



New Features Guide

EDI Products

5.1

DOCUMENT ID: DC39126-01-0510-01

LAST REVISED: October 2007

Copyright © 1992-2007 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase trademarks can be viewed at the Sybase trademarks page at <http://www.sybase.com/detail?id=1011207>. Sybase and the marks listed are trademarks of Sybase, Inc. ® indicates registration in the United States of America.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book	vii	
CHAPTER 1	Introduction	1
	Sybase EDI Products	1
	EDI Products 5.1 new features	2
	Product licensing with SySAM 2.0 [EDI 5.0].....	3
	SySAM concepts	3
CHAPTER 2	ECMAP	5
	Support for new versions of standards [EDI 5.1].....	5
	System variables [EDI 5.0]	6
	EDI-to-EDI mapping [EDI 5.0]	6
	Creating an EDI-to-EDI map	7
	Defining an EDI-to-EDI map	8
	Developing the map	9
	EDI-to-EDI rule commands and system variables.....	13
	Updated reports.....	22
	Boolean logic [EDI 5.0].....	23
	Boolean datatype	23
	New Boolean rule commands	23
	Updated rule commands	26
	Global repository [EDI 5.0]	28
	Enabling the repository.....	29
	Importing and exporting rule definitions	29
	Importing and exporting record definitions	30
	Importing and exporting memory variables	31
	Managing the repositories	32
	Save map options [EDI 5.0]	32
	Save option	32
	Save As option	33
	Map recovery option.....	33
	Password security [EDI 5.0]	34
	Read_Uncommitted option [EBF 4.2.5].....	34

- TA1 acknowledgements [EBF 4.2.4] 34
 - TA1XLG map [EBF 4.2.4]..... 35
 - LOGUPDATE map [EBF 4.2.4] 35
- Rules 36
 - Binary Data Placement command [EDI 4.2]..... 36
 - Enhanced rules import [EDI 4.2] 37
- RunMap options 37
- Compliance map acknowledgement options [EDI 4.2]..... 38
- Cross-reference table import utility [EDI 4.2] 38
- Record size increase [EBF 4.2.1]..... 38

CHAPTER 3 ECMAP Debugger 39

- ECMAP Debugger features [EDI 5.1]..... 39
- Accessing ECMAP Debugger 40
 - Accessing ECMAP Debugger in Windows 40
- ECMAP Debugger main window [EDI 5.1]..... 42
- Preparing to debug maps [EDI 5.1]..... 46
 - Obtaining map command line switches..... 46
 - Managing debugging sessions [EDI 5.1]..... 46
 - Starting EC RTP and loading map information 50
 - Setting and clearing breakpoints 51
 - Viewing specific real-time data values 52
 - Saving projects and debug sessions [EDI 5.1]..... 53
- Running an ECMAP Debugger session 54
 - Starting and controlling a debug session 54
 - Monitoring real-time data values 55
 - Monitoring the map run progress [5.1] 55
 - Viewing the trace file 55
 - Completing the debug session 56

CHAPTER 4 EC Gateway 57

- Reusing e-FTP scripts for multiple systems [EBF 4.2.3]..... 58
- Added security for files created by pfs/Async [EBF 4.2.3] 58
- Find and replace functions [EDI 4.2] 58
- Copying process scripts to other systems [EDI 4.2]..... 59
- Wildcard e-mail attachments [EDI 4.2]..... 60
- Passing input to the EC Gateway executable [EDI 4.2] 61
 - Passing input to and getting output
 from the EC Gateway DLL shared library [EDI 4.2]..... 62
- RunMap options 64
- Return command [EDI 4.2]..... 65
- Exit command [EDI 4.2] 65
- StringLength command [EBF 4.2.1] 65

