. You use the Trading Networks Console during design time to:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Define trading partner profiles</strong> to provide information about your trading partners so that the Transora Module can exchange Transora messages with them.</td>
</tr>
<tr>
<td></td>
<td>- <strong>Define trading partner agreements (TPAs)</strong> to provide additional information that governs how the Transora Module exchanges Transora messages.</td>
</tr>
<tr>
<td></td>
<td>- <strong>View the TN XML document types</strong> (transaction, response, and item confirmation) created in Trading Networks by the Transora Module during installation.</td>
</tr>
<tr>
<td>Trading Networks Database</td>
<td>Trading Networks saves trading partner profiles, TPAs, and TN document types, among other items, in its database and retrieves this information when needed.</td>
</tr>
</tbody>
</table>

For more information about Trading Networks, trading partner profiles, TPAs, and TN XML document types, see the *webMethods Trading Networks User’s Guide* guide. You also can find information about trading partner profiles in Chapter 4, “Defining Trading Networks Partner Profiles” in this guide and information about TPAs in Chapter 5, “Defining a Trading Partner Agreement” in this guide.
webMethods Modeler is a design-time tool (Java GUI) that you use to create and manage process models that define business processes. webMethods provides process models for Transora communications using the Transora Module.

During design time, you use webMethods Modeler to customize the process models that webMethods provides for Transora communications, thereby creating your own process models. Among other actions, to customize a process model template, you specify how the process model is to interact with your back-end systems and edit the services that the steps of the process model invoke. When you generate a process model, you generate the runtime elements (services, triggers, and process run-time scripts) that execute at run time. For more information about webMethods Modeler, see the webMethods Modeler User’s Guide. For information about customizing a process model template, see Chapter 7, “Customizing a Process Model Template” in this guide.

The webMethods Modeler Repository is a storage area that webMethods Modeler uses to save process model information. For more information about webMethods Modeler and the webMethods Modeler Repository, see the webMethods Modeler User’s Guide.

The Transora Module contains a process model sample for business processes that you might implement for Transora communications; that is, an item request/response process model. You can customize the process model sample to define your Transora transactions.

For detailed information about the samples, see Chapter 8, “webMethods Transora Module Services”.

### Component Description

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webMethods Modeler</td>
<td>webMethods Modeler is a design-time tool (Java GUI) that you use to create and manage process models that define business processes. webMethods provides process models for Transora communications using the Transora Module. During design time, you use webMethods Modeler to customize the process models that webMethods provides for Transora communications, thereby creating your own process models. Among other actions, to customize a process model template, you specify how the process model is to interact with your back-end systems and edit the services that the steps of the process model invoke. When you generate a process model, you generate the runtime elements (services, triggers, and process run-time scripts) that execute at run time. For more information about webMethods Modeler, see the webMethods Modeler User’s Guide. For information about customizing a process model template, see Chapter 7, “Customizing a Process Model Template” in this guide.</td>
</tr>
<tr>
<td>webMethods Modeler Repository</td>
<td>The webMethods Modeler Repository is a storage area that webMethods Modeler uses to save process model information. For more information about webMethods Modeler and the webMethods Modeler Repository, see the webMethods Modeler User’s Guide.</td>
</tr>
<tr>
<td>webMethods Transora Module</td>
<td>The Transora Module contains a process model sample for business processes that you might implement for Transora communications; that is, an item request/response process model. You can customize the process model sample to define your Transora transactions. For detailed information about the samples, see Chapter 8, “webMethods Transora Module Services”.</td>
</tr>
</tbody>
</table>
Run-Time Architecture and Components

The following figure illustrates the webMethods Transora Module run-time architecture, components, and the relationships between the run-time components. For further explanation, see the table that follows the figure.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webMethods Integration Server</td>
<td>The Integration Server contains the run-time elements (services, triggers, and process run-time scripts) that execute when exchanging communications with Transora. You use webMethods Modeler to generate these run-time elements during design time. When you generate the process model, it creates run-time elements on the Integration Server for the automated controlled steps within a process model. For more information about generating a process model, see webMethods Modeler User's Guide. For more information about the Integration Server, see the webMethods Integration Server Administrator's Guide.</td>
</tr>
</tbody>
</table>
### Component Description

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>webMethods</td>
<td>Trading Networks enables your enterprise to link with whom you want to exchange Transora messages, thereby forming a business-to-business trading network.</td>
</tr>
<tr>
<td>Trading Networks</td>
<td>During run time, Trading Networks uses its services and TN document types to recognize the Transora messages that it receives. For a Transora message, Trading Networks uses one of the TN XML document types that the Transora Module provides. After receiving and recognizing a Transora message, Trading Networks envelopes the message to create a Trading Networks BizDocEnvelope, which represents a routeable transaction within Trading Networks. Trading Networks saves this BizDocEnvelope to its database. The Transora Module, using the trading partner IDs, obtains the TPA. Based on the TPA, the Transora Module determines which Transora DTDs to use, whether to validate the outgoing message, and whether to sign and encrypt the outgoing message. After creating the message, the Transora Module constructs an EDIINT envelope (based on the sign and encrypt flag in the TPA and corresponding certificates stored in Trading Networks by the webMethods EDIINT Module) for the message. All outgoing and incoming messages created by the Transora Module are saved in the Trading Networks database. For more information about Trading Networks, trading partner profiles, TN document types, and TPAs, see the webMethods Trading Networks User's Guide guide. You also can find information about trading partner profiles in Chapter 4, “Defining Trading Networks Partner Profiles” in this guide and information about TPAs in Chapter 5, “Defining a Trading Partner Agreement” in this guide.</td>
</tr>
</tbody>
</table>

---

For more information about Trading Networks, trading partner profiles, TN document types, and TPAs, see the <em>webMethods Trading Networks User's Guide</em> guide. You also can find information about trading partner profiles in Chapter 4, “Defining Trading Networks Partner Profiles” in this guide and information about TPAs in Chapter 5, “Defining a Trading Partner Agreement” in this guide.
The Transora Module leverages various features of the webMethods EDI Module:

- Provides a mechanism to securely exchange business documents using EDIINT AS2.
- Enables business partners to set encryption types and key lengths at the business-partner level using the extended fields of the trading partner’s Trading Networks profile. The choices include Triple DES, DES, RC2 (40 bits), RC2 (64 bits), and RC2 (128 bits).
- Provides the standard outbound encryption permutations (signed, encrypted, signed and encrypted, or plain) at the send-service level.
- Can send receipts of received business documents back to the sender as well as receive business document receipts. EDIINT receipts are known as MDNs (message disposition notifications). webMethods EDI Module can send and receive synchronous or asynchronous, signed or unsigned MDNs.

For more information about webMethods EDI Module, see the chapters about the WmEDIINT package in the webMethods EDI Module documentation. For information about setting up EDIINT interaction with Transora, see “Update the ACL Settings for the WmEDIINT Package” on page 33 in this guide.

**Component** | **Description**
--- | ---
webMethods EDI Module | Provides a mechanism to securely exchange business documents using EDIINT AS2.
| Enables business partners to set encryption types and key lengths at the business-partner level using the extended fields of the trading partner’s Trading Networks profile. The choices include Triple DES, DES, RC2 (40 bits), RC2 (64 bits), and RC2 (128 bits).
| Provides the standard outbound encryption permutations (signed, encrypted, signed and encrypted, or plain) at the send-service level.
| Can send receipts of received business documents back to the sender as well as receive business document receipts. EDIINT receipts are known as MDNs (message disposition notifications). webMethods EDI Module can send and receive synchronous or asynchronous, signed or unsigned MDNs.

During run time, the Transora Module receives a Transora message from a back-end system or trading partner. The Transora Module passes the Transora message to Trading Networks.

Trading Networks recognizes the Transora message, creates a BizDocEnvelope, extracts a unique ID named the ConversationID from the message, and saves the BizDocEnvelope to its database.

Because Trading Networks extracts a ConversationID, after performing Trading Networks processing, Trading Networks passes the Transora message to the process run time (PRT). For more information about the PRT, see the PRT row in this table on page 19 and the *webMethods Modeler User’s Guide*. 

---
Run-Time Architecture and Components

The PRT is part of the Integration Server. It is the run-time engine that manages running instances of business processes. That is, it manages the Transora business processes that you defined during design time when you customized the process models provided for Transora communications. The PRT:

- Accepts Transora messages from Trading Networks.
- Determines which process model to use for a given Transora message.
- Processes a Transora message based on the type of Transora message and who sent it.
- Records the status of the Transora message to the Process Audit Log Database.

The PRT uses the ConversationID that Trading Networks extracted from the message to determine whether the message belongs to a new or existing business process. The PRT looks for a matching ConversationID among the running business processes. If it does not find a matching ConversationID, the PRT determines the process model to use for the message and starts a new instance of a business process that uses the matching model. If the PRT finds a running business process with a matching ConversationID, the business document rejoins the business process.

The PRT ensures the integrity, traceability, observability, and controllability of Transora business processes by tracking each transaction and its state. For example, suppose that you are implementing a publish items transaction and you send a Publish message to Transora. The PRT initiates the transaction, tracks the state of the transaction, and joins the transaction when it receives the response document from Transora.

For more information about the PRT, see the webMethods Modeler User’s Guide.

webMethods Monitor

You use webMethods Monitor to manage and monitor business processes. webMethods Monitor displays information about a business process by retrieving information from the Process Audit Log Database.

You can manage a business process by performing such commands as suspend, resume, restart, and terminate. For more information about webMethods Monitor, see the webMethods Integration Platform Logging and Monitoring Guide.
The Transora Item Synchronization Process

The following figure illustrates the Transora item synchronization process for a supplier (Publisher) using Transora as a data pool. The diagram represents a subset of the entire GDSN (Global Data Synchronisation Network) message choreography.

Suppliers are typically the manufacturers who send item information to their trading partners. Because they are “publishing” their item information, they are also referred to as Publishers. The trading partners are typically the retailers who purchase and then sell items from their suppliers. Because they are receiving the published item information, they are also referred to as Subscribers.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process Audit Log Database</td>
<td>webMethods Monitor and the PRT log audit data about running business processes to the Process Audit Log Database.</td>
</tr>
<tr>
<td>Trading Networks Database</td>
<td>Trading Networks stores TN document types, TPAs, and trading partner profile information, among other things, in its database.</td>
</tr>
</tbody>
</table>

Note: All messages have an implied response message.
When suppliers use the webMethods Transora Module with the Transora XML release 4.2 messaging capabilities for item synchronization, the webMethods Transora Module performs steps 1, 2, 3, and 4, thus enabling them to automate their Transora transactions.

The diagram above depicts the following scenario between a Publisher and TDSN Item Management (IM).

**Note:** Each message sent to TDSN IM will receive a technical acknowledgement message (AS2 MDN).

<table>
<thead>
<tr>
<th>Step</th>
<th>Action</th>
</tr>
</thead>
</table>
| 1    | Item Add  
   a The Publisher adds Item Information (GTIN Variation) to TDSN IM. TDSN IM moves the GTIN variation to the Production database immediately after the GTIN variation is successfully received and validated by TDSN IM, and if applicable registered with the GS1 Global Registry.  
   b TDSN IM returns an Item Add Data Catalogue Response to the Publisher. If the GTIN was registered, TDSN IM will forward GS1 GlobalRegistry item add response to the Publisher.  

Subsequent to Item Add, a Publisher can make a modification, update, or correction to a production Item, which is then automatically synchronized to the Subscribers that did not reject the original Item publication. TDSN IM returns an Item Modify/Correct Data Catalogue Response. |
| 2    | Item Link  
   a The Publisher adds Item Link to TDSN. The links create a logical hierarchy of the items. For example, pallets linked to cases; cases to consumer units.  
   b TDSN IM returns Item Link Data Catalogue Response. |
| 3    | Item Publish  
   a The Publisher publishes item hierarchies (linked items).  
   b TDSN IM returns Item Publication Data Catalogue Response. |
| 4    | Item Confirmation  
   The Publisher receives notification of subscriber Accept/Reject of the item publication. |
Item Synchronization with the webMethods Transora Module

As a supplier, your item synchronization process from the perspective of the Trading Networks looks similar: you send item request business documents to Transora and receive response business documents from Transora. Additionally, you can receive item confirmation messages from Transora.

The following sections describe:

- The process model for sending an item request message and receiving an item response from Transora
- Receiving an item confirmation message from Transora via Trading Networks routing rules

Processing Transora Item Request/Response Messages

The following figure illustrates your enterprise (acting as a supplier) sending a request business document to and receiving a response business document from Transora. For further explanation, see the table that follows the figure.
### Step 1
Your enterprise (acting as a supplier) submits an internal request business document from your back-end system to the Integration Server. The request business document can be a Trading Networks document or an Integration Server publishable record, with a pre-defined ConversationID.

### Step 2
The PRT looks for a matching ConversationID among the enabled process models. If the PRT does not find a matching ConversationID, it starts a new business process.

