Administrator’s Guide

iPlanet™ TradingXpert

Version 3.5.1

December 2001

The current product names and latest release versions listed in this documentation may have changed and have not necessarily been updated. Please refer to http://docs.iplanet.com/docs/manuals/ for the most current product release information.

Contents

About this Book

This Guide provides an introduction to the operation and functionality of the iPlanet TradingXpert System. TradingXpert provides companies with a comprehensive software solution for setting up and operating a cost-effective and easy-to-use electronic commerce system that is built upon Internet technologies.

The following topics are covered in this section:

- Before You Begin
- Audience and Roles
- Organization
- Documentation Set
- Conventions

TradingXpert is subject to the terms detailed in the license agreement accompanying it.

Before You Begin

Downloading the Latest Version of Any TradingXpert Document

We continuously update the TradingXpert documentation. You can download the latest version from:

http://docs.iplanet.com/docs/manuals/trade.html

Audience and Roles

This Guide is written for the TradingXpert System's site administrator.

What You Need to Know

The *iPlanet ECXpert Installation Guide* is written with the assumption that you understand the basics of the operating system on which you are running this software as well as iPlanet ECXpert and the iPlanet Application Server (iAS). In addition, the site administrator must have Administrator privileges.

The documentation also assumes you have the following basic background:

- a general understanding of the Internet and the World Wide Web
- experience in the setup and management of web services
- an understanding of your company's electronic commerce system architecture
- basic knowledge of the NAS architecture and functionality

Note For administrators planning to extend the utility of TradingXpert by building customized forms, a strong background in Java, C++, and NAS is recommended.

Organization

The *iPlanet ECXpert Installation Guide* is structured as follows:

- Chapter 1, “Introducing the TradingXpert System,” introduces the major features of the TradingXpert System.
- Chapter 2, “Customizing TradingXpert’s Capabilities,” describes how to customize the TradingXpert System to build your own forms.
- Chapter 3, “System Utilities,” documents the system utilities that are available for use with the TradingXpert System.
- Appendix A, “TradingXpert EDI Document Formats,” documents the ANSI X12 EDI document formats supported by TradingXpert.
- Appendix B, “TradingXpert API Reference,” provides details about the TradingXpert API.
- “Glossary” defines the various terms and concepts of the TradingXpert System as well as other iPlanet and Netscape applications and software.
- “Index” lists important terms and the page numbers where information about those terms appears in the text.

Documentation Set

The following is a list of iPlanet TradingXpert 3.5.1 documentation:

- *iPlanet TradingXpert Installation Guide*
- *iPlanet TradingXpert Administrator's and Developer's Guide*
- Any accompanying Release Notes. (Release Notes contain documentation errata, software patches, and a known bugs list.)

ECXpert and Other Documentation

- *iPlanet ECXpert Administrator's Guide*
- *iPlanet ECXpert Developer's Guide*
- *iPlanet ECXpert Operations Reference Guide*
- *iPlanet ECXpert Installation Guide*
- *iPlanet Application Server (iAS) Overview and Installation Guide*
- *Developing iPlanet Application Server (iAS) Extensions*

If you are using TradingXpert for electronic data interchange (EDI), the following documentation is also useful:

- the UN/EDIFACT on-line documentation on the World Wide Web at URL http://www.unece.org/trade/untdid/, for detailed information about EDI standards
- the *Electronic Data Interchange X12 Standards*, for a technical reference on EDI implementation (document no. ASC X12S/95-533, available from the ASC X12 Secretariat, Data Interchange Standards Association, Inc., 1800 Diagonal Road, Suite 200, Alexandria, VA 22314-2852, 703.548.7005), or at: http://www.disa.org.

Conventions

A number of typographic conventions are used throughout this manual to help you recognize special terms and instructions. These conventions are summarized in the table below.

Convention	Meaning	Example
boldface	items on the screen that you manipulate	Click the **Submit** button to save your changes.
	names of keys	Press **Enter** to clear the message.
boldface numbered steps	higher level descriptions of tasks you perform (more detailed instructions follow)	**1. Enter the group information.** Enter the name in the **Group Name** field, and a short description in the **Description** field.
italics	key words, such as terms that are defined in the text	The notices posted on an electronic BBS are called *articles*.
	names of books	For more information, refer to the *iPlanet ECXpert Installation Guide*.
	file names	The associated data is stored in the Dispatcher section of the *ecx.ini* file.
`courier font`	command line input or output	Enter the following command: `ls *.mle`

	text file content (HTML templates, config files)	`<TITLE>Password Check</TITLE>`
		`<IMG SRC="/ui/icons/hd_svcs.gif">`

	code samples	**Syntax** `const char* getName() const`
square brackets, [...]	In command syntax, items within square brackets are *optional*.	In the following example: nsusrgrp **`[-v]`** `insert` **`[-l] -k`** *`key arguments`* **`-r act=`***`relation action,`* **`usrid=`***`ID`***`|usrlogin=`***`login`* • **-v** and **-r** are optional. • You may specify either **usrid** or **usrlogin**. • You must substitute valid values for italicized items.
vertical line (|)	In command syntax, items on either side are valid *alternatives*.	
`courier italics`	In command syntax, items in italics are *not* literals.	
em dash (—)	“none” or “nothing”	**Arguments** —

Chapter

1

Introducing the TradingXpert System

This chapter introduces the major features and concepts of the iPlanet TradingXpert System as well as an overview of the TradingXpert architecture. It covers the following topics:

- What is the TradingXpert System?
- Key Features of TradingXpert
- Architecture Overview
- TradingXpert Usage Scenario

What is the TradingXpert System?

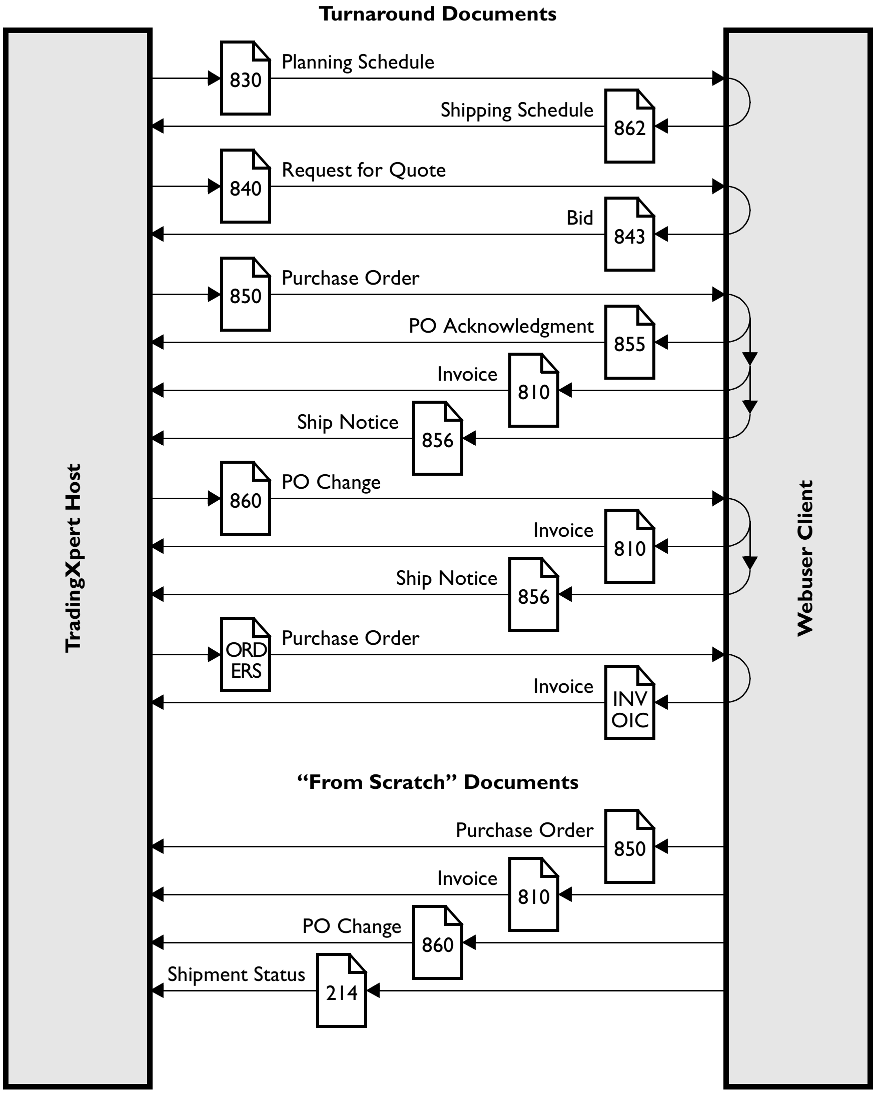
iPlanet TradingXpert, an Internet Commerce Exchange application, is packaged trading community software that enables a corporation to trade with any size partner, securely, over the Internet. TradingXpert allows customers to create Internet trading communities across multiple, diverse boundaries.

Customers buy TradingXpert to:

- Streamline cross-company processes
- Solidify trading relationships
- Empower low-tech trading partners

Figure 1.1 shows the various types of document exchanges between trading partners that are supported through TradingXpert without any customization. You can customize TradingXpert to add other documents as necessary to meet your specific needs. See Chapter 2 for more information on customizing TradingXpert.

Figure 1.1 TradingXpert System interaction with trading partners



Key Features of TradingXpert

TradingXpert provides a wealth of benefits to its users. Its flexible user interface, message center and form composition capabilities make it a valuable and unique business process solution. Some key features are listed below.

Flexible User Interface

- Fully customizable
- Fully extensible

Message Center

- Inbox
- Outbox
- Personalized profile
- Templates
- Attachments
- Event notifications

Form Composition

- Intelligent turnaround capability
- Pre-packaged templates for standard EDI transactions
  - — Planning Schedule with Release Capability
  - — Shipping Schedule
  - — Request for Quotation
  - — Response to Request for Quotation
  - — Purchase Order (ANSI 850 and EDIFACT ORDERS)
  - — Purchase Order Acknowledgment
  - — Purchase Order Change Request - Buyer Initiated
  - — Invoice (ANSI 810 and EDIFACT INVOIC)

- — Ship Notice/Manifest
- — Transportation Carrier Shipment Status Message

- Netscape Application Builder (NAB) — NT version only

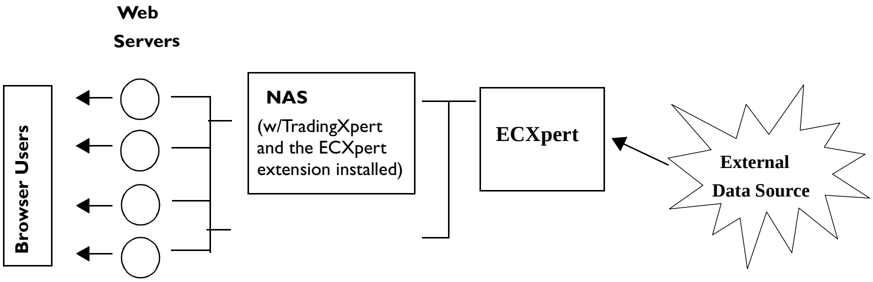
Built on the Netscape Application Server (NAS)

- 24 X 7 application availability
- Unmatched scalability and performance
- Extensible to additional data sources

Architecture Overview

Figure 1.2 presents a high-level diagram of the TradingXpert System architecture. You can refer to this diagram for the general discussion of TradingXpert components that follows.

Figure 1.2 TradingXpert System architecture



Note The ECXpert extension must be installed on TradingXpert. Optionally, TradingXpert can be on one machine and ECXpert on a different machine.

TradingXpert Usage Scenario

TradingXpert enables customers to conduct business transactions more effectively and efficiently. The following scenario introduces you to a typical TradingXpert session, beginning by logging in to the system, then generating a document to send to a trading partner, viewing the document, tracking the progress of a document and finally, logging out of the session. Please refer to the figures indicated in each step for detail about the screens and interfaces encountered during the scenario.

1. **Log in to the TradingXpert System using your User ID and password.** (See Figure 1.3)

   To open the TradingXpert login screen, enter the following URL in your browser:

   http://<*TradingXperthostname*>:<*port*>/TX

   where <hostname> is the name of the host machine where TradingXpert is installed, and <port> is the port number TradingXpert is using.

Note If new member self-registration is enabled, the login screen will include a **Register** button. Users who are not registered can click the button, complete the registration form that appears, and submit it. (See Figure 1.4) Depending on the parameters in the *FXproperties* file, self-registered users may be added as either active or inactive members; only active members can access TradingXpert. See Chapter 2, "Customizing TradingXpert's Capabilities," for instructions to enable member self-registration.

2. **Check the Inbound Folder for new documents received.**

   To check for new documents received, select the **Inbound** option from the sidebar menu on the Trade Center screen. (See Figure 1.5) This will display your Inbound Folder. The **Unread** field will tell you how many documents have not been viewed; select All Partners and Unread Documents Only from the pull-down menus and click **Search** to display the list of all unread documents. (See Figure 1.6)

3. **Choose a document to view.**

   At the bottom of this screen, notice the list of documents and the information provided for each. Select a Purchase Order (ANSI 850) to view, and click the **View** button for that Purchase Order. (See Figure 1.15) The document you select will appear on the screen.

4. **Send an Acknowledgment that you received the Purchase Order.**

   To generate a Purchase Order Acknowledgment (ANSI 855), click **Create PO Ack** at the bottom of the Purchase Order screen. An acknowledgment form will appear, pre-populated with data from the purchase order. Modify the data if necessary and submit the document.

5. **Generate an Invoice and send it to your trading partner.**

   Repeat the steps from the Acknowledgment process, but click **Create Invoice** at the bottom of the Purchase Order screen to automatically generate an Invoice (ANSI 810) from the purchase order.

   Alternately, you can manually create an Invoice. Select the **New Document** option from the sidebar menu to display the New Document screen. (See Figure 1.9) Then either select a trading partner and 810 from the pull-down menus, or select a template for invoices that is applicable to the trading partner. Complete the form and click **Submit**.

6. **Create a template for Shipment Status Notices, then use the template to create and send a notice to your trading partner.**

   Select the **Templates** option from the sidebar menu. From the Templates screen, select **Create New Template**. (See Figure 1.11) A new screen will appear; from the Create Template from Scratch section, select the trading partner and document type (Ship Status) and click **Create New Template**. (See Figure 1.12) Edit the Shipment Status Notice (ANSI 214) form that appears, and click **Save as Template**. Enter a template name when prompted for one, and click **Save Template**.

   Select the **New Document** option from the sidebar menu to display the New Document screen again. (See Figure 1.9) Select the template you just created and click **Create New Document**. Complete the form that appears, and click **Submit**.

7. **Track your documents.**

   Select the **Outbound** option from the sidebar menu. From the Outbound Folder, select **Search** to list all outbound documents. (See Figure 1.7) Choose a document from the list and click the **Tracking Info** button. The Tracking Information screen appears, detailing the status of the document.

8. **Log out of the TradingXpert System.**

   End your TradingXpert session by clicking **Logout** at the top of the screen.

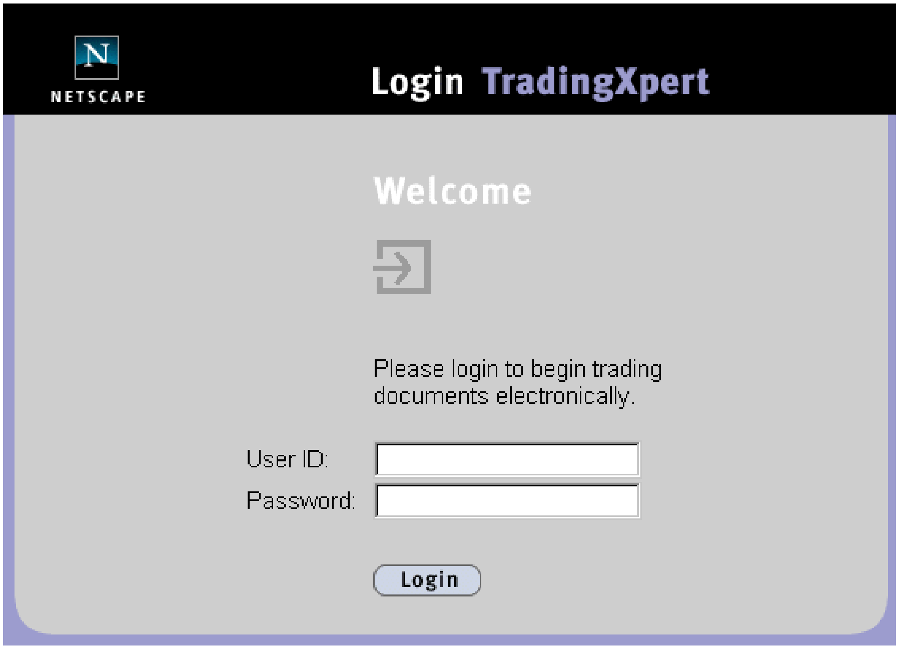
Although each interface has a unique purpose and content, each screen within TradingXpert shows the same main menu bar and sidebar to navigate the system. This section will introduce the different interfaces of the TradingXpert System.

Login

The Login screen is the first you will encounter as you begin a session in TradingXpert. Figure 1.3 illustrates this screen.

Note If new member self-registration is enabled, the login screen will also include a **Register** button. Users who are not registered can click the button, complete the registration form that appears, and submit it. (See Figure 1.4)

Figure 1.3 Login screen



Please login to begin trading documents electronically.

User ID:

Password:

Table 1.1 Information on the Login screen

Item	Description
User ID	The trading partner member ID.
Password	The corresponding password for the User ID.
Login button	Logs user into the TradingXpert User Interface (UI).
Register button	Opens the New Member Registration screen.

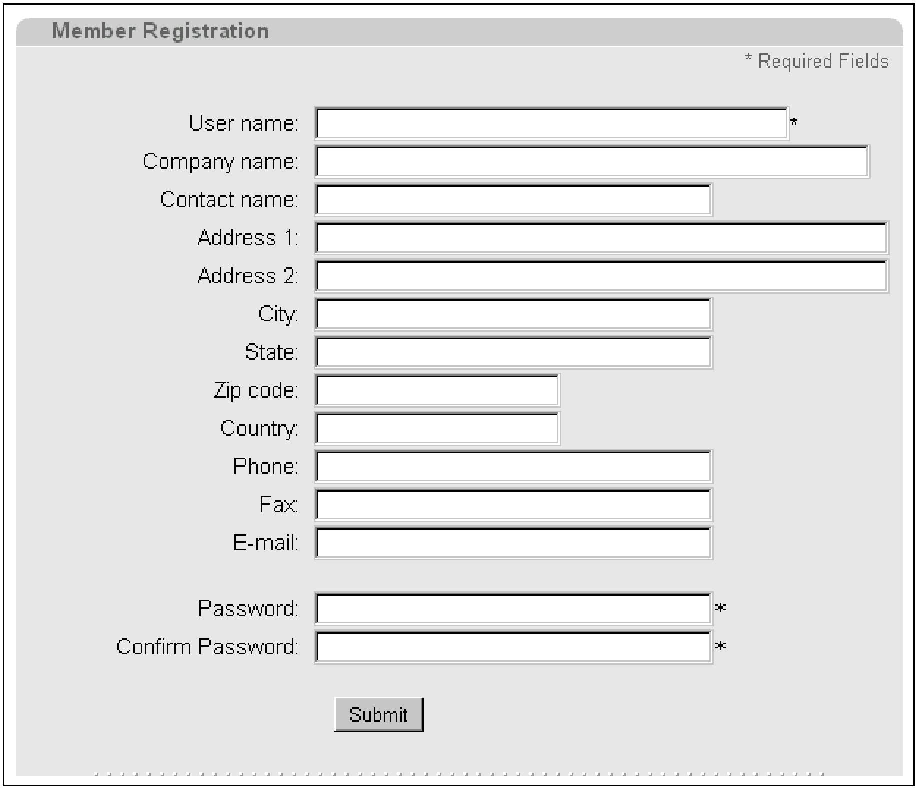
New Member Registration

Optionally, TradingXpert can be customized to allow individuals who have not previously registered as TradingXpert users to see the New Member Registration screen (see Figure 1.4), which they can complete and submit. The required fields on this screen are **Username**, **Password**, and **Confirm Password**.

After a user submits a registration, the screen will display a confirmation that the registration was completed successfully. Depending on the parameters set in the *FXproperties* file, self-registered users may be added as either active or inactive members; only active members can access the remaining TradingXpert screens.

Note See "Enabling New Member Self-Registration" on page 88 for instructions to enable member self-registration.

Figure 1.4 New Member Registration screen



Member Registration

* Required Fields

User name: *
Company name:
Contact name:
Address 1:
Address 2:
City:
State:
Zip code:
Country:
Phone:
Fax:
E-mail:

Password: *
Confirm Password: *

Table 1.2 Information on New Member Registration screen

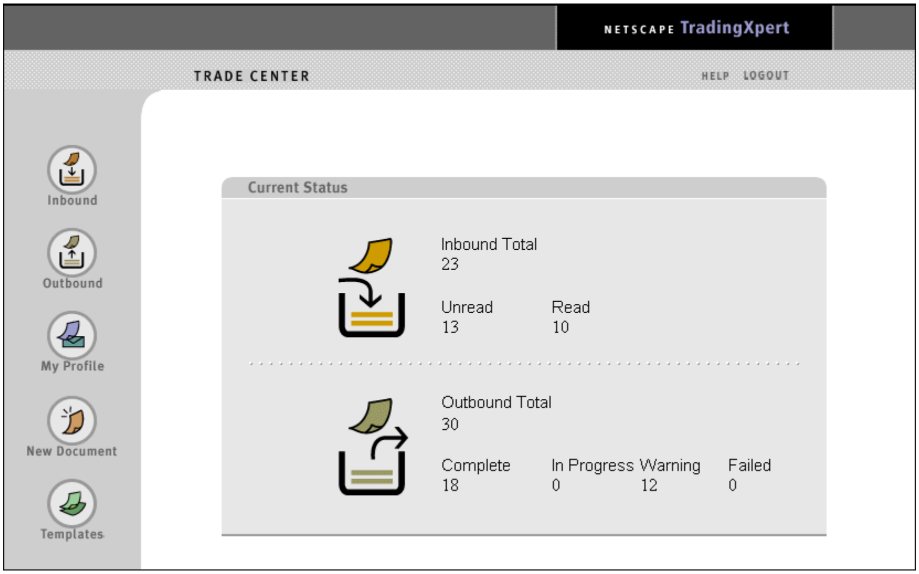
Item	Description
User name	User's login name. (required)
Company name	Name of user's company.
Contact name	User's full name.
Address 1	Address of Contact.
Address 2	Additional space for address of Contact.
City	Name of city.
State	Name of state.
Zip code	Postal zip code.
Country	Name of country.
Phone	Telephone number of Contact.
Fax	Facsimile number of Contact.
Email	Electronic mail address for Contact.
Password	Enter your password (does not display). (required)
Confirm Password	Enter your password again (does not display). (required) If your entries in this field and the **Password** field do not match exactly, you will be prompted to re-enter both password fields.
Submit button	Submits registration information, registering the user.

Trade Center

Once the user has logged in to the TradingXpert System, the Trade Center main screen (Figure 1.5) appears.

Note The menu options on the top and sidebar menus appear on all TradingXpert screens except the Login and New Member Registration screens.

Figure 1.5 Trade Center screen



Current Status

Inbound Total
23

Unread 13 Read 10

Outbound Total
30

Complete 18 In Progress 0 Warning 12 Failed 0

Table 1.3 Information on the Trade Center main screen

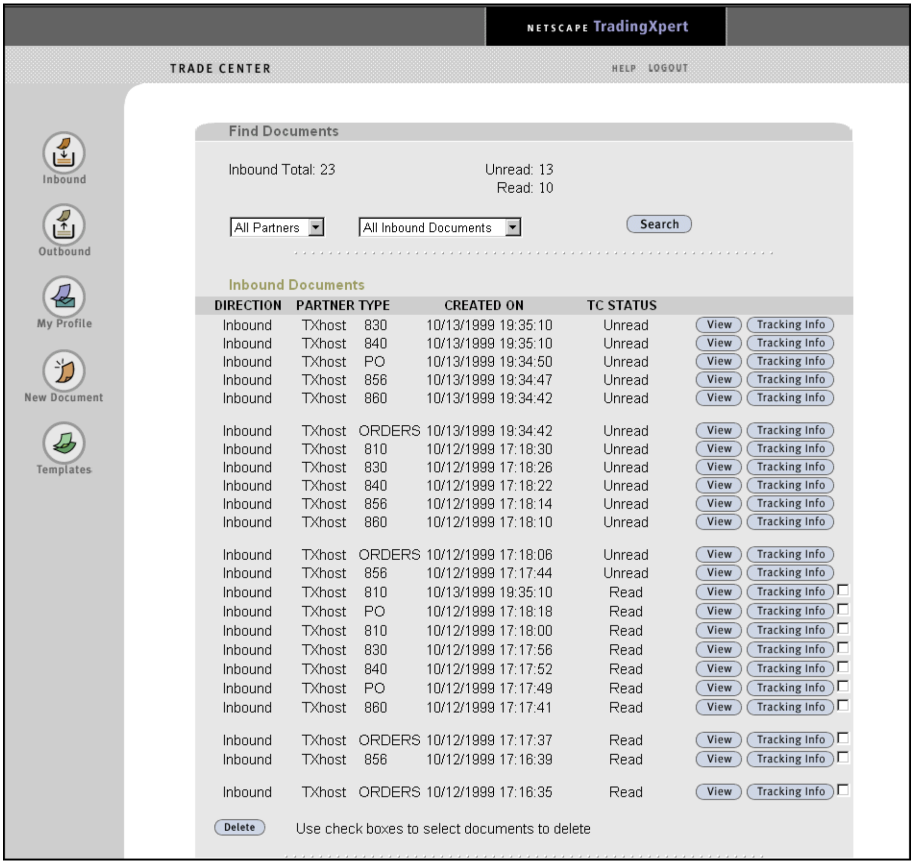
Item	Description
Links at Top	
Trade Center	Displays the Trade Center main screen (Figure 1.5).
Help	Displays online help information.
Logout	Closes the TradingXpert session and logs the user out of the system.
Sidebar Menu Options	
Inbound	Link to Inbound folder.
Outbound	Link to Outbound folder.
My Profile	Link to user's contact information and password maintenance screen.
New Document	Link to New Document screen where user can create a new document.
Templates	Link to the Templates screen where the user can manage and create document templates.

Item	Description
Current Status Fields	
Inbound Total	Displays number and status of incoming documents received by user.
Unread	The number of documents that have not been viewed by the user.
Read	The number of Inbound documents that have been viewed by the user.
Outbound Total	Displays number and status of outgoing documents sent by user.
Complete	The number of successful outbound documents.
In Progress	The number of documents still being processed.
Warning	The number of documents with a warning.
Failed	The number of documents that did not reach trading partner.

Inbound and Outbound Folders

The Inbound and Outbound folders look similar; however, the information reported in each differs. The Inbound folder offers the TradingXpert user a glimpse of documents that are being received, while the Outbound folder allows the TradingXpert user to view the status of any documents being sent to trading partners.

Figure 1.6 Inbound screen



Find Documents

Inbound Total: 23

Unread: 13
Read: 10

All Partners ▾ All Inbound Documents ▾

Inbound Documents

DIRECTION	PARTNER	TYPE	CREATED ON	TC STATUS			
Inbound	TXhost	830	10/13/1999 19:35:10	Unread	View	Tracking Info	
Inbound	TXhost	840	10/13/1999 19:35:10	Unread	View	Tracking Info	
Inbound	TXhost	PO	10/13/1999 19:34:50	Unread	View	Tracking Info	
Inbound	TXhost	856	10/13/1999 19:34:47	Unread	View	Tracking Info	
Inbound	TXhost	860	10/13/1999 19:34:42	Unread	View	Tracking Info	
Inbound	TXhost	ORDERS	10/13/1999 19:34:42	Unread	View	Tracking Info	
Inbound	TXhost	810	10/12/1999 17:18:30	Unread	View	Tracking Info	
Inbound	TXhost	830	10/12/1999 17:18:26	Unread	View	Tracking Info	
Inbound	TXhost	840	10/12/1999 17:18:22	Unread	View	Tracking Info	
Inbound	TXhost	856	10/12/1999 17:18:14	Unread	View	Tracking Info	
Inbound	TXhost	860	10/12/1999 17:18:10	Unread	View	Tracking Info	
Inbound	TXhost	ORDERS	10/12/1999 17:18:06	Unread	View	Tracking Info	
Inbound	TXhost	856	10/12/1999 17:17:44	Unread	View	Tracking Info	
Inbound	TXhost	810	10/13/1999 19:35:10	Read	View	Tracking Info	□
Inbound	TXhost	PO	10/12/1999 17:18:18	Read	View	Tracking Info	□
Inbound	TXhost	810	10/12/1999 17:18:00	Read	View	Tracking Info	□
Inbound	TXhost	830	10/12/1999 17:17:56	Read	View	Tracking Info	□
Inbound	TXhost	840	10/12/1999 17:17:52	Read	View	Tracking Info	□
Inbound	TXhost	PO	10/12/1999 17:17:49	Read	View	Tracking Info	□
Inbound	TXhost	860	10/12/1999 17:17:41	Read	View	Tracking Info	□
Inbound	TXhost	ORDERS	10/12/1999 17:17:37	Read	View	Tracking Info	□
Inbound	TXhost	856	10/12/1999 17:16:39	Read	View	Tracking Info	□
Inbound	TXhost	ORDERS	10/12/1999 17:16:35	Read	View	Tracking Info	□

Use check boxes to select documents to delete

Table 1.4 Information on the Inbound screens

Item	Description
	Find Documents
Inbound Total	The total number of documents in the user's Inbound folder. As read files are deleted from the folder, the total is decremented accordingly.
Unread	The number of documents that have not been viewed by the user.

Table 1.4 Information on the Inbound screens (Continued)

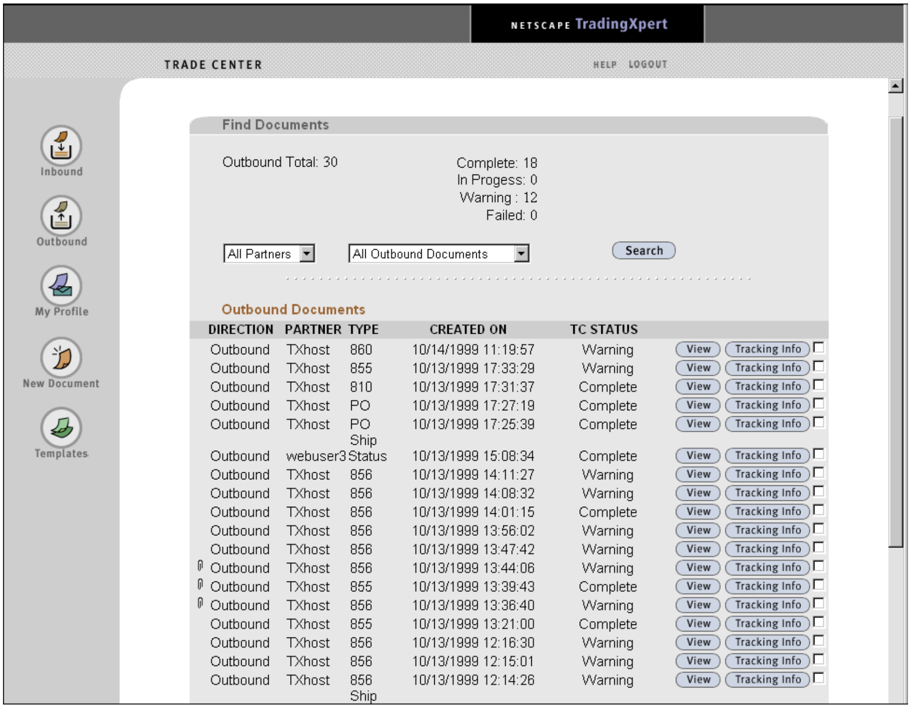
Item	Description
Read	The number of Inbound documents that have been viewed by the user. As read files are deleted from the folder, this figure is decremented accordingly.
Trading Partner drop-down list	Choose the trading partner about whom you want to obtain Outbound document status information.
Document Selection drop-down list	Choose the document type about which you want to obtain Outbound document status information.
Search button	Displays documents matching user selections from **Trading Partner** and **Document Selection** lists.
Inbound Documents	
Check box	Click to select a document for manual deletion from the list. A check box will appear next to only those documents that have been read.
DIRECTION	Whether a document is incoming or outgoing.
PARTNER	User name of trading partner.
TYPE	Format of the document that was sent or received.
CREATED ON	The date on which the document was created.
TC STATUS	Trade Center status. The icon represents the status of the document. See the Outbound Status List (above) for icon definitions.
View button	View the document. See "View" on page 40 for further information.
Tracking Info button	Lists progress activity of document.
Delete button	Click to manually delete all read documents whose check boxes are marked for deletion. A tracking ID will be displayed, and the number of read documents will be decremented.

Note Read documents remain in the Inbound folder until they are older than the number of days specified by the user's profile (see "Days before deleting messages" on page 30). Unread messages will remain in the Inbound folder past the deletion date, until they are read *and* the number of days before deletion has passed. If an unread message is past the deletion date, it will be removed from the Inbound folder as soon as it is read.

To restore messages that have been removed from the Inbound folder, change the **Days before deleting messages** parameter (see page 30) to zero (0).

Note "Deleting" an incoming document removes it from the list of inbound documents on the screen; it does not delete it from the database. To delete an item from the database, the administrator must use ECXpert's `bdgrealpurge` utility. See the *iPlanet ECXpert Administrator's Guide* for more information about `bdgrealpurge`.

Figure 1.7 Outbound screen



Find Documents

Outbound Total: 30

Complete: 18
In Progess: 0
Warning : 12
Failed: 0

Outbound Documents

	DIRECTION	PARTNER	TYPE	CREATED ON	TC STATUS			
	Outbound	TXhost	860	10/14/1999 11:19:57	Warning	View	Tracking Info	□
	Outbound	TXhost	855	10/13/1999 17:33:29	Warning	View	Tracking Info	□
	Outbound	TXhost	810	10/13/1999 17:31:37	Complete	View	Tracking Info	□
	Outbound	TXhost	PO	10/13/1999 17:27:19	Complete	View	Tracking Info	□
	Outbound	TXhost	PO	10/13/1999 17:25:39	Complete	View	Tracking Info	□
	Outbound	webuser3	Ship Status	10/13/1999 15:08:34	Complete	View	Tracking Info	□
	Outbound	TXhost	856	10/13/1999 14:11:27	Warning	View	Tracking Info	□
	Outbound	TXhost	856	10/13/1999 14:08:32	Warning	View	Tracking Info	□
	Outbound	TXhost	856	10/13/1999 14:01:15	Complete	View	Tracking Info	□
	Outbound	TXhost	856	10/13/1999 13:56:02	Warning	View	Tracking Info	□
	Outbound	TXhost	856	10/13/1999 13:47:42	Warning	View	Tracking Info	□
📎	Outbound	TXhost	856	10/13/1999 13:44:06	Warning	View	Tracking Info	□
📎	Outbound	TXhost	855	10/13/1999 13:39:43	Complete	View	Tracking Info	□
📎	Outbound	TXhost	856	10/13/1999 13:36:40	Warning	View	Tracking Info	□
	Outbound	TXhost	855	10/13/1999 13:21:00	Complete	View	Tracking Info	□
	Outbound	TXhost	856	10/13/1999 12:16:30	Warning	View	Tracking Info	□
	Outbound	TXhost	856	10/13/1999 12:15:01	Warning	View	Tracking Info	□
	Outbound	TXhost	856	10/13/1999 12:14:26	Warning	View	Tracking Info	□
			Ship					

Table 1.5 Information on the Outbound screen

Item	Description
	Find Documents
Outbound Total	The total number of documents in your Outbound folder.
Complete	The number of successful outbound documents.
In Progress	The number of documents still processing.
Warning	The number of documents with a warning.

Table 1.5 Information on the Outbound screen (Continued)

Item	Description
Failed	The number of documents that did not reach trading partner.
Trading Partner drop-down list	Choose the trading partner about whom you want to obtain Outbound document status information.
Document Selection drop-down list	Choose the document type about which you want to obtain Outbound document status information.
Search button	Displays documents matching user selections from **Trading Partner** and **Document Selection** lists.
Outbound Documents	
Check box	Click to select a document for manual deletion from the list.
DIRECTION	Whether a document is incoming or outgoing.
PARTNER	User name of trading partner.
TYPE	Format of the document that was sent or received.
CREATED ON	The date on which the document was created.
TC STATUS	Trade Center status. The icon represents the status of the document. See the Outbound Status List (above) for icon definitions.
View button	View the selected document. See "View" on page 40 for further information.
Tracking Info button	Lists progress activity of document.
Delete button	Click to manually delete all documents whose check boxes are marked for deletion. A tracking ID will be displayed, and the number of outbound documents will be decremented.

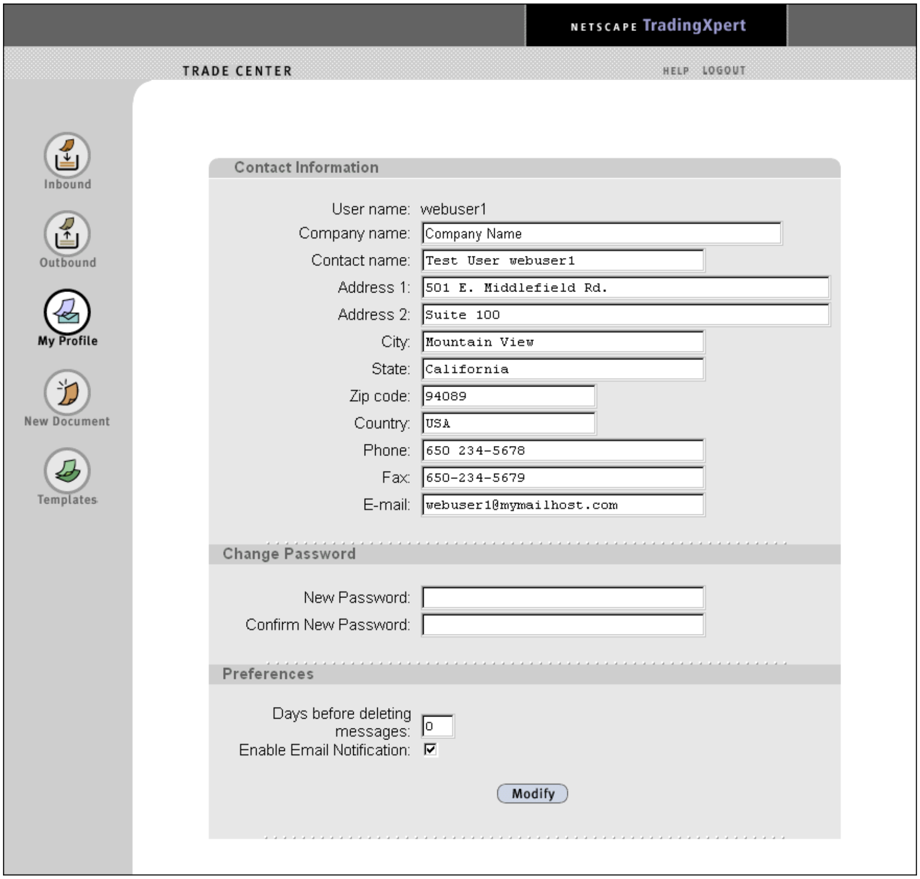
Note Outbound documents remain in the Outbound folder until they are older than the number of days specified by the user's profile (see "Days before deleting messages" on page 30). To restore messages that have been removed from the Outbound folder, change the **Days before deleting messages** parameter (see page 30) to zero (0).

Note "Deleting" an outbound document removes it from the list of outbound documents on the screen; it does not delete it from the database. To delete an item from the database, the administrator must use ECXpert's `bdgrealpurge` utility. See the *iPlanet ECXpert Administrator's Guide* for more information about `bdgrealpurge`.

My Profile

This user interface allows the TradingXpert user to view, update or delete information in a contact information form.

Figure 1.8 My Profile screen



Contact Information

User name: webuser1
Company name: Company Name
Contact name: Test User webuser1
Address 1: 501 E. Middlefield Rd.
Address 2: Suite 100
City: Mountain View
State: California
Zip code: 94089
Country: USA
Phone: 650 234-5678
Fax: 650-234-5679
E-mail: webuser1@mymailhost.com

Change Password

New Password:
Confirm New Password:

Preferences

Days before deleting messages: 0
Enable Email Notification: ☑

Table 1.6 Information on My Profile screen

Item	Description
Contact Information	
User name	User's login name.

Item	Description
Company name	Name of user's company.
Contact name	User's full name.
Address 1	Address of Contact.
Address 2	Additional space for address of Contact.
City	Name of city.
State	Name of state.
Zip code	Postal zip code.
Country	Name of country.
Phone	Telephone number of Contact.
Fax	Facsimile number of Contact.
Email	Electronic mail address for Contact.
Change Password	
New Password	Enter your new password (does not display).
Confirm New Password	Enter your new password again (does not display). If your entries in these two fields do not match exactly, you will be prompted to re-enter both password fields.
Preferences	
Days before deleting messages	Enter the number of days after which an unread, incoming document is received or an outgoing document is created that it no longer appears in the Inbound or Outbound folder. To keep all messages, enter a value of zero (0).
Enable Email Notification check box	Click the check box to be notified daily when the Inbound folder contains unread messages.
Modify button	Applies changes or entries made to contact information.

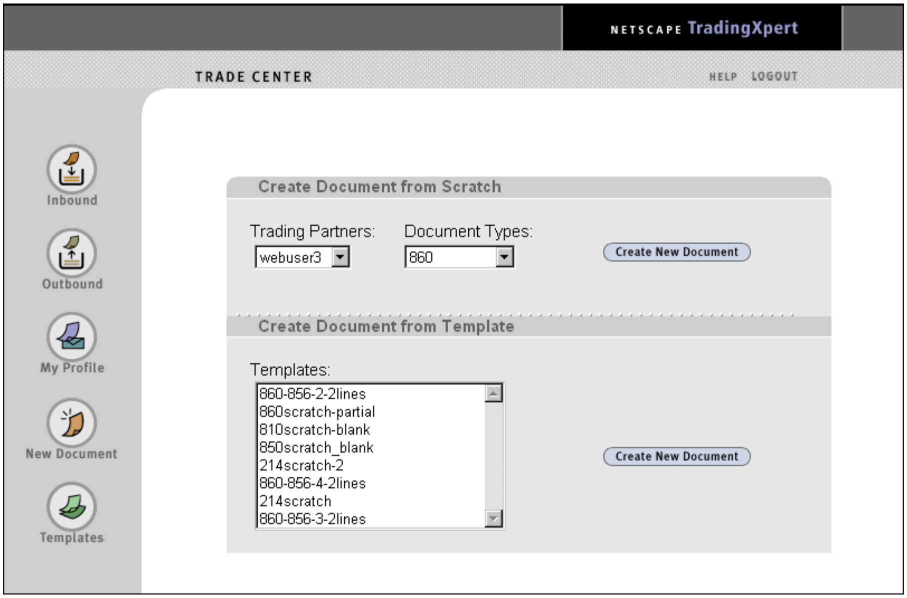
Note Read documents remain in the Inbound folder until they are older than the number of days specified by the user's profile (see "Days before deleting messages" on page 30). Unread messages will remain in the Inbound folder past the deletion date, until they are read *and* the number of days before deletion has passed. If an unread message is past the deletion date, it will be removed from the Inbound folder as soon as it is read.

Note "Deleting" messages removes them from the lists of inbound and outbound documents on the screen; it does not delete them from the database. To delete items from the database, the administrator must use ECXpert's `bdgrealpurge` utility. See the *iPlanet ECXpert Administrator's Guide* for more information about `bdgrealpurge`.

New Document

The new document screen (Figure 1.9) allows the logged-in user to generate a new document to send to a trading partner, either from scratch or from an existing document template.