CHAPTER 5	ECRTP	67
	ECRTP Web services API [EDI 5.1]	67
	Large file handling [EDI 5.1]	68
	Log bulk insert [EBF 4.2.4]	68
	JNI wrapper for ECRTP [EBF 4.2.3]	69
	PADEDI environment variable [EDI 4.2]	70
	RunMap options	70
	Character checks for compliance maps [EBF 4.2.5]	71
	Setting maximum waiting time for ECRTP to get a database connection [EBF 4.2.5]	72
	Controlling updates and lookups for control numbers [EBF 4.2.4]	73
	Validating the Interchange (ISA) and Group (GS) control number sequence [EDI 4.2]	75
	Override for compliance map results [EDI 4.2]	76
	Specifying map name extensions [EDI 4.2]	78
	-mns switch removed for ECRTP 4.2 [EDI 4.2]	78
	Return codes [EDI 4.2]	78
	NCPDP file processing [EDI 4.2]	79
	Literal numeric data mapping [EDI 4.2]	81
	Leading zeros in HIPAA compliance maps [EDI 4.2]	81
	ST control number validation	82
	Padding EDI data to minimum length [EDI 4.2]	83
	Outbound trading partner lookup option [EDI 4.2]	83
	Specifying maximum number of cached ODBC connections [EDI 4.2]	84
	Command line switches [EDI 4.2]	84
	ODBC switches [EDI 4.2]	87
APPENDIX A	Runtime Parameters and Switches	89
	Introduction	89
	Alphabetical listing of parameters and switches for ODBC and non-ODBC trading partner files	90
	Parameters and switches for ODBC trading partners only	95
	Required parameters	97
	Outbound required parameters	97
	Outbound required parameters for ODBC database users	98
	Inbound required parameters	99
	Inbound required parameters for ODBC database users	99
	Inbound/outbound optional parameters	100
	Inbound/outbound optional parameters for ODBC database users	103
	Outbound-only optional parameters	103
	Inbound-only optional parameters	104

	Inbound-only optional trading partner lookup switches	106
	Inbound-only optional trading partner lookup switches for ODBC database users	106
APPENDIX B	System Variables	109
	Alphabetical listing of ECMAP system variables	109
	Index	151

About This Book

Audience

This book is intended for customers who are installing and using Sybase® EDI Products version 5.1 and subsequent EBFs.

How to use this book

This book provides information about the features and system changes in EDI Products 5.1, EDI Products 5.0, and in EDI Products 4.2 and its subsequent EBF releases. For ease of reference, each new or updated feature is *tagged* with the main release or EBF version in which it is released. This book is organized into these chapters:

- Chapter 1, “Introduction,” provides general information about EDI Products, which is applicable to all the components in EDI Products.
- Chapter 2, “ECMAP,” describes the new and updated ECMAP™ features developed for the EDI Products 5.1 release, for EDI Products 5.0, and for EDI Products 4.2 and its subsequent EBFs.
- Chapter 3, “ECMAP Debugger,” contains a complete description of ECMAP Debugger and includes the new and updated ECMAP Debugger features developed for the the EDI Products 5.1 release. ECMAP Debugger is an add-on product which can help developers to debug maps.
- Chapter 4, “EC Gateway,” describes the EC Gateway™ features developed for EDI Products 4.2 and subsequent EBFs.
- Chapter 5, “ECRTP,” describes the new ECRTP™ features developed for the EDI Products 5.1 release, and for EDI Products 4.2 and its subsequent EBFs.
- Appendix A, “Runtime Parameters and Switches,” describes all the ECRTP parameters and switches available in EDI Products 4.2 and later.
- Appendix B, “System Variables,” describes all the system variables available in EDI Products 4.2 and later, including the new system variables introduced with EDI Products 5.0.

Related documents

The Sybase EDI Products documentation set consists of:

-
- *New Features Guide EDI Products 5.1* (this book) – describes the new features in EDI Products 5.1, EDI Products 5.0, EDI Products 4.2, and EDI Products 4.2 EBF releases.
 - *Installation Guide EDI Products 5.1* – describes system requirements and installation procedures for all EDI Products components.
 - *Release Bulletin EDI Products 5.1* – contains descriptions of change requests (CR) that have been addressed, documentation clarifications, and last-minute information that was too late to be included in the books or online help.
 - *ECMAP 4.2 Getting Started* – contains tutorials that provide a brief hands-on introduction to ECMAP.
 - *ECMAP 4.2 User Guide* – contains instructions for creating, generating, and running maps.
 - *ECMAP 4.2 Reference Guide* – contains information such as, database formats, informational messages, and EDI envelope types.
 - *ECRTP 4.2 Reference Guide* – describes ECRTP runtime parameters, and contains conceptual information and examples for map developers.
 - *ECRTP/TP 4.2 Reference Guide* – contains information on working with ECRTP/TP, which is a customized version ECRTP.
 - *EC Gateway 4.2 Reference Guide* for Windows and for UNIX – contains information on configuring and managing EC Gateway.
 - *EC Gateway Operations Viewer 4.2 Reference Guide* – describes how to use EC Gateway Operations Viewer to monitor operations of EC Gateway.