### Step 3
A mapping service maps the internal information to the Transora-related information.

### Step 4
A Trading Networks send service sends the Transora request business document to Transora.

### Step 5
Transora receives the Transora request business document.

### Step 6
Transora responds by sending a response business document to your enterprise by way of the Trading Networks. The response business document references the request business document, which contains the ConversationID.

### Step 7
The response business document is received by the webMethods EDIINT Module, which triggers the \texttt{wm.EDIINT.rules:processMsg} service. The EDIINT service extracts the Transora response business document and submits it to Trading Networks. Trading Networks extracts the ConversationID from response business document before passing the document to the PRT.

The PRT matches the ConversationID from the Transora response business document with the ConversationID associated with a specific process model. The business process resumes.

### Step 8
A mapping service maps the Transora-related information to internal information recognizable by your back-end system.

### Step 9
At the \textbf{Send to Back-End} step, a service sends the internal response business document to your back-end system.

### Step 10
Your back-end system receives and processes the internal response business document.

## Receiving Item Confirmation Messages

Transora sends the resulting confirmations to the Trading Networks in a response business document. The response business document is recognized as a “Transora Response” TN XML document type and triggers the “GDSN Item Registry Response” processing rule. The processing rule changes the document status in the Transaction Analysis.
Installing webMethods Transora Module 6.1.1

- System Requirements ................................................................. 26
- Install the webMethods Transora Module ................................. 27
- Uninstall the webMethods Transora Module ............................ 28
System Requirements

This section describes the system requirements that must be met before you can install the webMethods Transora Module.

Platform and Operating System Requirements

The Transora Module supports the following platforms and uses the same JVM as its host Integration Server.

<table>
<thead>
<tr>
<th>Platform and Operating System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 2000, 2003, XP</td>
</tr>
<tr>
<td>Solaris 2.8, 2.9</td>
</tr>
<tr>
<td>HP-UX 11i (64-bit)</td>
</tr>
<tr>
<td>Red Hat Enterprise Linux 4.0</td>
</tr>
</tbody>
</table>

webMethods Software Requirements

webMethods Components

The table below lists the webMethods components you must install before or at the same time you install the Transora Module. The table also lists the webMethods components you must install at some point for the Transora Module to operate fully.

<table>
<thead>
<tr>
<th>Required for Installation</th>
<th>Required for Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>webMethods Integration Server 6.1</td>
<td>webMethods Developer 6.1</td>
</tr>
<tr>
<td>Trading Networks Server 6.1</td>
<td>Trading Networks Console 6.1</td>
</tr>
<tr>
<td>WmEDIINT package of the webMethods EDI Module 6.1</td>
<td>webMethods Modeler 6.1.5</td>
</tr>
</tbody>
</table>

Hardware Requirements

The Transora Module has no hardware requirements beyond those of the Integration Server.
Install the webMethods Transora Module

Install webMethods Transora Module 6.1.1 on the same machine as the Integration Server. The webMethods Installer automatically installs the Transora Module in the Integration Server installation directory.

Important! This section provides specific instructions for installing the Transora Module. For complete instructions for using the webMethods Installer, see the webMethods Installation Guide.

To install the webMethods Transora Module

2. Specify the installation directory as the webMethods installation directory (by default, webMethods6).
3. In the component selection list, navigate to webMethods Integration Platform ➤ eStandards ➤ webMethods Transora Module. Choose to install the Program Files, as well as any required webMethods components you have not already installed. You also can choose to install the Samples and Documentation.
4. If you do not already have the webMethods EDIINT Module installed, you can install it now. To do so, in the component selection list, navigate to webMethods Integration Platform ➤ EDI ➤ webMethods EDI Module ➤ EDIINT. Choose to install the Program Files. You also can choose to install the Documentation.
5. Review the list of components you selected to install.
6. The webMethods Installer installs the selected components.
Uninstall the webMethods Transora Module

Important! This section provides specific instructions for uninstalling the Transora Module. For complete instructions for uninstalling webMethods components, see the webMethods Installation Guide.

To uninstall the webMethods Transora Module

1. Shut down the Integration Server.

Note: Although the instructions in the webMethods Installation Guide indicate that to uninstall you should shut down all webMethods components and applications that are running on your machine, you need to shut down only the Integration Server to uninstall the Transora Module.

2. Use the Windows Add/Remove Programs utility, available from the Control Panel, to start the uninstaller.

3. Select webMethods 6.1 as the program to uninstall. The webMethods Uninstaller starts.

4. In the webMethods Uninstaller, select to uninstall webMethods Transora Module.

5. The uninstaller removes all Transora Module-related files that were installed into the Integration Server_directory\packages\Transora directory. The uninstaller does not delete files created after you installed the Transora Module (for example, user-created or configuration files), nor does it delete the directory structure that contains the files.

6. If you do not want to save the files that the uninstaller did not delete, navigate to the Integration Server_directory\packages\Transora directory and delete the subdirectories.
Configuring the webMethods Transora Module

- Overview ................................................................. 30
- Sign up with Transora for Data Synchronization Service ......................... 30
- Define Trading Networks Profiles .................................... 30
- Define a Trading Partner Agreement .................................. 30
- Define TN Document Types and/or IS Document Types ..................... 31
- Integrate Back-End Applications with Transora ............................. 31
- Set Up Business Document Processing .................................. 32
- Update the ACL Settings for the WmEDIINT Package ..................... 33
- Run the webMethods Transora Module Sample ............................ 33
- Items Configured For You When You Install the Transora Module .......... 34
Overview

This chapter lists the steps that you need to complete to configure the webMethods Transora Module. The steps appear in the order that webMethods recommends you complete them. Additionally, this chapter contains information about items that are configured automatically for you when you install the Transora Module.

Step 1: Sign up with Transora for Data Synchronization Service

To participate in the Transora community, you first must sign up with Transora by completing and submitting a Transora application for a TDSN Subscription Agreement and pay your subscription fee.

To complete a Transora agreement and to find out about subscription fees, access the Transora Web site at http://www.transora.com. When you access the site, choose to contact Transora Customer Support.

Although a Global Location Number (GLN) is not required to complete a TDSN Subscription Agreement, it is required for service activation. If you do not enter your GLN the first time you submit your agreement, you will need to enter it later and resubmit your agreement for service activation.

Step 2: Define Trading Networks Profiles

You use the Trading Networks Console to define Trading Networks profiles for your enterprise and for Transora, with whom you want to exchange Transora messages. A trading partner profile includes such parameters as the GLN and the destination URL.

For more information about defining trading partner profiles, see Chapter 4, “Defining Trading Networks Partner Profiles” in this guide and the webMethods Trading Networks User’s Guide.

Step 3: Define a Trading Partner Agreement

You must define a trading partner agreement (TPA) to provide parameters that govern how to exchange Transora messages with Transora. You define a TPA using the Trading Networks Console.

For information about how to define a TPA, see Chapter 5, “Defining a Trading Partner Agreement” in this guide and the webMethods Trading Networks User’s Guide.
Step 4: Define TN Document Types and/or IS Document Types

You must define document types for the internally-formatted business documents that your back-end system will send to start a Transora business process. For example, your back-end system might send an internally-formatted item request document that will be sent to Transora as a TDC Item Request message. In this case, you would define a document type for the internally-formatted item request document.

You can create either:

- **A TN document type.** Use a TN document type if your back-end system sends documents via Trading Networks. For more information, see:
  - Chapter 6, “Defining TN Document Types”
  - “Customizations for the Wait for Internal Request Step” on page 66 in Chapter 7, “Customizing a Process Model Template”
  - *webMethods Trading Networks User’s Guide*

- **An IS document type.** Use an IS document type if you plan to use the publish/subscribe feature of the Integration Server. For more information, see:
  - “Customizations for the Wait for Internal Request Step” on page 66 in Chapter 7, “Customizing a Process Model Template”
  - *webMethods Developer User’s Guide*
  - *Publish-Subscribe Developer’s Guide*

Step 5: Integrate Back-End Applications with Transora

To integrate your back-end applications with Transora, you must create:

- Services that work with the business documents that your back-end system sends to the Integration Server. This type of service is known as an *outbound mapping service*, and its function is to map data from an internally-formatted request document into a Transora-formatted request document. For more information, see “Customizations for the Map to TDC Request Step” on page 67 in Chapter 7, “Customizing a Process Model Template”.

- Services that send response documents to your back-end system. For more information, see “Customizations for the Process Response and End Step” on page 68 in Chapter 7, “Customizing a Process Model Template”.
CHAPTER 3 Configuring the webMethods Transora Module

Step 6: Set Up Business Document Processing

For complex request/response scenarios, use a process model to define a business process. For simple request/response scenarios, webMethods recommends that you define the actions you want to perform using only Trading Networks processing rules, not a process model.

Using a Process Model

The Transora Module provides process model templates that define a Transora business process. That is, the process model defines the steps to take to send a Transora request message to Transora and receive a response back from Transora. You can customize a process model template to meet your site-specific requirements.

To customize a process model template, you update some of the steps in the process model. For example, the first step of the business process waits for an internally-formatted document from the back-end system. You will update this step to specify the TN document type or publishable IS document type that defines the internally-formatted business document.

For information about customizing a process model template, see Chapter 7, “Customizing a Process Model Template” in this guide, the Getting Started with webMethods Business Partner Management guide, and the webMethods Modeler User’s Guide.

Using Trading Networks Processing Rules

The Transora documents can also be sent using Trading Networks processing rules for better performance. Processing rules specify how you want Trading Networks to process documents. For example, you might want Trading Networks to send an alert e-mail message to a contact, and then deliver the document to the receiver that is identified in the document.

To use processing rules, define a doctype in Trading Networks and define a corresponding processing rule with a desired action. When a document is sent to Trading Networks that matches the doctype, the processing rule will be invoked.

The action for this rule can be either of the following:

- Create Transora payload defined under
  `wm.ip.transora.recs.CatalogueRequest:docType_Payload` and call the
  `wm.ip.transora.cm.handlers:send` service.

- Create Transora payload defined under
  `wm.ip.transora.recs.CatalogueRequest:docType_Payload` and call the
  `wm.ip.transora.cm.handlers:send` service without defining an extra doctype and processing rule.
Update the ACL Settings for the WmEDIINT Package

For a sample service, see the wm.ip.transora.samples.service:itemAddUsingRoutingRule service which is in the WmTransoraSamples package to add items to Transora. In this sample service, input the GTIN to add, the GLN of the sender, and the receiver id. The Transora response documents are recognized by pre-loaded response documents.

For more information about defining and managing processing rules, see the webMethods Trading Networks User’s Guide.

Step 7: Update the ACL Settings for the WmEDIINT Package

Transora posts responses to your server via EDIINT. To post a response, Transora should invoke the wm.EDIINT:receive service, which is in the WmEDIINT package.

When you create a Trading Networks profile for Transora, Trading Networks creates a user account for Transora on the Integration Server and includes it in the TNPartners group. For Transora to use the wm.EDIINT:receive service to post a response, the Transora user account must have authority to execute this service.

To grant the Transora user account the privilege to execute the wm.EDIINT:receive service, update the Execute ACL for the wm.EDIINT:receive service to use the TNPartners ACL.

Note: The TNPartners group and TNPartners ACL are added to the Integration Server when you install Trading Networks.

To update the ACL for the wm.EDIINT:receive service

1. Start the Developer if it is not already started.
2. In the Navigation panel, browse to and open the WmEDIINT package.
3. In the Navigation panel, browse to and select the wm.EDIINT:receive service.
4. In the editor, click the Permissions tab.
5. Select TNPartners for the Execute ACL.
6. Click Save.

Step 8: Run the webMethods Transora Module Sample

The sample demonstrates each step in a Transora business process:

- Receiving a request document from a back-end system
- Creating a Transora-formatted request message from the data in the request document sent by the back-end system
Waiting for and handling an EDIINT Message Delivery Notification (MDN) from Transora to acknowledge the receipt of the Transora request message.

Waiting for and verifying the Transora response document that Transora sends in response to the Transora request message.

Sending a document back to the back-end system with the outcome of the Transora request.

It is a good idea to run the sample early in your implementation of the business process, as well as to familiarize yourself with some of the implementation tasks you will need to complete.

For information about how to set up and run the sample, see Appendix A, “webMethods Transora Module Sample”.

### Items Configured For You When You Install the Transora Module

When you install the Transora Module, it automatically configures some of the items required to use the Transora Module. These items include:

- **TN document type to use for Transora response messages**. For more information, see “TN Document Types Provided with the Transora Module” on page 53 in Chapter 6, “Defining TN Document Types”.

- **Document attributes that are extracted from Transora documents**. For more information, see “How Information Extracted from the Transora Messages Is Used” on page 59 in Chapter 6, “Defining TN Document Types”.