Figure 1.9 New Document screen, initial selections



Trading Partners: webuser3

Document Types: 860

Templates:

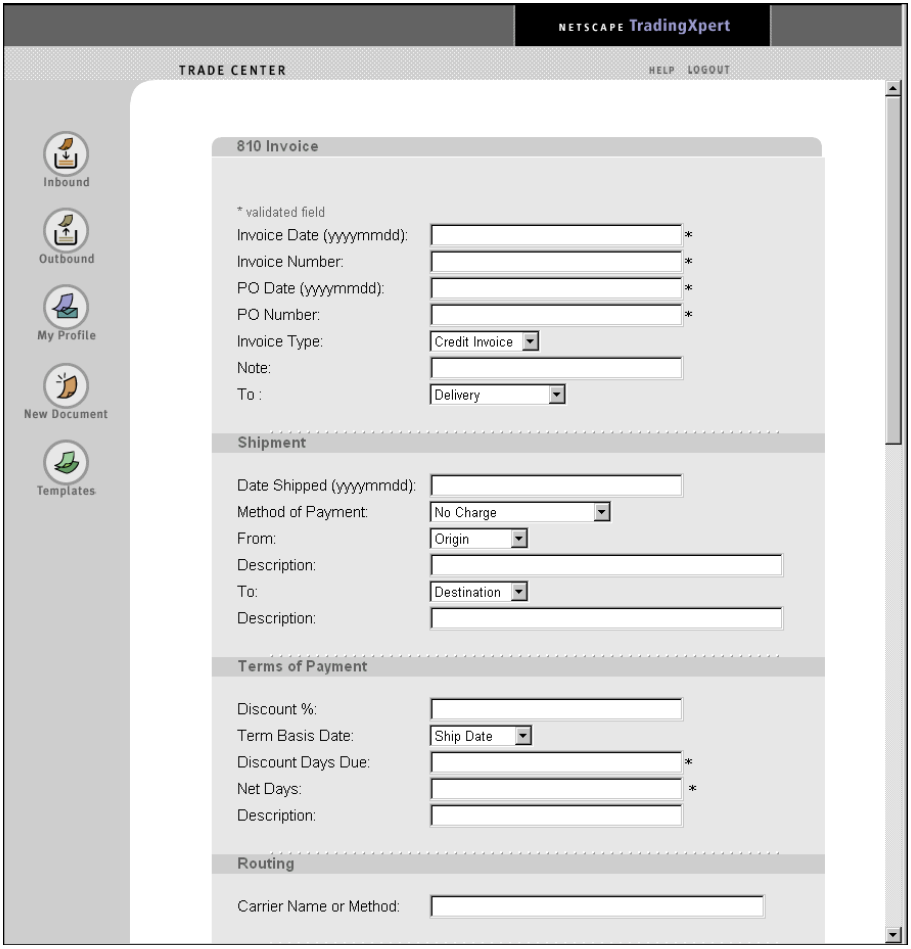
860-856-2-2lines
860scratch-partial
810scratch-blank
850scratch_blank
214scratch-2
860-856-4-2lines
214scratch
860-856-3-2lines

Table 1.7 Information about the New Document screen

Item	Description
	Create Document from Scratch
Trading Partners drop-down menu	Choose a trading partner.
Document Types drop-down menu	Choose the type of document to create.
Create New Document	Click this button to display the appropriate form for the selected document type.
	Create Document from Template
Templates select list	Choose an existing template. See "Templates" on page 35 to create and manage templates.
Create New Document	Click this button to create a new document from the specified template.

After the user has made selections from either the **Create Document from Scratch** or the **Create Document from Template** frame and clicked the corresponding **Create New Document** button, the appropriate form for the selected document type or template will be displayed. An example, for an EDI 810 document (Invoice) created from scratch, is shown in Figure 1.10.

Figure 1.10 New Document screen, Create New Document from Scratch form



810 Invoice

* validated field

Invoice Date (yyyymmdd): *

Invoice Number: *

PO Date (yyyymmdd): *

PO Number: *

Invoice Type: Credit Invoice

Note:

To : Delivery

Shipment

Date Shipped (yyyymmdd):

Method of Payment: No Charge

From: Origin

Description:

To: Destination

Description:

Terms of Payment

Discount %:

Term Basis Date: Ship Date

Discount Days Due: *

Net Days: *

Description:

Routing

Carrier Name or Method:

Table 1.8 Information common to most new document form screens

Item	Description
[Document Type]	
[Data fields]	The information displayed on the new document form varies, depending on the document type being created as well as whether you have created the document from scratch or from a template.
Attach File	
Input field	Enter the fully qualified pathname for the file to be attached to the document.
Browse	Click this button to locate the file on your desktop.
File type drop-down menu	Choose the type of file to attach. Note that the file type must match the file type for the ECXpert partnership.
[Tasks]	
Add Line Item	Click this button to add data fields for a new line item in the document.
Delete Line Item	Click this button to delete the selected line items from the document. Note that you cannot delete the only line item in a document.
Submit Document	Click to send the document to TradingXpert.
Save as Template	Click this button to save the current document as a template.

Note By default, all editable document forms allow users to attach one file in any format (**Any Type** in the pull-down menu). TradingXpert's document forms can be customized to allow web users to attach different file types to an outgoing document. To modify the types as well as the number of files that can be attached to a document, you must change the HTML for that document form. See Chapter 2, "Customizing TradingXpert's Capabilities," for instructions on modifying TradingXpert's file attachment capabilities.

Note If either partner is using a browser's e-mail program (e.g., Netscape Messenger) to submit EDI documents and attachments to ECXpert, the EDI documents must be attached before non-EDI files in order for the documents to be displayed and be viewable in the TradingXpert user interface.

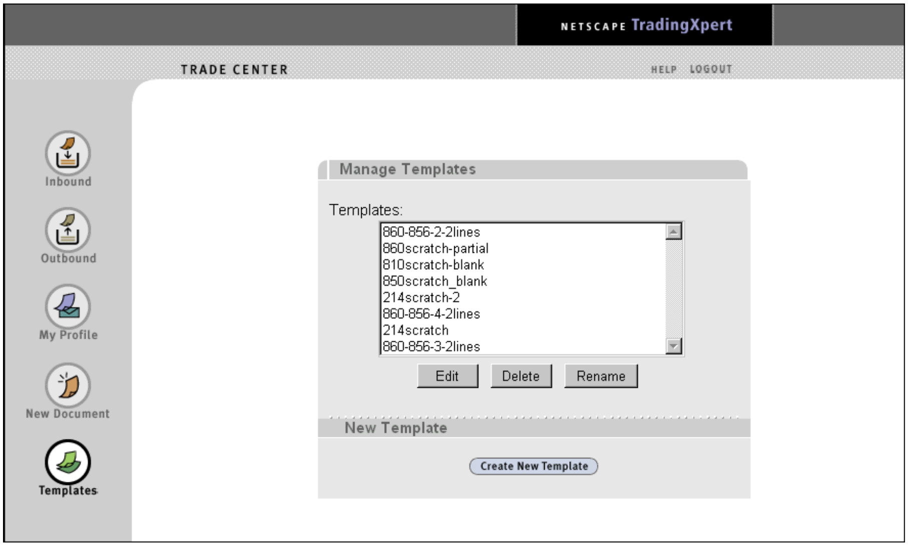
Note When the user creates a new document and clicks the **Save as Template** button, the user will be prompted for a new template name. *Template names should not include spaces.* After the user enters a name and clicks the **Save Template** button, the screen will display a message that the template was saved successfully.

Note New document templates can also be created from the Templates screen.

Templates

The template screen allows the logged-in user to manage existing document templates as well as to create new ones.

Figure 1.11 Templates screen, initial selections



Templates:

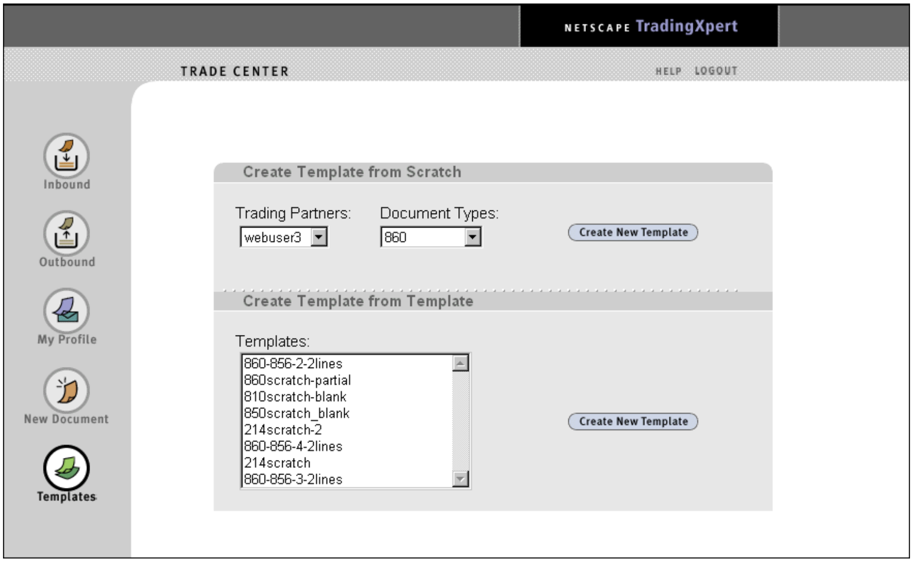
860-856-2-2lines
860scratch-partial
810scratch-blank
850scratch_blank
214scratch-2
860-856-4-2lines
214scratch
860-856-3-2lines

Table 1.9 Information about the Templates screen

Item	Description
	Manage Templates
Templates select list	Choose an existing template.
Edit	Click this button to edit the selected template.
Delete	Click this button to delete the selected template.
Rename	Click this button to rename the selected template.
	New Template
Create New Template	Click this button to display the Create Template screen (see “Create Template screen” on page 36).

When you click the **Create New Template** button on the Templates screen, a new screen will appear (Figure 1.12), from which you can select further criteria for creating the template.

Figure 1.12 Create Template screen



Create Template from Scratch

Trading Partners: webuser3

Document Types: 860

Create Template from Template

Templates:
- 860-856-2-2lines
- 860scratch-partial
- 810scratch-blank
- 850scratch_blank
- 214scratch-2
- 860-856-4-2lines
- 214scratch
- 860-856-3-2lines

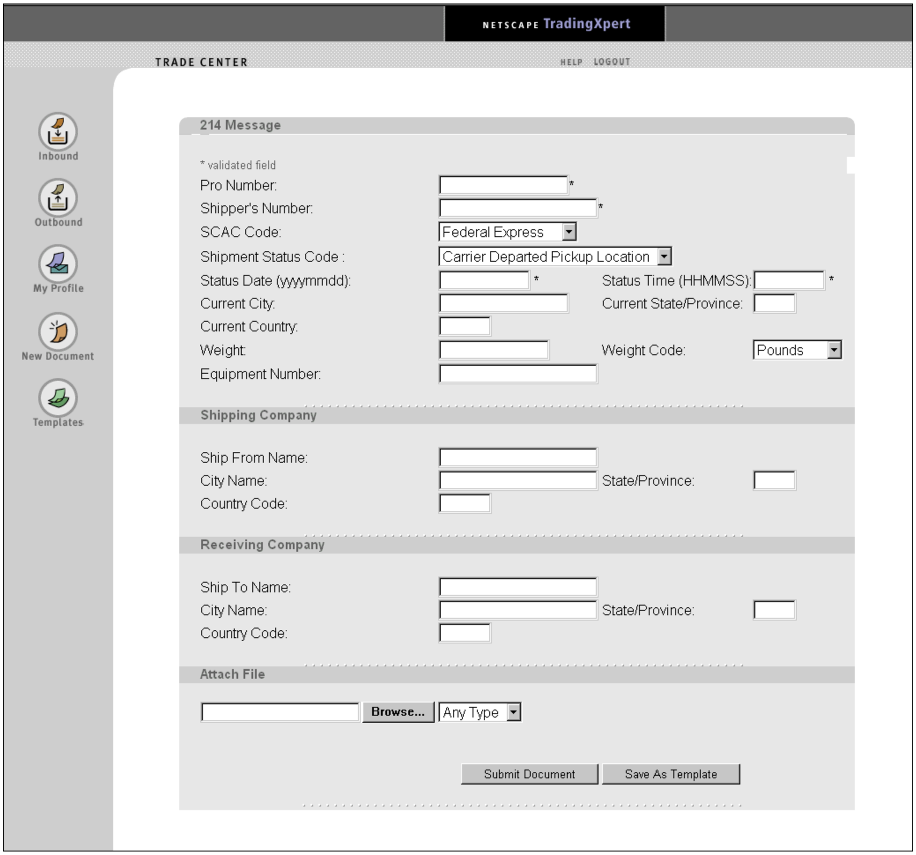
Table 1.10 Information about the Create Template screen

Item	Description
	Create Template from Scratch
Trading Partners drop-down menu	Choose a trading partner.
Document Types drop-down menu	Choose the type of document for which the template will be created.
Create New Template	Click this button to display the appropriate form for the selected document type.
	Create Template from Template
Templates select list	Choose an existing template from which to create a new template. See "Templates" on page 35 to create and manage templates.
Create New Template	Click this button to open the specified existing template, which you can modify and save as a new template.

After the user has made selections from either the **Create Template from Scratch** or the **Create Template from Template** frame and clicked the corresponding **Create New Template** button, the appropriate form for the selected document type or template will be displayed. An example, for an EDI 214 (shipment status notice) template created from scratch, is shown in Figure 1.13.

The user can then modify the document and click **Save as Template**, then enter a template name when prompted for one and click **Save Template**.

Figure 1.13 New Template screen, Create New Template from Scratch form



214 Message

* validated field

Pro Number: *

Shipper's Number: *

SCAC Code: Federal Express

Shipment Status Code : Carrier Departed Pickup Location

Status Date (yyyymmdd): * Status Time (HHMMSS): *

Current City: Current State/Province:

Current Country:

Weight: Weight Code: Pounds

Equipment Number:

Shipping Company

Ship From Name:

City Name: State/Province:

Country Code:

Receiving Company

Ship To Name:

City Name: State/Province:

Country Code:

Attach File

Any Type

Table 1.11 Information common to most new template forms

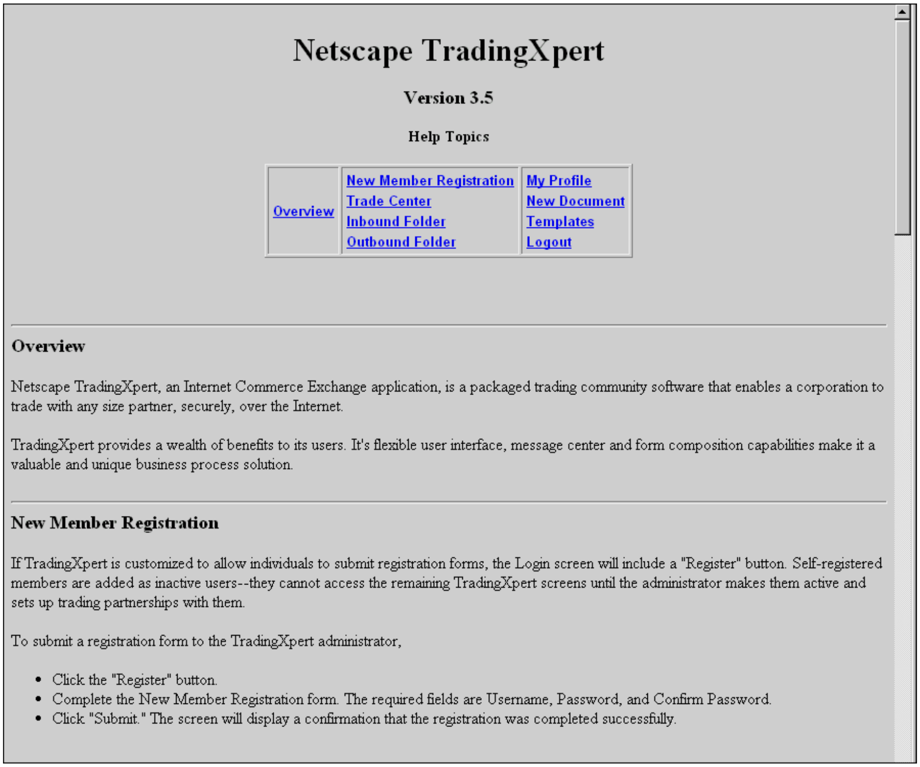
Item	Description
[Document Type]	
[Data fields]	The information displayed on the new template form varies, depending on the document type being created as well as whether you have created the template from scratch or from a template.
Attach File	
Input field	Enter the fully qualified pathname for the file to be attached to the template.
Browse	Click this button to locate the file on your desktop.
File type drop-down menu	Choose the type of file to attach. Note that the file type must match the file type for the ECXpert partnership.
[Tasks]	
Add Line Item	Click this button to add data fields for a new line item in the template.
Delete Line Item	Click this button to delete the selected line items from the template. Note that you cannot delete the only line item in a template.
Submit Document	Click to send the document to TradingXpert.
Save as Template	Click this button to save the current document as a template.

Note Template names should not include spaces. If you include spaces when you name a new template, you will be prompted to rename the template.

Help

The Help link at the top of the screen sends the user to an online help page (Figure 1.14). From this page you can click the link, under "Help Topics," for the TradingXpert screen for which you want to get help.

Figure 1.14 Online help, initial screen



Netscape TradingXpert

Version 3.5

Help Topics

Overview	New Member Registration Trade Center Inbound Folder Outbound Folder	My Profile New Document Templates Logout

Overview

Netscape TradingXpert, an Internet Commerce Exchange application, is a packaged trading community software that enables a corporation to trade with any size partner, securely, over the Internet.

TradingXpert provides a wealth of benefits to its users. It's flexible user interface, message center and form composition capabilities make it a valuable and unique business process solution.

New Member Registration

If TradingXpert is customized to allow individuals to submit registration forms, the Login screen will include a "Register" button. Self-registered members are added as inactive users--they cannot access the remaining TradingXpert screens until the administrator makes them active and sets up trading partnerships with them.

To submit a registration form to the TradingXpert administrator,

- Click the "Register" button.
- Complete the New Member Registration form. The required fields are Username, Password, and Confirm Password.
- Click "Submit." The screen will display a confirmation that the registration was completed successfully.

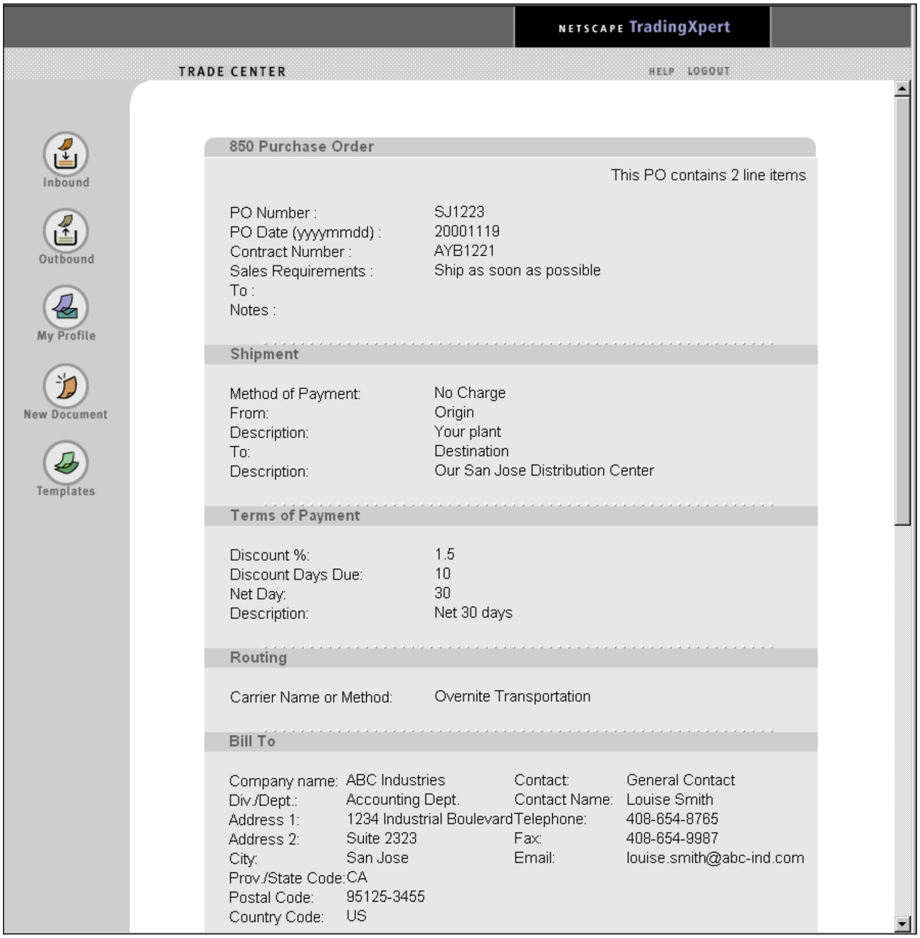
View

The View screen (Figure 1.15) can be reached through the View button for a document that is listed on either the Inbound or Outbound screen.

The View screen enables the TradingXpert user to look at the details for any document that has been selected. The format appearing on your screen may differ from the format in the example in Figure 1.15, depending on the document type and your organization's defined format.

Note To save files attached to inbound or outbound documents, click the name of the attachment and save it to your hard drive or other media, as prompted.

Figure 1.15 View screen (from View button on Inbound or Outbound screen)



850 Purchase Order

This PO contains 2 line items

PO Number :	SJ1223
PO Date (yyyymmdd) :	20001119
Contract Number :	AYB1221
Sales Requirements :	Ship as soon as possible
To :	
Notes :	

Shipment

Method of Payment:	No Charge
From:	Origin
Description:	Your plant
To:	Destination
Description:	Our San Jose Distribution Center

Terms of Payment

Discount %:	1.5
Discount Days Due:	10
Net Day:	30
Description:	Net 30 days

Routing

Carrier Name or Method: Overnite Transportation

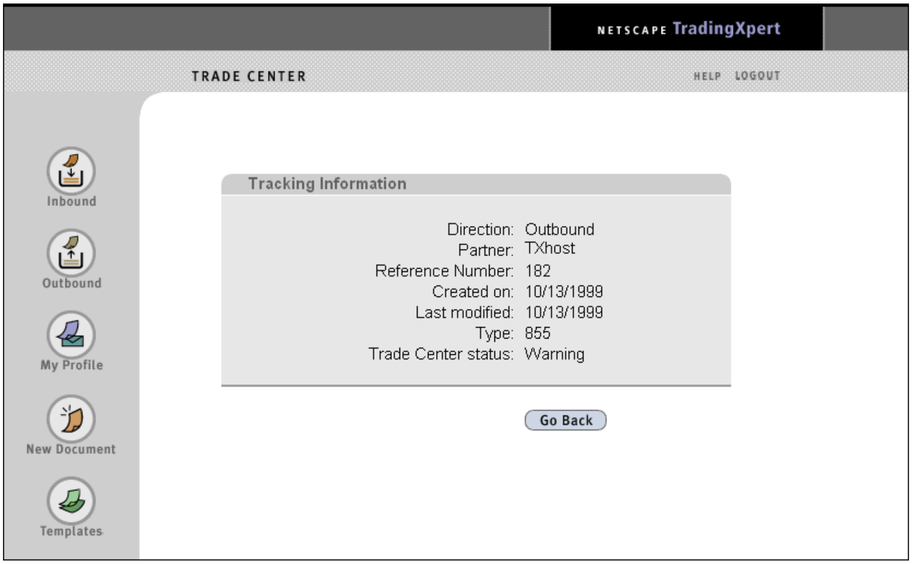
Bill To

Company name:	ABC Industries	Contact:	General Contact
Div./Dept.:	Accounting Dept.	Contact Name:	Louise Smith
Address 1:	1234 Industrial Boulevard	Telephone:	408-654-8765
Address 2:	Suite 2323	Fax:	408-654-9987
City:	San Jose	Email:	louise.smith@abc-ind.com
Prov./State Code:	CA		
Postal Code:	95125-3455		
Country Code:	US		

Tracking Information

The Tracking Information screen allows the TradingXpert user to view various tracking data for a selected Inbound or Outbound document.

Figure 1.16 Tracking Information screen



Tracking Information

Direction:	Outbound
Partner:	TXhost
Reference Number:	182
Created on:	10/13/1999
Last modified:	10/13/1999
Type:	855
Trade Center status:	Warning

Table 1.12 Information on the Tracking Information screen

Item	Description
Direction	Whether the document is incoming to or outgoing from the logged-in user.
Partner	Trading partner User ID.
Reference Number	The ECXpert Tracking ID for the document.
Created on	Date on which the document was created.
Last modified	Date on which the document most recently had changes incorporated.
Type	Type of document.
Trade Center Status	Document status within TradingXpert.
Go Back button	Returns user to the previous screen.

Chapter

2

Customizing TradingXpert's Capabilities

TradingXpert is not a comprehensive turnkey solution, but a paradigm that can be altered to match your own needs. This chapter describes the extension capabilities of the TradingXpert System. The following topics are presented:

- Viewing Additional Types of Documents
- Turning Around Additional Types of Documents
- Creating Custom Forms
- Using Custom Forms to View Documents from Different Senders
- Enabling Certificate-Based Logins
- Enabling Events
- Enabling New Member Self-Registration
- Modifying File Attachment Options for Document Forms
- Compiling Java Code (Solaris)

Note Throughout this chapter, when a file name is applicable to all operating systems, the Solaris version of the file path is given (e.g., *$NASDIR/APPS).* If you are using a different operating system, modify the path name according to the conventions of your system (e.g., *<NASDIR>\APPS* for Windows NT).

Viewing Additional Types of Documents

The EDI document types listed in Appendix A can be viewed without customizing TradingXpert. To view other types, you need:

- An ECXpert map to translate the document to the TradingXpert Format. Place the map with the other maps in ECXpert's *$NSBASE/maps* directory.
- One or more partnerships in ECXpert that use this map.
- A form to view the document. The form must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory, and it must have the same name as the **`doc type`** in the partnerships you created. For example, the *850.html* form is used to view documents of type 850.
- An additional entry for the `FX.viewableDocTypes` parameter in the *FXproperties* file, located in your *$NASDIR/APPS/FX/common/* directory.
- Optionally, an additional `FX.viewableDocTypes.<`*`document_type`*`>` parameter in the *FXproperties* file to specify a name to be displayed to users.

Turning Around Additional Types of Documents

We'll use the terminology *source document* and *destination document* to describe the inbound document a trading partner sent to the user and the document that the user sends back to the trading partner, respectively. To perform document turnaround, you need the following pieces.

- An ECXpert map to translate the source document to the TradingXpert Format. Place the map with the other maps in ECXpert's *$NSBASE/maps* directory.
- One or more partnerships in ECXpert that use this map.
- A data value file to assign values to data. (Optional; use only for from-scratch documents.) The form must reside in the *$NASDIR/APPS/FX/forms/<locale>/from_scratch_data* directory, and it must have the same name as the **`doc type`** in the partnerships you created. For example, *850.dat* defines the data values for 850 (Invoice) documents.

- A rules file for from-scratch documents to map data values from *cinfo.dat* to data labels in the document being created; this file is necessary only if the source document will be created from scratch. The form must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory, and the name must consist of a zero (0) followed by a dash and the name of the **`doc type`** in the partnerships you created. For example, *0-810.rules* defines the values that are assigned to data in from-scratch purchase orders.

- A form to view the source document. (This form is not mandatory, but it is helpful for the trading partners to see what they are turning around.) The form must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory, and it must have the same name as the **`doc type`** in the partnerships you created. For example, the *850.html* form is used to view documents of type 850.

- A rules file for turned-around documents to define how the fields from the source document are copied to the destination document. For example, *850-810.rules* defines how data from purchase orders is turned around into data on corresponding invoices.

- A form to view or edit the destination document before submitting it to ECXpert. The form must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory, and it must have the same name as the **`doc type`** in the partnership you created, plus a *_e* (signifying "editable"). For example, TradingXpert has an *810_e.html* form. It is up to you whether or not you want the user to actually edit anything in the document before submitting it. For example, the *810_e.html* form allows some of the fields to be edited, the *810.html* form does not allow any fields to be edited.

- An ECXpert map to translate the submitted file from TradingXpert format to the appropriate EDI standard or application data of the trading partner to whom you are sending the destination document. Place the map with the other maps in ECXpert's *$NSBASE/maps* directory.

- A companion order file to ease the job of the ECXpert mapper. (Optional) For example, *810.order* orders the elements of an invoice for submission to ECXpert.

Here are the files provided by TradingXpert to perform document turnaround on an 850 Purchase Order, turning it into an 810 Invoice.

File	Description
850html.sun (Solaris only) or *850html.mmc (NT only)*	ECXpert map to translate an X.12 850 Purchase Order to TradingXpert format.
850.html	Form to view Purchase Order (optional, but good to have).
810.dat	Defines data values for the document type.
0-810.rules	Maps data values from *810.dat* to data labels in a Purchase Order.
850-810.rules	Defines document turnaround from Purchase Order to Invoice.
810_e.html	Template so trading partner can view or edit the Invoice before submitting it.
html810.sun (Solaris only) or *html810.mmc (NT only)*	ECXpert map to translate TradingXpert format Invoice to X.12 810 format.
810.order	Companion file to ease the burden on *html810.sun* (Solaris only) or *html810.mmc* (NT only).

To accompany these files, there would be two partnerships in ECXpert. For two partners *TXhost* and *webuser1*, the relationships would look like this:

Sender	Receiver	Doc type	Map
TXhost	webuser1	850	*850html.sun* (Solaris only) or *850html.mmc* (NT only)
webuser1	TXhost	810	*html810.sun* (Solaris only) or *html810.mmc* (NT only)

Creating Custom Forms

Customization Example

The TradingXpert CD includes some sample files that you can use as templates when you are customizing TradingXpert. The sample customization involves receiving contact information from a non-EDI application, allowing the user to add phone numbers, and then sending the information to update a database.

The sample customization files are located in the *$NASDIR*/*APPS*/*FX*/*maps_etc*/*Customization* directory.

To implement this customization, you would perform the following sequence of tasks:

1. Assign a name (`cinfo`) to the data type to be processed.
2. Make entries for your data type in the *FXproperties* file.
3. Create an application data format file (*cinfo.app*) to help you map sample incoming data to data labels. (Optional)
4. Create a data value file (*cinfo.dat*) for from-scratch documents to assign data values for the document type. (Optional; use if creating from-scratch documents)
5. Create a rules file for from-scratch documents (*0-cinfo.rules*) to map data values from the data file (*cinfo.dat)* to data labels in the document being created. (Optional)
6. Create an HTML page (*cinfo.html*) to display the incoming data.
7. Create a rules file for turned-around documents (*cinfo-cinfo.rules*) to map incoming data labels to the outgoing data labels.
8. Create an HTML page (*cinfo_e.html*) that allows editing of the phone numbers in the outgoing data, and then updates a database with the edited information.
9. Create an order file (*cinfo.order*) to order the outgoing data. (Optional)
10. Set up partnerships and service lists in ECXpert to support the exchange of documents between the TradingXpert host and the web user client.

After you complete these tasks, a user can view an incoming document (*cinfo.html*) and, from it, create a turnaround document to return to the sender with the help of a rules file (*cinfo-cinfo.rules*) that populates the editable document (*cinfo_e.html*). When the user submits the turnaround document, an order file (*cinfo. order*) orders the elements of the document for submission to ECXpert.

Each of these tasks is detailed in the sections that follow, using the TradingXpert sample customization files that are provided.

Note If you want to add the ability to create templates from your customized document, you must also complete additional tasks. See Steps 5-7 in "Customization Summary" on page 76 for detailed instructions.

Assigning a Name to the Data Type

Before you start creating files to customize TradingXpert, you should assign a name to the data type that is to be processed. You will use this name as part of the file names in the series of customization files that you create, and you must specify it as the Document Type for the ECXpert partnership that is involved. The sample customization files use the name "cinfo."

The name should:

- Include only letters (not case sensitive) and numbers; no spaces or punctuation characters.
- Easily identify the data type that your customization files are processing. If the data type is an EDI 850, you might use `850` as the name. The sample customization files will be used to process contact information and thus use the name "cinfo."
- Be short enough to easily incorporate into the file names. You will create most of the file names by simply adding an extension onto the name you have chosen. From "cinfo," the sample customization files created are *cinfo.app*, *cinfo.html*, *cinfo-cinfo.rules*, *cinfo.order*, and *cinfo_e.html.*
- Be unique; if you have several customizations processing the same data type, you should make the names unique by adding a short suffix. For example, if you had several customizations processing contact information, you could use "cinfo," "cinfo2," "cinfoUS," and so on.

Note For simplicity, the sample customization files use `cinfo` for both the inbound and outbound (relative to the web user client) document types. If you want to use different custom document types for inbound and outbound documents, you would have to assign a name to the second document type and create an application data format for it as well. If you want to use one of the pre-defined EDI document types, these files are already provided.

Making Entries for Your Data Type in the FXproperties File

In order for your custom application data format to be processed correctly by TradingXpert, you must add some entries to the *FXproperties* file, located in your *$NASDIR/APPS/FX/common/* directory. For the sample customization, using `cinfo` as the data format name, you would add the bolded entries shown in Figure 2.1:

Figure 2.1 FXproperties file entries to support your customization

```
...
FX.viewableDocTypes: 862;860;856;855;850;843;840;830;810;214;ORDERS;
INVOIC;cinfo

FX.viewableDocTypes.850: PO
FX.viewableDocTypes.214: Ship Status
FX.viewableDocTypes.cinfo: Contact Information

FX.outboundDocTypes: 862;860;856;850;810;214;cinfo
...
```

- `FX.viewableDocTypes` list—adding "cinfo" here specifies that a template (*cinfo.html*) exists for displaying incoming `cinfo` data.
- `FX.viewableDocTypes.cinfo`—specifies a name ("Contact Information") to be displayed to users for the `cinfo` data type. This entry is optional; if you omit it, the data type name will be displayed to users.
- `FX.outboundDocTypes` list—adding "cinfo" here specifies that a template (*cinfo_e.html*) exists for displaying and editing outgoing `cinfo` data.

If documents based on your application data format can be created from scratch by a TradingXpert user (vs. only being created in response to an incoming document), you would also add it to the `FX.fromScratchDocTypes` entry, which is a subset of the `FX.outboundDocTypes` list:

```
...
FX.fromScratchDocTypes: 860;850;810;214;cinfo
...
```

Creating an Application Data Format File (Optional)

The logical first task to perform after you have selected a data type name is to create an application data format file that maps the incoming data to data labels, which you can then use to manipulate the data. In the TradingXpert sample customization files, the *cinfo.app* file, shown in Figure 2.2, serves this purpose.

An application data format file is optional, but it uses the same data format that TradingXpert uses to display files and can thus be used as a guide to create the *.html*, *.order*, and *.rules* files. The actual files processed through TradingXpert will be created automatically under the control of the ECXpert trading partnership under which the documents are exchanged.

Figure 2.2 The sample application data format file, cinfo.app

```
TAName_1_1=Naren
TAAddress_1_1=501 East Middlefield
TACity_1_1=Mtn View
TAState_1_1=CA
TAPostalCode_1_1=94040
TEL_1.AreaCode_1_1=800
TEL_1.Number_1_1=555-1234
TEL_2.AreaCode_1_1=888
TEL_2.Number_1_1=555-6789
```

The application data format file consists of a list of name-value pairs. Data labels on the left are associated with the values on the right and must conform to the following specification:

- Labels for non-repeating items (`TAName_1_1` through TA`PostalCode_1_1` in *cinfo.app*) must end with "`_1_1`" or "`_1_2,`" or something that has "_n_," where n is a single digit. The general form is `keyA_n_keyB`. In EDI, this could be `N4_1_4` for country code, where N4/4 is the EDI element.

- Labels for repeating items (last four lines for phone numbers in *cinfo.app*) must be prefixed by a label that ends in "`_#`" where # is the row number. These row numbers must increment in an ascending sequence with no breaks ("`TEL_1`" preceding "`TEL_2`" in *cinfo.app*).

- Items that are logically part of the same row must have the same prefix label ("`TEL_1`" vs. "`TEL_2`" in *cinfo.app*).

- Repeating items may be nested one additional level beyond what is shown in *cinfo.app*. For example, multiple extensions could be included for each phone number. The additional level of nesting adds an additional prefix label that ends in “`_#,`” where # is a grouping number. It is assumed that there are data elements at each level of nesting. This would look something like the example below:

```
...
TEL_1.AreaCode_1_1=800
TEL_1.Number_1_1=555-1234
TEL_1.EXT_1.Extension_1_1=8365
TEL_1.EXT_2.Extension_1_1=8366
TEL_2.AreaCode_1_1=888
TEL_2.Number_1_1=555-6789
TEL_2.EXT_1.Extension_1_1=9355
...
```

Creating a Data Value File for From-Scratch Documents (Optional)

If `cinfo` documents have to be created from scratch by web users, you must create a data value file to assign data values for the document type; it must reside in the *$NASDIR/APPS/FX/forms/<locale>/from_scratch_data* directory. In the TradingXpert sample customization directory, the *cinfo.dat* file, shown in Figure 2.4, serves this purpose. The fields on the left indicate the data labels for the document type.

Figure 2.3 Sample data value file for from-scratch documents, *cinfo.dat*

```
TAName_1_1=
TAAddress_1_1=
TACity_1_1=
TAState_1_1=
TAPostalCode_1_1=
TEL_1.AreaCode_1_1=650
TEL_1.Number_1_1=
```

Creating a Rules File for From-Scratch Documents (Optional)

A rules file is necessary to map data values from the data file (*cinfo.dat)* to data labels in the document being created; it must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory. In the TradingXpert sample customization directory, the *0-cinfo.rules* file, shown in Figure 2.4, serves this purpose. The fields on the left correspond to the fields in the editable outgoing file, *cinfo_e.html*, and the fields on the right correspond to fields in the data value file, *cinfo.dat*.

Figure 2.4 Sample rules file, *0-cinfo.rules*

```
TAName_1_1=TAName_1_1
TAAddress_1_1=TAAddress_1_1
TACity_1_1=TACity_1_1
TAState_1_1=TAState_1_1
TAPostalCode_1_1=TAPostalCode_1_1
TEL_*.AreaCode_1_1=TEL_*.AreaCode_1_1
TEL_*.Number_1_1=TEL_*.Number_1_1
```

Notes When your customization involves different document types for inbound and outbound (relative to the web user client) documents, the name of the rules file must consist of the inbound data type name before the dash, and the outbound data type name after the dash. When you create rules files for from-scratch documents, the inbound data type before the dash should be zero (0) to indicate that the file is created from scratch (e.g., *0-cinfo.rules*).

The rules file simply pairs labels data values from *cinfo.dat* to labels for the document being created:

- For repeating items in loops, you should use the asterisk wildcard character to indicate an unspecified total number of items. As shown above (last two lines for TEL_*), the number of asterisks and their positions must match exactly on the right and left sides of the equal sign. If you had an additional level of nesting—for example, multiple extension numbers per phone number, the last two lines would be extended into three lines that look something like the example below:

```
...
TEL_*.AreaCode_1_1=TEL_*.AreaCode_1_1
TEL_*.Number_1_1=TEL_*.Number_1_1
TEL_*.EXT_*.Extension_1_1=TEL_*.EXT_*.Extension_1_1
```

Note that repeating items are looped in column-major order in the rules file. In the sample files, this means that all the area codes are processed in one loop, all numbers are processed in a second loop, and all the extensions are processed in a third loop.

Creating an HTML Page to Display the Incoming Data

Next you need an HTML page to display the incoming data; this page must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory. In the TradingXpert sample customization directory, the *cinfo.html* file, shown in Figures 2.5-2.11, serves this purpose. This file uses a table to control the layout of fields and labels.