Other sources of information

Use the Sybase Getting Started CD, the SyBooks™ CD, and the Sybase Product Manuals Web site to learn more about your product:

- The Getting Started CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the SyBooks CD. It is included with your software. To read or print documents on the Getting Started CD, you need Adobe Acrobat Reader, which you can download at no charge from the Adobe Web site using a link provided on the CD.
- The SyBooks CD contains product manuals and is included with your software. The Eclipse-based SyBooks browser allows you to access the manuals in an easy-to-use, HTML-based format.

Some documentation may be provided in PDF format, which you can access through the PDF directory on the SyBooks CD. To read or print the PDF files, you need Adobe Acrobat Reader.

Refer to the *SyBooks Installation Guide* on the Getting Started CD, or the *README.txt* file on the SyBooks CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is an online version of the SyBooks CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Maintenance, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Sybase Product Manuals Web site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click Certification Report.
- 3 In the Certification Report filter select a product, platform, and timeframe and then click Go.
- 4 Click a Certification Report title to display the report.

❖ Finding the latest information on component certifications

- 1 Point your Web browser to Availability and Certification Reports at <http://certification.sybase.com/>.
- 2 Either select the product family and product under Search by Base Product; or select the platform and product under Search by Platform.
- 3 Select Search to display the availability and certification report for the selection.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.

-
- 2 Click MySybase and create a MySybase profile.

Sybase EBFs and software maintenance

❖ Finding the latest information on EBFs and software maintenance

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Maintenance. If prompted, enter your MySybase user name and password.
- 3 Select a product.
- 4 Specify a time frame and click Go. A list of EBF/Maintenance releases is displayed.

Padlock icons indicate that you do not have download authorization for certain EBF/Maintenance releases because you are not registered as a Technical Support Contact. If you have not registered, but have valid information provided by your Sybase representative or through your support contract, click Edit Roles to add the “Technical Support Contact” role to your MySybase profile.

- 5 Click the Info icon to display the EBF/Maintenance report, or click the product description to download the software.

Conventions

The following style conventions are used in this book:

- In a sample screen display, commands you should enter exactly as shown are in:

`this font`

- In a sample screen display, words that you should replace with the appropriate value for your installation are shown in:

this font

- In the regular text of this document, the names of files and directories appear in italics:

/usr/u/sybase

- The names of programs, utilities, procedures, and commands appear in this type:

mapinrun

The conventions for syntax statements in this manual are as follows:

Table 1: Syntax conventions

Key	Definition
command	Command names, command option names, utility names, utility flags, and other keywords are in a sans serif font, such as Arial.
<i>variable</i>	Variables, or words that stand for values that you fill in, are in italics.
{ }	Curly braces indicate that you choose at least one of the enclosed options. Do not include braces in your option.
[]	Brackets mean choosing one or more of the enclosed options is optional. Do not include brackets in your option.
()	Parentheses are to be typed as part of the command.
	The vertical bar means you can select only one of the options shown.
,	The comma means you can choose as many of the options shown as you like, separating your choices with commas to be typed as part of the command.

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.



Introduction

This chapter provides general information applicable to all the components in EDI Products 5.1.

Topic	Page
Sybase EDI Products	1
EDI Products 5.1 new features	2
Product licensing with SySAM 2.0 [EDI 5.0]	3

Electronic Data Interchange (EDI) is the computer-to-computer exchange of business documents in a structured, predefined, and standard format. The main purpose of EDI is to automatically read and process information between trading partners by establishing a standard data format. As a result, traditional processing delays and errors caused by human intervention can be eliminated.

EDI can be used to transmit documents such as invoices, purchase orders, health care claims, remittance advice, status inquiries, receipts, shipping documents, and other standard business correspondence between organizations. EDI can also be used to transmit financial information and electronic payments, which is generally referred to as electronic funds transfer. The presence of EDI becomes more and more widespread in today's market, especially with the growth of electronic commerce across the Internet. EDI improves the traditional way of exchanging information between trading partners, increasing productivity and efficiency.

Sybase EDI Products

Sybase EDI Products is a set of powerful components that meets the needs of any organization processing EDI: the ECMAP Development Workbench, ECRTTP runtime environment with trading partner management capabilities, and EC Gateway to support message management, Value Added Network (VAN) connectivity, and operations management.

The Sybase EDI Products 5.1 product set includes:

- ECMAP
- ECMAP Network Client
- ECMAP Debugger
- EC Gateway
- EC Gateway Operations Viewer
- ECRTTP
- ECRTTP/TP

The Sybase EDI Server 5.0 product bundle includes:

- ECMAP
- EC Gateway
- EC Gateway Operations Viewer
- ECRTTP

See the *EDI Products 5.1 Installation Guide* and the *EDI Products 5.1 Release Bulletin* for the latest information on system requirements.