- **TDC Accept Reject Response processing rule**, which processes an Accept Reject response document from Transora. Trading Networks uses this processing rule when it receives a TDC Accept/Response response message from Transora. The processing rule defines the following actions:
  - Save the TDC Accept/Response response message to the Trading network database.
  - Set the Processing Status to AcceptReject:NEW.
  - Invoke the `wm.ip.transora.rules:processResponse` service, which is a sample service. You should create your own service to handle the TDC Accept/Response response message, and update the processing rule to invoke the service you create.
Defining Trading Networks Partner Profiles

- What is a Trading Partner and a Trading Partner Profile? 36
- Defining the Profile for Your Corporation 37
- Defining a Profile for Transora 39
What is a Trading Partner and a Trading Partner Profile?

A trading partner is any person or organization with whom you want to conduct business electronically. You define a profile to provide information about a trading partner.

Using the Trading Networks Console, you will need to define profiles for:

- **Your Corporation**. You create the Trading Networks Enterprise profile to identify information about your own corporation.
- **Transora**. You create a Trading Networks partner profile to supply information about Transora.

### Information that You Supply in a Profile

Trading Networks provides a set of standard fields in a profile that you use to supply information about a trading partner, including the corporation name and identifying information, contact information, and preferred delivery methods.

Trading Networks allows profiles to be extended to include additional fields. The webMethods EDI Module takes advantage of this feature by adding extended profile fields for EDIINT. When you install the WmEDIINT package of the webMethods EDI Module, it adds these EDIINT-specific fields to profiles. Because the Transora Module uses the EDIINT transport when defining profiles to use with the Transora Module, you must supply this EDIINT information.

The Transora Module also takes advantage of the extended profile fields feature by including its own extended field. When you install the Transora Module, it automatically adds the UserID extended field to profiles. This field contains the batch user ID that the Transora Module uses to send documents to the Transora Data Catalog (TDC) exchange. You obtain the value for this batch user ID from Transora.

### External ID Type Field in a Profile

A standard field in the profile is the external ID type. The external ID type defines how Trading Networks identifies a corporation within a message or document. That is, Trading Networks uses the external ID within a document to determine who are the document’s sender and receiver. For example, a corporation that sends a document might place its D-U-N-S number within the field of the document that represents the sender.

Trading Networks provides several external ID types (for example, DUNS for a D-U-N-S number). However, the external ID types supplied by Trading Networks do not include the external ID types that are used in Transora messages.
The Transora Module adds the following additional external ID types to Trading Networks that you can use when defining profiles. When creating profiles for use with the Transora Module, you must specify values for these external ID types.

- **GLN** for the Global Location Number (GLN)
- **AS2-ID** for the AS2 Routing ID used when communicating with Transora via EDIINT AS2

## Defining the Profile for Your Corporation

When defining a profile for your corporation, you define the Enterprise profile in Trading Networks. To define the Enterprise profile, use the Trading Networks Profile Assistant. The Profile Assistant is a wizard that takes you step-by-step through the process of creating your own profile. For complete instructions about filling in the fields in the Profile Assistant and about creating the Enterprise profile in general, see the *webMethods Trading Networks User’s Guide*.

### Important! After defining the Enterprise profile, you must enable it in Trading Networks.

The following table lists the minimum information that you must supply in the profile.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field or Sub-Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate</td>
<td>Corporation Name</td>
<td>Specify the name of your corporation.</td>
</tr>
<tr>
<td></td>
<td>Partner Type</td>
<td>Specify the type of software that you use to interact with Transora, that is <em>webMethods Trading Networks</em>.</td>
</tr>
</tbody>
</table>
## CHAPTER 4  Defining Trading Networks Partner Profiles

### Corporate (continued)

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field or Sub-Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>External ID Type</td>
<td>Value</td>
<td>Specify values for the following external ID types.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>External ID Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLN</td>
<td>The Global Location Number (GLN) for your corporation.</td>
</tr>
<tr>
<td>AS2-ID</td>
<td>The type of identification you use for EDIINT AS2 messages, (for example, your GLN or D-U-N-S number). This value should correspond to the values in the ‘AS2-To:’ and ‘AS2-From:’ fields in the header of the EDIINT messages. Typically, AS2-ID and GLN are the same.</td>
</tr>
</tbody>
</table>

**required external ID type**

One external ID type is required for all profiles. This is known as the *required external ID type*, and it is configurable. By default, the required external ID type is **DUNS**. Trading Networks uses the required external ID type to create an Integration Server user account for a partner.

For more information about the required external ID type and how to configure it, see the configuring chapter of the [webMethods Trading Networks User’s Guide](#).

### Delivery Method Protocol

Specify information for the *Primary HTTP* delivery method.

**Important!** Make sure to select the *Use as Preferred Protocol* check box.

<table>
<thead>
<tr>
<th>Field or Sub-Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Specify the host name or IP address for your Integration Server.</td>
</tr>
<tr>
<td>Port</td>
<td>Specify the port number on which your Integration Server listens for incoming requests.</td>
</tr>
<tr>
<td>Location</td>
<td>Specify the following URL: <code>/invoke/wm.EDIINT/receive</code></td>
</tr>
</tbody>
</table>

**Note:** You may need to contact Transora Customer Support to verify the Host, Port, and Location values in the event that Transora has updated them.
Defining a Profile for Transora

When defining a profile for Transora, you define a partner profile using the Trading Networks Console. To define a partner profile, use the Trading Networks New Profile screen. You cannot create a partner profile until after you have created the Enterprise profile for your corporation (as discussed in “Defining the Profile for Your Corporation” on page 37).

For complete instructions to create a partner profile, see the webMethods Trading Networks User’s Guide.

**Important!** After defining a partner profile, you must enable it in Trading Networks.

---

### Extended Fields

**Group**

**Transora UserID**

Specify the batch user ID that Transora supplied to your corporation for sending and viewing data in the Transora Data Catalog. Specify the userID in this format:

GLN@xls

For example, a corporation whose GLN is 4561237890001 would specify this userID: 4561237890001@xls

### Security

**Sign and Decrypt**

Specify information for the following fields on both the Sign and Decrypt sub-tabs.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate</td>
<td>Your digital certificate that contains your public key.</td>
</tr>
<tr>
<td><strong>Important!</strong></td>
<td>Your partner must have this certificate installed on their system and the certificate should be associated with your corporation on their system.</td>
</tr>
<tr>
<td>CA Chain</td>
<td>Digital certificate chain of the certifying authority (CA) that signed your public certificate. If you do not have the CA's digital certificate chain, you can export it from your public certificate.</td>
</tr>
<tr>
<td><strong>Important!</strong></td>
<td>If you are using self-signed certificates, your partner must have your CA Certificate installed in their systems trusted root directory.</td>
</tr>
<tr>
<td>Private Key</td>
<td>The private key that corresponds to the public key in your digital certificate.</td>
</tr>
</tbody>
</table>
The following table lists the minimum information that you must supply in the profile.

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field or Sub-Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corporate</td>
<td>Corporation Name</td>
<td>Specify Transora.</td>
</tr>
<tr>
<td></td>
<td>Partner Type</td>
<td>Specify Other/Unknown.</td>
</tr>
<tr>
<td>Corporate (continued)</td>
<td>External ID Type</td>
<td>Specify values for the</td>
</tr>
<tr>
<td></td>
<td>and Value</td>
<td>following external ID types.</td>
</tr>
<tr>
<td>GLN</td>
<td>Description</td>
<td>Specify 8380160030003, which is Transora’s Global Location Number.</td>
</tr>
<tr>
<td>AS2-ID</td>
<td>Description</td>
<td>The type of identification that Transora uses for EDIINT AS2 messages, (for example, your GLN or D-U-N-S number). This value should correspond to the values in the ‘AS2-To:’ and ‘AS2-From:’ fields in the header of the EDIINT messages. Typically, AS2-ID and GLN are the same.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For testing with Transora, specify TRANSORA_QA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For production, specify TRANSORA</td>
</tr>
<tr>
<td>required</td>
<td>Description</td>
<td>Specify the value of the Trading Networks required external ID type. If the required external ID type is DUNS, specify 838016003, which is Transora’s D-U-N-S number.</td>
</tr>
<tr>
<td>external ID type</td>
<td></td>
<td>For more information about the required external ID type and how to configure it, see the configuring chapter of the webMethods Trading Networks User’s Guide.</td>
</tr>
</tbody>
</table>
### Defining a Profile for Transora

#### Delivery Method

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field or Sub-Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td></td>
<td>Specify information for the <strong>Primary HTTP</strong> delivery method.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Important!</strong> Make sure to select the <strong>Use as Preferred Protocol</strong> check box.</td>
</tr>
</tbody>
</table>

#### Host

- Transora’s IP address.
  - HTTP or HTTPS:
    - For testing with Transora, specify `transhub.preprod.transora.com`
    - For production, specify `trashub.transora.com`

#### Port

- The port number on which Transora listens for incoming requests.
  - For HTTP, specify 4080
  - For HTTPS, specify 1443

#### Location

- Specify the following URL:
  - HTTP or HTTPS:
    - For testing with Transora, specify `/exchange/8380160030003`
    - For production, specify `/exchange/0838016003001`

#### Extended Fields

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field or Sub-Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extended Fields</td>
<td>EDIINT Group AS2MDNUR URL field</td>
<td>The URL to which Transora is to send asynchronous MDNs. The format of the URL is typically <code>http://userID:password@host:port/invoke/wm.EDIINT/receive</code>, where: <code>userID</code> is the user name for the user account that Trading Networks generates when you create the Transora profile. <code>password</code> is the password that Trading Networks generates when you create the Transora profile. <code>host</code> is the host name or IP address for your Integration Server. <code>port</code> is the port number on which your Integration Server listens for incoming requests. <strong>Note:</strong> You can leave this field blank if you do not want to request asynchronous MDNs.</td>
</tr>
</tbody>
</table>

#### Security

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field or Sub-Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>Sign and Decrypt sub-tabs</td>
<td>Use the default settings on the <strong>Sign</strong> and <strong>Decrypt</strong> sub-tabs.</td>
</tr>
</tbody>
</table>
CHAPTER 4  Defining Trading Networks Partner Profiles

<table>
<thead>
<tr>
<th>Tab</th>
<th>Field or Sub-Tab</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Encrypt and Verify</em> sub-tabs</td>
<td></td>
<td>Specify information for the following fields on both the <em>Encrypt</em> and <em>Verify</em> sub-tabs.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>Certificate</td>
<td>Transora’s digital certificate that contains Transora’s public key.</td>
<td></td>
</tr>
<tr>
<td>CA Chain</td>
<td>Digital certificate of the CA that signed Transora’s public certificate. If you do not have the CA's digital certificate chain, you can export this from the public certificate.</td>
<td></td>
</tr>
</tbody>
</table>

*Important!* Transora has two layers of CA certificates in their public certificate chain. You must export both of these CA certificates into individual certificates files and then import these files in the correct order into Trading Networks.
Defining a Trading Partner Agreement

- What is a Trading Partner Agreement? ............................................. 44
- Defining a Trading Partner Agreement ............................................ 44
- Parameter Settings for the TPA Data .............................................. 45
What is a Trading Partner Agreement?

A Trading Partner Agreement (TPA) is a set of parameters that is used to govern how two trading partners exchange messages. The webMethods Transora Module uses a TPA to govern how your enterprise and Transora exchange Transora messages. The TPAs include such information as:

- The DTD to use to insert in the DOCTYPE header of messages being sent to Transora
- Whether to validate an outbound Transora message before sending it to Transora
- Whether to sign and/or encrypt an outbound Transora message before sending it to Transora
- Whether you want Transora to return an MDN in response to the outbound Transora message

A TPA augments webMethods Trading Networks profiles and offers a flexible way to process and manage transactions between your enterprise and Transora.

For more information about TPAs, see the webMethods Trading Networks User’s Guide.

Defining a Trading Partner Agreement

You can define as many as TPAs as you need. Each TPA in the Trading Networks system is uniquely identified by its combination of a sender ID, a receiver ID, and an agreement ID. If a matching TPA is found for the condition, then that TPA is used; otherwise, a default TPA is used. The default TPA is used when the sender ID = UNKNOWNS, receiver ID = UNKNOWNS and agreement ID = ‘Default-Transora-TPA’.

You use the Trading Networks Console to define a TPA. For complete instructions about creating a TPA, see the webMethods Trading Networks User’s Guide.

The following table lists information to supply when creating a TPA to govern the exchange of Transora messages between your enterprise and Transora.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sender</td>
<td>This field identifies the partner that has the sender role for the TPA. You can select the Enterprise profile or any sender from the profiles defined on your Trading Networks system. Alternatively you can specify UNKNOWNS, but this field cannot be blank.</td>
</tr>
<tr>
<td>Receiver</td>
<td>This field identifies the partner that has the receiver role for the TPA. Select the Transora profile because it is the receiver. Alternatively you can specify UNKNOWNS, but this field cannot be blank.</td>
</tr>
<tr>
<td>Agreement ID</td>
<td>Specify Default-Transora-TPA.</td>
</tr>
</tbody>
</table>
Parameter Settings for the TPA Data

The `wm.ip.transora.cm.recs:userParameters` IS document type defines the structure of the TPA data. The TPA data contains the fields that the Transora Module uses to govern the exchange of Transora messages between your enterprise and Transora.