Keep in mind the following when you are examining this file:

- The file is broken into sections, with notes on the important details in each section following the HTML.
- The longer lines of HTML have been wrapped, with continuation lines indented, to improve readability.
- Details of the HTML markup that are purely "cosmetic"—not affecting the function of the customization example—may change in the copy of the file that is on your TradingXpert installation CD.
- The required NAS markup is beyond the scope of this document to explain. For information on NAS markup, consult the *Netscape Application Server System Administration Guide.*

Figure 2.5 Sample HTML page to display incoming data (*cinfo.html*)—Part 1

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<BASE HREF="%gx type=cell id=BASEHREF%%/gx%">
</head>

<body bgcolor="#FFFFFF" text="#000000" link="#336699" vlink="#6699CC" alink="#996600">
<p>%gx type=tile id=defaultValues% %gx type=tile id=DocumentContent%</p>

<form METHOD="POST" ACTION=
      "/cgi-bin/gx.cgi/GUIDGX-{6400c704-54ac-11d2-b7c0-00104b605a87}">
  <input type="hidden" name="PARTNER" value="%gx type=cell id=PARTNER%%/gx%">
  <input type="hidden" name="DOC_TYPE" value="cinfo">
  <input type="hidden" name="FILE_TYPE" value="cinfo">
  <input type="hidden" name="PARTNER_QUAL" value="%gx type=cell id=PARTNER_QUAL%%/gx%">
  <input type="hidden" name="SENDER_QUAL" value="%gx type=cell id=SENDER_QUAL%%/gx%">
  <input type="hidden" name="PARTNER_ADDR" value="%gx type=cell id=PARTNER_ADDR%%/gx%">
  <input type="hidden" name="SENDER_ADDR" value="%gx type=cell id=SENDER_ADDR%%/gx%">
  <img src="clear.gif" width="3" height="55" align="top">
  <img src="clear.gif" width="48" height="3" align="left">
```

In the first HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

The only customization items you can *change* in this section are the two occurrences of "cinfo." You would replace these with the application data format name you have chosen for your customization.

You can *add* additional fields to this section as necessary, using any one of the `<input type...>` tags as a model. These would be fields that your customization required, not additional fields from ECXpert.

Figure 2.6 Sample HTML page to display incoming data (*cinfo.html*)—Part 2

```
<table border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><img src="corner_l_g.gif" width = "9" height="19"></td>
    <td bgcolor="#CCCCCC"><img src="clear.gif" width="15" height="3"></td>
    <td bgcolor="#CCCCCC" colspan="4">
         <font face="Arial, Helvetica, sans-serif"><b>
         <font color="#666666">Contact Info</font></b></font></td>
    <td><img src="corner_r_g.gif" width="9" height="19"></td>
  </tr>
  <tr>
    <td bgcolor="#FFFFFF"><img src="white.gif" width="9" height="1"></td>
    <td bgcolor="#FFFFFF"><img src="white.gif" width="5" height="1"></td>
    <td bgcolor="#FFFFFF" colspan="4"><img src="white.gif" width="10" height="1"></td>
    <td bgcolor="#FFFFFF"><img src="white.gif" width="9" height="1"></td>
  </tr>
  <tr align="right" valign="top">
    <td align="left" bgcolor="#E6E6E6">
         <img src="clear.gif" width="3" height="25"></td>
    <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
    <td bgcolor="#E6E6E6" align="left" valign="middle">
         <img src="clear.gif" width="156" height="3"></td>
    <td align="left" bgcolor="#E6E6E6" colspan="1" valign="middle">
         <img src="clear.gif" width="110" height="3"></td>
    <td bgcolor="#E6E6E6" align="left" valign="middle">
         <img src="clear.gif" width="150" height="3"></td>
    <td align="left" bgcolor="#E6E6E6" valign="middle">
         <img src="clear.gif" width="101" height="3"></td>
    <td bgcolor="#E6E6E6">&nbsp;</td>
  </tr>
```

The second HTML section, above, begins the table and contains three table rows that are purely cosmetic. You may change the HTML in this section any way you like for your customization.

Figure 2.7 Sample HTML page to display incoming data (*cinfo.html*)—Part 3

```
    <tr align="right" valign="top">
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td bgcolor="#E6E6E6" align="left" valign="middle">
           <font face="Arial, Helvetica, sans-serif">Contact Name:</font></td>
      <td align="left" bgcolor="#E6E6E6" colspan="1" valign="middle">
           <font face="Arial, Helvetica, sans-serif">
           %GX type=cell id=DocumentContent.TAName_1_1%%/GX%</font></td>
      <td bgcolor="#E6E6E6" align="left" valign="middle">
           <font face="Arial, Helvetica, sans-serif">Address:</font></td>
      <td align="left" bgcolor="#E6E6E6" valign="middle">
           <font face="Arial, Helvetica, sans-serif">
           %GX type=cell id=DocumentContent.TAAddress_1_1%%/GX%</font></td>
      <td bgcolor="#E6E6E6">&nbsp;</td>
```

In the third HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

This section contains a table row that displays two fields, the name and address labels taken from the sample application data format file. For any top level, non-repeating data items, this section illustrates the NAS markup that you must use to display the field. In this section you would substitute field names from your application data format file for `TAName_1_1` and `TAAddress_1_1`.

Figure 2.8 Sample HTML page to display incoming data (*cinfo.html*)—Part 4

```
    </tr>
    <tr align="right" valign="top">
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td bgcolor="#E6E6E6" align="left" valign="middle">
           <font face="Arial, Helvetica, sans-serif">City:</font></td>
      <td align="left" bgcolor="#E6E6E6" colspan="3" valign="middle">
           <font face="Arial, Helvetica, sans-serif">
           %GX type=cell id=DocumentContent.TACity_1_1%%/GX%</font></td>
      <td bgcolor="#E6E6E6">&nbsp;</td>
    </tr>
    <tr align="right" valign="top">
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td bgcolor="#E6E6E6" align="left" valign="middle">
           <font face="Arial, Helvetica, sans-serif">State :</font></td>
      <td align="left" bgcolor="#E6E6E6" colspan="3">
           <font face="Arial, Helvetica, sans-serif">
           %GX type=cell id=DocumentContent.TAState_1_1%%/GX%</font></td>
      <td bgcolor="#E6E6E6">&nbsp;</td>
    </tr>
    <tr align="right" valign="top">
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td bgcolor="#E6E6E6" align="left" valign="middle">
           <font face="Arial, Helvetica, sans-serif">Zip Code:</font></td>
      <td align="left" bgcolor="#E6E6E6" colspan="1" valign="middle">
           <font face="Arial, Helvetica, sans-serif">
           %GX type=cell id=DocumentContent.TAPostalCode_1_1%%/GX%</font></td>
      <td bgcolor="#E6E6E6">&nbsp;</td>
    </tr>
```

In the fourth HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

This section contains three table rows, each displaying one field, the city, state, and postal code labels taken from the sample application data format file. In this section you would substitute field names from your application data format file for `TACity_1_1` and TA`State_1_1` and TA`PostalCode_1_1`.

Figure 2.9 Sample HTML page to display incoming data (*cinfo.html*)—Part 5

```
    <tr align="right" valign="bottom" bgcolor="#CCCCCC">
      <td align="left" valign="top"><img src="clear.gif" width = "3" height="19"></td>
      <td align="left" valign="top">&nbsp;</td>
      <td align="left" valign="middle" colspan="4">
           <font face="Arial, Helvetica, sans-serif">
           <b><font color="#666666">Telephone Numbers</font></b></font>
      </td>
```

The fifth HTML section, above, contains one table row that displays a label (`Telephone Numbers`) for the phone numbers that will appear below. You may change the HTML in this section any way you like for your customization.

Figure 2.10 Sample HTML page to display incoming data (*cinfo.html*)—Part 6

```
      <td>&nbsp;</td>
    </tr>
%GX type=tile id=TEL_loop% <!-- TEL_loop -->
    <tr align="right valign="bottom">
      <td align="left valign="top" bgcolor="#E6E6E6">
           <img src="clear.gif" width="3" height="25"></td>
      <td align="left valign="top" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left valign="middle" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left valign="top" bgcolor="#E6E6E6" colspan="3">&nbsp;</td>
      <td bgcolor="#E6E6E6">&nbsp;</td>
    </tr>
    <tr align="right" valign="top">
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td bgcolor="#E6E6E6" align="left" valign="middle">
           <font face="Arial, Helvetica, sans-serif">
           Number-%GX type=cell id=TEL_loop.rowid%%/GX%:</font></td>
      <td align="left" bgcolor="#E6E6E6" colspan="3" valign="middle">
           <font face="Arial, Helvetica, sans-serif">
           (%GX type=cell id=TEL_loop.AreaCode_1_1%%/GX%)
           %GX type=cell id=TEL_loop.Number_1_1%%/GX%</font></td>
      <td bgcolor="#E6E6E6">&nbsp;</td>
    </tr>
%/GX% <!-- TEL_loop -->
```

In the sixth HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.

- Plain courier text may be changed any way you like to control the page display.

This section contains the loop that processes the multiple phone numbers. It contains two table rows, the first of which is purely cosmetic. The second table row displays the two phone number fields. In this section you would substitute field names from your application data format file for TEL_loop.AreaCode_1_1 and TEL_loop.Number_1_1. You would also want to change the "TEL" in the comments.

Note how the loop data labels are created from the data labels for repeating items in your application data format file: in the top level part of the label the "_1" is replaced by "_loop" (TEL_1.AreaCode_1_1 changes to TEL_loop.AreaCode_1_1 and TEL_1.Number _1_1 changes to TEL_loop.Number_1_1 in our example).

If you had an additional level of nesting in your repeating items, such as multiple extension numbers for a single phone number, the looping portion of this section would look something like this:

```
...
  %GX type=tile id=TEL_loop% <!-- TEL_loop -->
...
  %GX type=tile id=TEL_$ROWID_loop% <!-- TEL_$ROWID_loop -->
  %GX type=tile id=EXT_loop% <!-- EXT_loop -->
...
      (%GX type=cell id=TEL_loop.AreaCode_1_1%%/GX%)
      %GX type=cell id=TEL_loop.Number_1_1%%/GX%
      Extension-%GX type=cell id=EXT_loop.EXT_loop.rowid%%/GX%:
           %GX type=cell id=EXT_loop.EXT_1_1%%/GX%
...
  %/GX% <!-- EXT_loop -->
  %/GX% <!-- TEL_$ROWID_loop -->
...
  %/GX% <!-- TEL_loop -->
...
```

Figure 2.11 Sample HTML page to display incoming data (*cinfo.html*)—Part 7

```
  <tr><!-- Button to create a turnaround to change telephone numbers -->
     <td><A HREF="/cgi-bin/gx.cgi/GUIDGX-{15048130-5955-11d2-b7cb-00104b605a87}
     ?FROM_TYPE=%gx type=cell id="DOC_TYPE"%%/gx%
     &TO_TYPE=cinfo&PARTNER=%gx type=cell id=PARTNER%%/gx%
     &DOC_PATH=%gx type=cell id="DOC_PATH"%%/gx%">
     <img src="btn_change.gif" ALT="Change Telephone Numbers" width="102" height="17"
      border="0"></A></td>
  </tr>
  </table>
</form>
%/GX% <!-- DocumentContent -->
%/GX% <!-- defaultValues -->
</body>
</html>
```

In the seventh HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

This section contains the button that takes the user to a second HTML page where the phone numbers can be changed. It contains a single table row, plus the HTML markup to end the table and the form, plus the NAS markup to end the DocumentContent and DefaultValues tiles.

Creating a Rules File for Turned-Around Documents

A rules file is necessary to map the incoming data labels to the outgoing data labels; it must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory. In the TradingXpert sample customization directory, the *cinfo-cinfo.rules* file, shown in

Figure 2.12, serves this purpose. The fields on the left correspond to the fields in the editable outgoing file, *cinfo_e.html*, and the fields on the right correspond to fields in the incoming file, *cinfo.html*.

Figure 2.12 Sample rules file, *cinfo-cinfo.rules*

```
TAName_1_1=TAName_1_1
TAAddress_1_1=TAAddress_1_1
TACity_1_1=TACity_1_1
TAState_1_1=TAState_1_1
TAPostalCode_1_1=TAPostalCode_1_1
TEL_*.AreaCode_1_1=TEL_*.AreaCode_1_1
TEL_*.Number_1_1=TEL_*.Number_1_1
```

Notes When your customization involves different document types for inbound and outbound (relative to the web user client) documents, the name of the rules file must consist of the inbound data type name before the dash, and the outbound data type name after the dash.

The rules file simply pairs labels for the incoming data, as defined in your application data format file, with labels for the outgoing data:

- The labels for the outgoing data must conform to the same specifications described for labels in the application data format file.
- For repeating items in loops, you should use the asterisk wildcard character to indicate an unspecified total number of items. As shown above (last two lines for TEL_*), the number of asterisks and their positions must match exactly on the right and left sides of the equal sign. If you had an additional level of nesting—for example, multiple extension numbers per phone number, the last two lines would be extended into three lines that look something like the example below:

```
...
TEL_*.AreaCode_1_1=TEL_*.AreaCode_1_1
TEL_*.Number_1_1=TEL_*.Number_1_1
TEL_*.EXT_*.Extension_1_1=TEL_*.EXT_*.Extension_1_1
```

Note that repeating items are looped in column-major order in the rules file. In the sample files, this means that all the area codes are processed in one loop, all numbers are processed in a second loop, and all the extensions are processed in a third loop.

Creating an HTML Page for Adding Data and Sending It to TradingXpert

You need to create an HTML page for editing the outgoing (relative to the web user client) data and updating a database with the edited information. This file must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory. In the TradingXpert sample customization directory, the *cinfo_e.html* file, shown in Figures 2.13-2.19, serves this purpose.

Figure 2.13 Sample HTML page to edit outgoing data (*cinfo_e.html*)—Part 1

```
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<BASE HREF="%gx type=cell id=BASEHREF%%/gx%">
</head>

<body bgcolor="#FFFFFF" text="#000000" link="#336699" vlink="#6699CC" alink="#996600">
<p>%gx type=tile id=defaultValues% %gx type=tile id=DocumentContent%</p>

<form METHOD="POST" ACTION=
      "/cgi-bin/gx.cgi/GUIDGX-{6400c704-54ac-11d2-b7c0-00104b605a87}">
  <input type="hidden" name="PARTNER" value="%gx type=cell id=PARTNER%%/gx%">
  <input type="hidden" name="DOC_TYPE" value="cinfo">
  <input type="hidden" name="FILE_TYPE" value="cinfo">
  <input type="hidden" name="PARTNER_QUAL" value="%gx type=cell id=PARTNER_QUAL%%/gx%">
  <input type="hidden" name="SENDER_QUAL" value="%gx type=cell id=SENDER_QUAL%%/gx%">
  <input type="hidden" name="PARTNER_ADDR" value="%gx type=cell id=PARTNER_ADDR%%/gx%">
  <input type="hidden" name="SENDER_ADDR" value="%gx type=cell id=SENDER_ADDR%%/gx%">

  <input type="hidden" name="looptoaddto" value="TEL_">
  <input type="hidden" name="numbertoadd" value="1">

  <img src="clear.gif" width="3" height="55" align="top">
       <img src="clear.gif" width="48" height="3" align="left">
```

In the first HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

The only customization items you can *change* in this section are the two occurrences of "cinfo" and the one occurrence of "TEL." You would replace these with the application data format name you have chosen for your customization and the incoming data label for telephone numbers.

You can *add* additional fields to this section as necessary, using any one of the `<input type...>` tags as a model. These would be fields that your customization required, not additional fields from ECXpert.

Figure 2.14 Sample HTML page to edit outgoing data (*cinfo_e.html*)—Part 2

```
<table border="0" cellspacing="0" cellpadding="0">
  <tr>
    <td><img src="corner_l_g.gif" width = "9" height="19"></td>
    <td bgcolor="#CCCCCC"><img src="clear.gif" width="15" height="3"></td>
    <td bgcolor="#CCCCCC" colspan="4">
         <font face="Arial, Helvetica, sans-serif"><b>
         <font color="#666666">Contact Info</font></b></font></td>
    <td><img src="corner_r_g.gif" width="9" height="19"></td>
  </tr>
  <tr>
    <td bgcolor="#FFFFFF"><img src="white.gif" width="9" height="1"></td>
    <td bgcolor="#FFFFFF"><img src="white.gif" width="5" height="1"></td>
    <td bgcolor="#FFFFFF" colspan="4"><img src="white.gif" width="10" height="1"></td>
    <td bgcolor="#FFFFFF"><img src="white.gif" width="9" height="1"></td>
  </tr>
  <tr align="right" valign="top">
    <td align="left" bgcolor="#E6E6E6"><img src="clear.gif" width="3" height="25"></td>
    <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
    <td bgcolor="#E6E6E6" align="left" valign="middle">
         <img src="clear.gif" width="156" height="3"></td>
    <td align="left" bgcolor="#E6E6E6" colspan="1" valign="middle">
         <img src="clear.gif" width="110" height="3"></td>
    <td bgcolor="#E6E6E6" align="left" valign="middle">
         <img src="clear.gif" width="150" height="3"></td>
    <td align="left" bgcolor="#E6E6E6" valign="middle">
         <img src="clear.gif" width="101" height="3"></td>
    <td bgcolor="#E6E6E6">&nbsp;</td>
  </tr>
```

The second HTML section, above, begins the table and contains three table rows that are purely cosmetic. You may change the HTML in this section any way you like for your customization.

Figure 2.15 Sample HTML page to edit outgoing data (*cinfo_e.html*)—Part 3

```
    <input type="hidden" name="TAName_1_1"
         value="%GX type=cell id=DocumentContent.TAName_1_1%%/GX%">
    <input type="hidden" name="TAAddress_1_1"
         value="%GX type=cell id=DocumentContent.TAAddress_1_1%%/GX%">
    <tr align="right" valign="top">
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
      <td bgcolor="#E6E6E6" align="left" valign="middle">
           <font face="Arial, Helvetica, sans-serif">Contact Name:</font></td>
      <td align="left" bgcolor="#E6E6E6" colspan="1" valign="middle">
           <font face="Arial, Helvetica, sans-serif">
           %GX type=cell id=DocumentContent.TAName_1_1%%/GX%</font></td>
      <td bgcolor="#E6E6E6" align="left" valign="middle">
           <font face="Arial, Helvetica, sans-serif">Address:</font></td>
      <td align="left" bgcolor="#E6E6E6" valign="middle">
           <font face="Arial, Helvetica, sans-serif">
           %GX type=cell id=DocumentContent.TAAddress_1_1%%/GX%</font></td>
      <td bgcolor="#E6E6E6">&nbsp;</td>
    </tr>
```

In the third HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

This section contains a table row that displays two fields, the name and address labels taken from the left side of the sample rules file. For any top level, non-repeating data items, this section illustrates the NAS markup that you must use to display the field (without edit) and return the field value. In this section you would substitute field names from the left side of your rules file for `TAName_1_1` and TA`Address_1_1`.

Figure 2.16 Sample HTML page to edit outgoing data (*cinfo_e.html*)—Part 4

```
<input type="hidden" name="TACity_1_1"
     value="%GX type=cell id=DocumentContent.TACity_1_1%%/GX%">
<tr align="right" valign="top">
  <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
  <td bgcolor="#E6E6E6" align="left" valign="middle">
       <font face="Arial, Helvetica, sans-serif">City:</font></td>
  <td align="left" bgcolor="#E6E6E6" colspan="3" valign="middle">
       <font face="Arial, Helvetica, sans-serif">
       %GX type=cell id=DocumentContent.TACity_1_1%%/GX%</font></td>
  <td bgcolor="#E6E6E6">&nbsp;</td>
</tr>
<input type="hidden" name="TAState_1_1"
     value="%GX type=cell id=DocumentContent.TAState_1_1%%/GX%">
<tr align="right" valign="top">
  <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
  <td bgcolor="#E6E6E6" align="left" valign="middle">
       <font face="Arial, Helvetica, sans-serif">State:</font></td>
  <td align="left" bgcolor="#E6E6E6" colspan="3">
       <font face="Arial, Helvetica, sans-serif">
       %GX type=cell id=DocumentContent.TAState_1_1%%/GX%</font></td>
  <td bgcolor="#E6E6E6">&nbsp;</td>
</tr>
<input type="hidden" name="TAPostalCode_1_1"
     value="%GX type=cell id=DocumentContent.TAPostalCode_1_1%%/GX%">
<tr align="right" valign="top">
  <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" bgcolor="#E6E6E6">&nbsp;</td>
  <td bgcolor="#E6E6E6" align="left" valign="middle">
       <font face="Arial, Helvetica, sans-serif">Zip Code:</font></td>
  <td align="left" bgcolor="#E6E6E6" colspan="1" valign="middle">
       <font face="Arial, Helvetica, sans-serif">
       %GX type=cell id=DocumentContent.TAPostalCode_1_1%%/GX%</font></td>
  <td bgcolor="#E6E6E6">&nbsp;</td>
</tr>
```

In the fourth HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

This section contains three table rows, each displaying (without edit) and returning one field: the city, state, and postal code labels taken from the left side of the sample rules file. In this section you would substitute field names from the left side of your rules file for `TACity_1_1` and `TAState_1_1` and `TAPostalCode_1_1`.

Figure 2.17 Sample HTML page to edit outgoing data (*cinfo_e.html*)—Part 5

```
    <tr align="right" valign="bottom" bgcolor="#CCCCCC">
      <td align="left" valign="top"><img src="clear.gif" width = "3" height="19"></td>
      <td align="left" valign="top">&nbsp;</td>
      <td align="left" valign="middle" colspan="4">
           <font face="Arial, Helvetica, sans-serif">
           <b><font color="#666666">Telephone Numbers</font></b></font>
      </td>
      <td>&nbsp;</td>
    </tr>
```

The fifth HTML section, above, contains one table row that displays a label (`Telephone Numbers`) for the phone numbers that will appear below. You may change the HTML in this section any way you like for your customization.

Figure 2.18 Sample HTML page to edit outgoing data (*cinfo_e.html*)—Part 6

```
%GX type=tile id=TEL_loop% <!-- TEL_loop -->
  <tr align="left" valign="top">
    <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
    <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
    <td align="left" valign="top" bgcolor="#E6E6E6">
         <font face="Arial, Helvetica, sans-serif">
         Number-%GX type=cell id=TEL_loop.rowid%%/GX%:</font></td>
    <td align="left" valign="top" bgcolor="#E6E6E6" colspan="3">
          <input TYPE="text" name="TEL_%GX type=cell
               id=TEL_loop.rowid%%/GX%.AreaCode_1_1" SIZE="40" MAXLENGTH="80"
               value="%GX type=cell id=TEL_loop.AreaCode_1_1%%/GX%"></td>
    <td align="left" valign="top" bgcolor="#E6E6E6" colspan="3">
          <input TYPE="text" name="TEL_%GX
               type=cell id=rowid%%/GX%%GX type=cell
               id=TEL_loop.rowid%%/GX%.Number_1_1" SIZE="40" MAXLENGTH="80"
               value="%GX type=cell id=TEL_loop.Number_1_1%%/GX%"></td>
    <td align="right" valign="bottom" bgcolor="#E6E6E6">&nbsp;</td>
  </tr>
    <tr align="right valign="bottom">
      <td align="left valign="top" bgcolor="#E6E6E6">
           <img src="clear.gif" width="3" height="25"></td>
      <td align="left valign="top" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left valign="middle" bgcolor="#E6E6E6">&nbsp;</td>
      <td align="left valign="top" bgcolor="#E6E6E6" colspan="3">&nbsp;</td>
      <td bgcolor="#E6E6E6">&nbsp;</td>
    </tr>
%/GX% <!-- TEL_loop -->
```

In the sixth HTML section, above:

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

This section contains the loop that processes the multiple phone numbers. It contains two table rows, the second of which is purely cosmetic. The first table row displays the two phone number fields. In this section you would substitute corresponding portions of field names from the left side of your rules file for `TEL`, `AreaCode_1_1`, and `Number_1_1`. You would also want to change the "`TEL`" elsewhere in the NAS markup as well as in the comments.

If you had an additional level of nesting in your repeating items, such as a regional grouping of area codes, the looping portion of this section would look something like this:

```
<!-- TELROWID replace -->
%GX type=replace id=TEL_loop.rowid value=TELROWID%
%GX type=tile id=TEL_$ROWID_loop%
%gx type=tile id=EXT_loop%
<!-- EXTROWID replace -->
%GX type=replace id=EXT_loop.rowid value=EXTROWID%
...
Extension-%GX type=cell id=EXT_loop.rowid%%/GX%:
     <input type=text name="TEL_TELROWID.EXT_EXTROWID.EXT_1_1"
     value=%GX type = cell id=EXT_loop.EXT_1_1%%/GX%>
...
%/GX% <!-- end EXTROWID replace -->
%/gx% <!-- EXT_loop -->
%/GX% <!-- TEL_$ROWID_loop -->
%/GX% <!-- end TELROWID replace -->
```

Figure 2.19 Sample HTML page to edit outgoing data (*cinfo_e.html*)—Part 7

```
    <tr align="left" valign="top">
      <td align="left" valign="top" bgcolor="E6E6E6">
           <img src="clear.gif" width="3" height="45"></td>
      <td align="left" valign="top" bgcolor="E6E6E6">&nbsp;</td>
      <td align="right" valign="bottom" bgcolor="E6E6E6">
           <input type="SUBMIT" name="submit" value="Submit Document">
           <img src="clear.gif" width="10" height="10"></td>
      <td align="left" valign="bottom" bgcolor="E6E6E6" colspan="3">
           <input type="SUBMIT" name="submit" value="Add Line Items">
           <img src="clear.gif" width="10" height="10"></td>
      <td align="right" valign="bottom" bgcolor="E6E6E6">&nbsp;</td>
    </tr>
  <tr align="left" valign="top">
    <td align="left" valign="top" bgcolor="#E6E6E6">
         <img src="clear.gif" width="3" height="25"></td>
    <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
    <td align="center" valign="bottom" bgcolor="#E6E6E6" colspan="4">
         <img src="dots.gif" width="498" height="3"></td>
    <td align="right" valign="bottom" bgcolor="#E6E6E6">&nbsp;</td>
  </tr>
  </table>
</form>
%/GX% <!-- DocumentContent -->
%/GX% <!-- defaultValues -->
</body>
</html>
```

In the seventh HTML section, above:

- `Bolded` text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

This section contains a button that allows the user to add a phone number, using the hidden `looptoaddto` and `numbertoadd` fields from Figure 2.13, and another button that submits the form. For the button allowing the user to add a phone number (in italics in Figure 2.19), specifying any text value other than “Submit Document” causes a phone number to be added. This section contains two table rows, plus the HTML markup to end the table and the form, plus the NAS markup to end the Document-Content and DefaultValues tiles.

Creating an Order File (Optional)

An order file orders the outgoing (relative to the web user client) data; it must reside in the *$NASDIR/APPS/FX/forms/<locale>/default* directory. An order file is not required, but it simplifies the development of the associated *Mercator* map.

In the TradingXpert sample customization directory, the *cinfo.order* file, shown in Figure 2.20, serves this purpose. The fields on the left correspond to the fields in *cinfo_e.html*.

Figure 2.20 Sample order file, *cinfo.order*

```
TAName_1_1=
TAAddress_1_1=
TACity_1_1=
TAState_1_1=
TAPostalCode_1_1=
BEGINLOOP
TEL_*.AreaCode_1_1=
TEL_*.Number_1_1=
ENDLOOP
```

The order file simply lists the outgoing data labels, one per line, followed by an equal sign, in the order in which the data is to go out. Repeating items are listed between a `BEGINLOOP` and an `ENDLOOP` statement.

Note that repeating items are looped in row-major order in the order file. In the sample files, this means that the area code and number from each row are processed together in each iteration through a single loop.

If your data structure requires it, you can nest one loop within another. For example, if the example in the sample customization files allowed multiple extensions per phone number, the nested loops to order the data would look something like this:

```
...
BEGINLOOP
TEL_*.AreaCode_1_1=
TEL_*.Number_1_1=
    BEGINLOOP
    TEL_*.EXT_*.Extension_1_1=
    ENDLOOP
ENDLOOP
```

Setting Up ECXpert Partnerships and Service Lists

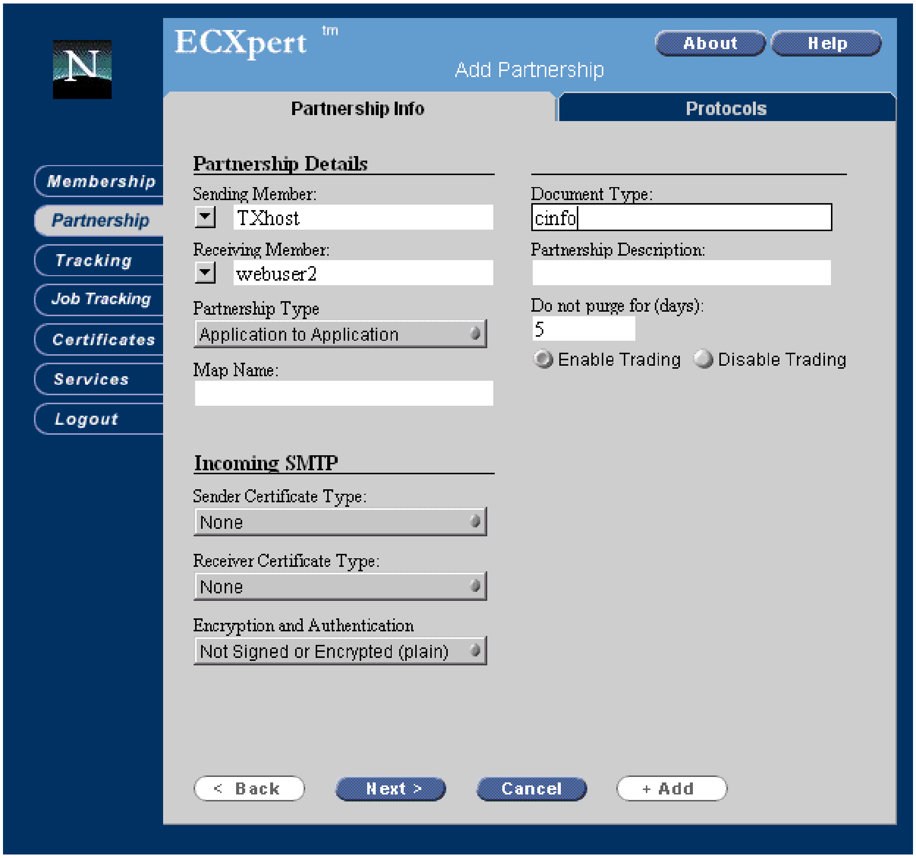
The last task to enable your TradingXpert customization is to set up a partnership and a service list in ECXpert for each document type to be exchanged in each direction. You must create a partnership and a service list to support the sending of `cinfo` documents from TradingXpert to the web user, and you must create a second partnership and service list to support the sending of `cinfo` documents from the web user to TradingXpert.

This section highlights the important values to fill in on the ECXpert tabs to create partnerships and service lists that support the sample customization files for the `cinfo` example. Refer to the *iPlanet ECXpert Administrator's Guide* for more detailed instructions on filling in these tabs.

1. **Create the TXhost-to-webuser (inbound) partnership.**

   The important values to supply on the Partnership Info tab are shown in Figure 2.21.

Figure 2.21 Details for inbound partnership (TXhost to webuser)



Partnership Details

Sending Member: TXhost

Receiving Member: webuser2

Partnership Type: Application to Application

Map Name:

Document Type: cinfo

Partnership Description:

Do not purge for (days): 5

○ Enable Trading ○ Disable Trading

Incoming SMTP

Sender Certificate Type: None

Receiver Certificate Type: None

Encryption and Authentication: Not Signed or Encrypted (plain)

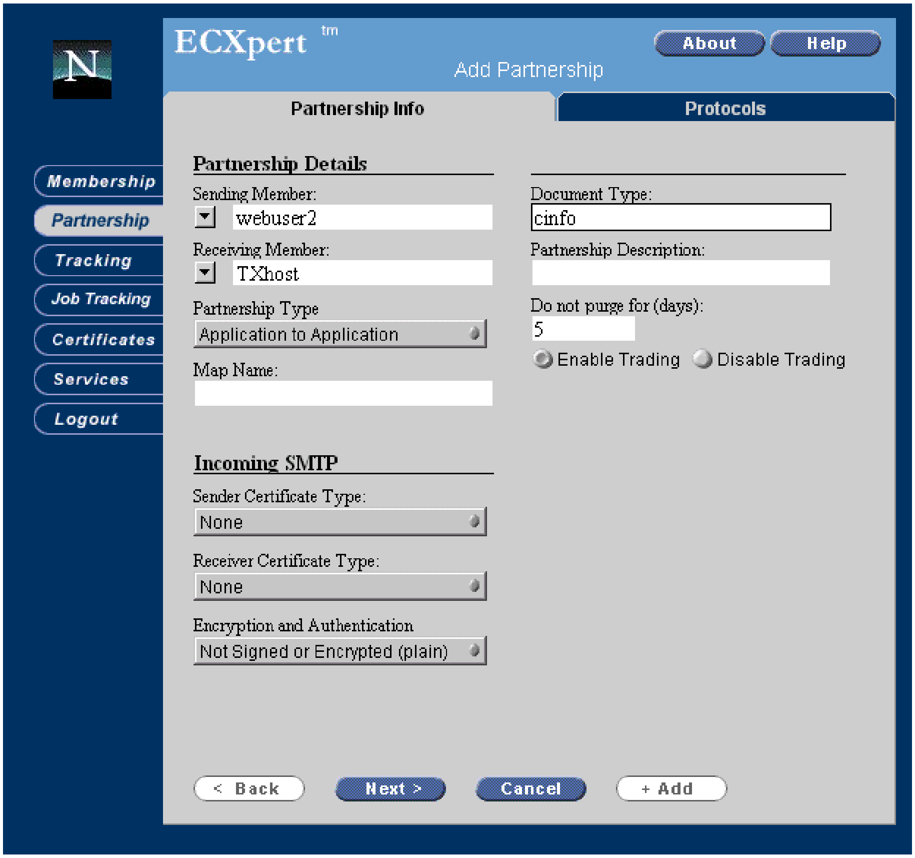
- **Sender:** TXhost—ECXpert member set up to communicate with TradingXpert
- **Receiver:** webuser2—trading partner who will be viewing your purchase orders to the associated company through TradingXpert via a browser
- **Document Type:** cinfo
- **Partnership Type:** Application to Application

On the Protocols tab, select **Poll**.

2. **Create the webuser-to-TXhost (outbound) partnership.**

   The important values to supply on the Partnership Info tab are shown in Figure 2.22.

Figure 2.22 Details for outbound partnership (webuser to TXhost)



Partnership Details

Sending Member: webuser2

Receiving Member: TXhost

Partnership Type: Application to Application

Map Name:

Document Type: cinfo

Partnership Description:

Do not purge for (days): 5

○ Enable Trading ○ Disable Trading

Incoming SMTP

Sender Certificate Type: None

Receiver Certificate Type: None

Encryption and Authentication: Not Signed or Encrypted (plain)

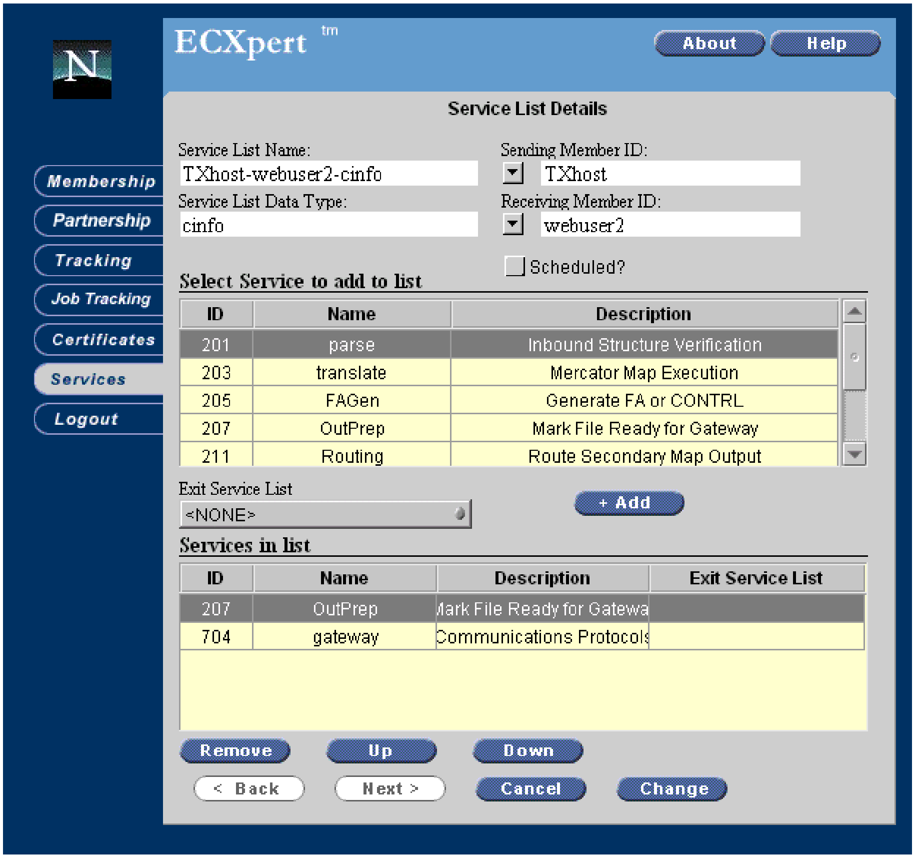
- **Sender:** webuser2—trading partner who will be viewing your purchase orders to the associated company through TradingXpert via a browser
- **Receiver:** TXhost—ECXpert member set up to communicate with TradingXpert
- **Document Type:** cinfo
- **Partnership Type:** Application to Application

On the Protocols tab, select **FTP** and set parameters as necessary.

3. **Create the TXhost-to-webuser (inbound) service list.**

   The important values to supply on the Service List Details tab are shown in Figure 2.23.

Figure 2.23 Details for inbound service list (TXhost to webuser)



Service List Details

Service List Name: TXhost-webuser2-cinfo

Sending Member ID: TXhost

Service List Data Type: cinfo

Receiving Member ID: webuser2

□ Scheduled?

Select Service to add to list

ID	Name	Description
201	parse	Inbound Structure Verification
203	translate	Mercator Map Execution
205	FAGen	Generate FA or CONTRL
207	OutPrep	Mark File Ready for Gateway
211	Routing	Route Secondary Map Output

Exit Service List: <NONE>

Services in list

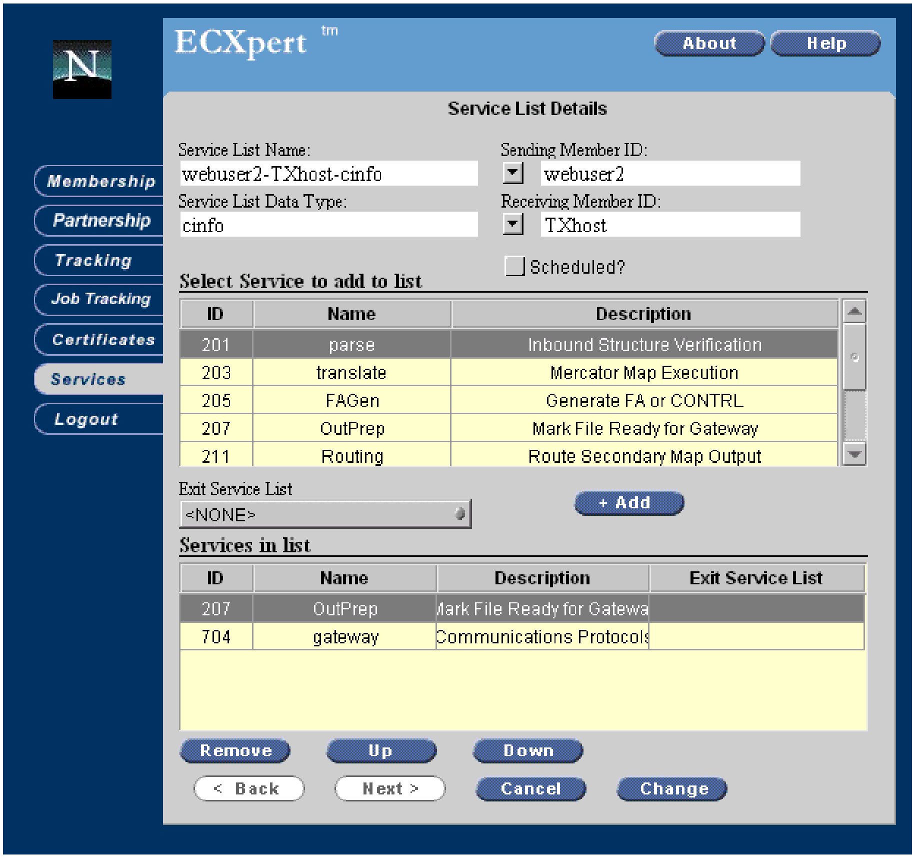
ID	Name	Description	Exit Service List
207	OutPrep	Mark File Ready for Gatewa	
704	gateway	Communications Protocols	

- **Sender:** TXhost—ECXpert member set up to communicate with TradingXpert
- **Receiver:** webuser2—trading partner who will be viewing your purchase orders to the associated company through TradingXpert via a browser
- **Data Type:** cinfo
- **Services in List:** OutPrep and Gateway

4. **Create the webuser-to-TXhost (outbound) service list.**

The important values to supply on the Service List Details tab are shown in Figure 2.24.

Figure 2.24 Details for outbound service list (webuser to TXhost)



Service List Details

Service List Name: webuser2-TXhost-cinfo

Sending Member ID: webuser2

Service List Data Type: cinfo

Receiving Member ID: TXhost

☐ Scheduled?

Select Service to add to list

ID	Name	Description
201	parse	Inbound Structure Verification
203	translate	Mercator Map Execution
205	FAGen	Generate FA or CONTRL
207	OutPrep	Mark File Ready for Gateway
211	Routing	Route Secondary Map Output

Exit Service List

<NONE>

Services in list

ID	Name	Description	Exit Service List
207	OutPrep	/ark File Ready for Gatewa	
704	gateway	Communications Protocols	

- **Sender:** webuser2—trading partner who will be viewing your purchase orders to the associated company through TradingXpert via a browser
- **Receiver:** TXhost—ECXpert member set up to communicate with TradingXpert
- **Data Type:** cinfo
- **Services in List:** OutPrep and Gateway

Customization Summary

To accept an incoming document type of `cinfo` and turn it around as another document of the same type:

1. Copy *cinfo.html*, *cinfo_e.html*, *cinfo.order*, and *cinfo-cinfo.rules* to the *$NASDIR/APPS/FX/forms/<locale>/default* directory. If you want to create the original `cinfo` document from scratch, you must also copy *0-cinfo.rules* to the */default* directory and *cinfo.dat* to the *$NASDIR/APPS/FX/forms/<locale>/from_scratch_data* directory.

2. Add `cinfo` to the following entries in the *FXproperties* file, which is located in the *$NASDIR/APPS/FX/common* directory:

   - `FX.viewableDocTypes`
   - `FX.outboundDocTypes`
   - `FX.fromScratchDocTypes (`if `cinfo` documents must be created from scratch)

   Optionally, create another property for `FX.viewableDocTypes.`*`cinfo`* with a descriptive label.

3. Add partnerships in ECXpert:

   - TXhost to Webuser (`cinfo` inbound)
   - Webuser to TXhost (`cinfo` outbound)

4. Restart NAS.

Note If you want to add the ability to create templates from your customized document, you must also complete Steps 5-7 before you restart NAS.

5. Add the Template Button to the *cinfo_e.html* file.

   — Remove `colspan=3` from the Add Line Items column (Line 8) in Figure 2.19.

   — Add the following code after the Add Line Items column (Line 9) in Figure 2.19:

   ```
   <td align="left" valign="bottom" bgcolor="E6E6E6"
   colspan="2"><input type="SUBMIT" name="submit"
   value="Save As Template"><img src="clear.gif"
   width="10" height="10"></td>
   ```

6. Add the Edit hidden field to the *cinfo_e.html* file.

Add the following code after the SENDER_ADDR hidden field code (Line 17) in Figure 2.13:

```
<input type="hidden" name="EDIT" value="%gx type=cell
id=EDIT%%/gx%">
```

7. Edit the *0-cinfo.rules* file.

   You must add all fields (editable and noneditable) that you add to the *cinfo_e.html* file to the *0-cinfo.rules* file.

Using Custom Forms to View Documents from Different Senders

It is possible to display an incoming document type using different HTML templates, depending on who sent the document to the trading partner.

When TradingXpert looks for a form to view a document, it first searches for a directory under the *forms* directory with the same name as the trading partner who sent the document, i.e., the *$NASDIR/APPS/FX/forms/<locale>/<partner>* directory. If this directory exists, TradingXpert then searches for the form in that directory. If the form is not there, or if the directory doesn't exist, TradingXpert uses the form in the *$NASDIR/APPS/FX/forms/<locale>/default* directory.

When TradingXpert is performing document turnaround, it performs the same search except that it looks for the directory with the same name as the trading partner who sent the original document, who is also the trading partner who is receiving the turned-around document.

You may use custom forms to make the form look different or to have different fields editable when you are performing document turnaround.

Whenever you add a new form for TradingXpert, make sure there is a generic version of that form in the *$NASDIR/APPS/FX/forms/<locale>/default* directory. This ensures that the document may be viewed even if there is no custom form for a particular trading partner.

Enabling Certificate-Based Logins

TradingXpert supports all of Netscape Enterprise Server's (NES) authentication capabilities to provide as much security as you want for your site. This includes the option to require each web user to have a certificate, in addition to a regular user ID and password, to log in to TradingXpert.

To integrate certificate-based security into TradingXpert,

1. **Set up the desired authentication in NES for the test user *webuser1*.**

   Refer to your NES documentation to set up authentication mechanisms.

2. **Open the TradingXpert login screen.**

   Using your browser, display the URL http://<*BASEHREF*>/TX/.

3. **Check the TradingXpert HTTP access log to verify that the web user's certificate is being authenticated.**

   When web server-based authentication is used, the HTTP access log, located at *$HTTPS-DIR/logs/access*, will indicate the web user's certificate.

Figure 2.25 Example of access log file with authentication enabled

```
[format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
"%Req->reqpb.clf-request%" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%]


222.222.222.143 - webuser1 [14/Oct/1999:17:39:42 -0700]
"GET /TX/txt_login_02.gif HTTP/1.0" 200 1064


222.222.222.143 - webuser1 [14/Oct/1999:17:39:42 -0700]
"GET /TX/icon_netscape_01.gif HTTP/1.0" 200 1542
```

When web server-based authentication is not used, the *Req->vars.auth-user* variable in the access log will have no value (see Figure 2.26).

Figure 2.26 Example of access log file without authentication enabled

```
[format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
"%Req->reqpb.clf-request%" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%]


222.222.222.143 - - [14/Oct/1999:17:39:42 -0700]
"GET /TX/txt_login_02.gif HTTP/1.0" 200 1064



222.222.222.143 - - [14/Oct/1999:17:39:42 -0700]
"GET /TX/icon_netscape_01.gif HTTP/1.0" 200 1542
```

4. **Log in as webuser1.**

   If either NES Basic Authentication or LDAP/Certificate-based Authentication is used, the User Id field on the TradingXpert login screen will be pre-filled—and non-modifiable—according to data in the user's certificate; the user must enter the Password to log in. To change the user ID, the user must exit the browser and start over again, using a different certificate.

   If NES authentication is not used, the TradingXpert login screen will require a User ID and Password.

5. **Set up the desired authentication for each web user.**

   Again, refer to your NES documentation to set up authentication mechanisms.

Enabling Events

TradingXpert has two optional event daemons that can generate and send e-mail messages to web users regarding their unread, inbound messages:

- **Email Notification**

  Email Notification gives web users the option to be notified daily, or by some other configurable interval, by e-mail when they have unread documents in their inbound mailbox.

  If the Email Notification daemon is enabled, it will check once a day for unread documents that have arrived for each user in the last 24 hours or some other configurable interval. It will then send an e-mail message, listing all documents that are unread, to each user who has provided an e-mail address and selected the **Enable Email Notification** parameter on the user profile screen.

Note The **Enable Email Notification** parameter will appear on user profile screens only if the Email Notification daemon is enabled.

- **Email Alert**

  Email Alert informs web users when they have not read incoming documents that are older than a specified period of time.

  If the Email Alert daemon is enabled, it will check for each user's unread documents according to specific time periods for each document type. It will then send an e-mail message, listing all documents of the specified type that remain unread, to each applicable user whose profile includes an e-mail address. For instance, if the administrator wants web users to read all 843s within six hours of their receipt, she can set the alert time to six hours; the web user will then receive an e-mail message if any 843s remain unread after the six-hour period.

  The Email Alert daemon runs as often as the smallest document alert time; if an 850 is set to one hour, then the process will run every hour, even if all of the other documents are set to 24 hours. Be aware that Email Alert can affect the performance of a system with many members and many document exchanges if it is set to run in small increments.

Registering Events

To enable event messages for your web users, perform the following steps:

1. **Set variables for each type of event to be enabled.**

   Edit the bolded entries in the *FXproperties* file, located in your *$NASDIR/APPS/FX/common/* directory.

Figure 2.27 FXproperties file entries to enable event notification

```
...
#Admin user for TradingXpert
FX.AdminUser:ECX_administrator_userID

#Admin password (encrypted) for TradingXpert
FX.AdminPassword:ECX_administrator_password

...