In addition, for the latest supported platforms and databases for EDI products, see the Availability and Certification Reports Web site at <http://certification.sybase.com>.

Throughout this book, each new feature refers to the version or EBF in which it has been released.

EDI Products 5.1 new features

The new features in EDI Products 5.1 are:

- ECMAP Debugger – you can define projects and use session controls to enable you to create and record multiple debugging scenarios. You can also view the map flow and view EDI inbound and outbound segment data. See Chapter 3, “ECMAP Debugger” for more information.
- HL7 – you can use more versions of the HL7 message formats. The ECMAP Standards Repository supports all versions between 2.1 and 2.5 inclusive—2.1, 2.2, 2.3, 2.3.1, 2.4, and 2.5.

- Large file handling – you can use ECRTP to handle files larger than 2GB on ECRTP for Windows only.
- Web services API for ECRTP – you can use the ECRTP Web services API to create Web applications that call ECRTP. For example, a payer can create a Web application with the API for providers to use for real-time eligibility verification. See “Installing and starting the ECRTP Web service,” in Chapter 8, “Installing ECRTP for Windows” in the *EDI Products 5.1 Installation Guide* to install and deploy the ECRTP Web service.

Product licensing with SySAM 2.0 [EDI 5.0]

Sybase EDI Products 5.0 and later uses Sybase Software Asset Management (SySAM) for license administration and asset management tasks.

SySAM concepts

SySAM-enabled components in EDI Products 5.0 and later use SySAM to perform runtime verification of licenses, every time you start the product. If the required license is not available, the product does not run. This differs from the license verification in earlier versions, which was performed only during installation.

Note SySAM-enabled products work for an initial grace period without a license. If you do not install a valid license for your configuration within the grace period, the product ceases to function. The initial grace period is generally 30 days but for each install SySAM optionally generates an email message confirming the actual expiration date.

See Chapter 1, “Before You Begin” in the *EDI Products 5.1 Installation Guide* for more information on product licensing with SySAM.

ECMAP includes mapping tools that allow you to develop complex maps for processing EDI transactions. ECMAP supports sequential and delimited file formats and standards, which include ANSI X12, UN/EDIFACT, NCPDP, and HL7.

Changes in ECMAP include support for new features, new versions of EDI standards, new rules, and new RunMap options.

Topic	Page
Support for new versions of standards [EDI 5.1]	5
System variables [EDI 5.0]	6
EDI-to-EDI mapping [EDI 5.0]	6
Boolean logic [EDI 5.0]	23
Global repository [EDI 5.0]	28
Save map options [EDI 5.0]	32
Password security [EDI 5.0]	34
Read_Uncommitted option [EBF 4.2.5]	34
TA1 acknowledgements [EBF 4.2.4]	34
Rules	36
RunMap options	37
Compliance map acknowledgement options [EDI 4.2]	38
Cross-reference table import utility [EDI 4.2]	38
Record size increase [EBF 4.2.1]	38

Support for new versions of standards [EDI 5.1]

With EDI Products version 5.1, EDI adds support for HL7 versions 2.1, 2.2, 2.3, 2.3.1, 2.4, and 2.5. Table 2-1 lists all the EDI standards that EDI Products 5.1 supports.

Table 2-1: EDI 5.1 support for EDI standards

Standard	Version
EDIFACT	<ul style="list-style-type: none"> • D.91A • D.92A • D.95B • D.96A to D.01A • D.96B to D.01B
HL7	All versions from 2.1 to 2.5 inclusive
NCPDP	<ul style="list-style-type: none"> • 5.1 • 5.2 • 5.3 • 5.4 • 5.6
X12	All versions from 2001 to 5032 inclusive

System variables [EDI 5.0]

The set of system variables has been updated. See “Alphabetical listing of ECMAP system variables” on page 109 for an alphabetical listing of all system variables available in EDI Products 5.1.

New system variables for outbound EDI-to-EDI transactions have been added to the set. These have the “EDIOUT” as a suffix in their names. EDI-to-EDI mapping is a feature of EDI 5.0 and later. See “System variables for outbound EDI-to-EDI transactions” on page 13 for more information.

EDI-to-EDI mapping [EDI 5.0]

ECMAP includes an EDI-to-EDI mapping capability. You no longer need to perform a two-step process using flat files as an intermediate format.

With EDI-to-EDI, you can:

- Convert data from one version of a standard to another.