Specify `wm.ip.transora.cm.recs:userParameters`

For a description of the values that you specify for the fields defined in the `wm.ip.transora.cm.recs:userParameters` IS document type, see “Parameter Settings for the TPA Data” below.

---

### Parameter Settings for the TPA Data

The `wm.ip.transora.cm.recs:userParameters` IS document type defines the structure of the TPA data. The Transora Module uses these fields when processing Transora messages being sent from your enterprise to Transora.

The Transora Module uses these fields by accessing them during the business process for a Transora transaction. The process models that the Transora Module provides, which define the Transora business processes, use these fields to select:

- The mapping service to use to format the outbound XML
- The transport to use to deliver the outbound message to Transora

For more information about the process models, see Chapter 7, "Customizing a Process Model Template".

The following table lists the TPA data parameters that the `wm.ip.transora.cm.recs:userParameters` IS document type defines and describes each parameter.

<table>
<thead>
<tr>
<th>Parameter in the TPA data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransoraGLN</td>
<td>Specify the Transora GLN to which you want the Transora Module to send a Transora message.</td>
</tr>
<tr>
<td>AltSenderGLN</td>
<td>Leave blank; this field is not used by the Transora Module.</td>
</tr>
</tbody>
</table>
### CHAPTER 5 Defining a Trading Partner Agreement

<table>
<thead>
<tr>
<th>Parameter in the TPA data</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Transport</strong></td>
<td>Specify EDIIN.</td>
</tr>
</tbody>
</table>

**Note:** The EDIIN transport delivers the outbound message using the Preferred Protocol defined in the Delivery Method tab of the receiving partner’s profile. The Preferred Protocol should be Primary HTTP. For more information about setting up a profile for use with the Transora Module, see Chapter 4, “Defining Trading Networks Partner Profiles”.

<table>
<thead>
<tr>
<th>DTD</th>
<th>This is the DTD SYSTEM URI that you want the Transora Module to insert in the DOCTYPE header of the outbound message that is being sent to Transora.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>■ For testing with Transora, specify <a href="http://www.preprod.transora.com/util/pi/TDC_XML/4.0/CatalogueRequest_Envelope.dtd">http://www.preprod.transora.com/util/pi/TDC_XML/4.0/CatalogueRequest_Envelope.dtd</a></td>
</tr>
<tr>
<td></td>
<td>■ For production, specify <a href="http://www.transora.com/util/pi/TDC_XML/4.0/CatalogueRequest_Envelope.dtd">http://www.transora.com/util/pi/TDC_XML/4.0/CatalogueRequest_Envelope.dtd</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Version</strong></th>
<th>Specify 4.2, which is the schema version of the payload to use.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ValidateOutput</strong></td>
<td>Whether you want the Transora Module to validate the outbound Transora message before sending it to Transora. Specify one of the following:</td>
</tr>
<tr>
<td></td>
<td>Yes Validate the outbound message.</td>
</tr>
<tr>
<td></td>
<td>No Do not validate the outbound message.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>MessageType</strong></th>
<th>Whether you want the Transora Module to sign and/or encrypt the outbound Transora message before sending it to Transora. Specify one of the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>signed</td>
<td>Sign the outbound message.</td>
</tr>
<tr>
<td>encrypted</td>
<td>Encrypt the outbound message.</td>
</tr>
<tr>
<td>signedAndEncrypted</td>
<td>Sign and encrypt the outbound message.</td>
</tr>
<tr>
<td>plain</td>
<td>Neither sign nor encrypt the outbound message.</td>
</tr>
</tbody>
</table>
### RequestMDN

Whether you want Transora to return an MDN in response to the outbound message. Specify one of the following:

- **none**: Do not request a return MDN.
- **synchronousMDN**: Request a return synchronous MDN.
- **asynchronousMDN**: Request a return asynchronous MDN.

**Note**: The process model that the Transora Module provides is designed assuming a synchronous MDN. If you specify **none** or **asynchronousMDN** you will need to update the process model appropriately.

### RequestSignedReceipt

If you specified that you want Transora to return an MDN, use this parameter to indicate whether you want the MDN to be signed. This parameter is ignored when **RequestMDN** is set to **none**. Specify one of the following for **RequestSignedReceipt**:

- **true**: Request a signed MDN.
- **false**: Request a plain MDN (not signed).
Defining TN Document Types

- What are TN Document Types? .............................................. 50
- Defining TN Document Types for Back-End System Documents .............................. 51
- TN Document Types Provided with the Transora Module .......................................... 53
- How Information Extracted from the Transora Messages Is Used ............................ 59
CHAPTER 6  Defining TN Document Types

What are TN Document Types?

TN document types are definitions that webMethods Trading Networks uses when it receives a business document. Trading Networks matches an incoming document against its TN document types to determine what type of document it received. After matching an incoming document to a TN document type, Trading Networks uses other information in the TN document type to determine:

- The attributes within the document that Trading Networks is to extract from the document.
- The pre-processing options that Trading Networks should use. The pre-processing options specify whether any or all of the following processing should occur:
  - Indicate whether Trading Networks is to verify the document’s digital signature
  - Validate the structure of the document
  - Determine whether the document has been received before
  - Save the document to the Trading Networks database

TN Document Types for Documents from Your Back-End Systems

When the webMethods Transora Module receives a business document from a back-end system, it invokes a Trading Networks service to recognize the type of business document. You must define the TN document type that Trading Networks will use to recognize the documents from your back-end system. Trading Networks can recognize both XML documents and flat files. For more information, see “Defining TN Document Types for Back-End System Documents” on page 51.

Note: Trading Networks also can receive an Integration Server publishable document from a back-end system. For information about defining Integration Server publishable documents, see the Publish-Subscribe Developer’s Guide.

To form the outbound Transora message, the Transora Module performs outbound mapping. That is, the Transora Module maps data from the documents received from a back-end system to the outbound Transora message. The resulting Transora message is in XML format. The process models provided with the Transora Module define the logic to perform these actions. For more information, see Chapter 7, “Customizing a Process Model Template”.

Defining TN Document Types for Transora Messages

The Transora Module processes response messages received from Transora using Trading Networks. As a result, Trading Networks must have TN document types for these Transora response messages. The Transora Module provides these TN document types for Transora messages. For more information, see “TN Document Types Provided with the Transora Module” on page 53.

Defining TN Document Types for Back-End System Documents

You must define TN document types for business documents that Trading Networks receives from your back-end system. These TN document types can be either of the following:

- **TN XML document types**, which define how to recognize XML documents
- **TN flat file document types**, which define how to recognize flat files

**Note:** When you customize your process models, you will assign a TN document type (or Integration Server publishable document) to the first step in the process model. By doing so, you indicate what document starts a Transora business process. For more information, see Chapter 7, “Customizing a Process Model Template”.

You define a TN document type using the Trading Networks Console. For step-by-step procedures for defining a TN document type, see the *webMethods Trading Networks User’s Guide*. The following table lists the tabs in the **Document Type Details** screen and a description of the type of information you supply on each tab when defining a TN document type.
### CHAPTER 6  Defining TN Document Types

<table>
<thead>
<tr>
<th>Tab</th>
<th>Description</th>
</tr>
</thead>
</table>
| Identify  | On the **Identify** tab, you supply information that Trading Networks uses to identify business documents. That is, Trading Networks uses the identification information to determine whether an incoming document matches the TN document type.  
For TN XML document types, you might specify the root tag that a business document must have and/or include XQL queries that identify specific elements that must be present in the business documents for the TN document type to match. You also might configure the TN XML document type to specify arbitrary pipeline elements to be used as identifying attributes.  
If your back-end system sends a flat file document, you must create a document gateway service in addition to defining a TN flat file document type. In the identification information of TN flat file document types, you might specify how Trading Networks is to recognize flat file business documents based on variables that the document gateway service places in the pipeline. |
| Extract   | On the **Extract** tab, you specify the attributes that you want Trading Networks to extract from business documents and how to locate the attributes within business documents.                                                                                      |
| Namespace | On the **Namespace** tab, you specify information about the namespaces that XML business documents might use. Trading Networks uses the namespace mappings when applying XQL queries against a business document.  
**Note:** This tab is not available for a TN flat file document type.                                                                                                                                  |
| Options   | On the **Options** tab, you specify options for pre-processing business documents. The options enable you to indicate whether you want Trading Networks to perform any or all of the following pre-processing actions:  
- Verify the digital signature of a business document.  
- Validate the structure of a business document (using an IS document type that you specify).  
- Check the uniqueness of a business document, that is, determine whether Trading Networks has already received the document.  
- Save a copy of the document content, attributes, and/or log information to the Trading Networks database.                                                                                 |
When you install the Transora Module, it automatically installs the following TN document types for Transora response messages in Trading Networks:

<table>
<thead>
<tr>
<th>TN Document Type</th>
<th>Description</th>
<th>See Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TDC Item Response</td>
<td>Represents a Transora Data Catalog Response message</td>
<td>53</td>
</tr>
<tr>
<td>TDC Publication Response</td>
<td>Represents a Transora Publication Response message</td>
<td>54</td>
</tr>
<tr>
<td>TDC Accept Reject Response</td>
<td>Represents a Transora Accept/Reject Response message</td>
<td>55</td>
</tr>
<tr>
<td>TDC Item Request</td>
<td>Represents a Transora Data Catalog Request message</td>
<td>56</td>
</tr>
<tr>
<td>TDC GDSN Item Registry Response</td>
<td>Represents a GS1 Global Registry item registration status response routed by Transora to publishers</td>
<td>57</td>
</tr>
<tr>
<td>TDC GDSN Party Registry Response</td>
<td>Represents a GS1 Global Registry party (GLN) contract confirmation response routed by Transora to publishers</td>
<td>58</td>
</tr>
</tbody>
</table>

The following sections provide additional detail about these TN document types provided with the Transora Module.

**TDC Item Response TN Document Type**

This TN document type provides the definition for a Transora Data Catalog Response message.

**Identification Information**

**Root Tag:** Envelope

<table>
<thead>
<tr>
<th>Identification Queries</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Envelope[0]/CatalogResponse[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/eb:MessageHeader[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td>ITEM</td>
</tr>
</tbody>
</table>
CHAPTER 6 Defining TN Document Types

Extraction Information

<table>
<thead>
<tr>
<th>Document Attribute</th>
<th>Extraction Queries</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>SenderID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:From[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>ReceiverID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:To[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>DocumentID</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseHeader[0]/MessageIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>ConversationID</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseHeader[0]/CorrelationIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>GTIN</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/ID1[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_ACTION</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Action[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_TYPE</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td></td>
</tr>
</tbody>
</table>

IS Document Type

When formatting this Transora message as an IData object, use the `wm.ip.transora.recs.CatalogueResponse:CatalogueResponse_Envelope` IS document type.

TDC Publication Response TN Document Type

This TN document type provides the definition for a Transora Publication Response message.

Identification Information

Root Tag: Envelope

<table>
<thead>
<tr>
<th>Identification Queries</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Envelope[0]/CatalogResponse[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/eb:MessageHeader[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td>PUBLICATION</td>
</tr>
</tbody>
</table>
TN Document Types Provided with the Transora Module

Extraction Information

<table>
<thead>
<tr>
<th>Document Attribute</th>
<th>Extraction Queries</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>SenderID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:From[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>ReceiverID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:To[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>DocumentID</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseHeader[0]/MessageIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>ConversationID</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseHeader[0]/CorrelationIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>GTIN</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/ID1[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_ACTION</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Action[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_TYPE</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td></td>
</tr>
</tbody>
</table>

IS Document Type

When formatting this Transora message as an IData object, use the `wm.ip.transora.recs.CatalogueResponse:CatalogueResponse_Envelope` IS document type.

TDC Accept Reject Response TN Document Type

This TN document type provides the definition for a Transora Accept/Reject Response message.

Identification Information

Root Tag: Envelope

<table>
<thead>
<tr>
<th>Identification Queries</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Envelope[0]/CatalogResponse[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/eb:MessageHeader[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td>AcceptRejectResponse</td>
</tr>
</tbody>
</table>
CHAPTER 6 Defining TN Document Types

Extraction Information

<table>
<thead>
<tr>
<th>Document Attribute</th>
<th>Extraction Queries</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>SenderID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:From[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>ReceiverID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:To[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>DocumentID</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseHeader[0]/MessageIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>ConversationID</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseHeader[0]/CorrelationIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>GTIN</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/ID1[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_ACTION</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Action[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_TYPE</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td></td>
</tr>
</tbody>
</table>

IS Document Type

When formatting this Transora message as an IData object, use the wm.ip.transora.recs.CatalogueResponse:CatalogueResponse_Envelope IS document type.

TDC Item Request TN Document Type

This TN document type provides the definition for a Transora Data Catalog Request message.