# Mail Server used for sending e-mail
FX.MailServer:<name.domain.com>

# Sending Email Address for Alert and Notification events
FX.EmailEventSenderAddress:fromuser@company.com

# Hour of day to have the Email notification event run
FX.EmailNotification:<hh:mm:ss>

# List of hours to wait before alerting users
FX.hours.214:<hh>

FX.hours.840:<hh>

FX.hours.843:<hh>

FX.hours.850:<hh>

FX.hours.855:<hh>

FX.hours.860:<hh>

FX.hours.INVOIC:<hh>

FX.hours.ORDERS:<hh>

FX.hours.856:<hh>

FX.hours.862:<hh>

FX.hours.810:<hh>

FX.hours.830:<hh>

...
```

where **FX.AdminUser** and **FX.AdminPassword** are the user ID and encrypted password for the administrator user, **FX.MailServer** is the mail server that will be sending email, **FX.EmailEventSenderAddress** is the e-mail address from which all event messages will be sent, **FX.EmailNotification** is the time of day the notification event should run (e.g., FX.EmailNotification:2:0:0 to run at 2:00 a.m.), and **FX.hours.N:<hh>** is the alert event time—in hours—for a specific document type (e.g., FX.hours.214:24 to send alerts about unread 214s 24 hours after their receipt, or FX.hours.214:0 to not send alerts about 214s). One **FX.hours** entry is included for each default document type; the default value is 24.

Note Encrypt your password for the **Fx.AdminPassword** entry using TradingXpert's `txsetpasswd` utility (see "Encrypting the TradingXpert Administrator Password (Solaris)" on page 100).

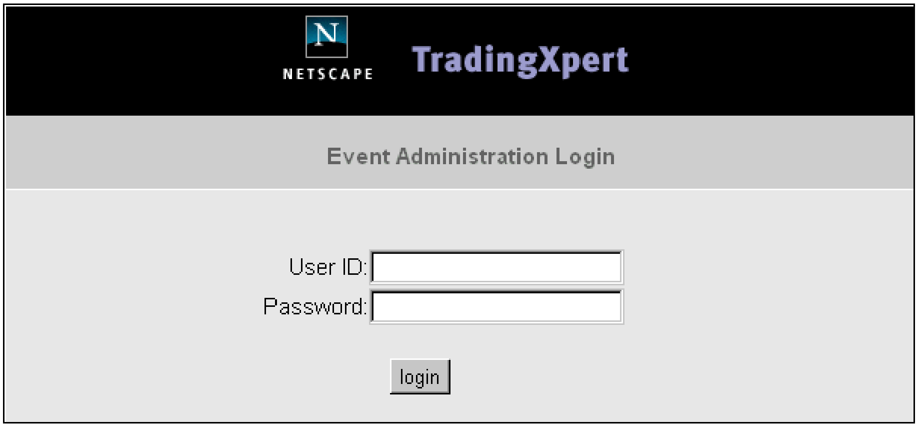
2. **Restart NAS.**

3. **Open the event registration login screen.**

   Using your browser, display the URL

   http://<*BASEHREF*>/TX/<*locale*>/events.html

Figure 2.28 Event Login screen



4. **Log in as administrator to the event registration interface.**

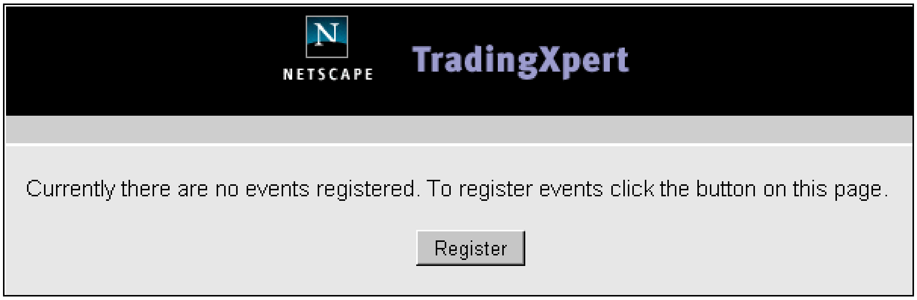
   Enter your administrator user ID and password on the login screen that appears, and click the **Login** button to log in.

   The Event Registration screen will appear, indicating that no events are registered (Figure 2.29).

Note You must be an ECXpert administrator user to log in to the Event Registration page.

5. **Click Register to register the events.**

Figure 2.29 Event Registration screen with no events registered

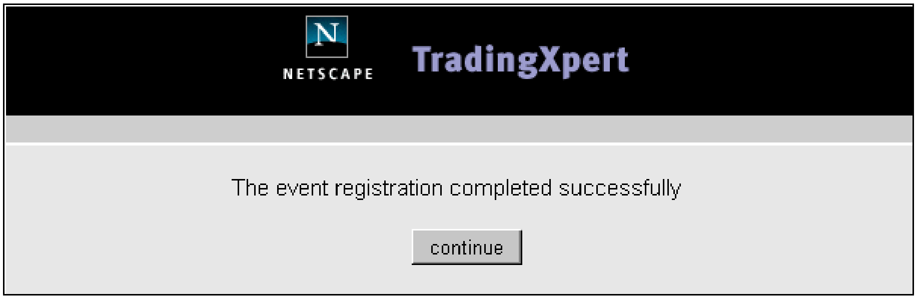


Currently there are no events registered. To register events click the button on this page.

A Confirmation screen will appear (Figure 2.30).

6. **Click Continue.**

Figure 2.30 Confirmation screen for event registration



The event registration completed successfully

The Event Information screen (Figure 2.31) will open, where you can view —and optionally modify—the events that are registered.

Modifying Event Parameters

To change the parameters for Email Notification or Email Alert:

1. **Edit variables for each type of event to be modified.**

   Edit the applicable parameters in the *FXproperties* file, located in your *$NASDIR/APPS/FX/common/* directory (see Figure 2.27).

2. **Restart NAS.**

3. **If you modify the value of the FX.EmailNotification parameter, delete the events and re-register them.**

   See "Modifying or Deleting Events" on page 84 and "Registering Events" on page 80.

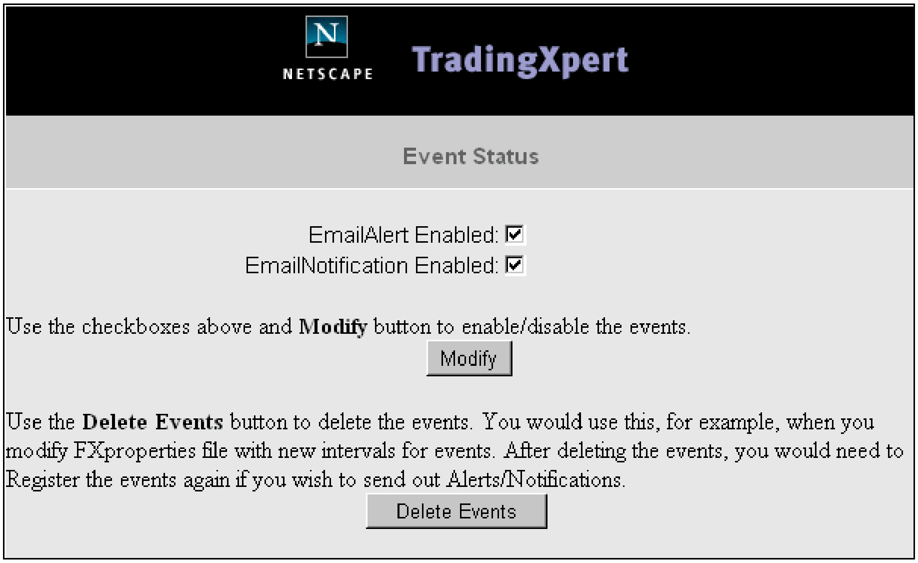
Note If you modify the value of any of the **FX.hours.N:<hh>** parameters, skip this step.

Modifying or Deleting Events

After you register events and press the **Continue** button, and each subsequent time you log in to the event registration interface, the Event Information screen (Figure 2.31) will appear.

The Event Information screen lists each event option—**EmailAlert Enabled** and **EmailNotification Enabled**— with a checkbox indicating whether or not the event is enabled. You can disable one or both of these events at any time. To disable an event, uncheck the checkbox and click the **Modify** button; a message on the screen will confirm that your modification was completed successfully. To completely delete the event daemon processes—regardless of whether they are currently enabled or not—click the **Delete Events** button; a message on the screen will confirm that your deletion was completed successfully.

Figure 2.31 Event Information screen



Event Status

EmailAlert Enabled: ☑
EmailNotification Enabled: ☑

Use the checkboxes above and **Modify** button to enable/disable the events.

Use the **Delete Events** button to delete the events. You would use this, for example, when you modify FXproperties file with new intervals for events. After deleting the events, you would need to Register the events again if you wish to send out Alerts/Notifications.

Customizing E-mail Templates for Email Notifications

To modify the wording and content of e-mail notifications,

1. **Edit variables in *$NASDIR/APPS/FX/templates/<locale>/message.txt.***

Figure 2.32 The default Email Notification variables in the message.txt file

```
...
EMAIL_NOTIFY_SUBJECT: TradingXpert Notification
...
EMAIL_NOTIFY: The following documents have arrived in the
last 24 hours:<br><br>To Access TradingXpert:
EMAIL_HEADER: <pre><b>Sender\t\tDocType\t\tTime</b></pre><br>
...
```

EMAIL_NOTIFY_SUBJECT is the subject header of the e-mail message, **EMAIL_NOTIFY** is the body of the e-mail message, and **EMAIL_HEADER** identifies the column headers for the list of documents included in the e-mail message.

Note The same EMAIL_HEADER parameter is used for both Email Notification and Email Alert messages.

2. **Edit the message layout at *$NASDIR/APPS/FX/forms/<locale>/default/emailhead.html* and/or */emailtail.html.***

Note The same *emailhead.html* and *emailtail.html* files are used for both Email Notification and Email Alert messages.

Figure 2.33 The default emailhead.html file for the Email Notification and Email Alert daemons

```
<html>

<head>

</head>

<BASE HREF="%gx type=cell id=BASEHREF%%/gx%">

<body bgcolor="#E6E6E6" width="100%">

<table width="100%"><tr><td></td><br></tr></table>

<br><br><br>
```

Figure 2.34 The default emailtail.html file for the Email Notification and Email Alert daemons

```
</body></html>
```

Customizing E-mail Templates for Email Alerts

To modify the wording and content of e-mail alerts,

1. **Edit variables in *$NASDIR/APPS/FX/templates/<locale>/message.txt*.**

Figure 2.35 The default Email Alert variables in the message.txt file

```
...
EMAIL_ALERT_SUBJECT: TradingXpert Alert
...
EMAIL_ALERT: The alert time has passed for the following doc-
uments<br><br>To Access TradingXpert:
...
EMAIL_HEADER: <pre><b>Sender\t\tDocType\t\tTime</b></pre><br>
...
```

`EMAIL_ALERT_SUBJECT` is the subject header of the e-mail message, **`EMAIL_ALERT`** is the body of the e-mail message, and **`EMAIL_HEADER`** identifies the column headers for the list of documents included in the e-mail message.

Note The same `EMAIL_HEADER` parameter is used for both Email Notification and Email Alert messages.

2. **Edit the message layout at *$NASDIR/APPS/FX/forms/<locale>/default/emailhead.html* and/or */emailtail.html*.**

Note The same *emailhead.html* and *emailtail.html* files are used for both Email Notification and Email Alert messages.

Figure 2.36 The default emailhead.html file for the Email Notification and Email Alert daemons

```
<html>

<head>

</head>

<BASE HREF="%gx type=cell id=BASEHREF%%/gx%">

<body bgcolor="#E6E6E6" width="100%">

<table width="100%"><tr><td></td><br></tr></table>

<br><br><br>
```

Figure 2.37 The default emailtail.html file for the Email Notification and Email Alert daemons

```
</body></html>
```

Enabling New Member Self-Registration

TradingXpert provides an optional feature to allow users to register a user ID and enter personal contact information themselves. TradingXpert will then automatically create an ECXpert membership for the new user.

If this feature is enabled, self-registered users can be added as one of the following types of users:

- **Inactive user**

  The user will not be able to access the system until the TradingXpert administrator manually activates the ECXpert membership and creates trading partnerships with the new user, as well as service lists as necessary.

- **Active user**

TradingXpert automatically activates the ECXpert membership for the new user; the user can then access remaining TradingXpert screens. The user cannot send or receive files, however, until the TradingXpert administrator manually creates trading partnerships with the user, as well as service lists as necessary.

To enable new member self-registration in TradingXpert,

1. **Replace the TradingXpert login template with one that allows New Member Registration.**

   Copy *$NASDIR/APPS/FX/maps_etc/newmember/index.html* to the *$NASDIR/APPS/FX/templates/<locale>/login/* directory.

   This file will replace the current *index.html* file in the */login* directory. The new login template will include a **Register** button (see Figure 2.38), which links the user to a New Member Registration screen (see Figure 2.39).

Figure 2.38 Login screen with a Register button

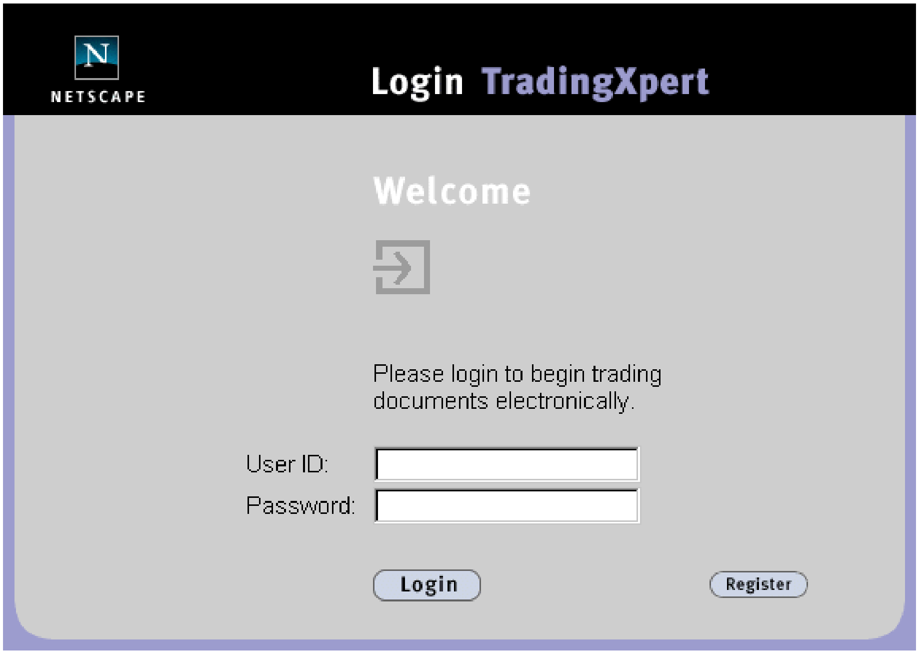
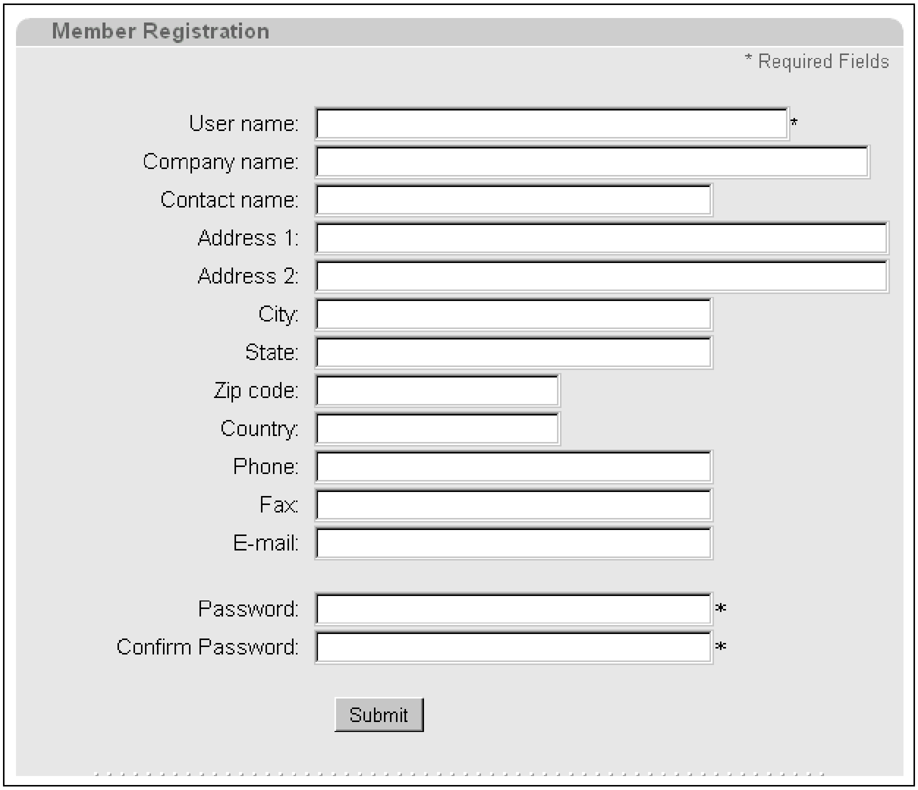


Figure 2.39 New Member Registration screen



Member Registration

* Required Fields

User name: *

Company name:

Contact name:

Address 1:

Address 2:

City:

State:

Zip code:

Country:

Phone:

Fax:

E-mail:

Password: *

Confirm Password: *

2. Set variables in the *FXproperties* file.

Edit the bolded entries to the *FXproperties* file, located in your *$NASDIR/APPS/FX/common/* directory.

Figure 2.40 FXproperties file entries to enable new member creation

```
...
# New members added are Active
FX.activeNewMember:True|False

# Send e-mail to administrator when new users add themselves
FX.sendEmailForNewMember: <True|False>

# Email Address from which to send email when new members add
themselves
FX.EmailNewMemberSenderAddress:<fromuser@company.com>

# Email Address to which to send email when new members add
themselves
FX.EmailAddress: <touser@company.com>

# Mail Server used for sending e-mail
FX.MailServer: <name.domain.com>

...
```

FX.activeNewMember indicates whether or not the new user's ECXpert membership will be automatically activated; **FX.sendEmailForNewMember**, which by default is set to "True," indicates whether an e-mail message will be sent to the administrator each time a new member is registered; **FX.EmailNewMember-SenderAddress** is the e-mail address from which this e-mail will be sent; **FX.Email-Address** is the address to which this e-mail message is directed; and **FX.MailServer** is the full domain name of the SMTP mail server used for the machine on which TradingXpert is running.

3. **If FX.activeNewMember is set to `False`, activate a new user's membership in ECXpert, and create partnerships and service lists.**

Note Skip Step 3 if **FX.activeNewMember** is set to `True`, and proceed to Step 4.

After each new web user registers as an *inactive* user, make that membership active in ECXpert; then create a partnership with the new member and a service list for each document type to be exchanged, in each direction, with the new member. Refer to the *iPlanet ECXpert Administrator's Guide* for more detailed instructions on activating memberships and on creating partnerships and service lists.

4. **If FX.activeNewMember is set to `True`, create a partnership and service lists in ECXpert.**

Note Skip Step 4 if **FX.activeNewMember** is set to `False`.

After each new web user registers as an *active* user, set up a partnership with the new member, as well as a service list for each document type to be exchanged with the new member, in each direction. Refer to the *iPlanet ECXpert Administrator's Guide* for more detailed instructions on setting up partnerships and service lists.

5. **If `FX.sendEmailForNewMember` is set to `True`, edit variables in *$NASDIR/APPS/FX/templates/<locale>/message.txt.***

   Modify the following variables to configure the e-mail messages that are sent to the administrator when a new member self-registers:

Figure 2.41 The default New Member variables in the message.txt file

```
...
NEW_MEMBER_SUBJECT: New TradingXpert Member
NEW_MEMBER_TEXT: The following username has been added to
TradingXpert:
```

`NEW_MEMBER_SUBJECT` is the subject header of the e-mail message, and **`NEW_MEMBER_TEXT`** is the body of the e-mail message.

Modifying File Attachment Options for Document Forms

TradingXpert forms for editable documents (e.g., *810_e.html*) allow web users to attach an application file to each outgoing document. These forms can be customized to specify allowable attachment types and to allow more than one attachment per document.

In order for ECXpert to be able to receive the file and forward it to the user's file system,

- The file type specified for the attachment must be the same file type allowed in the ECXpert partnership.
- The ECXpert partnership must be Application to Application.
- The ECXpert service list must include Outprep and Gateway to allow the file to pass through ECXpert without processing.

New service lists, and consequently new partnerships, may be required to accommodate the file types allowed as attachments. See the *iPlanet ECXpert Administrator's Guide* for more information about setting up service lists and partnerships.

To specify the types of attachments that can be sent with a specific type of document, or to change the maximum number of attachments that can be attached to the document, you must edit the editable version of that document.

Specifying the File Type of an Attachment Option

By default, all editable document forms allow users to attach one file in any format (**Any Type** in the pull-down menu). This file will be treated as the document type *ATTACH,* which is a generic file type for any attachment that is passed to ECXpert without processing.

Note An ECXpert partnership in which an ATTACH document type is exchanged must specify ATTACH as the document type.

You can modify the file types that are can be attached to an EDI document, making them as generic or as specific as you want. To specify a type of file that can be attached to a specific type of EDI document, edit the form for that editable EDI document type. Forms for editable EDI document types—e.g., *810_e.html* for invoices—are located in the *$NASDIR/APPS/FX/forms/<locale>/default* directory.

1. **Add HTML to create an attachment option.**

Figure 2.42 Sample HTML indicating an attachment option for a document form

```
<!-- Attachments -->
...

<tr align="left" valign="top">
  <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" valign="middle" bgcolor="#E6E6E6"><font face="Arial, Helvetica,
      sans-serif">
    <INPUT TYPE="file" NAME="ATTACH_N" SIZE=15 MAXLENGTH="80">
    </font></td>
  <td align="left" bgcolor="#E6E6E6" valign="middle"><font face="Arial, Helvetica,
      sans-serif">
    <select NAME="ATTFILETYPE_N">
    <option value="ATTACH">File Type</option>
        </font></td>
    </select>
  <td bgcolor="#E6E6E6">&nbsp;</td>
</tr>

...

<!-- End Attachments -->
```

In Figure 2.42,

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

2. **Increment the *`ATTACH_N`* and *`ATTFILETYPE_N`* values for the row of options, beginning with one (1).**

   `ATTACH_`*`N`* indicates a set of attachment options, and **`ATTFILETYPE_`***`N`* indicates a file type option for that set of attachment options. Both use the same value of *N*.

3. **Edit the value of the document type.**

   Replace *`ATTACH`* with the one-word name of the document type allowed in the ECXpert partnership.

4. **Modify the filetype name that should appear to web users.**

   Replace *`File Type`* with the wording that should appear in the pull-down menu of file types.

Note If either partner is using a browser's e-mail program (e.g., Netscape Messenger) to submit EDI documents and attachments to ECXpert, the EDI documents must be attached before non-EDI files in order for the documents to be displayed and be viewable in the TradingXpert user interface.

Adding Additional File Types

To allow an additional file type to be attached to the document,

1. **Add an additional `<option></option>` command in the HTML.**

   Figure 2.43 indicates two file type options.

2. **Modify *`ATTFILETYPE_N`* in the new option to reflect the value for the set of options.**

   Each instance of **`ATTFILETYPE_`***`N`* in a set of attachment options should have the same value of *N*. This value should reflect the value of **`ATTACH_`***`N`*.

3. **Edit the value of the document type in the new option.**

   Replace *`ATTACH`* with the one-word name of the document type allowed in the ECXpert partnership.

4. **Modify the file type name for the new option that should appear to web users.**

   Replace *`File Type`* with the wording that should appear in the pull-down menu of file types.

Figure 2.43 Sample HTML to add a second attachment type for a document form

```
...
    <select NAME="ATTFILETYPE_N">
      <option value="ATTACH">File Type</option>
      <option value="ANOTHER_DOCUMENT_TYPE">Another File Type</option>
    </select>
...
```

In Figure 2.43,

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

Allowing Additional Attachments

To allow more than one attachment to a specific type of document, edit the HTML for that document form:

1. **Add a new row for each additional attachment option.**

   Copy the default HTML in Figure 2.42—beginning with the <tr...> tag and ending with </tr> tag—and paste it immediately below to add the necessary additional input field, button, and pull-down menu to allow users to attach an additional file.

2. **Increment the *`ATTACH_N`* and *`ATTFILETYPE_N`* values in each additional row by one (1).**

In Figure 2.44, the comments around the original HTML have been changed from `<!-- Attachments -->` and `<!-- End Attachments -->` to `<!-- Attachment 1 -->` and `<!-- End Attachment 1 -->`. The section enclosed within the `<!-- Attachment 2 -->` and `<!-- End Attachment 2 -->` comments indicates a second row of e-mail attachment options.

In addition,

- **`Bolded`** text must appear in the file exactly as shown.
- *Italicized* text must be replaced by values suitable for your customization.
- `Plain courier` text may be changed any way you like to control the page display.

Figure 2.44 Sample HTML to modify the attachment options for a document form

```
<!-- Attachment 1 -->
...
<tr align="left" valign="top">
  <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" valign="middle" bgcolor="#E6E6E6"><font face="Arial, Helvetica,
        sans-serif">
<INPUT TYPE="file" NAME="ATTACH_1" SIZE=15 MAXLENGTH="80">
    </font></td>
  <td align="left" bgcolor="#E6E6E6" valign="middle"><font face="Arial, Helvetica,
        sans-serif">
    <select NAME="ATTFILETYPE_1">
    <option value="ATTACH">File Type</option>
    <option value="ANOTHER_DOCUMENT_TYPE">Another File Type</option>
    </select>
    </font></td>
  <td bgcolor="#E6E6E6">&nbsp;</td>
</tr>
...
<!-- End Attachment 1 -->

<!-- Attachment 2 -->
...
<tr align="left" valign="top">
  <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" valign="top" bgcolor="#E6E6E6">&nbsp;</td>
  <td align="left" valign="middle" bgcolor="#E6E6E6"><font face="Arial, Helvetica,
        sans-serif">
    <INPUT TYPE="file" NAME="ATTACH_2" SIZE=15 MAXLENGTH="80">
    </font></td>
  <td align="left" bgcolor="#E6E6E6" valign="middle"><font face="Arial, Helvetica,
        sans-serif">
    <select NAME="ATTFILETYPE_2">
    <option value="ATTACH">File Type</option>
    <option value="ANOTHER_DOCUMENT_TYPE">Another File Type</option>
    </select>
    </font></td>
  <td bgcolor="#E6E6E6">&nbsp;</td>
</tr>
...
<!-- End of Attachment 2 -->
```

Compiling Java Code (Solaris)

See the section on Compiling Java Code in Chapter 3, “Postinstallation Tasks,” of the *iPlanet TradingXpert Installation Guide*.

Chapter

3

System Utilities

This chapter documents the system utilities that are available for use with the TradingXpert System. The following topics are covered:

- Encrypting the TradingXpert Administrator Password (Solaris)
- Encrypting the TradingXpert Administrator Password (Windows NT)
- Generating Document Status Reports using Actuate

Encrypting the TradingXpert Administrator Password (Solaris)

TradingXpert's `txsetpasswd` utility, located in the *$NASDIR/APPS/FX/common/* directory, encrypts the FX.AdminPassword parameter, which is found in the *FXproperties* file located in your *$NASDIR/APPS/FX/common/* directory.

To encrypt the administrator's password, perform the following tasks.

1. **Add *$NASDIR/APPS/bin* to the `LD_LIBRARY_PATH`.**

   Use the following syntax to modify the environment variable in the administrator user's *.profile* or *.login* file:

   For the C shell:

   ```
   setenv LD_LIBRARY_PATH $NASDIR/APPS/bin:$LD_LIBRARY_PATH
   ```

   For the Bourne shell:

   ```
   LD_LIBRARY_PATH=$NASDIR/APPS/bin:$LD_LIBRARY_PATH
   export LD_LIBRARY_PATH
   ```

2. **Change to the *$NASDIR/APPS/FX/common/* directory.**

   ```
   cd $NASDIR/APPS/FX/common/
   ```

3. **Run the `txsetpasswd` utility with the desired password.**

   Type the following command at the command line:

   ```
   ./txsetpasswd <password>
   ```

   The encrypted password will be entered automatically as the **FX.AdminPassword** value in the *FXproperties* file.

Figure 3.1 FXproperties file entry for the administrator password

```
...

#Admin password (encrypted) for TradingXpert
FX.AdminPassword:ECX_administrator_password

...
```

Encrypting the TradingXpert Administrator Password (Windows NT)

TradingXpert's `txsetpasswd` utility, located in the *<NASDIR>\APPS\FX\common\* directory, encrypts the FX.AdminPassword parameter, which is found in the *FXproperties* file located in your *<NASDIR>\APPS\FX\common\* directory.

To encrypt the administrator's password, perform the following tasks.

1. **Add *<NASDIR>\APPS\bin* to the system variable `Path`.**

   From the Start Menu, open Settings, then Control Panel, then System. Select the Environment tab, and append the following syntax to the `Path` variable.

   ```
   :<NASDIR>\APPS\bin
   ```

   Click **Set**, then **Apply**; this will modify the variable in the administrator user's system properties.

2. **From a Command Prompt window, change to the *<NASDIR>\APPS\FX\common\* directory.**

   ```
   cd %NASDIR%\APPS\FX\common\
   ```

   where *%NASDIR%* is the directory in which NAS is installed.

3. **Run the `txsetpasswd` utility with the desired password.**

   Type the following command at the command prompt:

   ```
   txsetpasswd <password>
   ```

   The encrypted password will be entered automatically as the **FX.AdminPassword** value in the *FXproperties* file.

Figure 3.2 FXproperties file entry for the administrator password

```
...

#Admin password (encrypted) for TradingXpert
FX.AdminPassword:ECX_administrator_password