- Convert data from one standard to another. You can perform mapping between the EDIFACT, HL7, and X12 standards. You cannot perform EDI-to-EDI mappings with the NCPDP standard.
- Build a single map to process a query and response scenario. For example, you can build a map to take in an eligibility request and create the outgoing response. With EDI-to-EDI, only a single map is needed.
- Add information to a transaction and enrich the data that is received.

The Advance Shipment Notice (ASN) of a shipment contains information about the contents of a shipment. The radio frequency ID (RFID) tag of the shipment's pallet often contains additional information about the shipment. You can use EDI-to-EDI mapping to add information from the RFID tag to the original ASN.
- Map data elements from one map directly to other data elements in another map. This functionality allows ECMAP, for example, to be used in claim splitting and rejoining.
- Create more than one EDI output in a single EDI-to-EDI map. This functionality, for example, allows you to create both an X12 and XML output in one map.

Creating an EDI-to-EDI map

Creating an EDI-to-EDI map is similar to creating other types of maps. See “Task overview,” in Chapter 1, “ECMAP Overview” in the *ECMAP User Guide* for a comprehensive listing of all the steps involved in creating and running a map.

To create an EDI-to-EDI map, perform these additional steps:

- 1 Define an EDI-to-EDI map type and its properties. This involves:
 - a Selecting an EDI-to-EDI map type.
 - b Defining values for the inbound transaction.
 - c Creating the transaction for the outbound standard.
- 2 Select the segments for the map's inbound and outbound transactions.
- 3 Map the source EDI file to the destination EDI file.

Note You cannot run an EDI-to-EDI map as a compliance map.

Defining an EDI-to-EDI map

The procedure for defining an EDI-to-EDI map is similar to the procedure for defining other map types. See “Adding new maps,” in Chapter 3, “Working with Maps” in the *ECMAP User Guide* for detailed steps to create a map. There are several steps that are specific to EDI-to-EDI mapping, which are discussed in this section.

Once you have defined an EDI-to-EDI map, you can modify its properties in the Map Definition Properties window. See “Modifying a map,” in Chapter 3, “Working with Maps” in the *ECMAP User Guide*.

Selecting an EDI-to-EDI map type

In the New Map Definition window | Map Properties tab | Map Type field, select one of the EDI-to-EDI map types:

- EDIFACT to EDI
- HL7 to EDI
- X12 to EDI

For example, if the input is EDIFACT and the output is EDIFACT, HL7, or X12, select the EDIFACT to EDI map type.

Note When you select an EDI-to-EDI map type, the entry in the Direction field of the New Map Definition window automatically defaults to “IN” for inbound because all EDI-to-EDI maps are treated as inbound maps. There is no change in the rest of the fields in the Map Properties tab, Map Directories tab, and Map DSN tab of the New Map Definition window.

Defining the transaction for the outbound standard

You can add a maximum of 99 outbound EDI-to-EDI transactions in the EDI-to-EDI tab of the New Map Definition window.

❖ Creating outbound EDI-to-EDI transactions

- 1 Select the EDI-to-EDI tab to display the Outbound Standard Information view, which lists the outbound EDI-to-EDI transactions.
- 2 In the EDI-to-EDI tab:
 - a Select Add – to create new outbound transactions.

- b Select Properties – to modify the highlighted transaction.
 - c Select Delete – to delete an outbound EDI-to-EDI transaction. If you select this option, you must then confirm that you want to remove the transaction from the map.
- 3 In the Outbound Standard Options window, select the options for the outbound transactions that you want to create or modify:
 - Standard Type – select the EDIFACT, HL7, or X12 EDI standard types.
 - Transaction – type a valid transaction name or number.
 - Version – select a valid version for the standard you selected in the Standard Type field.
 - Standard Directory – enter, or browse for, the location of the standard database for the version you selected in the Version field.
 - Description – provide a description of the transaction. The description is important because you can use it to distinguish between several outbound transactions of the same kind, such as multiple 837s.
- 4 Click OK to save your outbound standard options to the database. The transaction code is also saved to the database. This information is used to create the outbound transactions later in the mapping process.

Developing the map

Once you have defined the map and added it to the list of maps in ECMAP, you can continue with the map development procedure outlined in “Task overview,” in Chapter 1, “ECMAP Overview” in the *ECMAP User Guide*. The major steps in the procedure are outlined in this section, along with more detailed additional steps that are specific to EDI-to-EDI mapping:

- 1 Set up a company ID. You can add multiple Company IDs if you have different outbound company profiles.
- 2 Set up a trading partner for the inbound EDI transaction.
- 3 Add a trading agreement for the trading partner you set up and add the EDI-to-EDI map to the inbound trading partner you have set up.
- 4 Set up a trading partner for the outbound EDI transaction. If the outbound transaction is for the same trading partner as the inbound transaction, skip this step.