Identification Information

Root Tag: Envelope

<table>
<thead>
<tr>
<th>Identification Queries</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Envelope[0]/CatalogRequest[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/eb:MessageHeader[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:Service[0]/@eb:type</td>
<td></td>
</tr>
</tbody>
</table>
TN Document Types Provided with the Transora Module

Extraction Information

<table>
<thead>
<tr>
<th>Document Attribute</th>
<th>Extraction Queries</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>SenderID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:From[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>ReceiverID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:To[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>DocumentID</td>
<td>/Envelope[0]/CatalogRequest[0]/RequestHeader[0]/MessageIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>ConversationID</td>
<td>/Envelope[0]/CatalogRequest[0]/RequestHeader[0]/CorrelationIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>GTIN</td>
<td>/Envelope[0]/CatalogRequest[0]/RequestDetail[0]/TxnDetail[0]/ID1[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_ACTION</td>
<td>/Envelope[0]/CatalogRequest[0]/Payload[0]/PayloadEntry[0]/@type</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_TYPE</td>
<td>/Envelope[0]/CatalogRequest[0]/Payload[0]/PayloadEntry[0]/@operation</td>
<td></td>
</tr>
</tbody>
</table>

Identification Information

TDC GDSN Item Registry Response TN Document Type

This TN document type provides the definition for GS1 Global Registry item registration status responses routed by Transora to publishers.

Identification Queries

<table>
<thead>
<tr>
<th>Identification Queries</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Envelope[0]/CatalogResponse[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/eb:MessageHeader[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td>GDSNITEMREGISTRYRESPONSE</td>
</tr>
</tbody>
</table>
CHAPTER 6 Defining TN Document Types

Extraction Information

<table>
<thead>
<tr>
<th>Document Attribute</th>
<th>Extraction Queries</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>SenderID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:From[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>ReceiverID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:To[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>DocumentID</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseHeader[0]/MessageIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>GTIN</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/ID1[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_TYPE</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_ACTION</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Action[0]</td>
<td></td>
</tr>
</tbody>
</table>

IS Document Type

When formatting this Transora message as an IData object, use the

TDC GDSN Party Registry Response TN Document Type

This TN document type provides the definition for GS1 Global Registry party (GLN) contract confirmation responses routed by Transora to publishers.

Identification Information

Root Tag: Envelope

<table>
<thead>
<tr>
<th>Identification Queries</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>/Envelope[0]/CatalogResponse[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/eb:MessageHeader[0]</td>
<td></td>
</tr>
<tr>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/Type[0]</td>
<td>GDSNPARTYREGISTRYRESPONSE</td>
</tr>
</tbody>
</table>

Identification Queries:

- /Envelope[0]/CatalogResponse[0]
- /Envelope[0]/eb:MessageHeader[0]
- /Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/Type[0]
How Information Extracted from the Transora Messages Is Used

Extraction Information

<table>
<thead>
<tr>
<th>Document Attribute</th>
<th>Extraction Queries</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>SenderID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:From[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>ReceiverID</td>
<td>/Envelope[0]/eb:MessageHeader[0]/eb:To[0]/eb:PartyId[0]</td>
<td>GLN</td>
</tr>
<tr>
<td>DocumentID</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseHeader[0]/MessageIDHeader[0]</td>
<td></td>
</tr>
<tr>
<td>GTIN</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/ID1[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_TYPE</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Type[0]</td>
<td></td>
</tr>
<tr>
<td>TDC_DOC_ACTION</td>
<td>/Envelope[0]/CatalogResponse[0]/ResponseDetail[0]/TxnDetail[0]/Action[0]</td>
<td></td>
</tr>
</tbody>
</table>

IS Document Type

When formatting this Transora message as an IData object, use the `wm.ip.transora.recs.CatalogueResponse:CatalogueResponse_Envelope` IS document type.

How Information Extracted from the Transora Messages Is Used

As specified in the previous Extraction Information sections, the TN document types provided with the Transora Module extract the document attributes. The following describes the document attributes that are extracted and how they are used:

- **SenderID** and **ReceiverID**. The SenderID and ReceiverID are standard document attributes that Trading Networks uses to locate the profiles of the sender and receiver of a document. Trading Networks then uses information in the profile to determine processing information (for example, delivery method information). The process run time (PRT) also uses the SenderID and ReceiverID when executing a business process.


- **ConversationID**. The ConversationID is a standard document attribute. Extracting the ConversationID triggers Trading Networks to pass a document to the PRT so that it can be processed in a business process. Trading Networks passes the document to the PRT after it completes its own actions, which the processing rules identify. A process model determines the actions performed in a business process. For more information about process models, see Chapter 7, “Customizing a Process Model Template” in this guide and the *webMethods Modeler User’s Guide*. 


The ConversationID uniquely identify all documents that are associated with the same instance of a business process. All documents that belong to a specific instance of a business process must have the same conversation ID.

- **GTIN.** The GTIN is a custom attribute that the Transora Module adds to Trading Networks when you install the Transora Module. The Transora Module uses this document attribute to hold the value of the Global Trade Identifier Number of the Item in a TDC Request/Response message.

- **TDC_DOC_ACTION.** The TDC_DOC_ACTION is a custom attribute that the Transora Module adds to Trading Networks when you install the Transora Module. The Transora Module uses this document attribute to identify the action of the TDC message (Add/Modify/Append/Link) that is being sent to or received from Transora.

- **TDC_DOC_TYPE.** The TDC_DOC_TYPE is a custom attribute that the Transora Module adds to Trading Networks when you install the Transora Module. The Transora Module uses this document attribute to identify the type of TDC message (Item/Publication) that is being sent to or received from Transora.
Customizing a Process Model Template

- What is a Process Model? ......................................................... 62
- Process Model Provided with the Transora Module .................................. 63
- What You Need to Create to Customize the Process Model .. 66
- Customizing a Process Model to Meet Your Site's Needs .... 68
- Making a Process Model Available for Use ..................................... 71
What is a Process Model?

A process model is a diagram that represents a business process. You create process models using webMethods Modeler. The following figure illustrates a sample process model. For further explanation, see the text that follows the figure.

Process Model Example

A process model consists of:

- **Steps.** The basic unit of work.
- **Transitions.** The lines between steps that indicate the execution order of steps.
- **Groups.** Clustering steps to represent different organizational boundaries.
- **Annotations (notes or text).** The labels, notes, and explanatory text.

The steps and transitions in a process model determine how the process run time (PRT) conducts a business process. For information about the process run time, see “Run-Time Architecture and Components” on page 16 in this guide and the webMethods Modeler User’s Guide.

The Transora Module provides a process model template that you can use to build your own process models for the various Transora business cases you want to implement. You customize a webMethods-provided process model template in webMethods Modeler by specifying how the process model is to interact with your back-end systems. You do this by editing the services that the steps of the process model invoke, specifying inbound and outbound mapping services, and enhancing the provided error and exception handling services, if necessary.
For more information about process models and webMethods Modeler, see the *Getting Started with webMethods Business Partner Management* and *webMethods Modeler User’s Guide*.

Note: For simple request/response scenarios, webMethods recommends that you define the actions you want to perform using only Trading Networks processing rules, not a process model. For more complex scenarios, use a process model to define a business process.

### Process Model Provided with the Transora Module

The Transora Module provides the following process model template to assist you in creating your own process models for Transora transactions:

- **TDC_Item_Message_Model.model**, which represents the Transora Item (Add/Modify/Link/Publish) business process. This business process begins when the Integration Server receives an internally-formatted document that represents an item request (for example, from a back-end system).

You can find the process model template in the following location, where `webMethods\IntegrationServer` is the directory in which the Integration Server is installed:

`webMethods\IntegrationServer\packages\WmTransoraSamples\pub\demo`

### Steps in the Process Model

The process model provided with the Transora Module define the steps to take during a business process to send a Transora request message to Transora.

The following diagram shows a process model that represents a business process for sending a Transora request message. See the table that appears after the diagram for more information about each step and what you will need to customize for a step, if necessary. For more information about how to customize this process model to meet your site’s requirements, see “What You Need to Create to Customize the Process Model” on page 66 and “Customizing a Process Model to Meet Your Site’s Needs” on page 68.
### Step 1: Wait for Internal Request

**An instance of this business process begins when the Integration Server receives an internally-formatted request document (for example, from a back-end system). The first step of the business process waits for this internal document.**

**Customization Required:** Update the inputs to this step to specify the TN document type or IS document type that represents the internally-formatted document from the back-end system. For more information, see “Customizations for the Wait for Internal Request Step” on page 66.

### Step 2: Map to TDC Request

**This step performs the outbound mapping; that is, it maps data from the internally-formatted request document to Transora-formatted request message. To do so, the step invokes an outbound mapping service that you create.**

**Customization Required:** Create the outbound mapping service. For more information, see “Customizations for the Map to TDC Request Step” on page 67.

**Update the process model step to reference the outbound mapping service that you create.**

### Step 3: Send TDC Request

**This step invokes the `wm.ip.Transora.cm.handlers:send` service to send the Transora-formatted request message to Transora. How the `wm.ip.Transora.cm.handlers:send` service delivers this document is governed by parameters you specified in the TPA. For more information, see “Parameter Settings for the TPA Data” on page 45 in Chapter 5, “Defining a Trading Partner Agreement”.**

**Customization Required:** Ensure that your outbound mapping service places the Transora-formatted request message as an XML String in the `outboundxml` pipeline variable. The `outboundxml` variable is an input to the `wm.ip.Transora.cm.handlers:send` service.
### Process Model Provided with the Transora Module

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
<th>Customization Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td><strong>Wait for MDN</strong> This step waits for a synchronous MDN that is sent from Transora in response to the Transora request message. The input to the step is set to wait for EDIINT MDN TN document type.</td>
<td>This step requires no customization if you defined the parameters in the TPA to indicate you want Transora to return a synchronous MDN. However, you will need to alter the process model if you defined the parameters to indicate either that you did not want an MDN or if you want an asynchronous MDN. For more information about setting these parameters, see “Parameter Settings for the TPA Data” on page 45 in Chapter 5, “Defining a Trading Partner Agreement”.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Verify MDN</strong> This step invokes the <code>wm.ip.transora.transport.EDIINT.service:verifyMDN</code> service to check the processing status of the EDIINT MDN message. It sets its <code>status</code> output accordingly either returning SUCCESS or FAILURE.</td>
<td>This step requires no customization if you receive a synchronous MDN. However, you will need to alter the process model if you either do not receive an MDN or if you receive an asynchronous MDN.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Wait for TDC Response</strong> This step waits for the Transora response message from Transora. The input to the step is set to wait for a TN document type that defines the Transora response message, which is a TN document type provided with the Transora Module.</td>
<td>This step requires no customization.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Verify Response</strong> This step verifies the Transora response message that Transora returns. To do so, the step invokes a service that you create.</td>
<td>Create a service to verify the Transora response message. Your service should use an output variable named <code>status</code> to return a value of either SUCCESS or FAILURE. For more information, see “Customizations for the Verify Response Step” on page 67. Update the process model step to reference the service that you create.</td>
</tr>
<tr>
<td>8</td>
<td><strong>Process Response and End</strong> This step returns a document to the back-end system to send the back-end system the outcome of the Transora request. To do so, the step invokes a service that you create.</td>
<td>Create a service that sends an outcome document back to your back-end system. For more information, see “Customizations for the Process Response and End Step” on page 68. Update the process model step to reference the service that you create.</td>
</tr>
</tbody>
</table>
What You Need to Create to Customize the Process Model

This section describes the items you need to create before you can customize the steps in a process model for a Transora business process.

Customizations for the Wait for Internal Request Step

For this first step of the business process, you need to create a document type that defines the internal request document that your back-end system will send to your Integration Server. To do so, create either a TN document type or an IS document type for a publishable IS document.

<table>
<thead>
<tr>
<th>Document type to Create</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>TN document type</td>
<td>Using the Trading Networks Console, define a TN document type for the internal request document. When you define the TN document type, be sure to extract the ConversationID system attribute. For more information, see the webMethods Trading Networks User’s Guide and “Defining TN Document Types for Back-End System Documents” on page 51 in this guide. Additionally, you might want to create a Trading Networks processing rule for the internal request document. If you do not create a processing rule, Trading Networks will use its default processing rule, which will ignore the document. However, because the TN document type instructs Trading Networks to extract a ConversationID, Trading Networks will pass the document to the process run time (PRT) so that it can be processed as part of a business process.</td>
</tr>
<tr>
<td>IS document type</td>
<td>Using webMethods Developer, create an IS document type that defines the structure of the internal request document. Be sure to update the properties to indicate that the document is publishable. For more information, see the webMethods Developer User’s Guide and the Publish-Subscribe Developer’s Guide.</td>
</tr>
</tbody>
</table>
Customizations for the Map to TDC Request Step

For this step you need to create a service to form the Transora request message. This service is referred to as an outbound mapping service. The service forms the Transora request message by mapping data from the internal request document that was received from the back-end system to the Transora request message that is to be sent to Transora.