...
```

Generating Document Status Reports using Actuate

If you have the Actuate Reporting System version 3.2 installed on a Windows NT computer, you can generate reports detailing the **read**/**unread** status of documents received by one or more trading partners. TradingXpert's `DocumentStatus.rox` report utility creates TradingXpert-specific reports using your Actuate system.

Note You must have the Actuate Reporting System version 3.2 installed on a Windows NT computer, whether your TradingXpert system is installed on NT or Solaris.

Note For more information about running the standard Actuate Reports included with ECXpert, and about making Actuate reports available in HTML, refer to the *iPlanet ECXpert Administrator's Guide.*

1. **If Actuate is installed on a different machine than TradingXpert, copy the *DocumentStatus.rox* file to the remote system.**

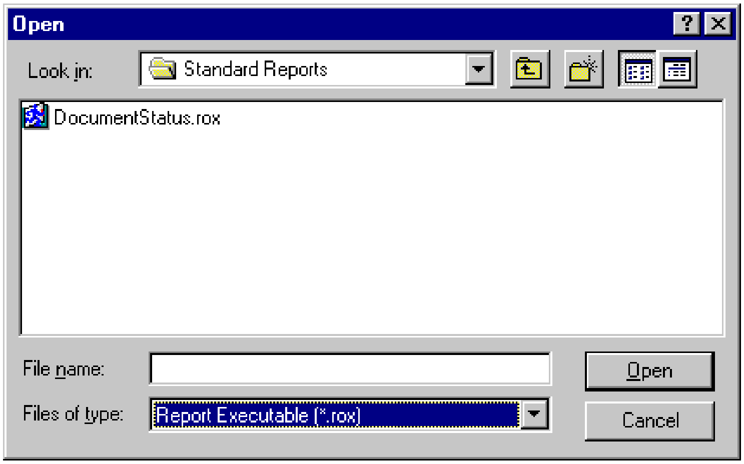
   DocumentStatus.rox is located at either *$NASDIR/APPS/FX/maps_etc/reports/DocumentStatus.rox* (Unix) or *<NASDIR>\APPS\FX\maps_etc\reports\DocumentStatus.rox* (Windows NT).

2. **Launch the Actuate End User Desktop.**

3. **Open the *DocumentStatus.rox* report file.**

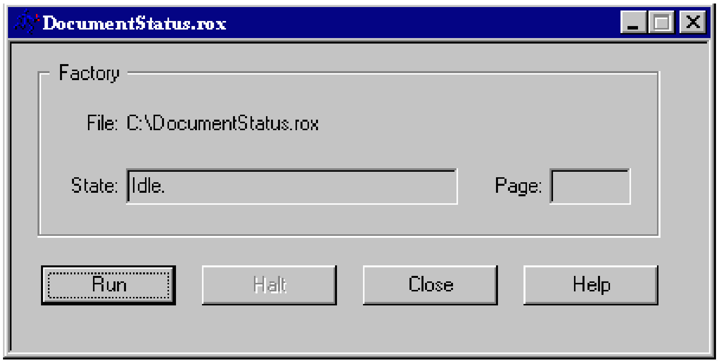
   Choose **File | Open** and set **Files of type** to **Report Executable (**.rox*)**. Set **Look in** to the directory in which you copied *DocumentStatus.rox*, and select the *DocumentStatus.rox* file.

Figure 3.3 Sample File | Open window



4. **In the DocumentStatus.rox window that pops up, click Run.**

Figure 3.4 DocumentStatus.rox utility

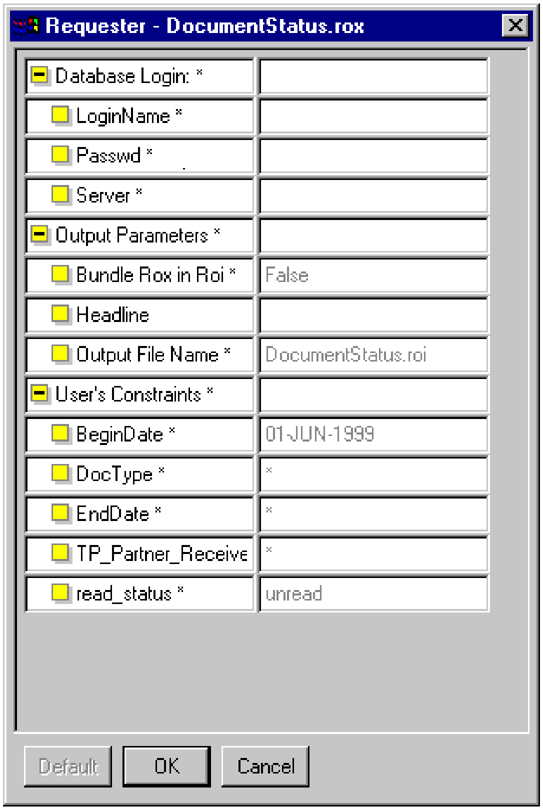


A pop-up window listing report parameters will open. (See Figure 3.5.)

5. **Select report parameters.**

In the parameters window, expand the menu options by clicking the items preceded by a plus sign (+). Once the main menu option are expanded, the plus sign will change to a minus sign (-). (See Figure 3.5.) Enter the appropriate values for each parameter field as indicated in Table 3.1.

Figure 3.5 Report parameters for the DocumentStatus.rox report utility



⊟ Database Login: *	
LoginName *	
Passwd *	
Server *	
⊟ Output Parameters *	
Bundle Rox in Roi *	False
Headline	
Output File Name *	DocumentStatus.roi
⊟ User's Constraints *	
BeginDate *	01-JUN-1999
DocType *	*
EndDate *	*
TP_Partner_Receive	*
read_status *	unread

Table 3.1 Report Parameters

Field	Description
	Database Login
Login Name	Your database user login name.
Passwd	Your database password.
Server	The name of the database server, as configured in the *tnsnames.ora* file.
	Output Parameters
Bundle Rox in Roi	`False`.

Field	Description (Continued)
Headline	This parameter is not used in the installation version of TradingXpert but is included to allow for extending the functionality of the report. The headline parameter is used when the reports are run through a web browser using the Actuate ReportCast channels. Refer to your Actuate documentation for more information.
Output File Name	The name of the report file that will be created.
User's Constraints	
BeginDate	The beginning date for the reporting period, in the format *dd-mmm-yyyy*.
DocType	The document type to be included in the report (e.g., 850).
EndDate	The ending date for the reporting period, in the format *dd-mmm-yyyy*.
TP_Partner_Receiver	The trading partner for whom you want to collect data.
read_status	The document status; the values are `read`,`unread` or `both`. The default value is `unread`.
[Buttons]	
Default button	Click **Default** to erase your changes to a field and replace it with the default value. This button is active only when you make changes to a parameter value.
OK button	Click this button to set the specified parameters.
Cancel button	Click this button to close the window without saving changes and exit the *DocumentStatus.rox* utility file.

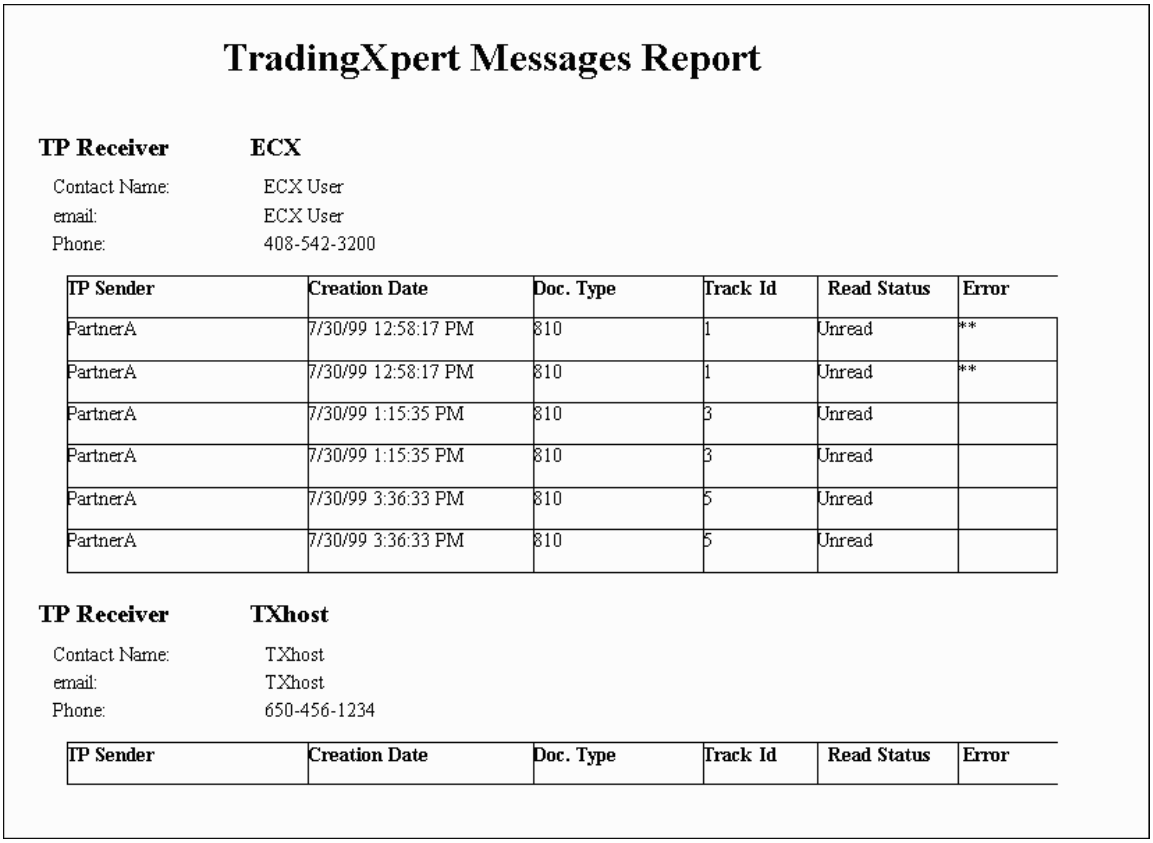
Note You can use an asterisk (*) to indicate all possible values for any field except for **Login Name**, **Passwd**, **Server**, and **Bundle Rox in Roi**.

6. **Click OK to set parameters and generate report.**

   Alternately, click **Cancel** to exit the screen without saving any changes.

   After you click **OK**, the system generates a report, the Actuate program starts, and the generated report is opened in the Actuate window.

Figure 3.6 Sample document status report generated by Actuate



TradingXpert Messages Report

TP Receiver **ECX**

Contact Name: ECX User
email: ECX User
Phone: 408-542-3200

TP Sender	Creation Date	Doc. Type	Track Id	Read Status	Error
PartnerA	7/30/99 12:58:17 PM	810	1	Unread	**
PartnerA	7/30/99 12:58:17 PM	810	1	Unread	**
PartnerA	7/30/99 1:15:35 PM	810	3	Unread	
PartnerA	7/30/99 1:15:35 PM	810	3	Unread	
PartnerA	7/30/99 3:36:33 PM	810	5	Unread	
PartnerA	7/30/99 3:36:33 PM	810	5	Unread	

TP Receiver **TXhost**

Contact Name: TXhost
email: TXhost
Phone: 650-456-1234

TP Sender	Creation Date	Doc. Type	Track Id	Read Status	Error

Appendix

A

TradingXpert EDI Document Formats

This appendix documents the ANSI X12 EDI document formats supported by TradingXpert. The following formats are covered:

- 214 - Transportation Carrier Shipment Status Message
- 810 - Invoice
- 830 - Planning Schedule with Release Capability
- 840 - Request for Quotation
- 843 - Response to Request for Quotation
- 850 - Purchase Order
- 855 - Purchase Order Acknowledgment
- 856 - Ship Notice/Manifest
- 860 - Purchase Order Change Request - Buyer Initiated
- 862 - Shipping Schedule

TradingXpert also supports the EDIFACT EDI document formats INVOIC and ORDERS, which are not documented in this appendix.

214 Transportation Carrier Shipment Status Message

Functional Group=QM

This Draft Standard for Trial Use contains the format and establishes the data contents of the Transportation Carrier Shipment Status Message Transaction Set (214) for use within the context of an Electronic Data Interchange (EDI) environment. This transaction set can be used by a transportation carrier to provide shippers, consignees, and their agents with the status of shipments in terms of dates, times, locations, route, identifying numbers, and conveyance.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	M	1		
020	B10	Beginning Segment for Transportation Carrier Shipment Status Message	M	1		
030	L11	Business Instructions and Reference Number	O	300		
LOOP ID - 0100					**10**	
050	N1	Name	O	1		
080	N4	Geographic Location	O	1		
LOOP ID - 0200					**999999**	
130	LX	Assigned Number	O	1		
LOOP ID - 0205					**10**	
140	AT7	Shipment Status Details	O	1		
143	MS1	Equipment, Shipment, or Real Property Location	O	1		
200	AT8	Shipment Weight, Packaging and Quantity Data	O	10		
610	SE	Transaction Set Trailer	M	1		

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	143	**Transaction Set Identifier Code**	M	ID	3/3
		Code Name			
		214 Transportation Carrier Shipment Status Message			
ST02	329	**Transaction Set Control Number**	M	AN	4/9

Semantics:

1. The transaction set identifier (ST01) is used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

B10 Beginning Segment for Transportation Carrier Shipment Status Message

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 3

To transmit identifying numbers and other basic data related to the transaction set.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
B1001	127	**Reference Identification**	C	AN	1/30
B1002	145	**Shipment Identification Number**	O	AN	1/30
B1003	140	**Standard Carrier Alpha Code**	M	ID	2/4

Syntax:

R0106 -- At least one of B1001 or B1006 is required.
E0105 -- Only one of B1001 or B1005 may be present.
P0506 -- If either B1005 or B1006 are present, then the others are required.

Semantics:

1. B1001 is the carrier-assigned reference number.
2. B1007 indicates if the reference numbers included in this transmission were transmitted to the carrier via EDI or were key-entered by the carrier. A "Y" indicates that the carrier received the reference numbers in an EDI transmission; an "N" indicates that the carrier did not receive the reference numbers in an EDI transmission and key entered the data from a shipper supplied document.

Comments:

1. B1001 is the carrier's PRO (invoice number) that identifies the shipment.
2. B1003 is required in Transaction Set 214.
3. B1006 is the carrier-assigned bar code identification or another carrier-assigned shipment identification, such as a manifest number.

L11 Business Instructions and Reference Number

Pos: 030 Max: 300
Heading - Optional
Loop: N/A Elements: 2

To specify instructions in this business relationship or a reference number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
L1101	**127**	**Reference Identification**	**C**	**AN**	**1/30**
L1102	**128**	**Reference Identification Qualifier**	**C**	**ID**	**2/3**
		Code Name			
		EQ Equipment Number			

Syntax:

1. L1101 R0103 -- At least one of L1101 or L1103 is required.
2. L1101 P0102 -- If either L1101 or L1102 are present, then the others are required.

N1 Name

Pos: 050 Max: 1
Heading - Optional
Loop: 0100 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	98	**Entity Identifier Code**	M	ID	2/3
		Code Name			
		SF Ship From			
		ST Ship To			
N102	93	**Name**	C	AN	1/60

Syntax:

1. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency, the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N4 Geographic Location

Pos: 080 Max: 1
Heading - Optional
Loop: 0100 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	**City Name**	O	AN	2/30
N402	156	**State or Province Code**	O	ID	2/2
N403	116	**Postal Code**	O	ID	3/15
N404	26	**Country Code**	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

LX Assigned Number

Pos: 130 Max: 1
Heading - Optional
Loop: 0200 Elements: 1

To reference a line number in a transaction set.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
LX01	554	Assigned Number	M	N0	1/6

AT7 Shipment Status Details

Pos: 140 Max: 1
Heading - Optional
Loop: 0205 Elements: 4

To specify the status of a shipment, the reason for that status, the date and time of the status, and the date and time of any appointments scheduled.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
AT701	1650	**Shipment Status Code**	C	ID	2/2
		Code Name			
		AF Carrier Departed Pick-up Location with Shipment			
		AJ Tendered for Delivery			
		K1 Arrived at Customs			
		S1 Trailer Spotted at Consignee's Location			
		X1 Arrived at Delivery Location			
AT702	1651	**Shipment Status or Appointment Reason Code**	C	ID	2/2
		Code Name			
		NS Normal Status			
AT705	373	**Date**	C	DT	8/8
AT706	337	**Time**	C	TM	4/8

Syntax:

1. AT701 E0103 -- Only one of AT701 or AT703 may be present.
2. AT701 P0102 -- If either AT701 or AT702 are present, then the others are required.
3. AT703 P0304 -- If either AT703 or AT704 are present, then the others are required.
4. AT706 C0605 -- If AT706 is present, then AT705 is required

5. AT707 C0706 -- If AT707 is present, then AT706 is required

Semantics:

1. If AT701 is present, AT705 is the date the status occurred. If AT703 is present, AT705 is a date related to an appointment.
2. If AT701 is present, AT706 is the time of the status. If AT703 is present, AT706 is the time of the appointment.
3. If AT707 is not present then AT706 represents local time of the status.

MS1 Equipment, Shipment, or Real Property Location

Pos: 143 Max: 1
Heading - Optional
Loop: 0205 Elements: 3

To specify the location of a piece of equipment, a shipment, or real property in terms of city and state or longitude and latitude

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
MS101	**19**	**City Name**	**C**	**AN**	**2/30**
MS102	**156**	**State or Province Code**	**C**	**ID**	**2/2**
MS103	**26**	**Country Code**	**C**	**ID**	**2/3**

Syntax:

L010203 -- If MS101 is present, then at least one of MS102 or MS103 is required.
E0104 -- Only one of MS101 or MS104 may be present.
C0201 -- If MS102 is present, then MS101 is required
C0301 -- If MS103 is present, then MS101 is required
P0405 -- If either MS104 or MS105 are present, then the others are required.
C0604 -- If MS106 is present, then MS104 is required
C0705 -- If MS107 is present, then MS105 is required

Semantics:

1. MS104 is the longitude expressed in Degrees, Minutes, and Seconds.
2. MS105 is the latitude expressed in Degrees, Minutes, and Seconds.
3. MS106 may only be 'E' or 'W'.
4. MS107 may only be 'N' or 'S'.

AT8 Shipment Weight, Packaging and Quantity Data

Pos: 200 Max: 10
Heading - Optional
Loop: 0200 Elements: 3

To specify shipment details in terms of weight, and quantity of handling units.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
AT801	**187**	**Weight Qualifier**	**C**	**ID**	**1/2**
		Code Name			
		G Gross Weight			
AT802	**188**	**Weight Unit Code**	**C**	**ID**	**1/1**
		Code Name			
		K Kilograms			
		L Pounds			
AT803	**81**	**Weight**	**C**	**R**	**1/10**

Syntax:

1. AT801 P010203 -- If either AT801, AT802 or AT803 are present, then the others are required.
2. AT806 P0607 -- If either AT806 or AT807 are present, then the others are required.

Semantics:

1. AT804 is the quantity of handling units that are not unitized (for example a carton). When added to the quantity in AT805, it is the total quantity of handling units in the shipment.
2. AT805 is the quantity of handling units that are unitized (for example on a pallet or slip sheet). When added to the quantity in AT804 it is the total quantity of handling units for the shipment.

SE Transaction Set Trailer

Pos: 610 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	96	Number of Included Segments	M	N0	1/10
SE02	329	Transaction Set Control Number	M	AN	4/9

Comments:

1. SE is the last segment of each transaction set.

810 Invoice

Functional Group=IN

This Draft Standard for Trial Use contains the format and establishes the data contents of the Invoice Transaction Set (810) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to provide for customary and established business and industry practice relative to the billing for goods and services provided.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	M	1		
020	BIG	Beginning Segment for Invoice	M	1		
030	NTE	Note/Special Instruction	O	100		
050	REF	Reference Identification	O	12		
LOOP ID - N1					**200**	
070	N1	Name	O	1		
090	N3	Address Information	O	2		
100	N4	Geographic Location	O	1		
120	PER	Administrative Communications Contact	O	3		
130	ITD	Terms of Sale/Deferred Terms of Sale	O	>1		
140	DTM	Date/Time Reference	O	10		
150	FOB	F.O.B. Related Instructions	O	1		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - IT1					**200000**	
010	IT1	Baseline Item Data (Invoice)	O	1		
030	IT3	Additional Item Data	O	5		
LOOP ID - PID					**1000**	
060	PID	Product/Item Description	O	1		

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	TDS	Total Monetary Value Summary	M	1		
030	CAD	Carrier Detail	O	1		
LOOP ID - ISS					>1	
060	ISS	Invoice Shipment Summary	O	1		
080	SE	Transaction Set Trailer	M	1		

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	**143**	**Transaction Set Identifier Code**	**M**	**ID**	**3/3**
		Code Name			
		810 Invoice			
ST02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BIG Beginning Segment for Invoice

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 5

To indicate the beginning of an invoice transaction set and transmit identifying numbers and dates

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BIG01	373	**Date**	M	DT	8/8
BIG02	76	**Invoice Number**	M	AN	1/22
BIG03	373	**Date**	O	DT	8/8
BIG04	324	**Purchase Order Number**	O	AN	1/22
BIG07	640	**Transaction Type Code**	O	ID	2/2
		Code Name			
		CN Credit Invoice			

Semantics:

1. BIG01 is the invoice issue date.
2. BIG03 is the date assigned by the purchaser to purchase order.
3. BIG10 indicates the consolidated invoice number. When BIG07 contains code CI, BIG10 is not used.

Comments:

1. BIG07 is used only to further define the type of invoice when needed.

NTE Note/Special Instruction

Pos: 030 Max: 100
Heading.6
- Optional
Loop: N/A Elements: 2

To transmit information in a free-form format, if necessary, for comment or special instruction

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
NTE01	363	**Note Reference Code**	O	ID	3/3
		Code Name			
		DEL Delivery			
NTE02	352	**Description**	M	AN	1/80

Comments:

1. The NTE segment permits free-form information/data which, under ANSI X12 standard implementations, is not machine processable. The use of the NTE segment should therefore be avoided, if at all possible, in an automated environment.

REF Reference Identification

Pos: 050 Max: 12
Heading - Optional
Loop: N/A Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		GT Goods and Service Tax Registration Number			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

N1 Name

Pos: 070 Max: 1
Heading - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	**98**	**Entity Identifier Code**	**M**	**ID**	**2/3**
		All valid X12 codes are used.			
N102	**93**	**Name**	**C**	**AN**	**1/60**

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N3 Address Information

Pos: 090 Max: 2
Heading - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 100 Max: 1
Heading - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

PER Administrative Communications Contact

Pos: 120 Max: 3
Heading - Optional
Loop: N1 Elements: 8

To identify a person or office to whom administrative communications should be directed.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PER01	366	**Contact Function Code**	M	ID	2/2
		Code Name			
		BI Bill Inquiry Contact			
		CN General Contact			
		DC Delivery Contact			
		IC Information Contact			
		SD Shipping Department			
		ZZ Mutually Defined			
PER02	93	**Name**	O	AN	1/60
PER03	365	**Communication Number Qualifier**	C	ID	2/2
		Code Name			
		TE Telephone			
PER04	364	**Communication Number**	C	AN	1/80
PER05	365	**Communication Number Qualifier**	C	ID	2/2
		Code Name			
		FX Facsimile			
PER06	364	**Communication Number**	C	AN	1/80
PER07	365	**Communication Number Qualifier**	C	ID	2/2
		Code Name			
		EM Electronic Mail			
PER08	364	**Communication Number**	C	AN	1/80

Syntax:

1. PER03 P0304 -- If either PER03 or PER04 are present, then the others are required.
2. PER05 P0506 -- If either PER05 or PER06 are present, then the others are required.
3. PER07 P0708 -- If either PER07 or PER08 are present, then the others are required.

ITD Terms of Sale/Deferred Terms of Sale

Pos: 130 Max: >1
Heading - Optional
Loop: N/A Elements: 5

To specify terms of sale.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ITD01	**336**	**Terms Type Code**	**O**	**ID**	**2/2**
		Code Name 01 Basic			
ITD03	**338**	**Terms Discount Percent**	**O**	**R**	**1/6**
ITD05	**351**	**Terms Discount Days Due**	**C**	**N0**	**1/3**
ITD07	**386**	**Terms Net Days**	**O**	**N0**	**1/3**
ITD12	**352**	**Description**	**O**	**AN**	**1/80**

Syntax:

1. ITD03 L03040513 -- If ITD03 is present, then at least one of ITD04, ITD05 or ITD13 is required.
2. ITD08 L08040513 -- If ITD08 is present, then at least one of ITD04, ITD05 or ITD13 is required.
3. ITD09 L091011 -- If ITD09 is present, then at least one of ITD10 or ITD11 is required.

Semantics:

1. ITD15 is the percentage applied to a base amount used to determine a late payment charge.

Comments:

1. If the code in ITD01 is "04", then ITD07 or ITD09 is required and either ITD10 or ITD11 is required; if the code in ITD01 is "05", then ITD06 or ITD07 is required.

DTM Date/Time Reference

Pos: 140 Max: 10
Heading - Optional
Loop: N/A Elements: 2

To specify pertinent dates and times.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
DTM01	374	**Date/Time Qualifier**	M	ID	3/3
		Code Name			
		011 Shippe			
DTM02	373	**Date**	C	DT	8/8

Syntax:

1. DTM02 R020305 -- At least one of DTM02, DTM03 or DTM05 is required.
2. DTM04 C0403 -- If DTM04 is present, then DTM03 is required
3. DTM05 P0506 -- If either DTM05 or DTM06 are present, then the others are required.

FOB F.O.B. Related Instructions

Pos: 150 Max: 1
Heading - Optional
Loop: N/A Elements: 5

To specify transportation instructions relating to shipment.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
FOB01	146	**Shipment Method of Payment**	M	ID	2/2
		Code Name			
		CC Collect			
		NC Service Freight,No Charge			
		PC Prepaid but Charged to Customer			
		PP Prepaid (by Seller)			
FOB02	309	**Location Qualifier**	C	ID	1/2
		Code Name			
		OP Other Unlisted Acceptance Point			
		OR Origin (Shipping Point)			
		WH Warehouse			

FOB03	**352**	**Description**	**O**	**AN**	**1/80**
FOB06	**309**	**Location Qualifier**	**C**	**ID**	**1/2**
		Code Name			
		DE Destination (Shipping)			
		FA Factory			
		OP Other Unlisted Acceptance Point			
		WH Warehouse			
FOB07	**352**	**Description**	**O**	**AN**	**1/80**

Syntax:

1. FOB03 C0302 -- If FOB03 is present, then FOB02 is required
2. FOB04 C0405 -- If FOB04 is present, then FOB05 is required
3. FOB07 C0706 -- If FOB07 is present, then FOB06 is required
4. FOB08 C0809 -- If FOB08 is present, then FOB09 is required

Semantics:

1. FOB01 indicates which party will pay the carrier.
2. FOB02 is the code specifying transportation responsibility location.
3. FOB06 is the code specifying the title passage location.
4. FOB08 is the code specifying the point at which the risk of loss transfers. This may be different than the location specified in FOB02/FOB03 and FOB06/FOB07.

IT1 Baseline Item Data (Invoice)

Pos: 010 Max: 1
Detail - Optional
Loop: IT1 Elements: 6

To specify the basic and most frequently used line item data for the invoice and related transactions

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
IT101	**350**	**Assigned Identification**	**O**	**AN**	**1/20**
IT102	**358**	**Quantity Invoiced**	**C**	**R**	**1/10**
IT103	**355**	**Unit or Basis for Measurement Code**	**C**	**ID**	**2/2**
		Code Name			
		EA Each			

IT104	**212**	**Unit Price**	**C**	**R**	**1/17**
IT106	**235**	**Product/Service ID Qualifier** **Code Name** BP Buyer's Part Number	**C**	**ID**	**2/2**
IT107	**234**	**Product/Service ID**	**C**	**AN**	**1/48**

Syntax:

1. IT102 P020304 -- If either IT102, IT103 or IT104 are present, then the others are required.
2. IT106 P0607 -- If either IT106 or IT107 are present, then the others are required.
3. IT108 P0809 -- If either IT108 or IT109 are present, then the others are required.
4. IT110 P1011 -- If either IT110 or IT111 are present, then the others are required.
5. IT112 P1213 -- If either IT112 or IT113 are present, then the others are required.
6. IT114 P1415 -- If either IT114 or IT115 are present, then the others are required.
7. IT116 P1617 -- If either IT116 or IT117 are present, then the others are required.
8. IT118 P1819 -- If either IT118 or IT119 are present, then the others are required.
9. IT120 P2021 -- If either IT120 or IT121 are present, then the others are required.
10. IT122 P2223 -- If either IT122 or IT123 are present, then the others are required.
11. IT124 P2425 -- If either IT124 or IT125 are present, then the others are required.

Semantics:

1. IT101 is the purchase order line item identification.

Comments:

1. Element 235/234 combinations should be interpreted to include products and/or services. See the Data Dictionary for a complete list of IDs.
2. IT106 through IT125 provide for ten different product/service IDs for each item. For example: Case, Color, Drawing No., U.P.C. No., ISBN No., Model No., or SKU.

IT3 Additional Item Data

Pos: 030 Max: 5
Detail - Optional
Loop: IT1 Elements: 4

To specify additional item details relating to variations between ordered and shipped quantities, or to specify alternate units of measures and quantities

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
IT301	382	**Number of Units Shipped**	C	R	1/10
IT302	355	**Unit or Basis for Measurement Code**	C	ID	2/2
		Code Name			
		BR Barrel			
		CT Carton			
		DZ Dozen			
		EA Each			
IT303	368	**Shipment/Order Status Code**	C	ID	2/2
		Code Name			
		BO Back Ordered			
		CP Partial Shipment on (Date),Considered No Backorder			
		CU Cancelled Line Item			
		DE Deleted Order			
		OR Temporarily Out of Stock - Reorder			
		UN Unavailable			
		ZZ Mutually Defined			
IT304	383	**Quantity Difference**	C	R	1/9

Syntax:

1. IT301 P0102 -- If either IT301 or IT302 are present, then the others are required.
2. IT301 R01030405 -- At least one of IT301, IT303, IT304 or IT305 is required.

PID Product/Item Description

Pos: 060 Max: 1
Detail - Optional
Loop: PID Elements: 2

To describe a product or process in coded or free-form format

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PID01	349	**Item Description Type**	M	ID	1/1
		Code Name			
		F Free-form			
PID05	352	**Description**	C	AN	1/80

Syntax:

1. PID04 C0403 -- If PID04 is present, then PID03 is required

2. PID04 R0405 -- At least one of PID04 or PID05 is required.
3. PID07 C0703 -- If PID07 is present, then PID03 is required
4. PID08 C0804 -- If PID08 is present, then PID04 is required
5. PID09 C0905 -- If PID09 is present, then PID05 is required

Semantics:

1. Use PID03 to indicate the organization that publishes the code list being referred to.
2. PID04 should be used for industry-specific product description codes.
3. PID08 describes the physical characteristics of the product identified in PID04. A "Y" indicates that the specified attribute applies to this item; an "N" indicates it does not apply. Any other value is indeterminate.
4. PID09 is used to identify the language being used in PID05.

Comments:

1. If PID01 equals "F", then PID05 is used. If PID01 equals "S", then PID04 is used. If PID01 equals "X", then both PID04 and PID05 are used.
2. Use PID06 when necessary to refer to the product surface or layer being described in the segment.
3. PID07 specifies the individual code list of the agency specified in PID03.

TDS Total Monetary Value Summary

Pos: 010 Max: 1
Summary - Mandatory
Loop: N/A Elements: 1

To specify the total invoice discounts and amounts.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
TDS01	**610**	**Amount**	**M**	**N2**	**1/15**

Semantics:

1. TDS01 is the total amount of invoice (including charges, less allowances) before terms discount (if discount is applicable).
2. TDS02 indicates the amount upon which the terms discount amount is calculated.
3. TDS03 is the amount of invoice due if paid by terms discount due date (total invoice or installment amount less cash discount).
4. TDS04 indicates the total amount of terms discount.

Comments:

1. TDS02 is required if the dollar value subject to discount is not equal to the dollar value of TDS01.

CAD Carrier Detail

Pos: 030 Max: 1
Summary - Optional
Loop: N/A Elements: 1

To specify transportation details for the transaction.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
CAD05	**387**	**Routing**	**C**	**AN**	**1/35**

Syntax:

1. CAD05 R0504 -- At least one of CAD05 or CAD04 is required.
2. CAD07 C0708 -- If CAD07 is present, then CAD08 is required

ISS Invoice Shipment Summary

Pos: 060 Max: 1
Summary - Optional
Loop: ISS Elements: 4

To specify summary details of total items shipped in terms of quantity, weight, and volume

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ISS01	**382**	**Number of Units Shipped**	**C**	**R**	**1/10**
ISS02	**355**	**Unit or Basis for Measurement Code**	**C**	**ID**	**2/2**
		Code Name			
		BR Barrel			
		DZ Dozen			
		EA Each			
ISS03	**81**	**Weight**	**C**	**R**	**1/10**
ISS04	**355**	**Unit or Basis for Measurement Code**	**C**	**ID**	**2/2**
		Code Name			
		01 Actual Pounds			

Syntax:

1. ISS01 R010305 -- At least one of ISS01, ISS03 or ISS05 is required.
2. ISS01 P0102 -- If either ISS01 or ISS02 are present, then the others are required.
3. ISS03 P0304 -- If either ISS03 or ISS04 are present, then the others are required.
4. ISS05 P0506 -- If either ISS05 or ISS06 are present, then the others are required.

Semantics:

1. ISS07 is the quantity of third party pallets.
2. ISS08 is the gross weight of third party pallets in pounds.

SE Transaction Set Trailer

Pos: 080 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	**96**	**Number of Included Segments**	**M**	**N0**	**1/10**
SE02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Comments:

1. SE is the last segment of each transaction set.

830 Planning Schedule with Release Capability

Functional Group=PS

This Draft Standard for Trial Use contains the format and establishes the data contents of the Planning Schedule with Release Capability Transaction Set (830) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to provide for customary and established business practice relative to the transfer of forecasting/material release information between organizations.The planning schedule transaction may be used in various ways or in a combination of ways, such as: (1) a simple forecast; (2) a forecast with the buyer's authorization for the seller to commit to resources, such as labor or material; (3) a forecast that is also used as an order release mechanism, containing such elements as resource authorizations, period-to-date cumulative quantities, and specific ship/delivery patterns for requirements that have been represented in "buckets," such as weekly, monthly, or quarterly. The order release forecast may also contain all data related to purchase orders, as required, because the order release capability eliminates the need for discrete generation of purchase orders.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	**M**	**1**		
020	BFR	Beginning Segment for Planning Schedule	**M**	**1**		
LOOP ID - N1					**200**	
230	N1	Name	**O**	**1**		
240	N2	Additional Name Information	**O**	**2**		
250	N3	Address Information	**O**	**2**		
260	N4	Geographic Location	**O**	**1**		
270	REF	Reference Identification	**O**	**12**		
280	PER	Administrative Communications Contact	**O**	**3**		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - LIN					**>1**	
010	LIN	Item Identification	**M**	**1**		
020	UIT	Unit Detail	**O**	**1**		
140	REF	Reference Identification	**O**	**12**		
LOOP ID - N1					**200**	
320	N1	Name	**O**	**1**		
330	N2	Additional Name Information	**O**	**2**		
340	N3	Address Information	**O**	**2**		
350	N4	Geographic Location	**O**	**1**		
360	REF	Reference Identification	**O**	**12**		

LOOP ID - FST					>1	
410	FST	Forecast Schedule	**O**	**1**		**N2/410**

LOOP ID - SHP					25	
470	SHP	Shipped/Received Information	**O**	**1**		

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
020	SE	Transaction Set Trailer	**M**	**1**		

Notes:

2/410 - At least one occurrence of segment FST is required, either in the FST loop or within the SDP loop. These two loops are mutually exclusive.

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	**143**	**Transaction Set Identifier Code**	**M**	**ID**	**3/3**
		Code Name			
		830 Planning Schedule with Release Capability			
ST02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BFR Beginning Segment for Planning Schedule

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 5

To indicate the beginning of a planning schedule transaction set; whether a ship or delivery based forecast; and related forecast envelope dates

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BFR01	**353**	**Transaction Set Purpose Code**	**M**	**ID**	**2/2**
		Code Name			
		00 Original			
		04 Change			
BFR03	**328**	**Release Number**	**C**	**AN**	**1/30**
BFR04	**675**	**Schedule Type Qualifier**	**M**	**ID**	**2/2**
		All valid X12 codes are used.			
BFR05	**676**	**Schedule Quantity Qualifier**	**M**	**ID**	**1/1**
		All valid X12 codes are used.			
BFR06	**373**	**Date**	**M**	**DT**	**8/8**

Syntax:

1. BFR02 R0203 -- At least one of BFR02 or BFR03 is required.

Semantics:

1. If BFR01 contains the value "04" (Net Change), BFR09 is required.
2. BFR02 is the identifying number for a forecast assigned by the orderer/purchaser.
3. BFR06 is the forecast horizon start date: The date when the forecast horizon (envelope) begins.
4. BFR07 is the forecast horizon end date: The date when the forecast horizon (envelope) ends.
5. BFR08 is the date forecast generated: The date the forecast data was generated.
6. BFR09 is the date forecast updated: The date the forecast was updated with "net change" data. (Used only when data element 353 in BFR01 contains the value "04", meaning net change.)

N1 Name

Pos: 230 Max: 1
Heading - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	**98**	**Entity Identifier Code**	**M**	**ID**	**2/3**
		Code Name			
		ST Ship To			
N102	**93**	**Name**	**C**	**AN**	**1/60**

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 240 Max: 2
Heading - Optional
Loop: N1 Elements: 1

To specify additional names or those longer than 35 characters in length

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	**93**	**Name**	**M**	**AN**	**1/60**

N3 Address Information

Pos: 250 Max: 2
Heading - Optional
Loop: N1 Elements: 2

To specify the location of the named party

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	**166**	**Address Information**	**M**	**AN**	**1/55**
N302	**166**	**Address Information**	**O**	**AN**	**1/55**

N4 Geographic Location

Pos: 260 Max: 1
Heading - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

REF Reference Identification

Pos: 270 Max: 12
Heading - Optional
Loop: N1 Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	128	Reference Identification Qualifier Code Name CT Contract Number DO Delivery Order Number IK Invoice Number	M	ID	2/3
REF02	127	Reference Identification	C	AN	1/30

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

PER Administrative Communications Contact

Pos: 280 Max: 3
Heading - Optional
Loop: N1 Elements: 8

To identify a person or office to whom administrative communications should be directed.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PER01	**366**	**Contact Function Code**	**M**	**ID**	**2/2**
		Code Name			
		IC Information Contact			
PER02	**93**	**Name**	**O**	**AN**	**1/60**
PER03	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		TE Telephone			
PER04	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER05	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		FX Facsimile			
PER06	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER07	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		EM Electronic Mail			
PER08	**364**	**Communication Number**	**C**	**AN**	**1/80**

Syntax:

1. PER03 P0304 -- If either PER03 or PER04 are present, then the others are required.
2. PER05 P0506 -- If either PER05 or PER06 are present, then the others are required.
3. PER07 P0708 -- If either PER07 or PER08 are present, then the others are required.

LIN Item Identification

Pos: 010 Max: 1	
Detail - Mandatory	
Loop: LIN Elements: 5	

To specify basic item identification data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
LIN01	**350**	**Assigned Identification**	**O**	**AN**	**1/20**
LIN02	**235**	**Product/Service ID Qualifier**	**M**	**ID**	**2/2**
		Code Name			
		BP Buyer's Part Number			
LIN03	**234**	**Product/Service ID**	**M**	**AN**	**1/48**
LIN04	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		VP Vendor's (Seller's) Part Number			
LIN05	**234**	**Product/Service ID**	**C**	**AN**	**1/48**

Syntax:

1. LIN04 P0405 -- If either LIN04 or LIN05 are present, then the others are required.
2. LIN06 P0607 -- If either LIN06 or LIN07 are present, then the others are required.
3. LIN08 P0809 -- If either LIN08 or LIN09 are present, then the others are required.
4. LIN10 P1011 -- If either LIN10 or LIN11 are present, then the others are required.
5. LIN12 P1213 -- If either LIN12 or LIN13 are present, then the others are required.
6. LIN14 P1415 -- If either LIN14 or LIN15 are present, then the others are required.
7. LIN16 P1617 -- If either LIN16 or LIN17 are present, then the others are required.
8. LIN18 P1819 -- If either LIN18 or LIN19 are present, then the others are required.
9. LIN20 P2021 -- If either LIN20 or LIN21 are present, then the others are required.
10. LIN22 P2223 -- If either LIN22 or LIN23 are present, then the others are required.
11. LIN24 P2425 -- If either LIN24 or LIN25 are present, then the others are required.
12. LIN26 P2627 -- If either LIN26 or LIN27 are present, then the others are required.
13. LIN28 P2829 -- If either LIN28 or LIN29 are present, then the others are required.
14. LIN30 P3031 -- If either LIN30 or LIN31 are present, then the others are required.

Semantics:

1. LIN01 is the line item identification

Comments:

1. See the Data Dictionary for a complete list of IDs.
2. LIN02 through LIN31 provide for fifteen different product/service IDs for each item. For example: Case, Color, Drawing No., U.P.C. No., ISBN No., Model No., or SKU.

UIT Unit Detail

Pos: 020 Max: 1
Detail - Optional
Loop: LIN Elements: 1

To specify item unit data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
UIT01	**C001**	**Composite Unit of Measure**	**M**	**Com p**	
	355	**Unit or Basis for Measurement Code**	**M**	**ID**	**2/2**
		Code Name			
		BR Barrel			
		CT Carton			
		DZ Dozen			
		EA Each			

Syntax:

1. UIT03 C0302 -- If UIT03 is present, then UIT02 is required

REF Reference Identification

Pos: 140 Max: 12
Detail - Optional
Loop: LIN Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		DK Dock Number			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

N1 Name

Pos: 320 Max: 1
Detail - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	**98**	**Entity Identifier Code**	**M**	**ID**	**2/3**
		Code Name			
		SF Ship From			
N102	**93**	**Name**	**C**	**AN**	**1/60**

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 330 Max: 2
Detail - Optional
Loop: N1 Elements: 1

To specify additional names or those longer than 35 characters in length.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	93	Name	M	AN	1/60

N3 Address Information

Pos: 340 Max: 2
Detail - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 350 Max: 1
Detail - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

REF Reference Identification

Pos: 360 Max: 12
Detail - Optional
Loop: N1 Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		LF Assembly Line Feed Location			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

FST Forecast Schedule

Pos: 410 Max: 1
Detail - Optional
Loop: FST Elements: 5

To specify the forecasted dates and quantities.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
FST01	**380**	**Quantity**	**M**	**R**	**1/15**
FST02	**680**	**Forecast Qualifier**	**M**	**ID**	**1/1**
		Code Name			
		A Immediate			
		C Firm			
		D Planning			

FST03	**681**	**Forecast Timing Qualifier** **Code Name** C Daily D Discrete M Monthly Bucket (Calendar Months)	**M**	**ID**	**1/1**
FST04	**373**	**Date**	**M**	**DT**	**8/8**
FST05	**373**	**Date**	**O**	**DT**	**8/8**

Syntax:

1. FST06 P0607 -- If either FST06 or FST07 are present, then the others are required.
2. FST08 P0809 -- If either FST08 or FST09 are present, then the others are required.

Semantics:

1. If FST03 equals "F" (indicating flexible interval), then FST04 and FST05 are required. FST04 would be used for the start date of the flexible interval and FST05 would be used for the end date of the flexible interval.

Comments:

1. As qualified by FST02 and FST03, FST04 represents either a discrete forecast date, the first date of a forecasted bucket (weekly, monthly, quarterly, etc.) or the start date of a flexible interval.
2. FST06 qualifies the time in FST07. The purpose of the FST07 element is to express the specific time of day in a 24-hour clock to satisfy "just-in-time" requirements. As an alternative, the ship/delivery pattern segment (SDP) may be used to define an approximate time, such as a.m. or p.m.

SHP Shipped/Received Information

Pos: 470 Max: 1
Detail - Optional
Loop: SHP Elements: 4

To specify shipment and/or receipt information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SHP01	**673**	**Quantity Qualifier** **Code Name** 02 Cumulative Quantity	**O**	**ID**	**2/2**

SHP02	**380**	**Quantity**	**C**	**R**	**1/15**
SHP03	**374**	**Date/Time Qualifier**	**C**	**ID**	**3/3**
		Code Name			
		011 Shipped			
SHP04	**373**	**Date**	**C**	**DT**	**8/8**

Syntax:

1. SHP01 C0102 -- If SHP01 is present, then SHP02 is required
2. SHP03 L030405 -- If SHP03 is present, then at least one of SHP04 or SHP05 is required.
3. SHP04 C0403 -- If SHP04 is present, then SHP03 is required
4. SHP05 C0503 -- If SHP05 is present, then SHP03 is required

Semantics:

1. SHP04 is the date shipped, delivered, received, or the cumulative quantity start date (as qualified by SHP03).
2. SHP06 is the cumulative quantity end date.

Comments:

1. The SHP segment is used to communicate shipment, delivery, or receipt information and may include discrete or cumulative quantities, dates, and times.
2. If SHP01 equals "02", "07", "08", "09", or "10" (indicating cumulative quantities), then SHP04 and SHP06 are required to identify the start and end dates of the quantity count.

SE Transaction Set Trailer

Pos: 020 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	**96**	**Number of Included Segments**	**M**	**N0**	**1/10**
SE02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Comments:

1. SE is the last segment of each transaction set.

840 Request for Quotation

Functional Group=RQ

This Draft Standard for Trial Use contains the format and establishes the data contents of the Request for Quotation Transaction Set (840) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to provide potential buyers with the ability to solicit price, delivery schedule, and other items from potential sellers of goods and services.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	**M**	**1**		
020	BQT	Beginning Segment for Request for Quotation	**M**	**1**		
050	REF	Reference Identification	**O**	**>1**		
060	PER	Administrative Communications Contact	**O**	**3**		
LOOP ID - N1					**10000**	
310	N1	Name	**O**	**1**		
320	N2	Additional Name Information	**O**	**2**		
330	N3	Address Information	**O**	**2**		
340	N4	Geographic Location	**O**	**>1**		
350	REF	Reference Identification	**O**	**12**		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - PO1					**100000**	
010	PO1	Baseline Item Data	**M**	**1**		
LOOP ID - PID					**1000**	
050	PID	Product/Item Description	**O**	**1**		
200	DTM	Date/Time Reference	**O**	**10**		

LOOP ID - N1					>1
340	N1	Name	O	1	
350	N2	Additional Name Information	O	2	
360	N3	Address Information	O	2	
370	N4	Geographic Location	O	1	
380	REF	Reference Identification	O	12	

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
020	SE	Transaction Set Trailer	M	1		

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	143	**Transaction Set Identifier Code**	M	ID	3/3
		Code Name 840 Request for Quotation			
ST02	329	**Transaction Set Control Number**	M	AN	4/9

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BQT Beginning Segment for Request for Quotation

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 5

To indicate the beginning of a Request for Quotation Transaction Set and transmit identifying numbers and dates.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BQT01	**353**	**Transaction Set Purpose Code**	**M**	**ID**	**2/2**
		Code Name			
		00 Original			
BQT02	**586**	**Request for Quote Reference Number**	**M**	**AN**	**1/45**
BQT03	**373**	**Date**	**M**	**DT**	**8/8**
BQT04	**374**	**Date/Time Qualifier**	**C**	**ID**	**3/3**
		Code Name			
		001 Cancel After			
BQT05	**373**	**Date**	**C**	**DT**	**8/8**

Syntax:

1. BQT04 P0405 -- If either BQT04 or BQT05 are present, then the others are required.

Semantics:

1. BQT03 is the date to be used for reference purposes in an RFQ and a response to RFQ.

REF Reference Identification

Pos: 050 Max: >1
Heading - Optional
Loop: N/A Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		65 Total Order Cycle Number			
		DS Defense Priorities Allocation System (DPAS) Priority Rating			
		RQ Purchase Requisition Number			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

PER Administrative Communications Contact

Pos: 060 Max: 3
Heading - Optional
Loop: N/A Elements: 8

To identify a person or office to whom administrative communications should be directed.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PER01	**366**	**Contact Function Code**	**M**	**ID**	**2/2**
		Code Name			
		IC Information Contact			
PER02	**93**	**Name**	**O**	**AN**	**1/60**
PER03	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		TE Telephone			
PER04	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER05	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		FX Facsimile			
PER06	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER07	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		EM Electronic Mail			
PER08	**364**	**Communication Number**	**C**	**AN**	**1/80**

Syntax:

1. PER03 P0304 -- If either PER03 or PER04 are present, then the others are required.
2. PER05 P0506 -- If either PER05 or PER06 are present, then the others are required.
3. PER07 P0708 -- If either PER07 or PER08 are present, then the others are required.

N1 Name

Pos: 310 Max: 1
Heading - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	98	**Entity Identifier Code**	M	ID	2/3
		Code Name			
		SF Ship From			
N102	93	**Name**	C	AN	1/60

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 320 Max: 2
Heading - Optional
Loop: N1 Elements: 1

To specify additional names or those longer than 35 characters in length.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	93	**Name**	M	AN	1/60

N3 Address Information

Pos: 330 Max: 2
Heading - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 340 Max: >1
Heading - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

REF Reference Identification

Pos: 350 Max: 12
Heading - Optional
Loop: N1 Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	128	**Reference Identification Qualifier**	M	ID	2/3
		Code Name			
		W7 Commercial and Government Entity (CAGE) Code			
REF02	127	**Reference Identification**	C	AN	1/30

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

2. REF04 contains data relating to the value cited in REF02.

PO1 Baseline Item Data

Pos: 010 Max: 1
Detail - Mandatory
Loop: PO1 Elements: 9

To specify basic and most frequently used line item data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PO101	350	**Assigned Identification**	O	AN	1/20
PO102	330	**Quantity Ordered**	C	R	1/15
PO103	355	**Unit or Basis for Measurement Code**	O	ID	2/2
		Code Name			
		BR Barrel			
		CT Carton			
		DZ Dozen			
		EA Each			
PO106	235	**Product/Service ID Qualifier**	C	ID	2/2
		Code Name			
		BP Buyer's Part Number			
PO107	234	**Product/Service ID**	C	AN	1/48
PO108	235	**Product/Service ID Qualifier**	C	ID	2/2
		Code Name			
		VP Vendor's (Seller's) Part Number			

PO109	**234**	**Product/Service ID**	**C**	**AN**	**1/48**
PO110	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		MF Manufacturer			
PO111	**234**	**Product/Service ID**	**C**	**AN**	**1/48**

Syntax:

1. PO103 C0302 -- If PO103 is present, then PO102 is required
2. PO105 C0504 -- If PO105 is present, then PO104 is required
3. PO106 P0607 -- If either PO106 or PO107 are present, then the others are required.
4. PO108 P0809 -- If either PO108 or PO109 are present, then the others are required.
5. PO110 P1011 -- If either PO110 or PO111 are present, then the others are required.
6. PO112 P1213 -- If either PO112 or PO113 are present, then the others are required.
7. PO114 P1415 -- If either PO114 or PO115 are present, then the others are required.
8. PO116 P1617 -- If either PO116 or PO117 are present, then the others are required.
9. PO118 P1819 -- If either PO118 or PO119 are present, then the others are required.
10. PO120 P2021 -- If either PO120 or PO121 are present, then the others are required.
11. PO122 P2223 -- If either PO122 or PO123 are present, then the others are required.
12. PO124 P2425 -- If either PO124 or PO125 are present, then the others are required.

Comments:

1. See the Data Element Dictionary for a complete list of IDs.
2. PO101 is the line item identification.
3. PO106 through PO125 provide for ten different product/service IDs per each item. For example: Case, Color, Drawing No., U.P.C. No., ISBN No., Model No., or SKU.

PID Product/Item Description

Pos: 050 Max: 1
Detail - Optional
Loop: PID Elements: 2

To describe a product or process in coded or free-form format.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PID01	349	**Item Description Type**	M	ID	1/1
		Code Name			
		F Free-form			
PID05	352	**Description**	C	AN	1/80

Syntax:

1. PID04 C0403 -- If PID04 is present, then PID03 is required
2. PID04 R0405 -- At least one of PID04 or PID05 is required.
3. PID07 C0703 -- If PID07 is present, then PID03 is required
4. PID08 C0804 -- If PID08 is present, then PID04 is required
5. PID09 C0905 -- If PID09 is present, then PID05 is required

Semantics:

1. Use PID03 to indicate the organization that publishes the code list being referred to.
2. PID04 should be used for industry-specific product description codes.
3. PID08 describes the physical characteristics of the product identified in PID04. A "Y" indicates that the specified attribute applies to this item; an "N" indicates it does not apply. Any other value is indeterminate.
4. PID09 is used to identify the language being used in PID05.

Comments:

1. If PID01 equals "F", then PID05 is used. If PID01 equals "S", then PID04 is used. If PID01 equals "X", then both PID04 and PID05 are used.
2. Use PID06 when necessary to refer to the product surface or layer being described in the segment.
3. PID07 specifies the individual code list of the agency specified in PID03.

DTM Date/Time Reference

Pos: 200 Max: 10
Detail - Optional
Loop: PO1 Elements: 2

To specify pertinent dates and times.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
DTM01	374	**Date/Time Qualifier**	M	ID	3/3
		Code Name			
		002 Delivery Requested			
DTM02	373	**Date**	C	DT	8/8

Syntax:

1. DTM02 R020305 -- At least one of DTM02, DTM03 or DTM05 is required.
2. DTM04 C0403 -- If DTM04 is present, then DTM03 is required
3. DTM05 P0506 -- If either DTM05 or DTM06 are present, then the others are required.

N1 Name

Pos: 340 Max: 1
Detail - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	98	**Entity Identifier Code**	M	ID	2/3
		Code Name			
		ST Ship To			
N102	93	**Name**	C	AN	1/60

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 350 Max: 2
Detail - Optional
Loop: N1 Elements: 1

To specify additional names or those longer than 35 characters in length.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	93	Name	M	AN	1/60

N3 Address Information

Pos: 360 Max: 2
Detail - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 370 Max: 1
Detail - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

REF Reference Identification

Pos: 380 Max: 12
Detail - Optional
Loop: N1 Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		RQ Purchase Requisition Number			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

SE Transaction Set Trailer

Pos: 020 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	**96**	**Number of Included Segments**	**M**	**N0**	**1/10**
SE02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Comments:

1. SE is the last segment of each transaction set.

843 Response to Request for Quotation

Functional Group=RR

This Draft Standard for Trial Use contains the format and establishes the data contents of the Response to Request for Quotation Transaction Set (843) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to provide potential buyers with price, delivery schedule, and other terms from potential sellers of goods and services, in response to a request for such information.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	**M**	**1**		
020	BQR	Beginning Segment for Response to Request for Quotation	**M**	**1**		
050	REF	Reference Identification	**O**	**>1**		
060	PER	Administrative Communications Contact	**O**	**3**		
080	FOB	F.O.B. Related Instructions	**O**	**>1**		
130	ITD	Terms of Sale/Deferred Terms of Sale	**O**	**5**		
LOOP ID - N9					**1000**	
290	N9	Reference Identification	**O**	**1**		
LOOP ID - N1					**10000**	
310	N1	Name	**O**	**1**		
330	N3	Address Information	**O**	**2**		
340	N4	Geographic Location	**O**	**1**		
350	REF	Reference Identification	**O**	**12**		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - PO1					**100000**	
010	PO1	Baseline Item Data	**O**	**1**		
100	REF	Reference Identification	**O**	**>1**		

200	DTM	Date/Time Reference	O	10		
LOOP ID - N1					>1	
420	N1	Name	O	1		
440	N3	Address Information	O	2		
450	N4	Geographic Location	O	1		

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
020	SE	Transaction Set Trailer	M	1		

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	143	**Transaction Set Identifier Code**	M	ID	3/3
		Code Name 843 Response to Request for Quotation			
ST02	329	**Transaction Set Control Number**	M	AN	4/9

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BQR Beginning Segment for Response to Request for Quotation

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 7

To indicate the beginning of a Response to Request for Quote Transaction Set and transmit identifying numbers and dates.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BQR01	**353**	**Transaction Set Purpose Code**	**M**	**ID**	**2/2**
		Code Name			
		00 Original			
BQR02	**586**	**Request for Quote Reference Number**	**M**	**AN**	**1/45**
BQR03	**373**	**Date**	**M**	**DT**	**8/8**
BQR04	**374**	**Date/Time Qualifier**	**C**	**ID**	**3/3**
		Code Name			
		002 Delivery Requested			
BQR05	**373**	**Date**	**C**	**DT**	**8/8**
BQR06	**379**	**Bid Type Response Code**	**O**	**ID**	**2/2**
		Code Name			
		BF Best and Final			
		BI Bid Without Exception			
		DQ Decline to Quote			
		RT Request for Time Extension			
BQR08	**327**	**Change Order Sequence Number**	**O**	**AN**	**1/8**

Syntax:

1. BQR04 P0405 -- If either BQR04 or BQR05 are present, then the others are required.

Semantics:

1. BQR03 is the date to be used for reference purposes in an RFQ and a response to RFQ.

REF Reference Identification

Pos: 050 Max: >1
Heading - Optional
Loop: N/A Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	128	**Reference Identification Qualifier**	M	ID	2/3
		Code Name			
		65 Total Order Cycle Number			
		DS Defense Priorities Allocation System (DPAS) Priority Rating			
		RQ Purchase Requisition Number			
REF02	127	**Reference Identification**	C	AN	1/30

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

PER Administrative Communications Contact

Pos: 060 Max: 3
Heading - Optional
Loop: N/A Elements: 8

To identify a person or office to whom administrative communications should be directed.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PER01	366	**Contact Function Code**	M	ID	2/2
		Code Name			
		IC Information Contact			
PER02	93	**Name**	O	AN	1/60
PER03	365	**Communication Number Qualifier**	C	ID	2/2
		Code Name			
		TE Telephone			
PER04	364	**Communication Number**	C	AN	1/80
PER05	365	**Communication Number Qualifier**	C	ID	2/2
		Code Name			
		FX Facsimile			

PER06	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER07	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		EM Electronic Mail			
PER08	**364**	**Communication Number**	**C**	**AN**	**1/80**

Syntax:

1. PER03 P0304 -- If either PER03 or PER04 are present, then the others are required.
2. PER05 P0506 -- If either PER05 or PER06 are present, then the others are required.
3. PER07 P0708 -- If either PER07 or PER08 are present, then the others are required.

FOB F.O.B. Related Instructions

Pos: 080 Max: >1
Heading - Optional
Loop: N/A Elements: 1

To specify transportation instructions relating to shipment.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
FOB01	**146**	**Shipment Method of Payment**	**M**	**ID**	**2/2**
		Code Name			
		BP Paid by Buyer			
		CC Collect			
		CD Collect on Delivery			
		CF Collect, Freight Credited Back to Customer			
		PP Prepaid (by Seller)			
		PS Paid by Seller			

Syntax:

1. FOB03 C0302 -- If FOB03 is present, then FOB02 is required
2. FOB04 C0405 -- If FOB04 is present, then FOB05 is required
3. FOB07 C0706 -- If FOB07 is present, then FOB06 is required
4. FOB08 C0809 -- If FOB08 is present, then FOB09 is required

Semantics:

1. FOB01 indicates which party will pay the carrier.
2. FOB02 is the code specifying transportation responsibility location.

3. FOB06 is the code specifying the title passage location.
4. FOB08 is the code specifying the point at which the risk of loss transfers. This may be different than the location specified in FOB02/FOB03 and FOB06/FOB07.

ITD Terms of Sale/Deferred Terms of Sale

Pos: 130 Max: 5
Heading - Optional
Loop: N/A Elements: 4

To specify terms of sale.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ITD01	**336**	**Terms Type Code**	**O**	**ID**	**2/2**
		Code Name			
		01 Basic			
ITD03	**338**	**Terms Discount Percent**	**O**	**R**	**1/6**
ITD05	**351**	**Terms Discount Days Due**	**C**	**N0**	**1/3**
ITD07	**386**	**Terms Net Days**	**O**	**N0**	**1/3**

Syntax:

1. ITD03 L03040513 -- If ITD03 is present, then at least one of ITD04, ITD05 or ITD13 is required.
2. ITD08 L08040513 -- If ITD08 is present, then at least one of ITD04, ITD05 or ITD13 is required.
3. ITD09 L091011 -- If ITD09 is present, then at least one of ITD10 or ITD11 is required.

Semantics:

1. ITD15 is the percentage applied to a base amount used to determine a late payment charge.

Comments:

1. If the code in ITD01 is "04", then ITD07 or ITD09 is required and either ITD10 or ITD11 is required; if the code in ITD01 is "05", then ITD06 or ITD07 is required.

N9 Reference Identification

Pos: 290 Max: 1
Heading - Optional
Loop: N9 Elements: 3

To transmit identifying information as specified by the Reference Identification Qualifier.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N901	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		CT Contract Number			
N902	**127**	**Reference Identification**	**C**	**AN**	**1/30**
N903	**369**	**Free-form Description**	**C**	**AN**	**1/45**

Syntax:

1. N902 R0203 -- At least one of N902 or N903 is required.
2. N906 C0605 -- If N906 is present, then N905 is required

Semantics:

1. N906 reflects the time zone which the time reflects.
2. N907 contains data relating to the value cited in N902.

N1 Name

Pos: 310 Max: 1
Heading - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	**98**	**Entity Identifier Code**	**M**	**ID**	**2/3**
		Code Name			
		SU Supplier/Manufacturer			
N102	**93**	**Name**	**C**	**AN**	**1/60**

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N3 Address Information

Pos: 330 Max: 2
Heading - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 340 Max: 1
Heading - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

REF Reference Identification

Pos: 350 Max: 12
Heading - Optional
Loop: N1 Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	128	**Reference Identification Qualifier**	M	ID	2/3
		Code Name			
		RQ Purchase Requisition Number			
		W7 Commercial and Government Entity (CAGE) Code			
REF02	127	**Reference Identification**	C	AN	1/30

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

PO1 Baseline Item Data

Pos: 010 Max: 1
Detail - Optional
Loop: PO1 Elements: 10

To specify basic and most frequently used line item data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PO101	350	**Assigned Identification**	O	AN	1/20
PO102	330	**Quantity Ordered**	C	R	1/15

PO103	**355**	**Unit or Basis for Measurement Code**	**O**	**ID**	**2/2**
		Code Name			
		BR Barrel			
		CT Carton			
		DZ Dozen			
		EA Each			
PO104	**212**	**Unit Price**	**C**	**R**	**1/17**
PO106	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		BP Buyer’s Part Number			
PO107	**234**	**Product/Service ID**	**C**	**AN**	**1/48**
PO108	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		VP Vendor’s (Seller’s) Part Number			
PO109	**234**	**Product/Service ID**	**C**	**AN**	**1/48**
PO110	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		MG Manufacturer’s Part Number			
PO111	**234**	**Product/Service ID**	**C**	**AN**	**1/48**

Syntax:

1. PO103 C0302 -- If PO103 is present, then PO102 is required
2. PO105 C0504 -- If PO105 is present, then PO104 is required
3. PO106 P0607 -- If either PO106 or PO107 are present, then the others are required.
4. PO108 P0809 -- If either PO108 or PO109 are present, then the others are required.
5. PO110 P1011 -- If either PO110 or PO111 are present, then the others are required.
6. PO112 P1213 -- If either PO112 or PO113 are present, then the others are required.
7. PO114 P1415 -- If either PO114 or PO115 are present, then the others are required.
8. PO116 P1617 -- If either PO116 or PO117 are present, then the others are required.
9. PO118 P1819 -- If either PO118 or PO119 are present, then the others are required.
10. PO120 P2021 -- If either PO120 or PO121 are present, then the others are required.
11. PO122 P2223 -- If either PO122 or PO123 are present, then the others are required.
12. PO124 P2425 -- If either PO124 or PO125 are present, then the others are required.

Comments:

1. See the Data Element Dictionary for a complete list of IDs.
2. PO101 is the line item identification.

3. PO106 through PO125 provide for ten different product/service IDs per each item. For example: Case, Color, Drawing No., U.P.C. No., ISBN No., Model No., or SKU.

REF Reference Identification

Pos: 100 Max: >1
Detail - Optional
Loop: PO1 Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		RQ Purchase Requisition Number			
		W7 Commercial and Government Entity (CAGE) Code			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

DTM Date/Time Reference

Pos: 200 Max: 10
Detail - Optional
Loop: PO1 Elements: 2

To specify pertinent dates and times.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
DTM01	**374**	**Date/Time Qualifier**	**M**	**ID**	**3/3**
		Code Name			
		002 Delivery Requested			
DTM02	**373**	**Date**	**C**	**DT**	**8/8**

Syntax:

1. DTM02 R020305 -- At least one of DTM02, DTM03 or DTM05 is required.

2. DTM04 C0403 -- If DTM04 is present, then DTM03 is required
3. DTM05 P0506 -- If either DTM05 or DTM06 are present, then the others are required.

N1 Name

Pos: 420 Max: 1
Detail - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	98	**Entity Identifier Code**	M	ID	2/3
		Code Name			
		ST Ship To			
N102	93	**Name**	C	AN	1/60

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N3 Address Information

Pos: 440 Max: 2
Detail - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	**Address Information**	M	AN	1/55
N302	166	**Address Information**	O	AN	1/55

N4 Geographic Location

Pos: 450 Max: 1
Detail - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

SE Transaction Set Trailer

Pos: 020 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	96	Number of Included Segments	M	N0	1/10
SE02	329	Transaction Set Control Number	M	AN	4/9

Comments:

1. SE is the last segment of each transaction set.

850 Purchase Order

Functional Group=PO

This Draft Standard for Trial Use contains the format and establishes the data contents of the Purchase Order Transaction Set (850) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to provide for customary and established business and industry practice relative to the placement of purchase orders for goods and services. This transaction set should not be used to convey purchase order changes or purchase order acknowledgment information.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	**M**	**1**		
020	BEG	Beginning Segment for Purchase Order	**M**	**1**		
080	FOB	F.O.B. Related Instructions	**O**	**>1**		
110	CSH	Sales Requirements	**O**	**5**		
130	ITD	Terms of Sale/Deferred Terms of Sale	**O**	**>1**		
240	TD5	Carrier Details (Routing Sequence/Transit Time)	**O**	**12**		
LOOP ID - N1					**200**	
310	N1	Name	**O**	**1**		
320	N2	Additional Name Information	**O**	**2**		
330	N3	Address Information	**O**	**2**		
340	N4	Geographic Location	**O**	**>1**		
360	PER	Administrative Communications Contact	**O**	**>1**		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - PO1					**100000**	
010	PO1	Baseline Item Data	**M**	**1**		**N2/010**
LOOP ID - PID					**1000**	
050	PID	Product/Item Description	**O**	**1**		

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - AMT					**>1**	
600	AMT	Monetary Amount	**O**	**1**		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - PO1					**100000**	
010	PO1	Baseline Item Data	**M**	**1**		**N2/010**
LOOP ID - PID					**1000**	
050	PID	Product/Item Description	**O**	**1**		
LOOP ID - AMT					**>1**	
600	AMT	Monetary Amount	**O**	**1**		

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - CTT					**1**	
010	CTT	Transaction Totals	**O**	**1**		**N3/010**
030	SE	Transaction Set Trailer	**M**	**1**		

Notes:

2/010 - PO102 is required.

3/010 - The number of line items (CTT01) is the accumulation of the number of PO1 segments. If used, hash total (CTT02) is the sum of the value of quantities ordered (PO102) for each PO1 segment.

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	**143**	**Transaction Set Identifier Code**	**M**	**ID**	**3/3**
		Code Name			
		850 Purchase Order			
ST02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BEG Beginning Segment for Purchase Order

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 6

To indicate the beginning of the Purchase Order Transaction Set and transmit identifying numbers and dates.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BEG01	**353**	**Transaction Set Purpose Code**	**M**	**ID**	**2/2**
		Code Name			
		00 Original			
BEG02	**92**	**Purchase Order Type Code**	**M**	**ID**	**2/2**
		Code Name			
		SA Stand-alone Order			
BEG03	**324**	**Purchase Order Number**	**M**	**AN**	**1/22**
BEG05	**373**	**Date**	**M**	**DT**	**8/8**
BEG06	**367**	**Contract Number**	**O**	**AN**	**1/30**
BEG07	**587**	**Acknowledgment Type**	**O**	**ID**	**2/2**
		Code Name			
		AC Acknowledge - With Detail and Change			

Semantics:

1. BEG05 is the date assigned by the purchaser to purchase order.

FOB F.O.B. Related Instructions