- 5 Add a trading agreement for each trading partner you set up, and add the EDI-to-EDI map to the outbound trading partner you have set up.

Note You must have a trading agreement for each outbound transaction before you can set options, such as requesting an acknowledgement, on outbound transactions.

In the Trade Agreement - Properties window | Map Type drop-down list, select EDI. This distinguishes outbound EDI-to-EDI transaction from a non-EDI-to-EDI outbound transaction.

You need not enter the map name since it is not a part of the trade agreement lookup when defining an outbound EDI-to-EDI transaction.

- 6 Create the inbound and outbound transactions to specify to ECMAP the segment and elements that make up the data you are exchanging with the trading partner.
 - a In the Segments tab, select File | Make Transaction to create the inbound transaction.
 - b A dialog box displays asking if you would like to create another transaction. Select Yes to create the outbound transaction.
 - c In the Create Transaction window, you can toggle between the inbound and outbound transactions. Select View | Transactions. The list of defined transactions displays in a menu for you to select.
 - d Select an outbound transaction from the list. The transaction displays.
You can modify the segments for a transaction if needed.
 - e In the Segments tab, select File | Make Transaction to create the outbound transaction. You can create additional outbound transactions if you have defined them.

Note If you selected an outbound EDIFACT transaction in step 6c, a dialog box displays with the prompt “Is this EDIFACT Syntax 4?” when you select File | Make Transaction.

Select Yes if it is a Syntax 4 EDIFACT transaction. The UNB and UNG segments are created based on Syntax 4.

Select No if it is *not* a Syntax 4 EDIFACT transaction. The UNB and UNG segments are created based on Syntax 3.

- f When you have completed creating the outbound transaction, select No when you are prompted to create another transaction.
- 7 Build the map in the Mapping window that displays after you have created the inbound and outbound transactions.

Figure 2-1: EDI-to-EDI Mapping window

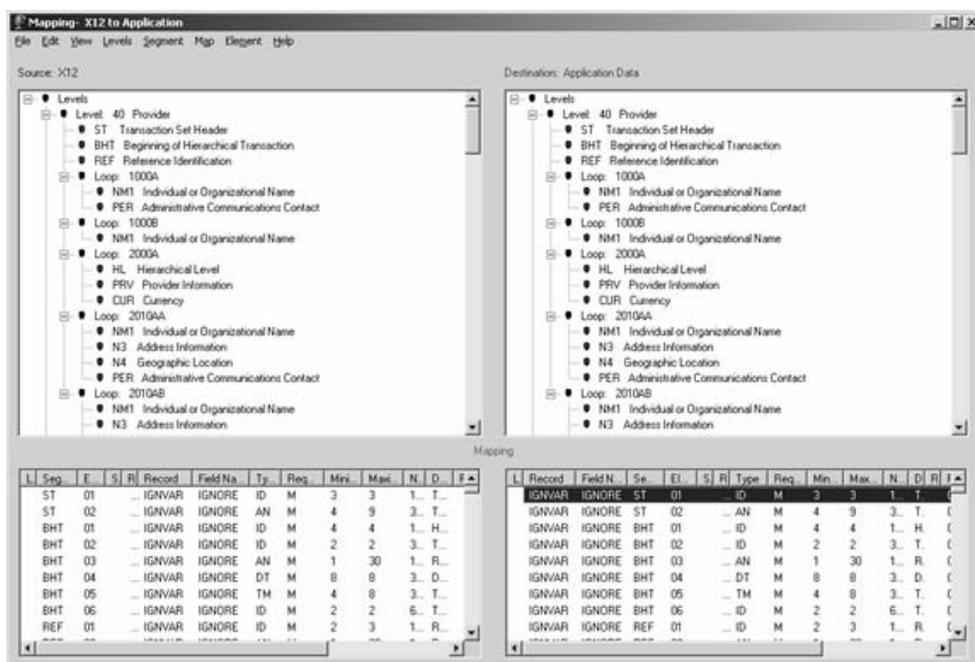


Figure 2-1 shows the EDI-to-EDI Mapping window. In the upper section of the Mapping window, you can view the source EDI file containing the inbound transaction on the left side pane and the destination EDI file containing the outbound transaction on the right side pane. You can map directly from the source EDI file to the destination EDI file.