The outbound mapping service should perform the following actions:

- **Convert the internal request document from the back-end system to an IData object.**

  If the internal request document is received from Trading Networks, you must request the content from Trading Networks, and then convert the content to an IData object. For more information about how to perform this action, see information about getting data from an external partner into a business process in the *Getting Started with webMethods Business Partner Management*.

  If the internal request document is a published IS document, it already will be in the business process pipeline as an IData object. For more information about getting data from internal systems into a business process, see the *Getting Started with webMethods Business Partner Management*.

- **Map data from the IData object for the internal request document to Transora Payload**

  (wm.ip.transora.recs.CatalogueRequest:docType_Payload). The Transora Module provides IS document types that define the payload of each Transora request messages.

  For an example of an outbound mapping service, see the wm.ip.transora.samples.service:mapOutboundDoc service, which is included in the WmTransoraSamples package.

Customizations for the Verify Response Step

For this step you need to create a service to verify the Transora response message. The service should check the userStatus in the response bizdoc and accordingly return SUCCESS or FAILURE in the TDCResponseStatus output variable.

The service should perform the following actions:

- **Invoke the wm.tn.doc:view service to obtain the response bizdoc and check for the userStatus variable.**

- **Based on the string PRODUCTION_SUCCESS, set the TDCResponseStatus variable to SUCCESS or FAILURE.**

  For an example of a service that verifies a Transora response message, see the wm.ip.transora.samples.service:verifyResponse service, which is included in the WmTransoraSamples package.
Customizations for the Process Response and End Step

For this step you need to create a service to send a document back to your back-end system that provides the back-end system with the outcome of its request.

The service should perform the following actions:

- Form the internal response document in the format that the back-end system expects.
- Send the internal response document to the back-end system.

For an example of a service that verifies a Transora response message, see the `wm.ip.transora.samples.service:processResponse` service, which is included in the `WmTransoraSamples` package.

Customizing a Process Model to Meet Your Site’s Needs

You customize a process model using webMethods Modeler. In webMethods Modeler, you import a process model provided by the Transora Module and make your changes, saving the process model with a new name.

To customize a Transora process model

1. From webMethods Modeler, import the template you want to customize:
   a. Select File ▶ Import.
   b. In the Select import file dialog box, navigate to `webMethods6\IntegrationServer\packages\WmTransoraSamples\pub\demo`, and select the TDC_Item_Message_Model process model.
   c. Click Open.
   d. In the Message dialog box, click OK.
   
   webMethods Modeler displays the process model.

2. Save the process model with a new name so that all customizations are saved with the new name. To do so:
   a. Select File ▶ Save as New Version.
   b. In the Choose new version name dialog box, enter a new name for the process model.
   c. Click OK.

3. Customize the **Wait for Internal Request** step to subscribe to the TN document type or IS document type for the internal request document. For more information, see “Customizations for the Wait for Internal Request Step” on page 66. To define the subscription:
a Right-click the **Wait for Internal Request** step, and select **Inputs/Outputs**.
b In the **Inputs and Outputs for step “Wait for Internal Request”** dialog box, click **Add Subscription**.
c In the **Add Subscription** dialog box, navigate to and select the TN document type or IS document type for the internal request document.
d Click **Add Document**.
e If you selected a TN document type, in the **Set TN Roles** dialog box:
   - For the **Sender Role**, select **Backend**.
   - For the **Receiver Role**, select **Transora**.
   Click **OK** to close the **Set TN Roles** dialog box.
f Click **OK**.

4 Customize the **Map to TDC Request** step to add the outbound mapping service to invoke. For more information, see “Customizations for the Map to TDC Request Step” on page 67. To add the service:

a Right-click the **Map to TDC Request** step, and select **Select service to invoke**.
b In the **Select service to invoke from server:DesignServer** dialog box, navigate to and select the outbound mapping service that you want to use.
c Click **Select**.

5 Update the handling of MDNs that Transora returns, if necessary:
   - If you want Transora to return a synchronous MDN, no customization is required.
   - If you want Transora to return an asynchronous MDN:
     - Update the TPA variables to indicate that you want an asynchronous MDN. For more information, see “Parameter Settings for the TPA Data” on page 45.
   - If you do not want Transora to return an MDN:
     - Update the TPA variables to indicate that you do not want an MDN. For more information, see “Parameter Settings for the TPA Data” on page 45.
     - Remove the **Wait for MDN** and **Verify MDN** steps from the process model and update the transitions accordingly.

6 Customize the **Verify Response** step to add the service to invoke. For more information, see “Customizations for the Verify Response Step” on page 67. To add the service:

a Right-click the **Verify Response** step, and select **Select service to invoke**.
b In the **Select service to invoke from server:DesignServer** dialog box, navigate to and select the service that you want to use.
c Click **Select**.
7 Customize the **Process Response and End** step to add the service to invoke. For more information, see “Customizations for the Process Response and End Step” on page 68. To add the service:

a Right-click the **Process Response and End** step, and select **Select service to invoke**.

b In the **Select service to invoke from server:DesignServer** dialog box, navigate to and select the service that you want to use.

c Click **Select**.

8 Save the process model by selected **File ▶ Save**.
Making a Process Model Available for Use

After you customize a process model, perform the following actions to make the process model available for selection by the PRT so that the process model can be used for a Transora business process.

To make a process model available for use

1. Generate the process model.
   
   From webMethods Modeler, generate the process model, which causes webMethods Modeler to create the run-time objects on the Integration Server. The run-time objects that webMethods Modeler generates are services, triggers, etc. The process model is simply a design-time tool that describes the business process. The run-time objects are required to execute a business process that uses the process model. For more information about how to generate a process model, see the webMethods Modeler User’s Guide.

Important! You may receive the following error message if the EDIINT MDN TN document type in your Trading Networks database is different than the one in the Trading Networks database that was used when the process model was created. The process model is still referring to the old document ID. For information about how to resolve the error, see “Resolving Document Not Found Errors” on page 88.

2. Update the process model for monitoring.
   
   The process model resides in the webMethods Modeler repository. However, to execute and monitor business processes that use the process model, webMethods Monitor also must have a copy of the process model. From webMethods Modeler, you update the process model for monitoring to have a copy of the process model.
information to be moved to webMethods Monitor. For more information about updating a process model for monitoring, see the webMethods Modeler User’s Guide.

3 Enable the process model.

After you update a process model for monitoring, webMethods Monitor has a copy, but the process model is disabled. Before the PRT can select the process model, you must enable it. For instructions for enabling a process model, see the webMethods Integration Platform Logging and Monitoring Guide.
webMethods Transora Module Services

- WmTransora Package Services ........................................... 74
- wm.ip.transora.cm.handlers ........................................... 74
- wm.ip.transora.service .................................................. 75
- wm.ip.transora.transport.edlint.service ......................... 78
WmTransora Package Services

The WmTransora package contains services used to implement and support the Transora-compliant functionality of the webMethods Transora Module, such as registering an item to the Transora GLOBAL registry and authorizing an item that a supplier has registered to a retailer. Only exposed services are documented.

wm.ip.transora.cm.handlers

wm.ip.transora.cm.handlers:send

Business processes invoke this service to send an XML document to Transora.

Based on the transport specified in the input parameters of the service, the appropriate send service is invoked. If the transport is:

- EDIINT, the business process invokes the wm.ip.Transora.transport.EDIINT:send service

**Input Parameters**

- **Payload**: Payload of the Transora Envelope that the user has mapped in the outbound Map.
- **SenderID**: Internal ID of the sender.
- **ReceiverID**: Internal ID of the receiver.
- **conversationID**: Conversation id for the process. If null, a new conversation id is created.
- **TPA**: Trading Partner Agreement between the sender and receiver.

**Output Parameters**

None.
wm.ip.transora.service

wm.ip.transora.service:createEnvelopeFromPayload

Creates a Transora Catalog Request Envelope from the internal payload. The output envelope conforms to the IS document type `wm.ip.transora.recs.CatalogueRequest:CatalogueRequest_Envelope`.

**Input Parameters**

<table>
<thead>
<tr>
<th>TPA</th>
<th><strong>Document</strong> Information about the current business process, including the sender profile, receiver profile, conversation ID, and input parameters to the business process. This IS document must conform to the IS document type <code>wm.ip.transora.cm.recs:TPAInfo</code>.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Payload</strong></td>
<td><strong>Document</strong> Data that you are sending to the partner. This IS document must conform to the IS document type <code>wm.ip.transora.recs.CatalogueRequest:record_Payload</code>.</td>
</tr>
</tbody>
</table>

**Output Parameters**

| **Envelope**      | **Document** The Transora Catalog Request Envelope. This IS document conforms to the IS document type `wm.ip.transora.recs.CatalogueRequest:CatalogueRequest_Envelope`. |

wm.ip.transora.service:getResponseDetails

Retrieves the result code, result message, and reason from the response envelope.

**Input Parameters**

<table>
<thead>
<tr>
<th>TPA</th>
<th>Trading Partner Agreement between the sender and receiver.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CatalogueResponse_Envelope</strong></td>
<td>Catalogue Response for Transora version 4.2.</td>
</tr>
</tbody>
</table>

**Output Parameters**

<table>
<thead>
<tr>
<th><strong>OverallResultCode</strong></th>
<th><strong>String</strong> The overall result code in the response received from partner.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OverallResultMessage</strong></td>
<td><strong>String</strong> The overall result message in the response received from partner.</td>
</tr>
<tr>
<td><strong>reason</strong></td>
<td><strong>String</strong> String built from the ResultCode and ResultMessage tags in the response document.</td>
</tr>
</tbody>
</table>
**wm.ip.transora.service:processRequest**

A back-end system invokes this service to submit an internally-formatted document to Trading Networks. This service invokes the `wm.ip.cm:processDocument` service to start a business process.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cid</code></td>
<td>String</td>
<td>The conversation ID that identifies the business process to which you are sending the document.</td>
</tr>
<tr>
<td><code>record</code></td>
<td>Document</td>
<td>The document that you are sending to the process run time (PRT) to start a business process. To start a business process with Transora the document must conform to the IS or TN document type.</td>
</tr>
</tbody>
</table>

**Output Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BizdocID</code></td>
<td>String</td>
<td>The internal ID that Trading Networks generated for the document that was submitted to Trading Networks.</td>
</tr>
</tbody>
</table>

**wm.ip.transora.service:processInit**

Performs initialization tasks. Call this service in the first wait step when triggering a process model.

**Input Parameters**

None.

**Output Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TPARecord</code></td>
<td></td>
<td>Trading Partner Agreement between the sender and receiver.</td>
</tr>
</tbody>
</table>

**wm.ip.transora.service:recordToOutboundXML**

Converts an IS document into an XML String, and then converts the XML String into node object.

**Input Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boundnode</code></td>
<td>Document</td>
<td>The outbound data to convert to an XML String, then a node object. This IS document must conform to the IS document type <code>wm.ip.transora.recs.CatalogueRequest:CatalogueRequest_Envelope</code>.</td>
</tr>
<tr>
<td><code>nsDecls</code></td>
<td>Document</td>
<td>An IS document that declares the namespaces associated with the namespace prefixes that you used in the key names in <code>boundNode</code>. In <code>nsDecls</code>, set the key names to the prefixes that you are used in <code>boundNode</code>. Then set the value of each key to the namespace that the prefix.</td>
</tr>
</tbody>
</table>
Output Parameters

- recordName: **String** The fully-qualified name of an IS document type that describes the format of the output document being created from data in the input `boundNode`. The IS document type should:
  - Include elements that might not be present in `boundNode`.
  - Specify the order in which elements are to appear in the output document.

- dtdHeaderInfo: **Document** An IS document that defines the contents of the DOCTYPE header that is to be inserted into the output document.

**Input Parameters**

- `xmldata`: **String** The outbound document in XML format.
- `node`: **Object** The parsed outbound XML object, which is an instance of `com.wm.lang.xml.Node`.

---

**wm.ip.transora.service:validateOutboundXML**

Validates a document against an IS document type or IS schema.

**Input Parameters**

- `node`: **Object** The outbound document to be validated. Pass the service the parsed outbound XML object, which must be an instance of `com.wm.lang.xml.Node`.
- `schemaName`: **String** (optional) The fully-qualified name of the IS schema that describes the format of the output document, including elements that might not be present in the document represented by `node`. It also specifies the order in which elements are to appear in the output document.

  You must specify either `schemaName` or `recordName`, but not both. Specify `schemaName` if you want to validate using an IS schema.

- `recordName`: **String** (optional) The fully-qualified name of an IS document type that describes the format of the output document, including elements that might not be present in the document represented by `node`. It also specifies the order in which elements are to appear in the output document.

  You must specify either `schemaName` or `recordName`, but not both. Specify `recordName` if you want to validate using an IS document type.

- `TPA`: **Document** Information about the current business process including the sender profile, receiver profile, conversation ID, and input parameters to the business process. This IS document must conform to the IS document type `wm.ip.transora.cm.recs:TPAInfo`. 