Pos: 080 Max: >1
Heading - Optional
Loop: N/A Elements: 5

To specify transportation instructions relating to shipment.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
FOB01	**146**	**Shipment Method of Payment**	**M**	**ID**	**2/2**
		Code Name			
		CC Collect			
		NC Service Freight,No Charge			
		PC Prepaid but Charged to Customer			
		PP Prepaid (by Seller)			
FOB02	**309**	**Location Qualifier**	**C**	**ID**	**1/2**
		Code Name			
		OR Origin (Shipping Point)			
		WH Warehouse			
		ZZ Mutually Defined			
FOB03	**352**	**Description**	**O**	**AN**	**1/80**
FOB06	**309**	**Location Qualifier**	**C**	**ID**	**1/2**
		Code Name			
		DE Destination (Shipping)			
		FA Factory			
		WH Warehouse			
		ZZ Mutually Defined			
FOB07	**352**	**Description**	**O**	**AN**	**1/80**

Syntax:

1. FOB03 C0302 -- If FOB03 is present, then FOB02 is required
2. FOB04 C0405 -- If FOB04 is present, then FOB05 is required
3. FOB07 C0706 -- If FOB07 is present, then FOB06 is required
4. FOB08 C0809 -- If FOB08 is present, then FOB09 is required

Semantics:

1. FOB01 indicates which party will pay the carrier.
2. FOB02 is the code specifying transportation responsibility location.
3. FOB06 is the code specifying the title passage location.

4. FOB08 is the code specifying the point at which the risk of loss transfers. This may be different than the location specified in FOB02/FOB03 and FOB06/FOB07.

CSH Sales Requirements

Pos: 110 Max: 5
Heading - Optional
Loop: N/A Elements: 1

To specify general conditions or requirements of the sale.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
CSH01	**563**	**Sales Requirement Code**	**O**	**ID**	**1/2**

Code	Name
N	No Back Order
Y	Back Order if Out of Stock
P2	Ship As Soon As Possible
SC	Ship Complete

Syntax:

1. CSH02 C0203 -- If CSH02 is present, then CSH03 is required
2. CSH06 P0607 -- If either CSH06 or CSH07 are present, then the others are required.
3. CSH09 P0910 -- If either CSH09 or CSH10 are present, then the others are required.

Semantics:

1. CSH03 is the maximum monetary amount value which the order must not exceed.
2. CSH04 is the account number to which the purchase amount is charged.
3. CSH05 is the date specified by the sender to be shown on the invoice.
4. CSH06 identifies the source of the code value in CSH07.

ITD Terms of Sale/Deferred Terms of Sale

Pos: 130 Max: >1
Heading - Optional
Loop: N/A Elements: 6

To specify terms of sale.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ITD01	**336**	**Terms Type Code**	**O**	**ID**	**2/2**
		Code Name			
		01 Basic			
ITD02	**333**	**Terms Basis Date Code**	**O**	**ID**	**1/2**
		Code Name			
		1 Ship Date			
		3 Invoice Date			
ITD03	**338**	**Terms Discount Percent**	**O**	**R**	**1/6**
ITD05	**351**	**Terms Discount Days Due**	**C**	**N0**	**1/3**
ITD07	**386**	**Terms Net Days**	**O**	**N0**	**1/3**
ITD12	**352**	**Description**	**O**	**AN**	**1/80**

Syntax:

1. ITD03 L03040513 -- If ITD03 is present, then at least one of ITD04, ITD05 or ITD13 is required.
2. ITD08 L08040513 -- If ITD08 is present, then at least one of ITD04, ITD05 or ITD13 is required.
3. ITD09 L091011 -- If ITD09 is present, then at least one of ITD10 or ITD11 is required.

Semantics:

1. ITD15 is the percentage applied to a base amount used to determine a late payment charge.

Comments:

1. If the code in ITD01 is "04", then ITD07 or ITD09 is required and either ITD10 or ITD11 is required; if the code in ITD01 is "05", then ITD06 or ITD07 is required.

TD5 Carrier Details (Routing Sequence/Transit Time)

Pos: 240 Max: 12
Heading - Optional
Loop: N/A Elements: 2

To specify the carrier and sequence of routing and provide transit time information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
TD501	**133**	**Routing Sequence Code**	**O**	**ID**	**1/2**
		Code Name			
		B Origin/Delivery Carrier (Any Mode)			
TD505	**387**	**Routing**	**C**	**AN**	**1/35**

Syntax:

1. TD502 R0204050612 -- At least one of TD502, TD504, TD505, TD506 or TD512 is required.
2. TD502 C0203 -- If TD502 is present, then TD503 is required
3. TD507 C0708 -- If TD507 is present, then TD508 is required
4. TD510 C1011 -- If TD510 is present, then TD511 is required
5. TD513 C1312 -- If TD513 is present, then TD512 is required
6. TD514 C1413 -- If TD514 is present, then TD513 is required
7. TD515 C1512 -- If TD515 is present, then TD512 is required

Semantics:

1. TD515 is the country where the service is to be performed.

Comments:

1. When specifying a routing sequence to be used for the shipment movement in lieu of specifying each carrier within the movement, use TD502 to identify the party responsible for defining the routing sequence, and use TD503 to identify the actual routing sequence, specified by the party identified in TD502.

N1 Name

Pos: 310 Max: 1
Heading - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	98	**Entity Identifier Code**	M	ID	2/3
		Code Name BT Bill-to-Party ST Ship To			
N102	93	**Name**	C	AN	1/60

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 320 Max: 2
Heading - Optional
Loop: N1 Elements: 1

To specify additional names or those longer than 35 characters in length.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	93	**Name**	M	AN	1/60

N3 Address Information

Pos: 330 Max: 2
Heading - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 340 Max: >1
Heading - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

PER Administrative Communications Contact

Pos: 360 Max: >1
Heading - Optional
Loop: N1 Elements: 8

To identify a person or office to whom administrative communications should be directed.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PER01	**366**	**Contact Function Code**	**M**	**ID**	**2/2**
		Code Name			
		BI Bill Inquiry Contact			
		CN General Contact			
		DC Delivery Contact			
		IC Information Contact			
		SD Shipping Department			
		ZZ Mutually Defined			
PER02	**93**	**Name**	**O**	**AN**	**1/60**
PER03	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		TE Telephone			
PER04	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER05	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		FX Facsimile			
PER06	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER07	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		EM Electronic Mail			
PER08	**364**	**Communication Number**	**C**	**AN**	**1/80**

Syntax:

1. PER03 P0304 -- If either PER03 or PER04 are present, then the others are required.
2. PER05 P0506 -- If either PER05 or PER06 are present, then the others are required.
3. PER07 P0708 -- If either PER07 or PER08 are present, then the others are required.

PO1 Baseline Item Data

Pos: 010 Max: 1
Detail - Mandatory
Loop: PO1 Elements: 9

To specify basic and most frequently used line item data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PO101	**350**	**Assigned Identification**	**O**	**AN**	**1/20**
PO102	**330**	**Quantity Ordered**	**C**	**R**	**1/15**

PO103	**355**	**Unit or Basis for Measurement Code**	**O**	**ID**	**2/2**
		Code Name			
		BR Barrel			
		CT Carton			
		DZ Dozen			
		EA Each			
PO104	**212**	**Unit Price**	**C**	**R**	**1/17**
PO105	**639**	**Basis of Unit Price Code**	**O**	**ID**	**2/2**
		Code Name			
		ES Estimated			
		PE Price per Each			
PO106	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		BP Buyer's Part Number			
PO107	**234**	**Product/Service ID**	**C**	**AN**	**1/48**
PO108	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		VP Vendor's (Seller's) Part Number			
PO109	**234**	**Product/Service ID**	**C**	**AN**	**1/48**

Syntax:

1. PO103 C0302 -- If PO103 is present, then PO102 is required
2. PO105 C0504 -- If PO105 is present, then PO104 is required
3. PO106 P0607 -- If either PO106 or PO107 are present, then the others are required.
4. PO108 P0809 -- If either PO108 or PO109 are present, then the others are required.
5. PO110 P1011 -- If either PO110 or PO111 are present, then the others are required.
6. PO112 P1213 -- If either PO112 or PO113 are present, then the others are required.
7. PO114 P1415 -- If either PO114 or PO115 are present, then the others are required.
8. PO116 P1617 -- If either PO116 or PO117 are present, then the others are required.
9. PO118 P1819 -- If either PO118 or PO119 are present, then the others are required.
10. PO120 P2021 -- If either PO120 or PO121 are present, then the others are required.
11. PO122 P2223 -- If either PO122 or PO123 are present, then the others are required.
12. PO124 P2425 -- If either PO124 or PO125 are present, then the others are required.

Comments:

1. See the Data Element Dictionary for a complete list of IDs.
2. PO101 is the line item identification.

3. PO106 through PO125 provide for ten different product/service IDs per each item. For example: Case, Color, Drawing No., U.P.C. No., ISBN No., Model No., or SKU.

PID Product/Item Description

Pos: 050 Max: 1
Detail - Optional
Loop: PID Elements: 2

To describe a product or process in coded or free-form format.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PID01	**349**	**Item Description Type**	**M**	**ID**	**1/1**
		Code Name F Free-form			
PID05	**352**	**Description**	**C**	**AN**	**1/80**

Syntax:

1. PID04 C0403 -- If PID04 is present, then PID03 is required
2. PID04 R0405 -- At least one of PID04 or PID05 is required.
3. PID07 C0703 -- If PID07 is present, then PID03 is required
4. PID08 C0804 -- If PID08 is present, then PID04 is required
5. PID09 C0905 -- If PID09 is present, then PID05 is required

Semantics:

1. Use PID03 to indicate the organization that publishes the code list being referred to.
2. PID04 should be used for industry-specific product description codes.
3. PID08 describes the physical characteristics of the product identified in PID04. A "Y" indicates that the specified attribute applies to this item; an "N" indicates it does not apply. Any other value is indeterminate.
4. PID09 is used to identify the language being used in PID05.

Comments:

1. If PID01 equals "F", then PID05 is used. If PID01 equals "S", then PID04 is used. If PID01 equals "X", then both PID04 and PID05 are used.
2. Use PID06 when necessary to refer to the product surface or layer being described in the segment.

3. PID07 specifies the individual code list of the agency specified in PID03.

AMT Monetary Amount

Pos: 600 Max: 1
Detail - Optional
Loop: AMT Elements: 2

To indicate the total monetary amount

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
AMT01	**522**	**Amount Qualifier Code**	**M**	**ID**	**1/3**
		Code Name			
		1 Line Item Total			
AMT02	**782**	**Monetary Amount**	**M**	**R**	**1/18**

CTT Transaction Totals

Pos: 010 Max: 1
Summary - Optional
Loop: CTT Elements: 2

To transmit a hash total for a specific element in the transaction set.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
CTT01	**354**	**Number of Line Items**	**M**	**N0**	**1/6**
CTT02	**347**	**Hash Total**	**O**	**R**	**1/10**

Syntax:

1. CTT03 P0304 -- If either CTT03 or CTT04 are present, then the others are required.
2. CTT05 P0506 -- If either CTT05 or CTT06 are present, then the others are required.

Comments:

1. This segment is intended to provide hash totals to validate transaction completeness and correctness.

SE Transaction Set Trailer

Pos: 030 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	**96**	**Number of Included Segments**	**M**	**N0**	**1/10**
SE02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Comments:

1. SE is the last segment of each transaction set.

855 Purchase Order Acknowledgment

Functional Group=PR

This Draft Standard for Trial Use contains the format and establishes the data contents of the Purchase Order Acknowledgment Transaction Set (855) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to provide for customary and established business and industry practice relative to a seller's acknowledgment of a buyer's purchase order. This transaction set can also be used as notification of a vendor generated order. This usage advises a buyer that a vendor has or will ship merchandise as prearranged in their partnership.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	**M**	**1**		
020	BAK	Beginning Segment for Purchase Order Acknowledgment	**M**	**1**		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - PO1					100000	
010	PO1	Baseline Item Data	O	1		N2/010
LOOP ID - PID					1000	
050	PID	Product/Item Description	O	1		
293	AMT	Monetary Amount	O	1		N2/293
LOOP ID - SCH					200	
310	SCH	Line Item Schedule	O	1		N2/310

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - CTT					1	
010	CTT	Transaction Totals	O	1		N3/010
030	SE	Transaction Set Trailer	M	1		

Notes:

2/010 - PO102 is required.

2/293 - If AMT is used in the detail area, then AMT01 will = 1 and AMT02 will indicate total line amount as calculated by the sender.

2/310 - The SCH segment is used to specify various quantities of items ordered that are to be scheduled. When this segment is used the unit of measurement code (SCH02) should always be identical to the unit of measurement code in the associated PO1 segment (PO103) and the sum of values of quantity (SCH01) should always equal the quantity ordered (PO102) in the PO1 segment.

3/010 - The number of line items (CTT01) is the accumulation of the number of PO1 segments. If used, hash total (CTT02) is the sum of the value of quantities ordered (PO102) for each PO1 segment.

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	143	**Transaction Set Identifier Code**	M	ID	3/3
		Code Name			
		855 Purchase Order Acknowledgment			
ST02	329	**Transaction Set Control Number**	M	AN	4/9

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BAK Beginning Segment for Purchase Order Acknowledgment

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 5

To indicate the beginning of the Purchase Order Acknowledgment Transaction Set and transmit identifying numbers and dates.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BAK01	353	**Transaction Set Purpose Code**	M	ID	2/2
		Code Name			
		00 Original			
BAK02	587	**Acknowledgment Type**	M	ID	2/2
		Code Name			
		AC Acknowledge - With Detail and Change			
		AT Accepted			
		RD Reject with Detail			
BAK03	324	**Purchase Order Number**	M	AN	1/22
BAK04	373	**Date**	M	DT	8/8
BAK07	367	**Contract Number**	O	AN	1/30

Semantics:

1. BAK04 is the date assigned by the purchaser to purchase order.

2. BAK08 is the seller's order number.
3. BAK09 is the date assigned by the sender to the acknowledgment.

PO1 Baseline Item Data

Pos: 010 Max: 1
Detail - Optional
Loop: PO1 Elements: 5

To specify basic and most frequently used line item data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PO102	**330**	**Quantity Ordered**	**C**	**R**	**1/15**
PO103	**355**	**Unit or Basis for Measurement Code**	**O**	**ID**	**2/2**
		Code Name			
		BR Barrel			
		CT Carton			
		DZ Dozen			
		EA Each			
PO104	**212**	**Unit Price**	**C**	**R**	**1/17**
PO108	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		VP Vendor’s (Seller’s) Part Number			
PO109	**234**	**Product/Service ID**	**C**	**AN**	**1/48**

Syntax:

1. PO103 C0302 -- If PO103 is present, then PO102 is required
2. PO105 C0504 -- If PO105 is present, then PO104 is required
3. PO106 P0607 -- If either PO106 or PO107 are present, then the others are required.
4. PO108 P0809 -- If either PO108 or PO109 are present, then the others are required.
5. PO110 P1011 -- If either PO110 or PO111 are present, then the others are required.
6. PO112 P1213 -- If either PO112 or PO113 are present, then the others are required.
7. PO114 P1415 -- If either PO114 or PO115 are present, then the others are required.
8. PO116 P1617 -- If either PO116 or PO117 are present, then the others are required.
9. PO118 P1819 -- If either PO118 or PO119 are present, then the others are required.
10. PO120 P2021 -- If either PO120 or PO121 are present, then the others are required.
11. PO122 P2223 -- If either PO122 or PO123 are present, then the others are required.
12. PO124 P2425 -- If either PO124 or PO125 are present, then the others are required.

Comments:

1. See the Data Element Dictionary for a complete list of IDs.
2. PO101 is the line item identification.
3. PO106 through PO125 provide for ten different product/service IDs per each item. For example: Case, Color, Drawing No., U.P.C. No., ISBN No., Model No., or SKU.

PID Product/Item Description

Pos: 050 Max: 1
Detail - Optional
Loop: PID Elements: 2

To describe a product or process in coded or free-form format.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PID01	**349**	**Item Description Type**	**M**	**ID**	**1/1**
		Code Name			
		F Free-form			
PID05	**352**	**Description**	**C**	**AN**	**1/80**

Syntax:

1. PID04 C0403 -- If PID04 is present, then PID03 is required
2. PID04 R0405 -- At least one of PID04 or PID05 is required.
3. PID07 C0703 -- If PID07 is present, then PID03 is required
4. PID08 C0804 -- If PID08 is present, then PID04 is required
5. PID09 C0905 -- If PID09 is present, then PID05 is required

Semantics:

1. Use PID03 to indicate the organization that publishes the code list being referred to.
2. PID04 should be used for industry-specific product description codes.
3. PID08 describes the physical characteristics of the product identified in PID04. A "Y" indicates that the specified attribute applies to this item; an "N" indicates it does not apply. Any other value is indeterminate.
4. PID09 is used to identify the language being used in PID05.

Comments:

1. If PID01 equals "F", then PID05 is used. If PID01 equals "S", then PID04 is used. If PID01 equals "X", then both PID04 and PID05 are used.
2. Use PID06 when necessary to refer to the product surface or layer being described in the segment.
3. PID07 specifies the individual code list of the agency specified in PID03.

AMT Monetary Amount

Pos: 293 Max: 1
Detail - Optional
Loop: PO1 Elements: 2

To indicate the total monetary amount.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
AMT01	**522**	**Amount Qualifier Code**	**M**	**ID**	**1/3**
		Code Name			
		1 Line Item Total			
AMT02	**782**	**Monetary Amount**	**M**	**R**	**1/18**

SCH Line Item Schedule

Pos: 310 Max: 1
Detail - Optional
Loop: SCH Elements: 4

To specify the data for scheduling a specific line-item.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SCH01	**380**	**Quantity**	**M**	**R**	**1/15**
SCH02	**355**	**Unit or Basis for Measurement Code**	**M**	**ID**	**2/2**
		Code Name			
		BR Barrel			
		CT Carton			
		DZ Dozen			
		EA Each			
SCH05	**374**	**Date/Time Qualifier**	**M**	**ID**	**3/3**
		Code Name			
		007 Effective			
SCH06	**373**	**Date**	**M**	**DT**	**8/8**

Syntax:

1. SCH03 C0304 -- If SCH03 is present, then SCH04 is required
2. SCH08 L080910 -- If SCH08 is present, then at least one of SCH09 or SCH10 is required.
3. SCH09 C0908 -- If SCH09 is present, then SCH08 is required
4. SCH10 C1008 -- If SCH10 is present, then SCH08 is required

Semantics:

1. SCH12 is the schedule identification.

Comments:

1. SCH05 specifies the interpretation to be used for SCH06 and SCH07.

CTT Transaction Totals

Pos: 010 Max: 1
Summary - Optional
Loop: CTT Elements: 1

To transmit a hash total for a specific element in the transaction set.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
CTT01	**354**	**Number of Line Items**	**M**	**N0**	**1/6**

Syntax:

1. CTT03 P0304 -- If either CTT03 or CTT04 are present, then the others are required.
2. CTT05 P0506 -- If either CTT05 or CTT06 are present, then the others are required.

Comments:

1. This segment is intended to provide hash totals to validate transaction completeness and correctness.

SE Transaction Set Trailer

Pos: 030 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	**96**	**Number of Included Segments**	**M**	**N0**	**1/10**
SE02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Comments:

1. SE is the last segment of each transaction set.

856 Ship Notice/Manifest

Functional Group=SH

This Draft Standard for Trial Use contains the format and establishes the data contents of the Ship Notice/Manifest Transaction Set (856) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to list the contents of a shipment of goods as well as additional information relating to the shipment, such as order information, product description, physical characteristics, type of packaging, marking, carrier information, and configuration of goods within the transportation equipment. The transaction set enables the sender to describe the contents and configuration of a shipment in various levels of detail and provides an ordered flexibility to convey information.The sender of this transaction is the organization responsible for detailing and communicating the contents of a shipment, or shipments, to one or more receivers of the transaction set. The receiver of this transaction set can be any organization having an interest in the contents of a shipment or information about the contents of a shipment.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	**M**	**1**		
020	BSN	Beginning Segment for Ship Notice	**M**	**1**		
040	DTM	Date/Time Reference	**O**	**10**		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - HL					**200000**	
010	HL	Hierarchical Level	**M**	**1**		**C2/010**
020	LIN	Item Identification	**O**	**1**		
030	SN1	Item Detail (Shipment)	**O**	**1**		
050	PRF	Purchase Order Reference	**O**	**1**		
120	TD5	Carrier Details (Routing Sequence/Transit Time)	**O**	**12**		
150	REF	Reference Identification	**O**	**>1**		
LOOP ID - N1					**200**	
220	N1	Name	**O**	**1**		
230	N2	Additional Name Information	**O**	**2**		
240	N3	Address Information	**O**	**2**		
250	N4	Geographic Location	**O**	**1**		
260	REF	Reference Identification	**O**	**12**		
270	PER	Administrative Communications Contact	**O**	**3**		

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
020	SE	Transaction Set Trailer	**M**	**1**		

Comments:

2/010 - The HL segment is the only mandatory segment within the HL loop, and by itself, the HL segment has no meaning.

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	143	**Transaction Set Identifier Code**	M	ID	3/3
		Code Name			
		856 Ship Notice/Manifest			
ST02	329	**Transaction Set Control Number**	M	AN	4/9

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BSN Beginning Segment for Ship Notice

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 4

To transmit identifying numbers, dates, and other basic data relating to the transaction set.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BSN01	353	**Transaction Set Purpose Code**	M	ID	2/2
		Code Name			
		00 Original			
BSN02	396	**Shipment Identification**	M	AN	2/30
BSN03	373	**Date**	M	DT	8/8
BSN04	337	**Time**	M	TM	4/8

Syntax:

1. BSN07 C0706 -- If BSN07 is present, then BSN06 is required

Semantics:

1. BSN03 is the date the shipment transaction set is created.
2. BSN04 is the time the shipment transaction set is created.
3. BSN06 is limited to shipment related codes.

Comments:

1. BSN06 and BSN07 differentiate the functionality of use for the transaction set.

DTM Date/Time Reference

Pos: 040 Max: 10
Heading - Optional
Loop: N/A Elements: 2

To specify pertinent dates and times.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
DTM01	**374**	**Date/Time Qualifier**	**M**	**ID**	**3/3**
		Code Name			
		011 Shipped			
DTM02	**373**	**Date**	**C**	**DT**	**8/8**

Syntax:

1. DTM02 R020305 -- At least one of DTM02, DTM03 or DTM05 is required.
2. DTM04 C0403 -- If DTM04 is present, then DTM03 is required
3. DTM05 P0506 -- If either DTM05 or DTM06 are present, then the others are required.

HL Hierarchical Level

Pos: 010 Max: 1
Detail - Mandatory
Loop: HL Elements: 3

To identify dependencies among and the content of hierarchically related groups of data segments.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
HL01	**628**	**Hierarchical ID Number**	**M**	**AN**	**1/12**
HL02	**734**	**Hierarchical Parent ID Number**	**O**	**AN**	**1/12**
HL03	**735**	**Hierarchical Level Code**	**M**	**ID**	**1/2**
		Code Name			
		I Item			
		O Order			
		S Shipment			

Comments:

1. The HL segment is used to identify levels of detail information using a hierarchical structure, such as relating line-item data to shipment data, and packaging data to line-item data.
2. The HL segment defines a top-down/left-right ordered structure.
3. HL01 shall contain a unique alphanumeric number for each occurrence of the HL segment in the transaction set. For example, HL01 could be used to indicate the number of occurrences of the HL segment, in which case the value of HL01 would be "1" for the initial HL segment and would be incremented by one in each subsequent HL segment within the transaction.
4. HL02 identifies the hierarchical ID number of the HL segment to which the current HL segment is subordinate.
5. HL03 indicates the context of the series of segments following the current HL segment up to the next occurrence of an HL segment in the transaction. For example, HL03 is used to indicate that subsequent segments in the HL loop form a logical grouping of data referring to shipment, order, or item-level information.
6. HL04 indicates whether or not there are subordinate (or child) HL segments related to the current HL segment.

LIN Item Identification

Pos: 020 Max: 1
Detail - Optional
Loop: HL Elements: 5

To specify basic item identification data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
LIN01	**350**	**Assigned Identification**	**O**	**AN**	**1/20**
LIN02	**235**	**Product/Service ID Qualifier**	**M**	**ID**	**2/2**
		Code Name			
		BP Buyer's Part Number			
LIN03	**234**	**Product/Service ID**	**M**	**AN**	**1/48**
LIN04	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		VP Vendor's (Seller's) Part Number			
LIN05	**234**	**Product/Service ID**	**C**	**AN**	**1/48**

Syntax:

1. LIN04 P0405 -- If either LIN04 or LIN05 are present, then the others are required.
2. LIN06 P0607 -- If either LIN06 or LIN07 are present, then the others are required.
3. LIN08 P0809 -- If either LIN08 or LIN09 are present, then the others are required.
4. LIN10 P1011 -- If either LIN10 or LIN11 are present, then the others are required.
5. LIN12 P1213 -- If either LIN12 or LIN13 are present, then the others are required.
6. LIN14 P1415 -- If either LIN14 or LIN15 are present, then the others are required.
7. LIN16 P1617 -- If either LIN16 or LIN17 are present, then the others are required.
8. LIN18 P1819 -- If either LIN18 or LIN19 are present, then the others are required.
9. LIN20 P2021 -- If either LIN20 or LIN21 are present, then the others are required.
10. LIN22 P2223 -- If either LIN22 or LIN23 are present, then the others are required.
11. LIN24 P2425 -- If either LIN24 or LIN25 are present, then the others are required.
12. LIN26 P2627 -- If either LIN26 or LIN27 are present, then the others are required.
13. LIN28 P2829 -- If either LIN28 or LIN29 are present, then the others are required.
14. LIN30 P3031 -- If either LIN30 or LIN31 are present, then the others are required.

Semantics:

1. LIN01 is the line item identification

Comments:

1. See the Data Dictionary for a complete list of IDs.
2. LIN02 through LIN31 provide for fifteen different product/service IDs for each item. For example: Case, Color, Drawing No., U.P.C. No., ISBN No., Model No., or SKU.

SN1 Item Detail (Shipment)

Pos: 030 Max: 1
Detail - Optional
Loop: HL Elements: 2

To specify line-item detail relative to shipment.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SN102	**382**	**Number of Units Shipped**	**M**	**R**	**1/10**
SN103	**355**	**Unit or Basis for Measurement Code**	**M**	**ID**	**2/2**

Code Name

BR	Barrel
CT	Carton
DZ	Dozen
EA	Each

Syntax:

1. SN105 P0506 -- If either SN105 or SN106 are present, then the others are required.

Semantics:

1. SN101 is the ship notice line-item identification.

Comments:

1. SN103 defines the unit of measurement for both SN102 and SN104.

PRF Purchase Order Reference

Pos: 050 Max: 1
Detail - Optional
Loop: HL Elements: 1

To provide reference to a specific purchase order.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PRF01	**324**	**Purchase Order Number**	**M**	**AN**	**1/22**

Semantics:

1. PRF04 is the date assigned by the purchaser to purchase order.

TD5 Carrier Details (Routing Sequence/Transit Time)

Pos: 120 Max: 12
Detail - Optional
Loop: HL Elements: 4

To specify the carrier and sequence of routing and provide transit time information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
TD501	**133**	**Routing Sequence Code**	**O**	**ID**	**1/2**
		Code Name			
		B Origin/Delivery Carrier (Any Mode)			
TD502	**66**	**Identification Code Qualifier**	**C**	**ID**	**1/2**
		Code Name			
		2 Standard Carrier Alpha Code (SCAC)			
TD503	**67**	**Identification Code**	**C**	**AN**	**2/80**
TD504	**91**	**Transportation Method/Type Code**	**C**	**ID**	**1/2**
		Code Name			
		A Air			
		D Parcel Post			
		M Motor (Common Carrier)			
		R Rail			
		U Private Parcel Service			

Syntax:

1. TD502 R0204050612 -- At least one of TD502, TD504, TD505, TD506 or TD512 is required.
2. TD502 C0203 -- If TD502 is present, then TD503 is required
3. TD507 C0708 -- If TD507 is present, then TD508 is required
4. TD510 C1011 -- If TD510 is present, then TD511 is required
5. TD513 C1312 -- If TD513 is present, then TD512 is required
6. TD514 C1413 -- If TD514 is present, then TD513 is required
7. TD515 C1512 -- If TD515 is present, then TD512 is required

Semantics:

1. TD515 is the country where the service is to be performed.

Comments:

1. When specifying a routing sequence to be used for the shipment movement in lieu of specifying each carrier within the movement, use TD502 to identify the party responsible for defining the routing sequence, and use TD503 to identify the actual routing sequence, specified by the party identified in TD502.

REF Reference Identification

Pos: 150 Max: >1
Detail - Optional
Loop: HL Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	128	**Reference Identification Qualifier**	M	ID	2/3
		Code Name			
		BM Bill of Lading Number			
REF02	127	**Reference Identification**	C	AN	1/30

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

N1 Name

Pos: 220 Max: 1
Detail - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	98	**Entity Identifier Code**	M	ID	2/3
		Code Name			
		SF Ship From			
		ST Ship To			
N102	93	**Name**	C	AN	1/60

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 230 Max: 2
Detail - Optional
Loop: N1 Elements: 1

To specify additional names or those longer than 35 characters in length.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	93	Name	M	AN	1/60

N3 Address Information

Pos: 240 Max: 2
Detail - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 250 Max: 1
Detail - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30

N402	**156**	**State or Province Code**	**O**	**ID**	**2/2**
N403	**116**	**Postal Code**	**O**	**ID**	**3/15**
N404	**26**	**Country Code**	**O**	**ID**	**2/3**

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

REF Reference Identification

Pos: 260 Max: 12
Detail - Optional
Loop: N1 Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		DK Dock Number			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

PER Administrative Communications Contact

Pos: 270 Max: 3
Detail - Optional
Loop: N1 Elements: 8

To identify a person or office to whom administrative communications should be directed.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PER01	**366**	**Contact Function Code**	**M**	**ID**	**2/2**
		Code Name			
		IC Information Contact			
PER02	**93**	**Name**	**O**	**AN**	**1/60**
PER03	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		TE Telephone			
PER04	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER05	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		FX Facsimile			
PER06	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER07	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		EM Electronic Mail			
PER08	**364**	**Communication Number**	**C**	**AN**	**1/80**

Syntax:

1. PER03 P0304 -- If either PER03 or PER04 are present, then the others are required.
2. PER05 P0506 -- If either PER05 or PER06 are present, then the others are required.
3. PER07 P0708 -- If either PER07 or PER08 are present, then the others are required.

SE Transaction Set Trailer

Pos: 020 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	**96**	**Number of Included Segments**	**M**	**N0**	**1/10**
SE02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Comments:

1. SE is the last segment of each transaction set.

860 Purchase Order Change Request - Buyer Initiated

Functional Group=PC

This Draft Standard for Trial Use contains the format and establishes the data contents of the Purchase Order Change Request - Buyer Initiated Transaction Set (860) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used to provide the information required for the customary and established business and industry practice relative to a purchase order change. This transaction can be used: (1) by a buyer to request a change to a previously submitted purchase order or (2) by a buyer to confirm acceptance of a purchase order change initiated by the seller or by mutual agreement of the two parties.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	M	1		
020	BCH	Beginning Segment for Purchase Order Change	M	1		
LOOP ID - N1					200	
300	N1	Name	O	1		
310	N2	Additional Name Information	O	2		
320	N3	Address Information	O	2		
330	N4	Geographic Location	O	>1		
350	PER	Administrative Communications Contact	O	>1		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - POC					>1	

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	POC	Line Item Change	O	1		
LOOP ID - PID					1000	
050	PID	Product/Item Description	O	1		
LOOP ID - SAC					25	
130	SAC	Service, Promotion, Allowance, or Charge Information	O	1		
200	DTM	Date/Time Reference	O	10		
LOOP ID - N1					200	
340	N1	Name	O	1		
350	N2	Additional Name Information	O	2		
360	N3	Address Information	O	2		
370	N4	Geographic Location	O	1		
380	REF	Reference Identification	O	12		
390	PER	Administrative Communications Contact	O	3		
420	TD5	Carrier Details (Routing Sequence/Transit Time)	O	12		

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
030	SE	Transaction Set Trailer	M	1		

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	143	**Transaction Set Identifier Code**	M	ID	3/3
		Code Name			
		860 Purchase Order Change Request - Buyer Initiated			
ST02	329	**Transaction Set Control Number**	M	AN	4/9

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BCH Beginning Segment for Purchase Order Change

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 6

To indicate the beginning of the Purchase Order Change Transaction Set and transmit identifying numbers and dates.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BCH01	**353**	**Transaction Set Purpose Code**	**M**	**ID**	**2/2**
		Code Name			
		00 Original			
BCH02	**92**	**Purchase Order Type Code**	**M**	**ID**	**2/2**
		Code Name			
		RL Release or Delivery Order			
BCH03	**324**	**Purchase Order Number**	**M**	**AN**	**1/22**
BCH04	**328**	**Release Number**	**O**	**AN**	**1/30**
BCH05	**327**	**Change Order Sequence Number**	**O**	**AN**	**1/8**
BCH06	**373**	**Date**	**M**	**DT**	**8/8**

Semantics:

1. BCH06 is the date assigned by the purchaser to purchase order.
2. BCH09 is the seller's order number.
3. BCH10 is the date assigned by the sender to the acknowledgment.
4. BCH11 is the date of the purchase order change request.

N1 Name

Pos: 300 Max: 1
Heading - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary;

Ref	Id	Element Name	Req	Type	Min/Max
N101	98	**Entity Identifier Code** **Code Name** SU Supplier/Manufacturer	M	ID	2/3
N102	93	**Name**	C	AN	1/60

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 310 Max: 2
Heading - Optional
Loop: N1 Elements: 1

To specify additional names or those longer than 35 characters in length.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	93	**Name**	M	AN	1/60

N3 Address Information

Pos: 320 Max: 2
Heading - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	**Address Information**	M	AN	1/55
N302	166	**Address Information**	O	AN	1/55

N4 Geographic Location

Pos: 330 Max: >1
Heading - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

PER Administrative Communications Contact

Pos: 350 Max: >1
Heading - Optional
Loop: N1 Elements: 8

To identify a person or office to whom administrative communications should be directed.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PER01	366	Contact Function Code Code Name IC Information Contact	M	ID	2/2
PER02	93	Name	O	AN	1/60

PER03	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		TE Telephone			
PER04	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER05	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		FX Facsimile			
PER06	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER07	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		EM Electronic Mail			
PER08	**364**	**Communication Number**	**C**	**AN**	**1/80**

Syntax:

1. PER03 P0304 -- If either PER03 or PER04 are present, then the others are required.
2. PER05 P0506 -- If either PER05 or PER06 are present, then the others are required.
3. PER07 P0708 -- If either PER07 or PER08 are present, then the others are required.

POC Line Item Change

Pos: 010 Max: 1
Detail - Optional
Loop: POC Elements: 10

To specify changes to a line item.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
POC01	**350**	**Assigned Identification**	**O**	**AN**	**1/20**
POC02	**670**	**Change or Response Type Code**	**M**	**ID**	**2/2**
		Code Name			
		CA Changes To Line Items			
		CT Change of Dates			
		DI Delete Item(s)			
POC03	**330**	**Quantity Ordered**	**O**	**R**	**1/15**
POC05	**C001**	**Composite Unit of Measure**	**C**	**Comp**	
	355	**Unit or Basis for Measurement Code**	**M**	**ID**	**2/2**
		Code Name			
		01 Actual Pounds			
POC06	**212**	**Unit Price**	**C**	**R**	**1/17**

POC07	**639**	**Basis of Unit Price Code**	**O**	**ID**	**2/2**
		Code Name			
		AA Bill			
POC08	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		BP Buyer's Part Number			
POC09	**234**	**Product/Service ID**	**C**	**AN**	**1/48**
POC10	**235**	**Product/Service ID Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		VP Vendor's (Seller's) Part Number			
POC11	**234**	**Product/Service ID**	**C**	**AN**	**1/48**

Syntax:

1. POC03 C030405 -- If 03 is present, then POC04 and POC05 are required.
2. POC07 C0706 -- If POC07 is present, then POC06 is required
3. POC08 P0809 -- If either POC08 or POC09 are present, then the others are required.
4. POC10 P1011 -- If either POC10 or POC11 are present, then the others are required.
5. POC12 P1213 -- If either POC12 or POC13 are present, then the others are required.
6. POC14 P1415 -- If either POC14 or POC15 are present, then the others are required.
7. POC16 P1617 -- If either POC16 or POC17 are present, then the others are required.
8. POC18 P1819 -- If either POC18 or POC19 are present, then the others are required.
9. POC20 P2021 -- If either POC20 or POC21 are present, then the others are required.
10. POC22 P2223 -- If either POC22 or POC23 are present, then the others are required.
11. POC24 P2425 -- If either POC24 or POC25 are present, then the others are required.
12. POC26 P2627 -- If either POC26 or POC27 are present, then the others are required.

Semantics:

1. POC01 is the purchase order line item identification.

PID Product/Item Description

Pos: 050 Max: 1
Detail - Optional
Loop: PID Elements: 2

To describe a product or process in coded or free-form format.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PID01	**349**	**Item Description Type**	**M**	**ID**	**1/1**
		Code Name			
		F Free-form			
PID05	**352**	**Description**	**C**	**AN**	**1/80**

Syntax:

1. PID04 C0403 -- If PID04 is present, then PID03 is required
2. PID04 R0405 -- At least one of PID04 or PID05 is required.
3. PID07 C0703 -- If PID07 is present, then PID03 is required
4. PID08 C0804 -- If PID08 is present, then PID04 is required
5. PID09 C0905 -- If PID09 is present, then PID05 is required

Semantics:

1. Use PID03 to indicate the organization that publishes the code list being referred to.
2. PID04 should be used for industry-specific product description codes.
3. PID08 describes the physical characteristics of the product identified in PID04. A "Y" indicates that the specified attribute applies to this item; an "N" indicates it does not apply. Any other value is indeterminate.
4. PID09 is used to identify the language being used in PID05.

Comments:

1. If PID01 equals "F", then PID05 is used. If PID01 equals "S", then PID04 is used. If PID01 equals "X", then both PID04 and PID05 are used.
2. Use PID06 when necessary to refer to the product surface or layer being described in the segment.
3. PID07 specifies the individual code list of the agency specified in PID03.

SAC Service, Promotion, Allowance, or Charge Information

Pos: 130 Max: 1
Detail - Optional
Loop: SAC Elements: 3

To request or identify a service, promotion, allowance, or charge; to specify the amount or percentage for the service, promotion, allowance, or charge.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SAC01	248	**Allowance or Charge Indicator**	M	ID	1/1
		Code Name			
		C Charge			
SAC02	1300	**Service, Promotion, Allowance, or Charge Code**	C	ID	4/4
		Code Name			
		B240 COD Charges			
SAC05	610	**Amount**	O	N2	1/15

Syntax:

1. SAC02 R0203 -- At least one of SAC02 or SAC03 is required.
2. SAC03 P0304 -- If either SAC03 or SAC04 are present, then the others are required.
3. SAC06 P0607 -- If either SAC06 or SAC07 are present, then the others are required.
4. SAC09 P0910 -- If either SAC09 or SAC10 are present, then the others are required.
5. SAC11 C1110 -- If SAC11 is present, then SAC10 is required
6. SAC13 L130204 -- If SAC13 is present, then at least one of SAC02 or SAC04 is required.
7. SAC14 C1413 -- If SAC14 is present, then SAC13 is required
8. SAC16 C1615 -- If SAC16 is present, then SAC15 is required

Semantics:

1. If SAC01 is "A" or "C", then at least one of SAC05, SAC07, or SAC08 is required.
2. SAC05 is the total amount for the service, promotion, allowance, or charge.
3. If SAC05 is present with SAC07 or SAC08, then SAC05 takes precedence.
4. SAC08 is the allowance or charge rate per unit.

5. SAC10 and SAC11 is the quantity basis when the allowance or charge quantity is different from the purchase order or invoice quantity.
6. SAC10 and SAC11 used together indicate a quantity range, which could be a dollar amount, that is applicable to service, promotion, allowance, or charge.
7. SAC13 is used in conjunction with SAC02 or SAC04 to provide a specific reference number as identified by the code used.
8. SAC14 is used in conjunction with SAC13 to identify an option when there is more than one option of the promotion.
9. SAC16 is used to identify the language being used in SAC15.

Comments:

1. SAC04 may be used to uniquely identify the service, promotion, allowance, or charge. In addition, it may be used in conjunction to further the code in SAC02.
2. In some business applications, it is necessary to advise the trading partner of the actual dollar amount that a particular allowance, charge, or promotion was based on to reduce ambiguity. This amount is commonly referred to as "Dollar Basis Amount". It is represented in the SAC segment in SAC10 using the qualifier "DO" - Dollars in SAC09.

DTM Date/Time Reference

Pos: 200 Max: 10
Detail - Optional
Loop: POC Elements: 3

To specify pertinent dates and times.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
DTM01	**374**	**Date/Time Qualifier**	**M**	**ID**	**3/3**
		Code Name			
		010 Requested Ship			
DTM02	**373**	**Date**	**C**	**DT**	**8/8**
DTM03	**337**	**Time**	**C**	**TM**	**4/8**

Syntax:

1. DTM02 R020305 -- At least one of DTM02, DTM03 or DTM05 is required.
2. DTM04 C0403 -- If DTM04 is present, then DTM03 is required

3. DTM05 P0506 -- If either DTM05 or DTM06 are present, then the others are required.

N1 Name

Pos: 340 Max: 1
Detail - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	**98**	**Entity Identifier Code**	**M**	**ID**	**2/3**
		Code Name			
		ST Ship To			
N102	**93**	**Name**	**C**	**AN**	**1/60**

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 350 Max: 2
Detail - Optional
Loop: N1 Elements: 2

To specify additional names or those longer than 35 characters in length.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	**93**	**Name**	**M**	**AN**	**1/60**
N202	**93**	**Name**	**O**	**AN**	**1/60**

N3 Address Information

Pos: 360 Max: 2
Detail - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 370 Max: 1
Detail - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

REF Reference Identification

Pos: 380 Max: 12
Detail - Optional
Loop: N1 Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		DK Dock Number			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

PER Administrative Communications Contact

Pos: 390 Max: 3
Detail - Optional
Loop: N1 Elements: 8

To identify a person or office to whom administrative communications should be directed.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
PER01	**366**	**Contact Function Code**	**M**	**ID**	**2/2**
		Code Name			
		IC Information Contact			
PER02	**93**	**Name**	**O**	**AN**	**1/60**
PER03	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		TE Telephone			
PER04	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER05	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name			
		FX Facsimile			

PER06	**364**	**Communication Number**	**C**	**AN**	**1/80**
PER07	**365**	**Communication Number Qualifier**	**C**	**ID**	**2/2**
		Code Name EM Electronic Mail			
PER08	**364**	**Communication Number**	**C**	**AN**	**1/80**

Syntax:

1. PER03 P0304 -- If either PER03 or PER04 are present, then the others are required.
2. PER05 P0506 -- If either PER05 or PER06 are present, then the others are required.
3. PER07 P0708 -- If either PER07 or PER08 are present, then the others are required.

TD5 Carrier Details (Routing Sequence/Transit Time)

Pos: 420 Max: 12
Detail - Optional
Loop: N1 Elements: 3

To specify the carrier and sequence of routing and provide transit time information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
TD501	**133**	**Routing Sequence Code**	**O**	**ID**	**1/2**
		Code Name B Origin/Delivery Carrier (Any Mode)			
TD502	**66**	**Identification Code Qualifier**	**C**	**ID**	**1/2**
		Code Name 2 Standard Carrier Alpha Code (SCAC)			
TD503	**67**	**Identification Code**	**C**	**AN**	**2/80**

Syntax:

1. TD502 R0204050612 -- At least one of TD502, TD504, TD505, TD506 or TD512 is required.
2. TD502 C0203 -- If TD502 is present, then TD503 is required
3. TD507 C0708 -- If TD507 is present, then TD508 is required
4. TD510 C1011 -- If TD510 is present, then TD511 is required
5. TD513 C1312 -- If TD513 is present, then TD512 is required
6. TD514 C1413 -- If TD514 is present, then TD513 is required
7. TD515 C1512 -- If TD515 is present, then TD512 is required

Semantics:

1. TD515 is the country where the service is to be performed.

Comments:

1. When specifying a routing sequence to be used for the shipment movement in lieu of specifying each carrier within the movement, use TD502 to identify the party responsible for defining the routing sequence, and use TD503 to identify the actual routing sequence, specified by the party identified in TD502.

SE Transaction Set Trailer

Pos: 030 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	96	Number of Included Segments	M	N0	1/10
SE02	329	Transaction Set Control Number	M	AN	4/9

Comments:

1. SE is the last segment of each transaction set.

862 Shipping Schedule

Functional Group=SS

This Draft Standard for Trial Use contains the format and establishes the data contents of the Shipping Schedule Transaction Set (862) for use within the context of an Electronic Data Interchange (EDI) environment. The transaction set can be used by a customer to convey precise shipping schedule requirements to a supplier, and is intended to supplement the planning schedule transaction set (830). The shipping schedule transaction set will supersede certain shipping and delivery information transmitted in a previous planning schedule transaction, but it does not replace the 830 transaction set. The shipping schedule transaction set shall not be used to authorize labor, materials or other resources.The use of this transaction set will facilitate the practice of Just-In-Time (JIT) manufacturing by providing the customer with a mechanism to issue precise shipping schedule requirements on a more frequent basis than with the issuance of a planning schedule transaction, e.g., daily shipping schedules versus weekly planning schedules. The shipping schedule transaction also provides the ability for a customer location to issue shipping requirements independent of other customer locations when planning schedule transactions are issued by a consolidated scheduling organization.

Heading:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
010	ST	Transaction Set Header	**M**	**1**		
020	BSS	Beginning Segment for Shipping Schedule/Production Sequence	**M**	**1**		
030	DTM	Date/Time Reference	**O**	**10**		
LOOP ID - N1					**200**	
050	N1	Name	**O**	**1**		
060	N2	Additional Name Information	**O**	**2**		
070	N3	Address Information	**O**	**2**		
080	N4	Geographic Location	**O**	**1**		

Detail:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
LOOP ID - LIN					**10000**	
010	LIN	Item Identification	**M**	**1**		
020	UIT	Unit Detail	**M**	**1**		
047	QTY	Quantity	**O**	**1**		
050	REF	Reference Identification	**O**	**12**		
070	SDP	Ship/Delivery Pattern	**O**	**1**		
LOOP ID - FST					**100**	
080	FST	Forecast Schedule	**O**	**1**		

Summary:

Pos	Id	Segment Name	Req	Max Use	Repeat	Notes
020	SE	Transaction Set Trailer	**M**	**1**		

ST Transaction Set Header

Pos: 010 Max: 1
Heading - Mandatory
Loop: N/A Elements: 2

To indicate the start of a transaction set and to assign a control number.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
ST01	143	**Transaction Set Identifier Code**	M	ID	3/3
		Code Name			
		862 Shipping Schedule			
ST02	329	**Transaction Set Control Number**	M	AN	4/9

Semantics:

1. The transaction set identifier (ST01) used by the translation routines of the interchange partners to select the appropriate transaction set definition (e.g., 810 selects the Invoice Transaction Set).

BSS Beginning Segment for Shipping Schedule/ Production Sequence

Pos: 020 Max: 1
Heading - Mandatory
Loop: N/A Elements: 7

To transmit identifying numbers, dates, and other basic data relating to the transaction set.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
BSS01	353	**Transaction Set Purpose Code**	M	ID	2/2
		Code Name			
		00 Original			
		04 Change			
BSS02	127	**Reference Identification**	M	AN	1/30
BSS03	373	**Date**	M	DT	8/8
BSS04	675	**Schedule Type Qualifier**	M	ID	2/2
		All valid X12 codes are used.			

BSS05	373	Date	M	DT	8/8
BSS06	373	Date	M	DT	8/8
BSS10	324	Purchase Order Number	O	AN	1/22

Syntax:

1. BSS07 R0708 -- At least one of BSS07 or BSS08 is required.

Semantics:

1. Use BSS02 to indicate a document number.
2. Use BSS03 to indicate the date of this document.
3. Use BSS05 to indicate the schedule horizon start date (the date when the schedule begins).
4. Use BSS06 to indicate the schedule horizon end date (the date when the schedule ends).
5. BSS08 is the identifying number for a forecast assigned by the orderer/purchaser.

DTM Date/Time Reference

Pos: 030 Max: 10
Heading - Optional
Loop: N/A Elements: 3

To specify pertinent dates and times.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
DTM01	374	Date/Time Qualifier	M	ID	3/3
		Code Name			
		002 Delivery Requested			
		003 Invoice			
		004 Purchase Order			
		011 Shipped			
		018 Available			
		035 Delivered			
DTM02	373	Date	C	DT	8/8
DTM03	337	Time	C	TM	4/8

Syntax:

1. DTM02 R020305 -- At least one of DTM02, DTM03 or DTM05 is required.

2. DTM04 C0403 -- If DTM04 is present, then DTM03 is required
3. DTM05 P0506 -- If either DTM05 or DTM06 are present, then the others are required.

N1 Name

Pos: 050 Max: 1
Heading - Optional
Loop: N1 Elements: 2

To identify a party by type of organization, name, and code

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N101	**98**	**Entity Identifier Code**	**M**	**ID**	**2/3**
		Code Name			
		ST Ship To			
N102	**93**	**Name**	**C**	**AN**	**1/60**

Syntax:

1. N102 R0203 -- At least one of N102 or N103 is required.
2. N103 P0304 -- If either N103 or N104 are present, then the others are required.

Comments:

1. This segment, used alone, provides the most efficient method of providing organizational identification. To obtain this efficiency the "ID Code" (N104) must provide a key to the table maintained by the transaction processing party.
2. N105 and N106 further define the type of entity in N101.

N2 Additional Name Information

Pos: 060 Max: 2
Heading - Optional
Loop: N1 Elements: 1

To specify additional names or those longer than 35 characters in length.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N201	**93**	**Name**	**M**	**AN**	**1/60**

N3 Address Information

Pos: 070 Max: 2
Heading - Optional
Loop: N1 Elements: 2

To specify the location of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N301	166	Address Information	M	AN	1/55
N302	166	Address Information	O	AN	1/55

N4 Geographic Location

Pos: 080 Max: 1
Heading - Optional
Loop: N1 Elements: 4

To specify the geographic place of the named party.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
N401	19	City Name	O	AN	2/30
N402	156	State or Province Code	O	ID	2/2
N403	116	Postal Code	O	ID	3/15
N404	26	Country Code	O	ID	2/3

Syntax:

1. N406 C0605 -- If N406 is present, then N405 is required

Comments:

1. A combination of either N401 through N404, or N405 and N406 may be adequate to specify a location.
2. N402 is required only if city name (N401) is in the U.S. or Canada.

LIN Item Identification

Pos: 010 Max: 1
Detail - Mandatory
Loop: LIN Elements: 5

To specify basic item identification data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
LIN01	350	**Assigned Identification**	O	AN	1/20
LIN02	235	**Product/Service ID Qualifier**	M	ID	2/2
		Code Name			
		BP Buyer's Part Number			
LIN03	234	**Product/Service ID**	M	AN	1/48
LIN04	235	**Product/Service ID Qualifier**	C	ID	2/2
		Code Name			
		VP Vendor's (Seller's) Part Number			
LIN05	234	**Product/Service ID**	C	AN	1/48

Syntax:

1. LIN04 P0405 -- If either LIN04 or LIN05 are present, then the others are required.
2. LIN06 P0607 -- If either LIN06 or LIN07 are present, then the others are required.
3. LIN08 P0809 -- If either LIN08 or LIN09 are present, then the others are required.
4. LIN10 P1011 -- If either LIN10 or LIN11 are present, then the others are required.
5. LIN12 P1213 -- If either LIN12 or LIN13 are present, then the others are required.
6. LIN14 P1415 -- If either LIN14 or LIN15 are present, then the others are required.
7. LIN16 P1617 -- If either LIN16 or LIN17 are present, then the others are required.
8. LIN18 P1819 -- If either LIN18 or LIN19 are present, then the others are required.
9. LIN20 P2021 -- If either LIN20 or LIN21 are present, then the others are required.
10. LIN22 P2223 -- If either LIN22 or LIN23 are present, then the others are required.
11. LIN24 P2425 -- If either LIN24 or LIN25 are present, then the others are required.
12. LIN26 P2627 -- If either LIN26 or LIN27 are present, then the others are required.
13. LIN28 P2829 -- If either LIN28 or LIN29 are present, then the others are required.
14. LIN30 P3031 -- If either LIN30 or LIN31 are present, then the others are required.

Semantics:

1. LIN01 is the line item identification

Comments:

1. See the Data Dictionary for a complete list of IDs.

2. LIN02 through LIN31 provide for fifteen different product/service IDs for each item. For example: Case, Color, Drawing No., U.P.C. No., ISBN No., Model No., or SKU.

UIT Unit Detail

Pos: 020 Max: 1
Detail - Mandatory
Loop: LIN Elements: 1

To specify item unit data.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
UIT01	**C001**	**Composite Unit of Measure**	**M**	**Comp**	
	355	**Unit or Basis for Measurement Code** **Code Name** EA Each	**M**	**ID**	**2/2**

Syntax:

1. UIT03 C0302 -- If UIT03 is present, then UIT02 is required

QTY Quantity

Pos: 047 Max: 1
Detail - Optional
Loop: LIN Elements: 2

To specify quantity information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
QTY01	**673**	**Quantity Qualifier** **Code Name** 01 Discrete Quantity 02 Cumulative Quantity	**M**	**ID**	**2/2**
QTY02	**380**	**Quantity**	**C**	**R**	**1/15**

Syntax:

1. QTY02 R0204 -- At least one of QTY02 or QTY04 is required.
2. QTY02 E0204 -- Only one of QTY02 or QTY04 may be present.

Semantics:

1. QTY04 is used when the quantity is non-numeric.

REF Reference Identification

Pos: 050 Max: 12
Detail - Optional
Loop: LIN Elements: 2

To specify identifying information.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
REF01	**128**	**Reference Identification Qualifier**	**M**	**ID**	**2/3**
		Code Name			
		DK Dock Number			
		LF Assembly Line Feed Location			
REF02	**127**	**Reference Identification**	**C**	**AN**	**1/30**

Syntax:

1. REF02 R0203 -- At least one of REF02 or REF03 is required.

Semantics:

1. REF04 contains data relating to the value cited in REF02.

SDP Ship/Delivery Pattern

Pos: 070 Max: 1
Detail - Optional
Loop: LIN Elements: 2

To identify specific ship/delivery requirements

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SDP01	**678**	**Ship/Delivery or Calendar Pattern Code**	**M**	**ID**	**1/2**
		Code Name A Monday through Friday M Immediately N As Directed			
SDP02	**679**	**Ship/Delivery Pattern Time Code**	**M**	**ID**	**1/1**
		Code Name A 1st Shift (Normal Working Hours) F As Directed G Any Shift			

Comments:

1. The intent of this segment is to define the routine ship or delivery patterns, as required, when order quantities are in "buckets", such as weekly, monthly. Ship/delivery patterns eliminate the need to transmit discrete quantities and dates for each required shipment or delivery. It is assumed that a "bucketed" quantity is to be divided equally by the ship/delivery pattern. For example, a weekly quantity of 100 with a delivery pattern of Monday and Wednesday would result in 50 to be delivered on Monday and 50 to be delivered on Wednesday.

FST Forecast Schedule

Pos: 080 Max: 1
Detail - Optional
Loop: FST Elements: 5

To specify the forecasted dates and quantities.

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
FST01	**380**	**Quantity**	**M**	**R**	**1/15**
FST02	**680**	**Forecast Qualifier**	**M**	**ID**	**1/1**
		Code Name A Immediate C Firm D Planning			

FST03	**681**	**Forecast Timing Qualifier**	**M**	**ID**	**1/1**
		Code Name			
		C Daily			
		D Discrete			
		W Weekly Bucket (Monday through Sunday)			
FST04	**373**	**Date**	**M**	**DT**	**8/8**
FST10	**783**	**Planning Schedule Type Code**	**O**	**ID**	**2/2**
		Code Name			
		AA Customized Ordering Forecast			
		PS Purchaser to Seller			
		XA Requirement Forecast			

Syntax:

1. FST06 P0607 -- If either FST06 or FST07 are present, then the others are required.
2. FST08 P0809 -- If either FST08 or FST09 are present, then the others are required.

Semantics:

1. If FST03 equals "F" (indicating flexible interval), then FST04 and FST05 are required. FST04 would be used for the start date of the flexible interval and FST05 would be used for the end date of the flexible interval.

Comments:

1. As qualified by FST02 and FST03, FST04 represents either a discrete forecast date, the first date of a forecasted bucket (weekly, monthly, quarterly, etc.) or the start date of a flexible interval.
2. FST06 qualifies the time in FST07. The purpose of the FST07 element is to express the specific time of day in a 24-hour clock to satisfy "just-in-time" requirements. As an alternative, the ship/delivery pattern segment (SDP) may be used to define an approximate time, such as a.m. or p.m.

SE Transaction Set Trailer

Pos: 020 Max: 1
Summary - Mandatory
Loop: N/A Elements: 2

To indicate the end of the transaction set and provide the count of the transmitted segments (including the beginning (ST) and ending (SE) segments).

Element Summary:

Ref	Id	Element Name	Req	Type	Min/Max
SE01	**96**	**Number of Included Segments**	**M**	**N0**	**1/10**
SE02	**329**	**Transaction Set Control Number**	**M**	**AN**	**4/9**

Comments:

1. SE is the last segment of each transaction set.

Appendix

B

TradingXpert API Reference

This appendix provides the following details about the TradingXpert API:

- Introduction
- TradingXpert API Interface Methods and References

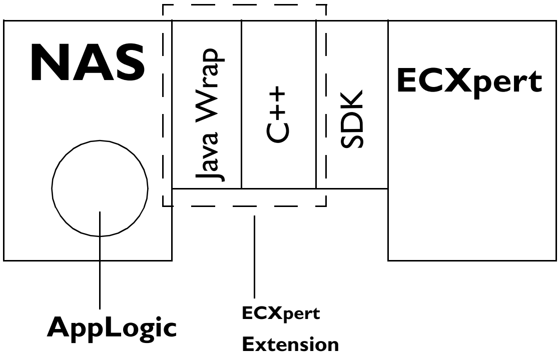
Introduction

The NAS extension of ECXpert contains Java interfaces to ECXpert objects, such as member, member address, partnership, document, tracking, log, service, service list, and submission. The extension is written in C++ with a Java wrapper, so that a developer can design Java application logic to directly use the extension to interface with the APIs in the ECXpert Software Developer's Kit (SDK).

Note to C++ programmers An *interface* in Java functions exactly as a *class* in C++.

The interfaces and methods in the extension have almost a one-to-one mapping to the classes and methods in the ECXpert SDK. Through this extension, most of the ECXpert functionality is exposed to any developer who wishes to design applications using ECXpert as a platform. For example, you can use the extension to administer user and partnership profiles, define services and service lists, submit documents into ECXpert, and track its workflow.

Figure B.1 Interaction with ECXpert



Note The Java classes wrap around the C++ interfaces.

TradingXpert API Interface Methods and References

The following fourteen NAS ECXpert extension interfaces are available for TradingXpert :

- IEcxAddress
- IEcxBase
- IEcxDocID
- IEcxDocument
- IEcxLog
- IEcxLogin
- IEcxMember
- IEcxMgr
- IEcxPartnerId
- IEcxPartnership
- IEcxService
- IEcxServiceList
- IEcxSubmit
- IEcxTracking

This section lists the methods for each interface available in the TradingXpert API and provides the command syntax, parameter descriptions, and return values for each TradingXpert API Interface that does not appear in the *iPlanet ECXpert Developer's Guide*.

Notes An asterisk (*) after a method indicates that a note for that method is included at the end of the method list.

In cases where the method references are documented in the *iPlanet ECXpert Developer's Guide*, a reference to the appropriate chapter within that document is provided. Reference details are provided there.

Important All *accessor* methods (those methods whose names start with "get") will return either an integer or a String. If any of the method calls fail, it will return null.

M*utator* methods (those methods whose names start with "set") will return zero (0) if they succeed or a non-zero error code if they fail.

The rest of the methods, such as *list*, *add*, *delete*, and *next,* also return zero (0) for success and an error code for failure--with the exception of *more*, which returns the number of records remaining in the list.

The `IEcxMgr` Interface is the only interface not mapped directly to an ECXpert SDK class. Similarly, all methods except those in the `EcxMgr` Interface map to the corresponding SDK class methods.

IEcxAddresses

Syntax `public interface IEcxAddresses extends com.kivasoft.IObject`

The `IEcxAdresses` interface represents trading address records in an ECXpert database. Administrators can manipulate any address record; non-administrators can add and delete only their own address records. A user must be logged in to the database before accessing a record.

Methods
`public int add()`
`public int clear()`
`public int delete()`
`public java.lang.String getMember()`
`public java.lang.String getQual()`
`public java.lang.String getQualId()`
`public int list()`
`public int more()`
`public int next()`
`public int setLogin(ecx.IEcxLogin pLogin)`
`public int setMember(java.lang.String pMember)`
`public int setQual(java.lang.String pQualifier)`
`public int setQualId(java.lang.String pId)`

For additional details about each method in the `IEcxAddress` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on The ECXAddresses Class.

IEcxBase

Syntax `public interface extends com.kivasoft.IObject`

The `IEcxBase` Interface defines the interface from which all ECXpert API interfaces are derived. For example, ECXpert's `IEcxSubmit` Interface is derived from the `IEcxBase` Interface. You may define a sub-interface of the `IEcxBase` Interface and create objects from the derived interface. The `IEcxBase` Interface is intended to be used as an abstract interface. You should never need to create `IEcxBase` objects.

The IEcxBase Interface defines methods that are common to the ECXpert API interfaces you use to interact programmatically with the ECXpert System. The interface provides methods that allow you to get, set, and clear the error number corresponding to the last error reported by ECXpert. For example, you can create an object from an interface derived from IEcxBase and call that object's errnum() method to determine whether or not an error occurred. You can call the object's clearErr() method to reset the error condition to the "no error" state.

For additional details about each method in the IEcxBase Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on The EcxBase Class.

IEcxDocID

Syntax public interface IEcxDocId extends com.kivasoft.IObject

The IEcxDocId Interface represents a key from which documents can be retrieved from the database. You must create an IEcxDocId object before you can call the document's get() and getDocId()methods. A document ID key consists of the following values:

- tracking ID
- interchange ID
- group ID
- document ID

Example:

```
/* an IEcxDocument object is required before this class file can
be used.
   See the example of the IEcxDocument class */

import java.util.*;

public int ShowDocId (IEcxDocument currentECXDocObj)
throws Exception
{
    IEcxDocId   docIDObj;
    docIDObj = currentECXDocObj.getDocId();
    if (docIDObj != null)
    {
        System.out.println ("DocIdObject retrieved, documentId 
is " + docIDObj.getDocumentId());
        System.out.println ("  tracking ID is " + 
docIDObj.getTrackingId());
        System.out.println ("  interchange ID is " + 
docIDObj.getInterchangeId());
        System.out.println ("  group ID is " + 
docIDObj.getGroupId());
    }
    else
    {
       System.out.println ("  getDocId returned a null IEcxDocId 
object, error number is " + 
((IEcxBase)currentECXDocObj).errnum());
        return FAIL;
    }

    return SUCCESS;
}
```

Methods
`public int getDocumentId()`
`public int getGroupId()`
`public int getInterchangeId()`
`public int getTrackingId()`
`public int setValues (int trkId,int intId,int grpId, int docId)`

For additional details about each method in the `IEcxDocId` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on Document-Related Classes.

IEcxDocument

Syntax `public interface IEcxDocument extends com.kivasoft.IObject`

The `IEcxDocument` Interface represents documents sent to the logged-in user via ECXpert. You can retrieve these document records and access information that identifies them, such as the file name that contains the document's content.

Example:

```
import java.util.*;

public int listDocuments (IEcxMgr ecxMgr, IEcxLogin ecxLogin)
{
     IEcxDocument  currentECXDoc =
ecxMgr.createDocument(ecxLogin);

    if (currentECXDoc.list () != 0)
    {
        System.out.println ("IECXDocment.list (unread) failed,
error number " + ((IEcxBase)currentECXDoc).errnum());
        return FAIL;
    }

    System.out.println ("FXDocList: " + currentECXDoc.more() + "
documents found");

    while (currentECXDoc.more() > 0)       // loop through each
document
    {
        System.out.println ("  Filename = " +
currentECXDoc.getFileName());

        if (currentECXDoc.more() >= 1)  // if there are still
more documents to retrieve
        {
            if (currentECXDoc.next() != 0)
            {
                System.out.println ("Retrieving next document
failed");
                return FAIL
            }
        }
    }
}
```

Methods

```
public ecx.IEcxDocId getDocId()
public int clear()
public int delete()
public int get(ecx.IEcxDocId pDocId,int markRead)
public java.lang.String getAttachedFileName(int index, int
trkId)*
public int getAttachmentCount(int trkId)
public int getCardCount()
public short getCardFlags(int cardNum)
public short getCardIOType(int cardNum)
public int getCreationDate()*
public short getDataState()
public java.lang.String getDocType()
public java.lang.String getFileName()
public int getModifyDate()*
public int getPartNum()*
public int getPartTotal()*
public short getRead()
public java.lang.String getRelease()
public java.lang.String getSecondaryTitle()
public java.lang.String getSecondaryValue()
public java.lang.String getSenderName()
public java.lang.String getStandard()
public short getState()
public java.lang.String getTitle()
public java.lang.String getTranslatedFileName()*
public java.lang.String getValue()
public java.lang.String getVersion()
public java.lang.String getXportParam()
public java.langString getXportType()
public int list()*
public int list (int flags)
public int listByReceiver(java.lang.String receiver, int
flags)*
public int more()
public int next()
public int setLogin (ecx.IEcxLogin pLogin)
public int setRead(short read)
public int setReadyForPurge(ecx.IEcxDocId pDocId)
public int setSenderName()
public int setTrackState(short state)
public int setXportType(java.lang.String xportType)
```

***Notes** When you use the `getAttachedFileName(int index, int trkId)` method, you must check if the returned value is null.

The `getCreationDate()` and `getModifyDate()` methods return the **UNIX** date, which is the number of seconds since January 1, 1970.

The `getPartNum()` method returns the part number of the document. A value of one (1) indicates the main document; subsequent values indicate attached documents.

The `getPartNum()` and `getPartTotal()` methods are used together to determine if a file is the main part, the attached part, or the dummy part of the submission. When a submission includes an attachment, it includes at least three parts: the first part is always the dummy part, to accommodate the fact that the first part of an e-mail attachment sent through a browser is always empty; the second part is the main document; and the remaining parts are attachments. If a user submits a document via the browser's email, the browser treats the document as an attachment, making the submission a two-part submission—a dummy part plus the main document. If a user submits a document via sendmail, that user must follow the same convention of sending first a dummy part and then the main document.

If you are unable to retrieve something (i.e. a translated file name) using one of the tracking IDs returned by the `list()` method, do not be alarmed. Some of the attributes of the retrieved document may not exist, depending on the nature of the document. Simply skip to the next one.

The `listByReceiver(java.lang.String receiver, int flags)` method is exactly the same as `list()`, except that it lists documents received by the receiver. Only admin can call this method.

For additional details about each method in the `IEcxDocument` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on Document-Related Classes.

IEcxLog

Syntax `public interface IEcxLog extends com.kivasoft.IObject`

The `IEcxLog` Interface represents entries in the ECXpert log. You can use an `IEcxLog` object to add an entry to the log.

Methods
`public int clear()`
`public java.lang.String getELCategory()`
`public int getELDocId()`
`public int getELEventId()`
`public java.lang.String getELEventShortMsg()`
`public int getELGrpId()`
`public int getELId()`
`public int getELIntgId()`
`public int getELSeverity()`
`public java.lang.String getELTDId()`
`public int getELTrkId()`
`public int logEvent(int errnum,int severity, java.lang.String message,int trkId)`
`public int more()`
`public int next()`
`public int retrieveLog(int trkId,java.lang.StringsndrMBName, java.lang.String rcvrMBName,int fromDt,int toDt,short stateBitMap)`*
`public int setLogin(ecx.IEcxLogin pLogin)`

*__Note__ The two arguments for `retrieveLog()`, `fromDt` and `toDt,` must be in the UNIX date format, which is the number of seconds since January 1, 1970.

For additional details about each method in the `IEcxLog` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on The EcxLog Class.

IEcxLogin

Syntax `public interface IEcxLogin extends com.kivasoft.IObject`

Objects of the `IEcxLogin` Interface represent connections to ECXpert. To log in to the database, you can create an `IEcxLogin` object and call the object's `login` method. When you no longer need the connection to ECXpert, you should call the object's l`ogout` method.

Example:

```
/* this routine retrieves the userName and the password from the
NAS session */

import java.util.*;

public IEcxLogin loginToEcx()
{
    IEcxMgr   ecxMgr;
    IEcxLogin ecxLogin;

    ecxMgr = access_cECX.getcECX (context, null, this);
    ecxLogin = ecxMgr.createLogin();

    if (ecxLogin.login (userSession.getUserName(),
userSession.getPassword() != 0);
    {
        System.out.println ("Invalid Login, error number = " +
((IEcxBase) ecxLogin).errnum());
        return null;
    }

    return ecxLogin;
}
```

Methods

```
public int login(java.lang.String pUsername,java.lang.String
pPassword)
public int logout()
public int memberType()
```

For additional details about each method in the `IEcxLogin` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on The EcxLogin Class.

IEcxMember

Syntax `public interface IEcxMember extends com.kivasoft.IObject`

The `IEcxMember` Interface represents member records in an ECXpert database. Administrators can manipulate any member record for their trading partnerships; non-administrators can only change contact information in their own record. A user must be logged-in to the database before accessing a record.

Example:

```
/* This example sets the Miscellaneous information about a
member.
   The Contact* variables are assumed to be supplied
   from somewhere else
*/

public void updateContactInfo(IEcxMgr ecxMgr, IEcxLogin
ecxLogin, String userName)
throws Exception
{

IEcxMember ecxMember = ecxMgr.createMember(ecxLogin);

ecxMember.clear();
ecxMember.setName(userName);
if(ecxMember.get()!=0)
   throw new IOException ("ECXmember.Get failed");
ecxMember.setContactName(contactName);
ecxMember.setContactAddress1(contactAddress1);
ecxMember.setContactAddress2(contactAddress2);
ecxMember.setContactCity(contactCity);
ecxMember.setContactState(contactState);
ecxMember.setContactCountry(contactCountry);
ecxMember.setContactZip(contactZip);
ecxMember.setContactFax(contactFax);
ecxMember.setContactPhone(contactPhone);
ecxMember.setContactEmailId(contactEmailId);

if(ecxMember.change()!=0)
   throw new Exception("ECXmember.Change failed on update
contactName, Error number " + ((IEcxBase)ecxMember).errnum());

}
```

Methods
`public int add()`
`public int change()`
`public int clear()`
`public int delete()`
`public int get()`
`public int getActive()`
`public java.lang.String getContactAddress1()`
`public java.lang.String getContactAddress2()`
`public java.lang.String getContactCity()`
`public java.lang.String getContactCompany()`
`public java.lang.String getContactCountry()`
`public java.lang.String getContactEmailId()`
`public java.lang.String getContactFax()`
`public java.lang.String getContactName()`
`public java.lang.String getContactPhone()`
`public java.lang.String getContactState()`
`public java.lang.String getContactZip()`

```
public java.lang.String getDescription()
public int getIsGroup()
public java.lang.String getModByGroup()
public java.lang.String getModByUser()
public java.lang.String getModDt()*
public java.lang.String getName()
public java.lang.String getParentName()
public java.lang.String getPassword()
public int getObjPerm()
public int getTrusted()
public int getType()
public int list()
public int more()
public int next()
public int setActive(int active)
public int setContactAddress1(java.lang.String
contactAddress1)
public int setContactAddress2(java.lang.String
contactAddress2)
public int setContactCity(java.lang.String
contactCity)
public int setContactCompany(java.lang.String
contactCompany)
public int setContactCountry(java.lang.String
contactCountry)
public int setContactEmailId(java.lang.String
contactEmailId)
public int setContactFax(java.lang.String contactFax)
public int setContactName(java.lang.String
contactName)
public int setContactPhone(java.lang.String
contactPhone)
public int setContactState(java.lang.String
contactState)
public int setContactZip(java.lang.String contactZip)
public int setDescription(java.lang.String
pDescription)
public int setIsGroup(int isGroup)
public int setLogin(ecx.IEcxLogin pLogin)
public int setName(java.lang.String pName)
public int setObjPerm(int objPerm)
public int setPassword(java.lang.String password)
public int setTrusted(int trusted)
public int setType(int type)
```

***Note** The `getModDt` accessor returns a date in text format, e.g., "Oct 21, 1998, 15:31:34."

For additional details about each method in the `IEcxMember` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on The EcxMember Class.

IEcxMgr

Syntax `public interface IEcxMgr extends com.kivasoft.IObject`

The `IEcxMgr` Interface represents the manager interface of all interfaces in the extension. It is used to create instances of the rest of the TradeXpert interfaces. The instance of `IEcxMgr` must be obtained before you can use any interface in the extension.

Given that ECX is the extension name:

```
IEcxMgr ecxMgr = access_cECX.getcECX(context, null, this)
```

The `IEcxMgr` object now may be used to create instances of all other interfaces.

Example:

See the sample code for IEcxLogin.

Method
`public ecx.IEcxAddresses createAddresses(ecx.IEcxLogin pLogin)`
`public ecx.IEcxDocId createDocId(int trkId,int intId,int grpId,int docId)`
`public ecx.IEcxDocument createDocument(ecx.IEcxLogin pLogin)`
`public ecx.IEcxLog createLog(ecx.IEcxLogin pLogin)`
`public ecx.IEcxLogin createLogin()`
`public ecx.IEcxMember createMember(ecx.IEcxLogin pLogin)`
`public ecx.IEcxPartnerId createPartnerId(int partnershipId,int standardId, java.lang.String docType,int groupId)`
`public ecx.IEcxPartnership createPartnership(ecx.IEcxLogin pLogin)`
`public ecx.IEcxService createService(ecx.IEcxLogin pLogin)`
`public ecx.IEcxServiceList createServiceList(ecx.IEcxLogin pLogin)`
`public ecx.IEcxSubmit createSubmit()`
`public ecx.IEcxTracking createTracking(ecx.IEcxLogin pLogin)`

createAddresses(ecx.IEcxLogin pLogin)

Creates an instance of `IEcxAddress`.

Syntax `public ecx.IEcxAddresses createAddresses(ecx.IEcxLogin pLogin)`

Returns A reference to that instance on success, null on failure.

Parameters

`pLogin`	A reference of a valid `IEcxLogin` object.

Discussion It must pass in a valid instance of `IEcxLogin`.

createDocId(int trkId, int intId, int grpId, int docId)

Creates an instance of `IEcxDocId`.

Syntax

```
public ecx.IEcxDocId createDocId(int trkId,int intId,int
grpId,int docId)
```

Returns The reference to that instance on success, null on failure.

Parameters

`docId`	An integer that is a valid document ID.
`grpId`	An integer that is a valid group ID.
`intId`	An integer that is a valid interchange Id.
`trkId`	An integer that is a valid tracking ID.

Discussion It must pass in valid tracking ID, interchange ID, group ID and document ID.

Example See the `IEcxDocId` Interface for sample code.

createDocument(ecx.IEcxLogin pLogin)

Creates an instance of `IEcxDocument`.

Syntax

```
public ecx.IEcxDocument createDocument(ecx.IEcxLogin pLogin)
```

Returns The reference to that instance on success, null on failure.

Parameters

`pLogin`	A reference of a valid `IEcxLogin` object.

Discussion It must pass in a valid instance of `IEcxLogin`.

Example See the `IEcxDocument` Interface for sample code.

createLog(ecx.IEcxLogin pLogin)

Creates an instance of `IEcxLog`.

Syntax `public ecx.IEcxLog createLog(ecx.IEcxLogin pLogin)`

Returns The reference to that instance on success, null on failure.

Parameters

`pLogin`	A reference of a valid `IEcxLogin` object.

Discussion It must pass in a valid instance of `IEcxLogin`.

createLogin( )

Creates an instance of `IEcxLogin`.

Syntax `public ecx.IEcxLogin createLogin()`

Returns The reference to that instance on success, null on failure.

Example See the `IEcxLogin` Interface for sample code.

createMember(ecx.IEcxLogin pLogin)

Creates an instance of `IEcxMember`.

Syntax `public ecx.IEcxMember createMember(ecx.IEcxLogin pLogin)`

Returns The reference to that instance on success, null on failure.

Parameters

`pLogin`	A reference of a valid `IEcxLogin` object.

Discussion It must pass in a valid instance of `IEcxLogin`.

createPartnerId(int partnershipId,int standardId,java.lang.String docType,int groupId)

Creates an instance of `IEcxPartnerId`.

Syntax

```
public ecx.IEcxPartnerId createPartnerId(int partnershipId,int
standardId,java.lang.String docType)
```

Returns The reference to that instance on success, null on failure.

Parameters

`partnershipId`	An integer that is a valid partnership ID.
`standardId`	An integer that is a valid standard ID.
`docType`	A string that is a valid document type.
`groupId`	An integer that is a valid group ID.

Discussion It must pass in valid partnership ID, standard ID, document type and group ID.

createPartnership(ecx.IEcxLogin pLogin)

Creates an instance of `IEcxPartnership`.

Syntax

```
public ecx.IEcxPartnership createPartnership(ecx.IEcxLogin
pLogin)
```

Returns The reference to that instance on success, null on failure.

Parameters

`pLogin`	A reference of a valid `IEcxLogin` object.

Discussion It must pass in a valid instance of `IEcxLogin`.

Example See the `IEcxPartnership` Interface for sample code.

createService(ecx.IEcxLogin pLogin)

Creates an instance of `IEcxService`.

Syntax

```
public ecx.IEcxService createService(ecx.IEcxLogin pLogin)
```

Returns The reference to that instance on success, null on failure.

Parameters

`pLogin`	A reference of a valid `IEcxLogin` object.

Discussion It must pass in a valid instance of `IEcxLogin`.

createServiceList(ecx.IEcxLogin pLogin)

Creates an instance of `IEcxServiceList`.

Syntax

```
public ecx.IEcxServiceList createServiceList(ecx.IEcxLogin
pLogin)
```

Returns The reference to that instance on success, null on failure.

Parameters

`pLogin`	A reference of a valid `IEcxLogin` object.

Discussion It must pass in a valid instance of `IEcxLogin`.

createSubmit( )

Creates an instance of `IEcxSubmit`.

Syntax `public ecx.IEcxSubmit createSubmit()`

Returns The reference to that instance on success, null on failure.

Example See the `IEcxSubmit` Interface for sample code.

createTracking(ecx.IEcxLogin pLogin)

Creates an instance of `IEcxTracking`.

Syntax `public ecx.IEcxTracking createTracking(ecx.IEcxLogin pLogin)`

Returns The reference to that instance on success, null on failure.

Parameters

`pLogin`	A reference of a valid `IEcxLogin` object.

Discussion It must pass in a valid instance of `IEcxLogin`.

Example See the `IEcxTracking` Interface for sample code.

IEcxPartnerId

Syntax `public interface IEcxPartnerId extends com.kivasoft.IObject`

The `IEcxPartnerId` interface represents a key from which partnership views can be retrieved from the database. You must create an `IEcxPartnerId` object before you can call the partnership's `get()` and `getPartnerId()` methods. A partner ID consists of the following values:

- document type
- partnership ID

- standard ID
- group ID

In general, values for a partnership ID and a standard ID are the same for each record in the view.

Methods
`public int getGroupId`
`public int getPartnershipId()`
`public int getStandardId()`
`public int setValues(int partnership_Id, intstandard_id,java.lang.String doctype,int group_Id)`
`public java.lang.String getDocType()`

For additional details about each method in the `IEcxPartnerId` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on Partnership-Related Classes.

IEcxPartnership

Syntax `public interface IEcxPartnership extends com.kivasoft.IObject`

The `EcxPartnership` interface represents a view on the following kinds of records in an ECXpert database:

- document types
- EDI standards information
- partnership groups
- partnerships

A record in the view represents a partnership record whose ID matches a standards information ID, a group ID and a document type ID and whose group type matches the document type.

Only administrators can add, change, or delete records using this view. An administrator can retrieve any record from the view; a non-administrator can only retrieve records from the view that includes the user as either a sender or receiver. A user must be logged in to the database before accessing a record through the view.

Example:

```
public void whoSendsMeDocuments(IEcxMgr ecxMgr, IEcxLogin
ecxLogin, String userName) throws IOException
{
    ecxPtnship = ecxMgr.createPartnership(ecxLogin);
    ecxPtnship.clear();

    if (ecxPtnship.list(userName)!=0)    //no partners were be
found, throw  IOException
        throw new IOException ("EcxPartnership.List failed.
Error number " + ((IEcxBase)ecxPtnship).errnum());

    //retrieve partnership contact information

    while(ecxPtnship.more()!=0)
    {
        System.out.println ("user " + userName + " receives " +
ecxPtnship.getDoctype()
                            + " type documents from " +
ecxPtnship.getSenderName());
        ecxPtnship.next();
    }
}
```

Method
`public int add()`
`public int change()`
`public int clear()`
`public int delete()`
`public int get(ecx.IEcxPartnerId pPartnerId)`
`public int getAckExpected()`