For EDI-to-EDI maps, two mappings occur. These are for inbound and outbound element mappings. The Mapping window has been updated to display two panes at the bottom for inbound and outbound element mapping. The left pane is for inbound element mapping and the right pane is for outbound element mapping.

Double-click a segment in any of the transaction panes to view the elements belonging to that segment. Click a segment or element to view its properties in the bottom pane. If the segment or element is mapped, it displays the link, as a check mark, to the linked item on the opposite pane.

You can perform drag-and-drop mappings between inbound and outbound transactions. As with other types of maps, you can also perform drag-and-drop mapping with memory variables, records, and fields. You can add rules and cross-reference tables to elements in inbound and outbound transactions. You can use the Make Field option with both the inbound and outbound transactions.

Note To map an inbound element to multiple outbound transactions, make copies of the inbound element and map a copy to each outbound element you want to map to that particular inbound element.

You can also map the inbound element to a record/field or memory variable and map that record/field or memory variable to multiple outbound elements. This allows you to perform rules and execute commands on records/fields or memory variables, allowing you to manipulate data or perform other business operations.

You must associate levels with the transaction. Levels have several properties:

- Levels can be the same across all mapped transactions. You can use the same level number in the source and destination panes.
- For outbound EDI, you must map the envelope level segments such as ISA, GS, ST, SE, GE, and IEA to their own levels.

The options in the main menu across the top of the Mapping window change to reflect the current active pane—source or destination.

- 8 Build the map logic with rule definitions using rule commands, including the new rule commands for EDI-to-EDI mapping. These are described in EDI-to-EDI rule commands and system variables.
- 9 Create the map flow.
- 10 Generate the map.
- 11 Run the map.

EDI-to-EDI rule commands and system variables

ECMAP includes rule commands and system variables for EDI-to-EDI mapping:

- Company Lookup command – optional
- EDI Filename command – required
- EDI-to-EDI Run Switches command – optional
- Map Envelope Level command – required
- Map Level command – optional
- Trade Partner Lookup command – optional

You can use these commands to assign information to a value for a specific transaction, or for all transactions. This means you can use the same variable and it can contain different information for different transactions.

System variables for outbound EDI-to-EDI transactions

New system variables have been added for outbound EDI-to-EDI transactions. These variables are called EDIOUT system variables and they allow inbound and outbound EDI information to be kept separate. These variables use an “EDIOUT”, for example SYS_DATE_EDIOUT.

These system variables display in the System Variable window only when you work with EDI-to-EDI maps. The EDIOUT system variables are in “Alphabetical listing of ECMAP system variables” on page 109.

Company Lookup command

The Company Lookup command allows you to find the company entry using its profile number.

Select Company Lookup from the Rule Command - New window. In the Parameters section, select the variables for each of the parameters of the Company Lookup command by clicking the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays. Company Lookup has these parameters:

- Profile Number – type in the profile number of the profile assigned to the company to perform a lookup for a particular company profile. A company may have multiple profiles so that it can be recognized by different trading partners.

- Transaction – select this parameter to associate with this command either a specific transaction or all the transactions:
 - Select a single outbound transaction to associate with the command.
 - ALL – associate all outbound transactions with the command. This is the default.

EDI Filename command

The EDI Filename command allows you to associate file names with EDI output transactions. This command is required for EDI-to-EDI maps.

Note You must use this command before you execute the Map Envelope Level command for a transaction.

You must associate a unique file name with each transaction. Once the EDI file has been written to, you cannot change the file name.

If you want the output to go to the specified file, use the EDI-to-EDI Run Switches command and select the Ignore Trade Partner Mailbox and Ignore Trade Agreement Mailbox options. If you do not select these two options, normal mailbox routing occurs and the file is not written to the specified location with the specified name.

Select EDI Filename from the Rule Command - New window. In the Parameters section, select the variables for each of the parameters of the EDI Filename command by clicking the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays. This command has these parameters:

- Filename – select one of:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Filename text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the Filename text box.
 - System Variable – the System Variables window displays next to the Rule Command window. Double-click a system variable to automatically enter it in the Filename text box.

- Constant – enter a value in the text box on the Constant Value window that displays and select OK. The value is entered in the Filename text box.
- Transaction – associate a specific outbound transaction with this command.

Note The ALL option is not allowed for the Transaction parameter in this command.

EDI-to-EDI Run Switches command

The EDI-to-EDI Run Switches command allows you to set the outbound run switches associated with the outbound EDI transactions. This is an optional command.

You can turn switches on and off throughout a map run. For example, you can enable a switch for a short period of the map run and then turn it off when the switch is not required.