---

**Output Parameters**

- `xmlData`: **String** The outbound document in XML format.
- `node`: **Object** The parsed outbound XML object, which is an instance of `com.wm.lang.xml.Node`.
Output Parameters

isValid String The validation result. The output parameter isValid will have one of the following values:

- true if the validation was successful.
- false if the validation failed. If isValid is false, reason contains the errors.

reason String (optional) The errors in the document. The output parameter reason is returned if the validation failed, that is, if isValid is false.

Note

For more information about how the document is validated, see the description of the pub.schema.validate service in the webMethods Integration Server Built-In Services Reference.

wm.ip.transora.transport.ediint.service

wm.ip.transora.transport.ediint.service:verifyMDN

Business processes invoke this service to verify an EDIINT MDN message.

Input Parameters

documents Document Information about documents received in the business process. For each document, the key is the event name (which depends on the process model) and the value is the internal ID that Trading Networks generated for document.

signals Document Information about signals received in the business process. For each signal, the key is the signal name (which depends on the process model) and the value is the signal.

roles Document Information about partners bound into roles in the business process. For each role, the key is the role name (which depends on the process model) and the value is the internal partner ID that Trading Networks generated for partner when its profile was defined.

parameters Document Additional state-type information for the business process.

Output Parameters

status String Status of the MDN. It sill have one of the following values:

- SUCCESS if the MDN represent a successful EDIINT message delivery
- FAILURE if the MDN represents error in EDIINT message delivery
webMethods Transora Module Sample

- About the webMethods Transora Module Sample ............................................. 80
- Before You Can Use the Sample ................................................................. 80
- What to Do to Run the Sample ................................................................. 80
- Run the Sample ......................................................................................... 86
- Resolving Document Not Found Errors ..................................................... 88
About the webMethods Transora Module Sample

The webMethods Transora Module sample illustrates a Transora business process. The business process starts when a back-end system sends a business document that represents a Transora request. The business process incorporates the request from the back-end system into a Transora-formatted request message and sends the Transora request message to Transora. The business process handles the receipt of both a synchronous EDIINT MDN and the Transora response message, which Transora sends in response to the Transora request message. Then, the business process creates a response document, which contains the outcome of the request, and sends the document to the back-end system.

Before You Can Use the Sample

This sample assumes that you are familiar with the Integration Server, Trading Networks, webMethods Modeler, webMethods Monitor, and Transora.

Make sure you have:

- Installed the Integration Server, Trading Networks, the WmEDIINT package of the webMethods EDI Module, webMethods Modeler, webMethods Monitor, and webMethods Developer. For more information, see the webMethods Integration Platform Installation Guide.
- Installed the webMethods Transora Module. For more information, see Chapter 2, “Installing webMethods Transora Module 6.1.1” in this guide.

What to Do to Run the Sample

To run the sample, you need to perform the following steps after you install the items identified in the previous section.

Step 1: Define the Trading Networks Profiles.
Step 2: Import the process models for the sample scenarios into webMethods Modeler.
Step 3: Generate the process models, and update them for monitoring.
Step 4: Enable the process models.

Important! Before going into production, you should delete or disable the WmTransoraSamples package.
Step 1: Define the Trading Networks Profiles

For this sample, you need a profile for your enterprise and a profile for Transora. To create the profiles, you use the Transora Module screens, and the Transora Module will create the profiles in Trading Networks.

1. From the Server Administrator, select Transora from the Adapters menu in the navigation panel. This will open a new Browser window.

2. Click Sample Setup. The Transora Sample Setup panel appears.

To create the profiles for the Transora Module sample

1. From the Server Administrator, select Transora from the Adapters menu in the navigation panel. This will open a new Browser window.

2. Click Sample Setup. The Transora Sample Setup panel appears.
Complete the **Enterprise** column. All fields are required.

<table>
<thead>
<tr>
<th>Field</th>
<th>Specify</th>
</tr>
</thead>
<tbody>
<tr>
<td>External ID</td>
<td>DUNS: The D-U-N-S number for your corporation.</td>
</tr>
<tr>
<td></td>
<td>GLN: The Global Location Number for your corporation.</td>
</tr>
<tr>
<td></td>
<td>AS2-ID: The type of identification you use for EDIINT AS2 messages, for example, your GLN or DUNS number. This value should correspond to the values in the 'AS2-To:' and 'AS2-From:' fields in the Header of the EDIINT messages. Typically, AS2-ID and GLN are the same value.</td>
</tr>
<tr>
<td>Delivery Method</td>
<td>Host: localhost, or the host name or IP address of your Integration Server</td>
</tr>
<tr>
<td>Primary HTTP</td>
<td>Port: 5555, or the port on which your Integration Server listens for incoming requests.</td>
</tr>
<tr>
<td></td>
<td>Location: /invoke/wm.EDIINT/receive</td>
</tr>
<tr>
<td>Extended Field</td>
<td>AS2 MDN: /invoke/wm.EDIINT/receive</td>
</tr>
<tr>
<td></td>
<td>Location: /invoke/wm.EDIINT/receive</td>
</tr>
<tr>
<td></td>
<td>Transora User ID: The Batch User ID that Transora provided for your corporation.</td>
</tr>
</tbody>
</table>
What to Do to Run the Sample

<table>
<thead>
<tr>
<th>Field</th>
<th>Specify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Info Certificate</td>
<td>Your digital certificate, which contains your public key. If you do not have a public certificate, use webMethods6\IntegrationServer\packages\WmTransoraSample\pub\demo\SenderCert.der.</td>
</tr>
<tr>
<td></td>
<td><strong>Important!</strong> Transora must have this certificate installed on its system, and the certificate should be associated with your corporation.</td>
</tr>
<tr>
<td>CA Certificate</td>
<td>The digital certificate of the certifying authority (CA), which signed your above public certificate. If you do not have the CA certificate, you can export it from your public certificate.</td>
</tr>
<tr>
<td></td>
<td>If you used the certificate provided with the Transora Module sample for your public certificate, use the webMethods6\IntegrationServer\packages\WmTransoraSample\pub\demo\SenderCert.der.</td>
</tr>
<tr>
<td>Private Key</td>
<td>The private key that corresponds to the public key in your digital certificate. If you used the certificate provided with the Transora Module sample for your public certificate, use the webMethods6\IntegrationServer\packages\WmTransoraSample\pub\demo\SenderPrivateKey.der.</td>
</tr>
</tbody>
</table>

- **Note:** The information in the Transora column is completed for you. You should not change it.

4 Record the values you specified for the following fields in the Enterprise column. Send an e-mail message with this information to Transora so that Transora can set up your profile in its system.

- GLN
- Host
- Location
- Password
- AS2-ID
- Port
- User ID

5 Click **Setup Profiles**. In Trading Networks, the Enterprise profile is updated and the Transora_QA profile is added.

- **Important!** This screen adds the profile only this first time. If you need to make changes to a profile, use the Trading Networks Console.
6 Record the user name and password of the Integration Server user account that Trading Networks creates for Transora_QA.

The password is returned only the first time the user account is created in Integration Server for a Trading Networks profile. Subsequent creation of profiles with the same D-U-N-S number will result in a dummy password, “dummy,” being returned.

**Step 2: Import the Process Model**

The Transora Module sample provides the following process model:

- **TDC_Item_Message_Model.model**. Use this process model for the Add Items, Modify Items, and Link Items samples.

To import the process model for the sample into webMethods Modeler

1 Start webMethods Modeler if it is not already running. When you start webMethods Modeler, connect it to the Design Server. Be sure that the webMethods Modeler’s Repository Server has been started before you start webMethods Modeler.

2 Select **File ➤ Import**.

3 In the **Select import file** dialog box, navigate to the following directory and select the process model you want to import.

   `webMethods6\IntegrationServer\packages\WmTransoraSamples\pub\demo`

4 Click **Open**, and then click **OK**.

5 Repeat steps 2 through 4 for each process model.

**Step 3: Generate the Process Models and Update Them for Monitoring**

To create the run-time objects for a process model, you generate the process model. Then, you update the model for monitoring to create a copy of the process model in the webMethods Monitor environment.

To generate the process models and update them for monitoring

1 From webMethods Modeler, click **File ➤ Open**.

2 Under **Business Processes**, select the process model you want to generate, and click **Open**.
3 Select **Tools ▶ Generate Business Process**. webMethods Modeler displays the **Implementation Generation Messages** screen. webMethods Modeler begins to generate the steps in the process model and displays a series of messages in the **Implementation Generation Messages** screen.

**Important!** If you receive the following error while generating the process model, you must resolve it before continuing. This error message occurs when the EDIINT MDN TN document type in your Trading Networks database is different than the one in the Trading Networks database that was used when the process model was created. The process model is still referring to the old document ID. For information about how to resolve the error, see “Resolving Document Not Found Errors” on page 88.

4 When you see a message that the process model generated successfully, click **Close** to close the **Implementation Generation Messages** screen.

5 Click **Tools ▶ Update Model for Monitoring**.

6 Repeat steps 1 through 5 for each process model.
Step 4: Enable the Process Models

You manually enable the process models in webMethods Monitor. The process models are not enabled automatically.

**Note:** To enable a process model, you must have started the Broker Server service prior to starting the Integration Server. If you cannot enable a process model, shut down the Integration Server, restart the Broker Server service, and then restart the Integration Server.

To enable the process models

1. Open the webMethods Monitor user interface if it is not already open. You can access the webMethods Monitor user interface with the following URL:

   \[http://<IntegrationServerHost>:<IntegrationServerPort>/WmMonitor\]

   For example, \[http://localhost:5555/WmMonitor/\].

2. Select **Process Models** from the **Processes** menu of the navigation panel.

3. In the **Process Models** table, in the **Enabled** column, click **No** in the row for the process model that you want to enable.

4. In the message box, click **OK**.

5. Above the **Process Models** table, click **Refresh this page**. In the **Enabled** column and in the row for the process model you enabled, **Yes** now displays.

6. Repeat steps 3 through 5 for each process model.

Run the Sample

You run the samples from the home page of the Transora Module sample package.

To run the sample

1. From the Server Administrator, select the **Transora** from the **Adapters** menu in the navigation panel. This will open a new Browser window.
2. Select the link in the navigation panel that corresponds to the sample that you want to run. For example, if you want to run the Add Items sample, click Add Items Sample.

3. Update the From (Sender GLN) in the Sync Items section of the screen to match your corporation’s GLN.

4. Change the Global Trade Item Number in the Item - Global Attributes section of the screen to a unique 14-digit number that is used every time a new item is sent to Transora.
5 Make any other changes you would like to the fields on the screen.
6 Click **Send Items**. You will receive the following in response:

```
Transora > Add Items Sample

Item submitted to TN successfully
```

7 Open the webMethods Monitor user interface if it is not already open. You can access the webMethods Monitor user interface with the following URL:

http://<IntegrationServerHost>:<IntegrationServerPort>/WmMonitor

For example, http://localhost:5555/WmMonitor/.

8 Select **Recent Activity** from the navigation panel.

You should see the TDC Item Message process model in progress.

**Resolving Document Not Found Errors**

While generating the process model you may receive the following error message:

```
Generating implementation for "TDC_Item_Message_Model"

Error: Document "0agU6r0uuuc1rikip00000005" does not exist...

"TDC_Item_Message_Model" generation failed with 1 errors.
```

This error occurs when the internal ID for the EDIINT MDN TN document type in your Trading Networks database is different from the one in the Trading Networks database that was used when the process model was created. You must resolve the error before you can complete generating the process model.

To resolve this error:

1 Click **Close** to close the **Implementation Generation Messages** screen if it is still open.
2 Update the subscription for the EDIINT MDN TN document type by doing the following:
   a Select the **Wait for MDN** step in the process model, and right-click.
   b Select **Input/Output**.
   c In the Inputs and Outputs dialog for the step, delete the input document **EDIINT MDN**.
   d Click **Add Subscription**.
   e In the Add Subscription dialog, choose **EDIINT MDN** from the **Trading Networks Documents** folder. Then click **Add Document**.
   f When prompted for roles, specify **Transora** for **Sender Role** and **Focal role** for **Receiver Role**, and then click **OK**.

3 Reset the properties for the **Wait for MDN** step by doing the following:
   a Select the **Wait for MDN** step in the process model, and right-click.
   b Select **Properties**.
c In the Properties dialog for the step, set Join Type to Complex.

![Properties dialog]

- Label: Wait for MDN
- Description: Edit
- Unique ID: N/A
- Logical Server: Design Server
- Folder: TDC_Temp_Message_Model.Design_Server
- Generated Flow Name: Wait_for_MDN
- Service to Invoke: wm.ip.transora.samples.service:waitForMDN
- Correlation Service: wm.ip.transora.samples.service.correlationService

**Transitions**
- Retry Count: 1
- Join Type: Complex
- Timeout Value: OR
- Publish/Subscribe Filter: XOR
- Inputs/Outputs: AND
- Enable Resubmission: Complex

**Transitions/Subscriptions**

- Left Parent: Transition-from-N7 Send TN...
- Transitions/Subscriptions: wm.ip.transora.comm:recs:Deliver
- Right Parent: wm.ip.transora.comm:recs:Deliver
- Generated Join Expression:

```
|Transition-from-N7 and (wm.ip.transora.comm:recs:DeliveryFailure xor 0xg06c03e110b5d0000085) |
```

d In the Complex Join Editor dialog box, select the last condition, and update the Transitions/Subscriptions column to select the internal document ID of the EDIINT MDN TN document type. It will be the only internal ID number in the list.