```
public int getActive()
public int getArchiveWaitPeriod()
public java.lang.String getDecimalPointCharacter()
public int getDeleteWaitPeriod()
public java.lang.String getDescription()
public java.lang.String getDocLastControlNumber()
public int getDocLock()
public int getDocPriority()
public java.lang.String getDoctype()
public java.lang.String getElementSeparator()
public int getGenOptEnv()
public int getGroupGenerateDocAck()
public java.lang.String getGroupLastControlNumber()
public int getGroupLock()
public java.lang.String getGroupType()
public int getIntchngAckWaitPeriod()
public int getIntchngGenerateAck()
public java.lang.String getIntchngLastControlNumber()
public int getIntchngLock()
public int getMapDirection()
public java.lang.String getMapName()
public java.lang.String getOutRelease()
public java.lang.String getOutStandard()
public java.lang.String getOutVersion()
public ecx.IEcxPartnerId getPartnerId()
public int getPreEnveloped()
public java.lang.String getPrimaryXportParam()
public java.lang.String getPrimaryXportType()
public java.lang.String getRcvrAppCode()
public java.lang.String getRcvrAppQual()
public int getReceiverCertificateType()
public java.lang.String getReceiverName()
public java.lang.String getReceiverQual()
public java.lang.String getReceiverQualId()
public java.lang.String getReleaseCharacter()
public java.lang.String getSecondaryXportParam()
public java.lang.String getSecondaryXportType()
public int getSecurity()
public java.lang.String getSegmentTerminator()
public int getSenderCertificateType()
public java.lang.String getSenderName()
public java.lang.String getSenderQual()
public java.lang.String getSenderQualId()
public int getSendType()
public java.lang.String getSndrAppCode()
public java.lang.String getSndrAppQual()
public java.lang.String getStandardName()
public java.lang.String getStandardRelease()
public java.lang.String getStandardVersion()
public java.lang.String getSubElementSeparator()
public int getTestProductionFlag()
public int list(java.lang.String partner)
public int more()
public int next()
public int setAckExpected(int ackExpected)
public int setActive(int active)
public int setArchiveWaitPeriod(int archiveWaitPeriod)
public int setDecimalPointCharacter(java.lang.String
decimalPointCharacter)
public int setDeleteWaitPeriod(int deleteWaitPeriod)
public int setDescription(java.lang.String description)
public int setDocLastControlNumber(java.lang.String
docLastControlNumber)
public int setDocLock(int docLock)
public int setDocPriority(int docPriority)
public int setDoctype(java.lang.String docType)
public int setElementSeparator(java.lang.String
elementSeparator)
public int setGenOptEnv(int genOptEnv)
public int setGroupGenerateDocAck(int groupGenerateDocAck)
public int setGroupLastControlNumber(java.lang.String
groupLastControlNumber)
public int setGroupLock(int groupLock)
public int setGroupType(java.lang.String groupType)
public int setIntchngAckWaitPeriod(int intchngAckWaitPeriod)
public int setIntchngGenerateAck(int intchngGenerateAck)
public int setIntchngLastControlNumber(java.lang.String
intchngLastControlNumber)
public int setIntchngLock(int intchngLock)
public int setLogin(ecx.IEcxLogin pLogin)
public int setMapDirection(int mapDirection)
public int setMapName(java.lang.String mapName)
public int setOutRelease(java.lang.String outRelease)
public int setOutStandard(java.lang.String outStandard)
public int setOutVersion(java.lang.String outVersion)
public int setPartnerId(ecx.IEcxPartnerId pPartnerId)
public int setPreEnveloped(int preEnveloped)
public int setPrimaryXportParam(java.lang.String
primaryXportParam)
public int setPrimaryXportType(java.lang.String
primaryXportType)
public int setRcvrAppCode(java.lang.String rcvrAppCode)
public int setRcvrAppQual(java.lang.String rcvrAppQual)
public int setReceiverCertificateType(int
receiverCertificateType)
public int setReceiverName(java.lang.String receiverName)
public int setReceiverQual(java.lang.String receiverQual)
public int setReceiverQualId(java.lang.String
receiverQualId)
public int setReleaseCharacter(java.lang.String
releaseCharacter)
```

```
public int setSecondaryXportParam(java.lang.String
secondaryXportParam)
public int setSecondaryXportType(java.lang.String
secondaryXportType)
public int setSecurity(int security)
public int setSegmentTerminator(java.lang.String
segmentTerminator)
public int setSenderCertificateType(int
senderCertificateType)
public int setSenderName(java.lang.String pSenderName)
public int setSenderQual(java.lang.String senderQual)
public int setSenderQualId(java.lang.String senderQualId)
public int setSendType(int sendType)
public int setSndrAppCode(java.lang.String sndrAppCode)
public int setSndrAppQual(java.lang.String sndrAppQual)
public int setStandardName(java.lang.String standardName)
public int setStandardRelease(java.lang.String
standardRelease)
public int setStandardVersion(java.lang.String
standardVersion)
public int setSubElementSeparator(java.lang.String
subElementSeparator)
public int setTestProductionFlag(int testProductionFlag)
```

For additional details about each method in the `IEcxPartnership` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on Partnership-Related Classes.

IEcxService

Syntax `public interface IEcxService extends com.kivasoft.IObject`

The `IEcxService` Interface represents service records in an ECXpert database. Only administrators can add, change, or delete a service record. A user must be logged-in to the database before viewing a service record.

Methods
`public int add()`
`public int change()`
`public int clear()`
`public int delete()`
`public int get()`
`public java.lang.String getEntryName()`
`public int getId()`
`public int getMaxThread()`
`public java.lang.String getModByGroup()`
`public java.lang.String getModByUser()`
`public java.lang.String getModDt()`*
`public java.lang.String getName()`
`public int getObjPerm()`
`public java.lang.String getParam()`
`public java.lang.String getPathName()`
`public int getType()`
`public int list()`
`public int more()`
`public int next()`
`public int setEntryName(java.lang.String entryName)`
`public int setId(int id)`
`public int setLogin(ecx.IEcxLogin pLogin)`
`public int setMaxThread(int maxThread)`
`public int setName(java.lang.String name)`
`public int setObjPerm(int objPerm)`
`public int setParam(java.lang.String param)`
`public int setPathName(java.lang.String pathName)`
`public int setType(int type)`

*****Note** The accessor `getModDt()` returns a date in text format, e.g., “Oct 21, 1998, 15:31:34.”

For additional details about each method in the `IEcxService` Interface, refer to the *iPlanet ECXpert Developer’s Guide* chapter on The EcxService Class.

IEcxServiceList

Syntax `public interface IEcxServiceList extends com.kivasoft.IObject`

The `IEcxServiceList` Interface represents service list records in an ECXpert database. Only administrators can add, change, or delete a service list record. A user must be logged-in to the database before viewing a service list record.

Methods
`public int add()`
`public int change()`
`public int clear()`
`public int delete()`
`public int get()`
`public java.lang.String getDesc()`
`public java.lang.String getErrorHandler()`
`public java.lang.String getModByGroup()`
`public java.lang.String getModByUser()`
`public java.lang.String getModDt()`*
`public int getObjPerm()`
`public java.lang.String getRcvrMBName()`
`public int getSeqNum()`
`public java.lang.String getServiceListName()`
`public java.lang.String getServiceParams()`
`public java.lang.String getSndrMBName()`
`public int getSvrId()`
`public java.lang.String getSvrName()`
`public java.lang.String getTypeName()`

```
public int list()
public int more()
public int next()
public int setDesc(java.lang.String desc)
public int setErrorHandler(java.lang.String
errorHandler)
public int setObjPerm(int objPerm)
public int setRcvrMBName(java.lang.String rcvrMBName)
public int setSeqNum(int seqNum)
public int setServiceListName(java.lang.String
serviceListName)
public int setServiceParams(java.lang.String
serviceParams)
public int setSndrMBName(java.lang.String sndrMBName)
public int setSvrId(int svrId)
public int setSvrName(java.lang.String svrName)
public int setTypeName(java.lang.String typeName)
```

***Note** The accessor `getModDt()` returns a date in text format, e.g., "Oct 21, 1998, 15:31:34."

For additional details about each method in the `IEcxServiceList` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on The EcxService Class.

IEcxSubmit

Syntax `public interface IEcxSubmit extends com.kivasoft.IObject`

The `IEcxSubmit` Interface defines methods that you use to submit a file to ECXpert. You can use these methods to provide a file submission capability within your application instead of requiring the user to execute a command or use ECXpert's HTML interface to submit an object.

You may create objects from the `IEcxSubmit` Interface and use them directly, or you may define a subinterface of the `IEcxSubmit` Interface and create objects from the derived interface. For example, you might define a sub-interface that handles much of the application logic associated with files to be submitted to ECXpert. Objects derived from your sub-interface would inherit the ability to submit files to ECXpert.

You call methods to specify this information. For example, you call the object's `setSender()` method to specify the sender's member ID. You must specify the files that you wish to submit to ECXpert. You build a submission list by calling the object's `addFile()` method to add a file to the list. You specify the following information when you add a file:

- Document name
- Document type, such as EDIFACT or EDIX12, or a non-EDI type

You can add as many files as you want. If you add more than one file, the files become part of a single multi-part file. When you finish adding the files to the submission list, you can call the object's `Submit()` method to submit the files.

By default, ECXpert moves the files being submitted to the directory specified by the `repository` entry in the configuration file's `tcpip-connector` section. Moving a file (copying the file and deleting the source file after copying) is the most efficient way to submit files if your application executes on the same server as ECXpert.

You can also submit files to ECXpert using a TCP/IP connection. You specify whether or not to use a TCP/IP connection when you call the object's `submit()` method. Using a TCP/IP connection causes ECXpert to stream the contents of the files through a socket to the server. This is a useful technique if your application runs on a remote computer and the files being submitted are relatively small. If you want to submit large files from a remote computer, you should consider using a protocol such as FTP to copy the files to the server and then submit them from the server.

Note If you stream data through a TCP/IP connection, the source file is not deleted after the data has been streamed to the server.

After you submit a file, you should check for errors. If no error occurred, you can call the object's `getFirstTrackingID()` method to determine the tracking ID of the first file submitted and the object's `getNextTrackingID()` method to determine the tracking ID for each additional file in the list.

Warning If the `submit()` method fails, the value returned by calling the `getFirstTrackingID()` or `getNextTrackingID()` method is undefined. When you no longer need references to these files, you can call the object's `clearFileList()` method to remove the files from the list.

Example:

```
WARNING:  This is a machine generated list, do not modify below
** WizardDictionaryValues={
**   CodeTemplate="/kiva/templates/DoInputWizard.javatmpl",
**   CodeFiles="*.java;Session:SessionAccessorInsert.java",
**   CodeProject="Input",
**   CodeDir="/kiva/APPS/ecx_demo/",
**   CodeLanguage="Java",
**   SessionOut=[
**     "sender"
**     "password"
**     "recipient"
**     "fileName"
**     "fileType"
**     "ecxIniFileName"
**   ],
**   BaseAgent="ecx_demo.BaseAppLogic",
**   CodeWizard="com.kivasoft.wizard.DoInputWizardFactory",
**   CodeFile="/kiva/APPS/ecx_demo/Input.java",
**   Input_filename="/kiva/APPS/DevXpert/web/ecx_demo/
index.html",
**   CodeGUID="{588779da-f69c-15e5-e4e3-080020794ab3}",
**   Project="/kiva/APPS/ecx_demo/ecx_demo.gxm",
**   ValIn={com.kivasoft.tools.KSVectorHash
**     ValIn=[
**       "sender"
**       "password"
**       "recipient"
**       "fileName"
**       "fileType"
**       "ecxIniFileName"
**       "remoteSubmission"
**     ],
**     ValIn_NotNull=[
**       "true"
**       "true"
**       "true"
**       "true"
**       "true"
**       "true"
**       "true"
**     ],
**   },
** }
** WARNING:  This is a machine generated list, do not modify
above
*/
package ecx_demo;

import java.util.*;

import com.kivasoft.*;
import com.kivasoft.applogic.*;
import com.kivasoft.session.*;
import com.kivasoft.types.*;
import com.kivasoft.util.*;

import ecx_demo.Session;

import ecx_demo.BaseAppLogic;

import ecx.*;

public class Input extends ecx_demo.BaseAppLogic
{

    public String guid()
    {
        return "{588779da-f69c-15e5-e4e3-080020794ab3}";
    }

    public int execute()
    {
        ecx_demo.Session session = getSessionProxy();

        if (session == null) {
          return result("<HTML>Call to getSessionProxy() failed
in Input</HTML>");
```

```
        }

        //
        // Verify correctness of valIn criteria
        //
        String sender = valIn.getValString("sender");

        if ( null == sender ||
             0 == sender.trim().length() )
        {
            log("Input error on sender");
            return result("<HTML><BODY>sender should not be
null!</BODY></HTML>");
        }
        String password = valIn.getValString("password");

        if ( null == password ||
             0 == password.trim().length() )
        {
            log("Input error on password");
            return result("<HTML><BODY>password should not be
null!</BODY></HTML>");
        }
        String recipient = valIn.getValString("recipient");

        if ( null == recipient ||
             0 == recipient.trim().length() )
        {
            log("Input error on recipient");
            return result("<HTML><BODY>recipient should not be
null!</BODY></HTML>");
        }
        String fileName = valIn.getValString("fileName");

        if ( null == fileName ||
             0 == fileName.trim().length() )
        {
            log("Input error on fileName");
            return result("<HTML><BODY>fileName should not be
null!</BODY></HTML>");
        }
        String fileType = valIn.getValString("fileType");

        if ( null == fileType ||
             0 == fileType.trim().length() )
        {
            log("Input error on fileType");
            return result("<HTML><BODY>fileType should not be
null!</BODY></HTML>");
        }
        String ecxIniFileName =
valIn.getValString("ecxIniFileName");

        if ( null == ecxIniFileName ||
             0 == ecxIniFileName.trim().length() )
        {
            log("Input error on ecxIniFileName");
           return result("<HTML><BODY>ecxIniFileName should not
be null!</BODY></HTML>");
        }
        String remoteSubmission =
valIn.getValString("remoteSubmission");

        if ( null == remoteSubmission ||
             0 == remoteSubmission.trim().length() )
        {
            log("Input error on remoteSubmission");
            return result("<HTML><BODY>remoteSubmission should
not be null!</BODY></HTML>");
        }

        //
        // Save login criteria into the session.
        //
        session.setsender(valIn.getValString("sender"));
        session.setpassword(valIn.getValString("password"));
        session.setrecipient(valIn.getValString("recipient"));
        session.setfileName(valIn.getValString("fileName"));
        session.setfileType(valIn.getValString("fileType"));

session.setecxIniFileName(valIn.getValString("ecxIniFileName"))
;
        session.saveSession();


    // Get the extension
    IEcxMgr ecxMgr = access_cECX.getcECX(context,null,this);
    IEcxSubmit ecxSubmit = ecxMgr.createSubmit();

    System.out.println("Got the extension...");

    ecxSubmit.setSender(sender);
    ecxSubmit.setRecipient(recipient);
    ecxSubmit.setPassword(password);
    ecxSubmit.addFile(fileName, fileType);
    ecxSubmit.setEcxIniFileName(ecxIniFileName);

    System.out.println("Set all parameters...");

    boolean remote;
    if (remoteSubmission.equals("yes"))
      remote = true;
    else
      remote = false;
    ecxSubmit.submit(remote);

    // Return screens
    if (((IEcxBase)ecxSubmit).errnum() == 0)
    {
       String successString = "Submission successful, the file’s
ECXpert tracking ID is " + ecxSubmit.getFirstTrackingID() + ".";
        return streamResult(successString);
    }
    else
    {
        String errorString = "Submission failed, error number "
+ ((IEcxBase)ecxSubmit).errnum() + ".";
```

```
            return streamResult(errorString);
        }

    } // execute

} // class
```

Methods

```
public int addFile(java.lang.String pFile,java.lang.String
pFileType)
public int clearFileList()
public java.lang.String getDeliveryMethod()
public java.lang.String getEcxIniFileName()
public int getFirstTrackingID()
public java.lang.String getMapName()
public java.lang.String getPassword()
public java.lang.String getRecipient()
public java.lang.String getSender()
public int getNextTrackingID()
public int setDeliveryMethod(java.lang.String
pDeliveryMethod)
public int setEcxIniFileName(java.lang.String pIniFileName)
public int setMapName(java.lang.String pMapName)
public int setPassword(java.lang.String pPassword)
public int setRecipient(java.lang.String pRecipient)
public int setSender(java.lang.String pSender)
public int submit(boolean bDataStreaming)
```

For additional details about each method in the `IEcxSubmit` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on The EcxSubmit Class.

IEcxTracking

Syntax `public interface IEcxTracking extends com.kivasoft.IObject`

The `IEcxTracking` Interface represents documents sent from the logged-in user via ECXpert. You can retrieve the tracking status of a document using an `IEcxTracking` object.

Example:

```
import java.util.*;

public void listOutBoundDocuments (IEcxMgr ecxMgr, IEcxLogin
ecxLogin, String userName)
throws Exception
{
    currentECXTrackObj = ecxMgr.createTracking(ecxLogin);
    if (currentECXTrackObj == null)
      throw new Exception ("Can’t initialize a tracking object.
Error number " + ((IEcxBase)currentEcxTrackObj).errnum());

    if (currentECXTrackObj.list ("", 0, 0, 0, userName) != 0)
        throw new Exception ("IECXTracking.list() failed, error
number = " + ((IEcxBase)currentEcxTrackObj).errnum());

   System.out.println ("currentECXTrackObj.more() + " documents
found");

    while (currentECXTrackObj.more() > 0)       // loop through
each document
    {
        System.out.println ("Sent " +
currentECXTrackObj.getDocType() + " type document to " +
currentECXTrackObj.getReceiverName());
        {
            if (currentECXTrackObj.next() != 0)
                throw new Exception ("Retrieving next tracking
object failed.  Error number " +
((IEcxBase)currentECXTrackObj).errnum());
        }
    }
}
```

Methods

```
public ecx.IEcxDocId getDocId()
public int clear()
public int delete()
public int get(ecx.IEcxDocId docId,int markRead)
public int getCreationDate()
public java.lang.String getDocType()
public short getDataState()
public java.lang.String getFileName()
public int getModifyDate()
public int getPartNum()*
public int getPartTotal()*
public int getProgress()
public java.lang.String getReceiverName()
public java.lang.String getRelease()
public java.lang.String getSecondaryTitle()
public java.lang.String getSecondaryValue()
public java.lang.String getStandard()
public short getState()
public java.lang.String getTitle()
public java.lang.String getTranslatedFileName()*
public java.lang.String getValue()
public java.lang.String getVersion()
public boolean isBundle()*
public int list(java.lang.String receiver,int
fromDt,int toDt,int stateFlag,java.lang.String
sender)*
public int list ()*
public int more()
public int next()
public int setLogin(ecx.IEcxLogin pLogin)
public int setReadyForPurge(ecx.IEcxDocId pDocId)
```

***Notes** The `getPartNum()` method returns the part number of the document. A value of one (1) indicates the main document; subsequent values indicate attached documents.

The `getPartNum()` and `getPartTotal()` methods are used together to determine if a file is the main part, the attached part, or the dummy part of the submission. When a submission includes an attachment, it includes at least three parts: the first part is always the dummy part, to accommodate the fact that the first part of an e-mail attachment sent through a browser is always empty; the second part is the main document; and the remaining parts are attachments. If a user submits a document via the browser's email, the browser treats the document as an attachment, making the submission a two-part submission—a dummy part plus the main document. If a user submits a document via sendmail, that user must follow the same convention of sending first a dummy part and then the main document.

The two arguments for `list()`, `fromDt` and `toDt`, must be in the UNIX date format, which is the number of seconds since January 1, 1970.

Also, if you are unable to retrieve something (i.e. a translated file name) using one of the tracking IDs returned by the `list()` method, do not be alarmed. Some of the attributes of the retrieved document may not exist, depending on the nature of the document. Simply skip to the next one.

The `isBundle()` method returns `true` if the tracking ID is generated by the bundle process, `false` if it is not. Actual tracking IDs generated by the bundle process are not displayed in TradingXpert's Inbound and Outbound folders.

Important The `getCreationDate()` and the `getModifyDate()` methods return the **UNIX** date, which is the number of seconds since January 1, 1970.

For additional details about each method in the `IEcxTracking` Interface, refer to the *iPlanet ECXpert Developer's Guide* chapter on The EcxTracking Class.

Glossary

997 A *confirmation message* in the *ANSI X12* standard. Comparable to the *CONTRL message* in the *EDIFACT* standard.

Administrative Interface The component of *TradingXpert* that provides access to TradingXpert's administrative functions. The Administrative Interface consists of forms where users enter data to update the TradingXpert *database*. In TradingXpert Version 3.5.1, the Administrative Interface is divided into two separate interfaces, the *Product Administrative Interface* and the *Server Administrative Interface*.

AIAG HTTP Automotive Industry Action Group's (AIAG) industry-specific implementation of *HTTP*.

ANSI X12 The ANSI (American National Standards Institute) ASC (Accredited Standards Committee) standard for *EDI*. X12 is used widely in North America. The *EDIFACT* EDI standard is used more widely internationally.

application format An application-specific data format for *documents*. Also referred to as a *proprietary format*.

archive To remove *TradingXpert* data from the production *database* and store it in another location. Such archived data can be restored to the TradingXpert database if necessary

authentication A process in which the recipient of an electronic transmission can verify that the sender's identity is as it is represented and the data received is as it was sent.

batch processing

Business data processing in which transaction data is accumulated until some time interval has passed or some trigger volume is exceeded. The accumulated transactions are then processed in a batch. The opposite end of the processing spectrum from *real-time transaction processing*.

bundling

Combining multiple documents for transmission as a unit. Generally synonymous with *enveloping*. Functionally the reverse of *parsing*.

CA

See *certificate authority (CA)*.

card

A logical unit in the *Map Definition Tool* software that specifies the details of the *EDI translation* between a *data element* in an input format (EDI or non-EDI) and the corresponding data element in an output format (EDI or non-EDI), or vice versa.

certificate

A digital document that supports a *trading partner's* claim to ownership of a *public key*. A certificate aids in verification of a trading partner's claim that a given public key does in fact belong to that trading partner, thus making it more difficult for a fraudulent third party to impersonate a trading partner.

certificate authority (CA)

An agency that issues certificates.

certificate revocation list (CRL)

A list of *certificate*s that have been revoked by the *certificate authority (CA)* (CA) that issued them and should no longer be accepted. If you receive a certificate revocation list (CRL), you can import it into ECXpert just as you would import a certificate from that CA. ECXpert stores it in the database and then rejects any certificate from that CA that appear on the CRL.

Communications

The component of *TradingXpert* that handles all incoming and outgoing communications sessions for your TradingXpert installation. In *inbound processing*, when the *submission unit* originates outside of the domain of your TradingXpert installation, the Communications component receives the submission unit and writes it to a disk file. In *outbound processing*, when the submission unit originates within the domain of your TradingXpert, the Transport component sends the submission unit.

Communications Agent

A subcomponent of the *Communications* component. A separate Communications Agent exists for each *communications protocol* that *TradingXpert* supports. A Communications Agent simply receives data and materializes a file on the TradingXpert system.

communications protocol

A standard set of rules that the systems on both the sending and receiving end follow in a communications session.

compliance

Conforming to the rules of a standard, as in *year 2000 compliance*. In *EDI*, being in agreement with the syntax rules of an EDI standard.

compliance check Examining a *submission unit* to ensure that it is in agreement with requirements of the standard used to create it.

compression The process of compacting data so that it is represented by fewer bytes, thereby reducing the size of the file that has to be stored on disk or transmitted over communications lines. The reverse of *decompression* or expansion.

confirmation message A message returned to the Sender, confirming that the Receiver has received the document(s) that were sent. In the *ANSI X12* standard, this is a *997* document. In the *EDIFACT* standard, this is a *CONTRL message*.

CONTRL message In the *EDIFACT* standard, a message sent to the sender of an EDIFACT message, acknowledging that a particular *interchange* has been received and *compliance checked*. A CONTRL message indicates the syntactical correctness of the business *documents* that have been received, informing the sender of any problems encountered. A CONTRL message does not deal with data content, which is application-specific. TradingXpert supports CONTRL message at the *interchange*, *functional group*, and document levels. In *ANSI X12*, the document type that is comparable to an EDIFACT CONTRL message is called a *functional acknowledgment*.

CRL See *certificate revocation list (CRL)*.

data delimiters Special characters in *EDI* that mark the boundary between data elements or subelements. In the *ANSI X12* standard, delimiters cannot appear anywhere in the data within an interchange. In the *EDIFACT* standard, delimiters appearing in the data must be preceded by a *release character*.

data element The basic unit of *EDI* data, roughly corresponding to a field in a database setting. Typical examples of data elements are unit price, quantity, and product code.

data element separator See *data delimiters*.

data segment In *EDI*, a structured sequence of *data elements*, separated by *data element separators*. A data segment is comparable to a record in a database. A data segment may also be comparable to a line item on a printed form, when similar data segments repeat within a *document*.

database The database containing all the data being processed by *TradingXpert*. Separate disk files are used to store each incoming *submission unit* from a *trading partner* that has been received and each outgoing submission unit to a trading partner that is ready to be sent. Audit information is also stored separately. All other data in the TradingXpert is maintained in the database. All access to the database is handled through application calls to the *ECXpert Data Store API*. The database is similar in concept to what is commonly called a “mailbox.”

decompression The process of restoring compacted data to its original format so that it can be read and processed. The reverse of *compression*.

decryption The process of decoding data that has been encrypted, or encoded, in such a way that it is only readable by someone who possesses a decoding key. The reverse, encoding process is called *encryption*. See also *public key encryption*.

digital certificate See *certificate*.

Dispatcher The component of the *TradingXpert* that acts as the "traffic cop". Except for receipt of incoming data by the *Communications* component, most of the processing that occurs within TradingXpert is initiated by the Dispatcher. A separate instance of the Dispatcher manages the processing of each *submission unit*. The total number of Dispatchers that are permitted to be active at one time can be set by the TradingXpert site administrator.

document A business document, such as a purchase order or an invoice. In *EDI*, a document is more narrowly defined as a collection of *data segments* with a defined structure that carries all the information necessary to complete a specified part of a business transaction. A document can also be referred to as a *message*, or a *transaction set*.

document type A specific *EDI standard document* definition. Each EDI document type has a document type number (*ANSI X12*) or name (*EDIFACT*). Many document type numbers also have version numbers.

document tracking Determining the current status of *documents* being processed by *TradingXpert*. This is done through the Tracking tabs of the *Product Administrative Interface*.

dual certificates In *public key encryption*, use of one *certificate* for *encryption* and another for *signing*, instead of using the same certificate for both.

TradingXpert Short name for the *Netscape TradingXpert System*.

ECXpert Data Store API The API that moves data back and forth between the TradingXpert *database* and the forms of the *Administrative Interface*.

EDI Electronic data interchange. A set of standardized formats for different types of business *documents* that allow otherwise incompatible business data processing systems to exchange documents without manual intervention.

EDI translation The conversion of data in *application formats* to and from *EDI standard formats*.

EDI Translator/ Mapper The component of the *TradingXpert System* that handles the translation of business *documents* between different *application formats* and *standard EDI formats*.

EDIFACT Electronic Data Interchange For Administration, Commerce, and Transportation. An international implementation of *EDI* sponsored by the United Nations and the European Union. The syntax rules are identified as ISO 9735. EDIFACT is widely used internationally, while *ANSI X12* is more widely used in North America.

EERP See *end-to-end response (EERP).*

encryption Encoding data in such a way that it is only readable by someone who possesses a key for decoding the data. The reverse, decoding process is called *decryption*. See also *public key encryption.*

end-to-end response (EERP) In *OFTP* transport, an end-to-end response, or EERP, notifies the original sender of a file that it has been successfully delivered to its final destination, no matter how many intermediate hops it made, or how it may have been split into multiple files or combined with other files. An EERP is an acknowledgment from the ultimate recipient that the data has been received. An EERP is comparable to *message disposition notification (MDN)* in *SMTP* transport.

envelope In *EDI* communications, the structural and communications data added onto the basic *document*(s) that are sent as a unit. Information about the *envelope* is added at the *interchange*, *functional group* (optional in *EDIFACT*), and *document* levels.

enveloping The *EDI* process of adding the *envelope* data, including *delimiter characters*, to basic *documents* that are sent as a unit. Usually used synonymously with *bundling*. Functionally the reverse of *parsing*.

ERP systems Enterprise Resource Planning systems. Major data processing applications that support resource management in an enterprise. Examples include Oracle Financials and SAP.

eXML Connector The ECXpert extension that supports *XML*.

expansion See *decompression.*

external member An TradingXpert *member* that operates outside your TradingXpert data processing domain. External members are usually other independent business entities outside of your organization. External members are often customers of or vendors to your organization.

format, proprietary See *proprietary format.*

format, standard EDI See *standard EDI format.*

FTP

File transfer protocol. A data communications *protocol* for transferring files directly between servers and clients over the Internet, without use of electronic mail. See also *TCP/IP*.

functional acknowledgment

In the *ANSI X12* standard (*document type* 997), a message sent to the sender of submission unit acknowledging that a particular *functional group* has been received and compliance checked. A functional acknowledgment indicates the syntactical correctness of the business *documents* that have been received, informing the sender of any problems encountered. A functional acknowledgment does not deal with data content, which is application-specific. TradingXpert supports functional acknowledgment at both the functional group and *document* levels. In *EDIFACT*, the document type that is comparable to an ANSI X12 functional acknowledgement is called a *CONTRL message*.

functional group

In *EDI*, a collection of one or more *documents* that is being sent to the same *trading partner* that share a logical correlation. One or more functional groups compose an *interchange*. In *ANSI X12*, functional groups are required and each functional group must consist of documents of the same *document type* and of the same *group type* as defined by the standard (e.g., PO, IN). In *EDIFACT*, functional groups are optional and, if present, must consist of documents of the same document type.

Gateway

A subcomponent of the *Communications* component. The Gateway is a multi-threaded server, started at TradingXpert startup, that is responsible for all communications between TradingXpert and other systems.

GEIS FTP

A *protocol* for accessing the General Electric Information Systems (GEIS) EDI*EXPRESS service using *FTP*.

GISB HTTP

Gas Industry Standards Board (GISB) industry-specific implementation of *HTTP*.

group type

In the *ANSI X12* standard, a family of related *document types*. Most group types contain only one document type, but a few contain as many as ten or twenty.

HTTP

Hypertext transfer protocol. A set of rules for exchanging web pages on the World Wide Web. A web page may be composed of multiple files, containing both text and graphics. See also *TCP/IP*.

IFC

Internet Foundation Class. A set of libraries that must be available to your browser in order to support the Java-based *Product Administrative Interface*.

inbound processing

The flow of processing that occurs when TradingXpert is receiving a *submission unit*. The reverse of *outbound processing*.

interchange

In *EDI*, the highest level of *enveloping*. An *ANSI X12* interchange is composed of one or more *functional groups*. In an *EDIFACT* interchange, functional groups are optional. A TradingXpert *submission unit* can contain multiple interchanges. Transport via *SMTP* requires one interchange per message file. ANSI X12 specifies that a new interchange be created whenever any of the following changes:

- either sender ID (From) or recipient ID (To)
- *EDI standard format*
- test/production flag
- delimiter or terminator

EDIFACT specifies that a new interchange be created, in addition to the above situations, whenever the detailed routing information changes within the same recipient.

internal member

An TradingXpert *member* that operates within your TradingXpert data processing domain. Internal members are usually departments or other administrative units within your organization.

ISO 9735

See *EDIFACT*.

job tracking

Determining the current status of jobs managed by the *Scheduler*. This is done through the Job Tracking tabs in the *Product Administrative Interface*.

LDAP

Lightweight Directory Access Protocol. An internet standard protocol for interfacing with directories.

legacy system

A business data processing system that existed before your TradingXpert was implemented and which may require data translation by TradingXpert. In ECXpert 3.5.1, the legacy systems that you can integrate with ECXpert are Oracle Financials, SAP, and MQSeries.

manifest

The list of documents contained in a *submission unit*.

map

A specification of the way in which *data* in one format, such as an *EDI standard format*, must be transformed into another format, such as an *application format*.

map, TradingXpert-cooperative

In *outbound processing*, a *map* that allows the outbound submission unit to be constructed *document* by document in separate files. The documents can then be bundled by recipient and transmitted directly to their respective recipients. This feature requires the application producing the outbound submission unit to place a header and trailer around each logical document.

Map Definition Tool

The TradingXpert program used to create a *map file* that the *Map Execution Engine* can use. *Maps* from other sources, such as maps that have been in use by *legacy systems*, have to be reconstructed through the Map Definition Tool in order to be used by TradingXpert.

Map Execution Engine

The TradingXpert program that uses a map file created by the *Map Definition Tool* to translate documents from an *application format* to a *standard EDI format*, or from a standard EDI format to an application format.

map file

A file supplied by the user that contains *map* information.

mapping

The process of specifying the way in which *data* in one format, such as an *EDI standard format*, must be transformed into another format, such as an *application format*.

MD5

Message digest (MD) hashing algorithm. Hashing algorithms are a key component of *public key encryption*. MD5 is a revision earlier of versions that improves the level of security. The MD5 algorithm is slightly faster than *SHA-1*, but it is less secure.

MDN

See *message disposition notification (MDN).*

member

A participant in your TradingXpert system that is fully defined via the *Administrative Interface*. A member is a data processing system and/or an individual that sends or receives business documents through your TradingXpert System. *Internal members* operate within your TradingXpert data processing domain. *External members* operate outside your TradingXpert data processing domain. *Trusted members* act as agents for others and handle data validation for those they represent. Specific *trading address* are assigned to individual members, and may only be assigned to a single member.

Mercator

The *Map Definition Tool* that is bundled with TradingXpert, developed by TSI International. It can produce *map files* that translate from any supported format to any other supported format: EDI to application, application to EDI, EDI to EDI, and application to application.

message

See *document.*

message disposition notification (MDN)

In *SMTP* transport, a process that provides a series of notifications to the sender of a *submission unit* about its current status as it is being received by the *Gateway*. The *Communications* component provides MDN to reliably track delivery of a submission unit via Internet mail. MDN supplements the *functional acknowledgment* of *ANSI X12*, or the *CONTRL message* of *EDIFACT*, but does not replace either.

MIME Multi-purpose internet mail extensions. The standard Internet protocol that lets users exchange application-specific file formats via Internet email. Web servers and clients use the data type specified in the MIME header that is inserted at the beginning of any Web transmission with an appropriate receiver application. The MIME specification is an amendment to the original SMTP mail protocol. The *S/MIME* specification adds security functions to MIME.

multiple body parts The TradingXpert feature that allows structured business documents to be accompanied by attached application-specific data files of any defined format. For example, an EDI request for quote might be accompanied by a CAD file containing related engineering drawings and a spreadsheet file that can be used as a worksheet in generating the quote. TradingXpert treats each "body part" as a separate *submission unit* with its own *tracking ID* and *service list*. TradingXpert cross-references all related body parts and none are sent until all have completed processing. Currently *SMTP* is the only protocol that can support this feature. The `multi_part` parameter in the `commsmtp-send` section of the system settings must be set to true to enable TradingXpert to send multiple body parts. No special settings are required for TradingXpert to be able to receive them.

OBI Open Buying on the Internet. An Internet standard providing for support of purchase transactions over the Internet. OBI is based on current standards, including *SSL* for secure Internet communications, *HTML* for content display, *SET* for credit card transactions, and *X.509* for *certificates*.

Odette FTP (OFTP) Organisation for Data Exchange by Tele Transmission in Europe ("Odette") file transfer protocol, originally developed for the European auto industry. OFTP is not a variation of *FTP*, but a completely separate protocol.

OFTP See *Odette FTP (OFTP)*.

outbound processing The flow of processing that occurs when TradingXpert is sending a *submission unit*. The reverse of *inbound processing*.

parsing The process of breaking out all the data components of a submission unit. Functionally the reverse of *bundling* or *enveloping*.

partnership See *trading partnership*.

partnership, reverse See *reverse partnership*.

pass-through Use of ECXpert as a simple gateway through which documents are passed without *parsing* or *translation*.

poll command The TradingXpert utility that polls a specified port or file location for the presence of data. When found, it initiates TradingXpert processing of a *submission unit* by TradingXpert.

primary service A *service* that is internal to the TradingXpert System, available as soon as the software is installed. Primary services process all *documents* within a *submission unit* the same way. Examples of primary TradingXpert services are *parsing*, *translation*, and *functional acknowledgment* generation. You may create *user-defined services* to supplement TradingXpert's primary services.

private key The key belonging to an intended message recipient that is never published. The message sender uses the intended recipient's *public key* to encrypt the message. A message encrypted using the intended recipient's public key can only be decrypted using the intended recipient's private key.

Product Administrative Interface In *TradingXpert* Version 3.5.1, the portion of the *Administrative Interface* that handles normal system functions involving maintenance of the information infrastructure that supports the automated processing of business *document*s in TradingXpert. Setting up users, trading partnerships, and EDI parameters are typical routine administrative functions. System administration functions in Version 3.5.1 are handled by the *Server Administrative Interface* portion of the Administrative Interface.

proprietary format An application-specific data format for *documents*. Also called an *application format*.

protocol A special pre-defined set of communication rules that both the sender and receiver in a telecommunication connection agree to use in a communication. The Internet is supported by the *TCP/IP* family of protocols.

public key The published key belonging to an intended message recipient. The message sender uses the intended recipient's public key to encrypt the message. A message encrypted using the intended recipient's public key can only be decrypted using the intended recipient's *private key*.

public key encryption An *encryption* method in which sender and receiver each have two keys, one public and one private. The *public key* is published as widely as necessary so that anyone can determine with certainty the correct public key for a given trading partner. The relationship between a trading partner's public and private keys is such that a message encrypted by the public key can only be correctly decrypted using the *private key*.

qualifier In *EDI*, a code at the beginning of a *data element* that specifies how the data in the remainder of the field is to be interpreted. The *trading address qualifier* is the one you encounter most frequently in the TradingXpert *Administrative Interface*.

real-time transaction processing
Business data processing in which data for a transaction is processed as soon as it is received. The opposite end of the processing spectrum from batch processing.

release character
In the *EDIFACT* standard, a character that is used to restore a character to its original meaning when it has been specified as a *data delimiter*. A release character allows a data delimiter to appear within the data.

reverse partnership
A *trading partnership* automatically created by ECXpert when the original partnership is set up to have ECXpert generate a *confirmation message*; either a *functional acknowledgment* (*ANSI X12* document type 997) or a *CONTRL message* (*EDIFACT*). The reverse partnership is required to support the exchange of the these messages. ECXpert automatically creates the reverse partnership for you when you request a confirmation message for a partnership. In order for the confirmation messages to be processed correctly, you must edit the reverse partnership and specify appropriate *envelope* and *protocol* information. Also the service list for the original partnership must contain FAGen.

A reverse partnership reverses the Sender and Receiver information from the original partnership, using the same EDI standard and version. The Partnership Type is Application to EDI. The Document Type is 997 for ANSI X12 or CONTRL for EDIFACT. Multiple partnerships with the same Sender and Receiver, but with different Document Types, all share the same reverse partnership.

A reverse partnership is also required to support end-to-end response (EERP) under Odette FTP, but ECXpert does not create it for you automatically.

scheduled service list
A *service list* for which processing is time-based, triggered by the *Scheduler*.

Scheduler
The component of *TradingXpert* that manages scheduling of time-based processing.

scenario
A specific example that illustrates a way in which *ECXpert* can be used.

segment
See *data segment*.

segment terminator
In *EDI*, A special character that is used to mark the end of a *data segment*.

Server Administrative Interface

In *TradingXpert* Version 3.5.1, the portion of the *Administrative Interface* that handles system administration functions, such as configuring the system, starting and stopping ECXpert servers, and setting up time-based processing jobs. Maintenance of the information infrastructure that supports the automated processing of business *document*s in TradingXpert is handled through the Product Administrative Interface portion of the Administrative Interface.

service

A specific action that can be performed on a *submission unit*, or a subset of *documents* in the submission unit, which changes, moves, or copies the data. A service is an external executable file. *TradingXpert* provides internal, or *primary services*, and supports external, or *user-defined services*.

service list

A list of *services* that are to be performed in sequence on a *submission unit*. Service list processing is managed by the *Dispatcher*.

service list, scheduled

See *scheduled service list*.

service, primary

See *primary service*.

service, user-defined

See *user-defined service*.

session

The entire sequence of TradingXpert processing of a *submission unit*, including the communications session in which it is received or sent.

SHA-1

Secure hashing algorithm (SHA). Hashing algorithms are a key component of *public key encryption*. SHA-1 is a revision of SHA that corrects a flaw in the original algorithm. The SHA-1 algorithm is slightly slower than *MD5*, but it is more secure.

signing

Use of a certificate for *authentication* of the sender of an electronic transmission.

site administrator

The person with primary responsibility for ongoing operation of your *TradingXpert* installation. This person may also be referred to as the system administrator.

S/MIME

Secure multi-purpose internet mail extensions. An extension to the *MIME* protocol that adds encryption, decryption, and authentication to prevent unauthorized recipients from being able to make use of the information.

SMTP

Simple mail transport protocol. The standard Internet protocol under which electronic mail is transmitted.

SNMP

Simple network management protocol. A set of rules governing the management of networks and the monitoring of network devices and functions. SNMP is used in conjunction with *TCP/IP*, although it has evolved independently.

SSL

Secure sockets layer. Netscape Communications Corporation's public key encryption and authentication software that can be used with *HTTP*.

standard EDI format

A specific standard format for *documents* defined under *EDI*.

Submission Agent

The *Gateway* calls a Submission Agent and passes it the output of the *Communications Agent*. The Submission Agent writes the file to a disk, and submits the file to the TradingXpert *Dispatcher* for processing.

submission unit

A collection of one or more business *documents* that is processed as a unit by the *TradingXpert System*. In *inbound processing* a submission unit is received from an *external member* and passed on to an *internal member*. In *outbound processing* a submission unit from an internal member is sent to an external member. TradingXpert allows a submission unit to be composed of whatever combination of data is required by the internal and external members that are involved. With *SMTP* transport, the *EDI* component of a submission unit must be composed of a single *interchange* sent to a single *trading partner*.

submit command

An TradingXpert command line utility that can be used to present a *submission unit* to the TradingXpert System for processing.

syntax (EDI)

The rules governing structure of *documents* transmitted under *EDI*, including the following:

- valid data types and relationships within a *data segment*
- valid order, position, and frequency of repetition of data segments in a document
- organization of documents composing *functional groups* and *interchanges*

TCP/IP

Transmission Control Protocol/Internet Protocol. The primary Internet *protocols* which govern the exchange of messages between Internet points at the information packet level and the Internet address level.

tracking ID

The unique identifier that TradingXpert generates and assigns to all *documents* in a *submission unit* so that all pieces of the submission unit can be tracked to completion of processing.

trading address

A unique identifier for a *trading partner*. A trading address *data element* consists of a *trading address qualifier* and the actual ID.

trading address qualifier The first characters of a trading address *data element*, the value of which specifies how the remainder of the trading address is to be interpreted. For example, in *ANSI X12* '12' indicates a phone number, while '01' indicates a Duns number, and 'ZZ' indicates a unique, mutually agreed upon ID established in a *trading partner agreement*.

trading partner Either one of the two *members* involved in a *trading partnership*.

trading partner agreement A contractual agreement between two business parties that specifies all legal and business requirements that are to be met when exchanging *EDI* transmissions.

trading partnership The set of data defining a relationship between an external trading partner and an internal trading partner on your TradingXpert.

transaction set See *document*.

trusted member A *member*, such as a *VAN*, who acts as an agent for other *trading partners*, sending *documents* on their behalf. A trusted member assumes the responsibility for validation of *trading addresses* referenced in documents sent on behalf of others.

user-defined service A *service* that a TradingXpert user defines to perform processing that is not provided by a *primary service*. User-defined services are defined by creating an executable program that uses the *ECXpert Data Store API* to access the *database*. Examples of user-defined services are encryption/decryption, compression/decompression, and data moving/copying.

VAN Value-added network. A third-party communications service that handles large volumes of *EDI* transmissions for its clients.

X12 See *ANSI X12*.

XML eXtensible Markup Language. A web standard defining an extensible markup language that can be used to encode complex document data. See also *eXML Connector*.

year 2000 compliance Being able to correctly process dates in different centuries. Much of the early software developed in the 1950's and '60's processed dates using only the last two digits of the year, which would not allow correct computation of the elapsed time between years beginning with 19 and years beginning with 20. Software such as *ECXpert* that is "year 2000 compliant" processes dates using all four digits of the year.

Index