![Complex Join Editor]

- Left Parent: { wm.ip.transora.sql:oid:... }
- Transitions/Subscriptions: (wm.ip.transora.comm:recs:DeliveryFailure xor 0xg06c03e110b5d0000085)
- Right Parent: wm.ip.transora.comm:recs:Deliver
- Generated Join Expression:

```
|Transition-from-N7 and (wm.ip.transora.comm:recs:DeliveryFailure xor 0xg06c03e110b5d0000085) |
```

e Click Generate. This will update the generated join expression to use the internal ID of the EDIINT MDN TN document type in your Trading Networks database.
1. Click **OK** to close the *Complex Join Editor* dialog box.

4. Select **File > Save** to save the process model.

5. Re-generate the process model again.
Index

A
Access Control Lists (ACLs)
  updating for wm.EDIINT:receive service 33
ACLs. See Access Control Lists
Agreement ID field
  in TPA 44
AltSenderGLN parameter
  in TPA data 45
AS2MDNURL field
  in partner profile 41
asynchronous MDNs
  URL for Transora 41

B
batch user ID
  in Enterprise profile 39
batch user ID in a profile 36
business documents
  extracting attributes 52
  identifying to TN 52
  pre-processing actions 52
  set up processing 32
  XML namespaces 52

C
CA Chain field
  in Enterprise profile 39
  in partner profile 42
Certificate field
  in Enterprise profile 39
  in partner profile 42
conventions used in this document 7
Corporate tab
  Enterprise profile fields 37
  partner profile fields 40
Corporation Name field
  in Enterprise profile 37
  in partner profile 40

D
Decrypt field
  in Enterprise profile 39
  in partner profile 41
default TPA
  defined 44
Delivery Method tab
  Enterprise profile fields 38
  partner profile fields 41
design-time
  architecture and components 13
digital certificate
  adding
    Enterprise profile 39
digital certificate chain
  in Enterprise profile 39
  in partner profile 42
document attributes, extracted 59
Document ID does not exist, error message 71, 85, 88
document types, TN
  defining for back-end systems 51
  described 50
  extracting information 59
  in back-end system 50
  pre-processing options 50
  provided by Transora Module 53
TDC Accept Reject Response 55
TDC GDSN Item Registry Response 57
TDC GDSN Party Registry Response 58
TDC Item Request 56
TDC Item Response 53
TDC Publication Response 54
Transora response messages 51
documentation
  additional 8
  conventions used 7
  feedback 8
  using effectively 7
DTD parameter
in TPA data 46

E
EDIINT
  encryption types and key lengths 18
  extended profiles 36
  receipts (MDNs) 18
  support for 11
  webmethods EDI Module at design-time 18
  enabling a process model 86
Encrypt sub-tab
  in partner profile 42
encryption
  setting for outbound Transora message 46
  types and key lengths 18
Enterprise profile
  defining 37
  enabling 37
error while generating process model 88
Execute ACL
  updating 33
Extended Fields tab
  Enterprise profile fields 39
  partner profile fields 41
External ID Type field
  in Enterprise profile 38
  in partner profile 40
External ID Types
  described 36, 37
  in partner profile 40
external ID types
  adding 38
Extract tab
  in Document Type Details screen (TN) 52

G
GDSN
  Transora item synchronization process 20
  generating process model 71
  for sample 84
Global Data Synchronisation Network. See GDSN

H
Host field
  in Enterprise profile 38
  in partner profile 41

I
Identify tab
  in Document Type Details screen (TN) 52
inbound message actions
  supported 11
installing Transora Module 27
IS document type
  customizing for process model 66
  defining 31
  identifying TPA data 45
IS Document Type field
  in TPA 45
IS. See webMethods Integration Server
item correction 11
item synchronization, Transora
  process 20
item synchronization, Transora Module
  process 22

L
Location field
  in Enterprise profile 38
  in partner profile 41

M
Map to TDC Request process model step
  customizing 67
described 64
MDNs (Message Disposition Notifications)
  requesting outbound message response 47
  requesting signed 47
  types 18
URL for asynchronous 41
message actions
  supported
    inbound 11
    outbound 11
messages
  processing Transora Item Confirmation 23
  processing Transora Item Request/Response 22
MessageType parameter
  in TPA data 46

N
Namespace tab
  in Document Type Details screen (TN) 52

O
Options tab
  in Document Type Details screen (TN) 52
outbound mapping service
  creating in process model step 67
  example location 67
outbound message actions
  supported 11
outbound messages
  requesting MDN 47
  requesting signed MDN 47
  signing/encrypting before sending 46
  TN document types 50
  TPA data parameters 45
  unsupported 11
  validating before sending 46
outboundxml pipeline variable 64

P
packages
  Transora Module
    WmIPRoot 12
    WmTransora 12
    WmTransoraDocuments 12
    WmTransoraSamples 13
  WmEDIINT 33
  WmTransora 74
partner profile
  defining 39
  enabling 39
Partner Type field
  in Enterprise profile 37
  in partner profile 40
payload, Transora
  as input to service 74
  in processing rules 32
  in service 75
  in TPA data 46
  mapping 67
Port field
  in Enterprise profile 38
  in partner profile 41
pre-processing options for TN document types 50
Primary HTTP delivery method
  in Enterprise profile 38
  in partner profile 41
private key
  adding in Enterprise profile 39
Private Key field
  in Enterprise profile 39
Process Audit Log Database 19, 20
process model
  customizing 32, 68
  described 62
  design-time 15
  enabling 86
  error while generating 71, 85, 88
  generating sample 84
  importing from sample 84
  making available 71
  Modeler Repository 15
  sending a Transora request message 64
  steps
    customizing Map to TDC Request 67
    customizing Process Response and End 68
    customizing Verify Response 67
    customizing Wait for Internal Request 66
    Map to TDC Request 64
    Send TDC Request 64
    Verify MDN 65
    Verify Response 65
    Wait for Internal Request 64
    Wait for MDN 65
    Wait for TDC Response 65
  TDC_Item_Message_Model.model template 63
Index

template location 63
when to use 32, 63
Process Response and End process model step 65
customizing 68
process run time. See PRT
processing rule
customizing process model step 66
defining at configuration 32
versus process model 32, 63
profiles, Trading Networks
defining 30
described 36
Enterprise
defining 37
extended profiles for EDIINT 36
external ID type field 36
information you supply 36
partner
defining 39
UserID (batch user ID field) 36
program code conventions in this document 7
Protocol field
in Enterprise profile 38
in partner profile 41
PRT
described 19
run-time use 19
Publishers
defined 20
See also Suppliers
Publisher-side functionality with Transora 10

R
Receiver field
in TPA 44
RequestMDN parameter
in TPA data 47
RequestSignedReceipt parameter
in TPA data 47
required external ID type
Enterprise profile 38
partner profile 40
requirements, Transora Module

webMethods components 26
run-time
architecture and components 16
run-time use 16

S
samples
package 13
running during configuration 33
wm.ip.transora.samples.service.mapOutboundDoc service 67
wm.ip.transora.samples.service.processResponse service 68
wm.ip.transora.samples.service.verifyResponse service 67
samples package 13
Security tab
Enterprise profile fields 39
partner profile fields 41
Send TDC Request process model step
described 64
Sender field
in TPA 44
services
wm.EDIINT.receive 33
wm.ip.transora.cm.handlers.send 74
wm.ip.transora.samples.service.itemAddUsingRoutingRule 33
wm.ip.transora.service.createEnvelopeFromPayload 75
wm.ip.transora.service.getResponseDetails 75
wm.ip.transora.service.processRequest 76
wm.ip.transora.service.processRequest 76
wm.ip.transora.service.recordToOutboundXML 76
wm.ip.transora.service.validateOutboundXML 77
wm.ip.transora.transport.ediint.service.verifyMDN 78
Sign sub-tab
in Enterprise profile 39
in partner profile 41
signed MDN
requesting 47
signing outbound Transora messages 46
storing process models 15
Subscribers
defined 20
Suppliers
defined 20
TDC Accept Reject Response processing rule 34
TDC Accept Reject Response TN document type 55
TDC GDSN Item Registry Response TN document type 57
TDC GDSN Party Registry Response TN document type 58
TDC Item Request TN document type 56
TDC Item Response TN document type 53
TDC Publication Response TN document type 54
TDC_Item_Message_Model.model template 63
TDCResponseStatus in Transora response message 67
TDSN Item Management (IM)
interacting with a Publisher 21
TDSN Subscription Agreement 30
templates, process model
described 62
location 63
TDC_Item_Message_Model.model 63
TN document type
customizing for process model 66
defining 31, 51
defining for Transora response messages 51
described 50
in back-end system 50
pre-configured 34
pre-processing options 50
provided by Transora Module 53
TDC Accept Reject Response 55
TDC GDSN Item Registry Response 57
TDC GDSN Party Registry Response 58
TDC Item Request 56
TDC Item Response 53
TDC Publication Response 54
TN flat file document type
identifying 52
TN XML document type
identifying 52
TNPartners ACL 33
TNPartners group 33
TPA data
AltSenderGLN parameter 45
defining IS document type 45
DTD parameter 46
MessageType parameter 46
RequestMDN parameter 47
RequestSignedReceipt parameter 47
TransoraGLN parameter 45
Transport parameter 46
ValidateOutput parameter 46
Version parameter 46
TPA. See trading partner agreement
Trading Networks Database
design-time use 14
run-time use 20
trading partner
described 36
trading partner agreement (TPA)
default 44
defining 30, 44
described 44
information required 44
trading partner profile. See profiles, Trading Networks 36
Transora
described 10
integrating back-end applications 31
item synchronization process 20
signing up 30
user account on IS 33
version supported by Transora Module 10
Web site 30
Transora Item (Add/Modify/Link) business process template 63
Transora Module
architecture and components (design-time) 13
architecture and components (run-time) 16
configuring 30
described 10
design-time use 15
EDIINT support 11
features 10
installing 27
item synchronization 22
packages 12
pre-configured items 34
process model templates 62
run-time use 18
supported inbound message actions 11
supported outbound message actions  11
supported Transora version  10
TN document types provided  53
uninstalling  28
Transora XML
no retailer support  10
unsupported outbound messages  11
TransoraGLN parameter
in TPA data  45
Transport parameter
in TPA data  46
troubleshooting information  8
typographical conventions in this document  7

U
uninstalling Transora Module  28
Use as Preferred Protocol
delivery method
in Enterprise profile  38
in partner profile  41
UserID field
in Enterprise profile  39

V
ValidateOutput parameter
in TPA data  46
Verify MDN process model step  65
Verify Response process model step  65
customizing  67
Verify sub-tab
in partner profile  42
Version parameter
in TPA data  46

W
Wait for Internal Request process model step
customizing  66
described  64
Wait for MDN process model step
described  65
Wait for TDC Response process model step  65
webMethods Developer
requirements  26
webMethods EDI Module
extended profile fields  36
requirements  26
run-time use  18
webMethods Installer  27
webMethods Integration Server  16
design-time use  14
requirements  26
webMethods Modeler
customizing process model  68
design-time use  15
requirements  26
sample process model  62
webMethods Modeler Repository
design-time use  15
webMethods Monitor
run-time use  19
webMethods Trading Networks
Console requirements  26
design-time use  14
processing rules  32, 63
run-time use  17
server requirements  26
TN document types
back-end systems  50
Transora response messages  51
wm.EDIINT:receive service
updating ACLs  33
wm.ip.Transora.cm.handlers:send service
in process model  64
wm.ip.transora.cm.handlers:send service  74
wm.ip.transora.cm.recs
userParameters IS document type  45
wm.ip.transora.samples.service:itemAddUsingRoutingRule
service  33
wm.ip.transora.samples.service:mapOutboundDoc service  67
wm.ip.transora.samples.service:processResponse service  68
wm.ip.transora.samples.service:verifyResponse service  67
wm.ip.transora.service:createEnvelopeFromPayload service  75
wm.ip.transora.service:getResponseDetails service  75
wm.ip.transora.service:processInit service  76
wm.ip.transora.service:processRequest service  76
Index

wm.ip.transora.service:recordToOutboundXML service 76
wm.ip.transora.service:validateOutboundXML service 77
wm.ip.transora.transport.ediint.service:verifyMDN service 78
wm.tn.doc
  view service 67
WmEDIINT package
  ACLs 33
  extended profile fields 36
WmIPRoot package 12
WmTransora package 12, 74
WmTransoraDocuments package 12
WmTransoraSamples package 13

X

XQL queries
  identifying namespace mappings 52