Select EDI-to-EDI Run Switches from the Rule Command - New window. In the Parameters section, select the variables for each of the parameters of the EDI-to-EDI Run Switches command by clicking the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays. This command has these parameters:

- Select Switch – parameter options are:
 - Company Override – allows you to override the location of the company directory. This is the same as the Substitute Company Directory option when you want to run outbound maps.

When you select Company Override, the Enter Override Value parameter field is enabled for you to provide a value for the substitute company directory using a variable or constant. See the Enter Override Value parameter description for more information. If you are running ECRTTP with a non-ODBC trading partner, use the switch to set a substitute directory.

Note Changing the Company Override field does not take effect until the next lookup occurs.

- Control Number Override – allows you to override the default value from ECRTP and disable control number lookups and updates in the trading partner database during outbound map runs. The value for control number is used for the outbound ISA.

When you select this option, the Enter Override Value parameter field is enabled for you to provide a control number value using a variable or constant. See the Enter Override Value parameter description for more information.

See “Controlling updates and lookups for control numbers [EBF 4.2.4]” on page 73 for information on the properties for the control number.

Note If you use the Control Number Override, you cannot turn it off during a map run as the On/Off option is disabled. If you change the Control Number Override field, it does not take effect until the next time you map the interchange.

- Ignore Trade Agreement Mailbox – when you select this option, ECRTP ignores trading agreement mailbox information.
- Ignore Trade Partner Mailbox – when you select this option, ECRTP ignores trading partner mailbox information.
- Map Numeric Zero – allows you write a zero to a numeric field.
- No UNG, UNE Segments – when you select this option, no UNG or UNE segments are written for an EDIFACT message.
- Pad EDI – checks that all data written to the EDI file meets the minimum length requirements of the element.
- Route EDI : In – routes the outbound file to the IN mailbox.
- Route EDI : Out – routes the outbound file to the OUT mailbox.
- Route EDI : Good – routes the outbound file to the GOOD mailbox.
- Route EDI : Bad – routes the outbound file to the BAD mailbox.
- Route EDI : Other – routes the outbound file to the OTHER mailbox.
- Enter Override Value – enabled when Company Override or Control Number Override options are selected in the Select Switch parameter field. You must use one of the following four options to provide the substitute value when Company Override or Control Number Override are selected:

- Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Enter Override Value text box.
- Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the Enter Override Value text box.
- System Variable – the System Variables window displays next to the Rule Command window. Double-click a system variable to automatically enter it in the Enter Override Value text box.
- Constant – enter a value in the text box on the Constant Value window that displays and select OK. The value is entered in the Enter Override Value text box.
- On/Off – allows you to enable and disable the switch you selected for the Select Switch parameter, any time in the map run.

Note If you use the Control Number Override, you cannot turn it off during a map run as the On/Off option is disabled.

- Transaction – select this parameter to associate with this command either a specific transaction or all the transactions:
 - Select a single outbound transaction to associate with the command.
 - ALL – associate all outbound transactions with the command. This is the default.

Map Envelope Level command

The Map Envelope Level command is for inbound EDI-to-EDI mapping only. This command allows you to map the envelope level segments for EDIFACT, HL7, and X12 outbound EDI transactions.

Note You must execute the command for each envelope level segment because each of these segments must be defined on its own level for mapping to occur.

Select Map Envelope Level from the Rule Command - New window. In the Parameters, select the variables for each of the parameters of the Map Envelope Level command by clicking on the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays. This command has these parameters:

- Envelope Type – the different envelope level segment options are:
 - Interchange
 - Group
 - Transaction
- Segment Type – the two segment options are:
 - Header
 - Trailer
- Level – the Levels window displays next to the Rule Command window. Double-click a level to automatically enter it in the Level text box.
- Control Number Options – the options are:
 - Keep Current Control Numbers and Update Database – keep the control numbers from the inbound transaction and update the outbound transaction’s trade partner entry, in the database, with these values.
 - Keep Current Control Numbers with No Database Update – keep the control numbers from the inbound transaction with no database update.
 - Get Control Numbers from Database – obtain the control numbers from the outbound transaction’s trade partner tables.
- Update ALL Trade Partner Record – the options are:
 - Yes – obtain the control numbers from the ALL trade partner record instead of obtaining an exact trade partner match.
 - No – control numbers are not obtained from the ALL trade partner.
- Status Variable – specify whether the Map Envelope Level command created the envelope successfully.

The only option you can select for this parameter is Memory Variable. The Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Status Variable text box.

If envelope creation is not successful, an “N” is returned in the memory variable you selected. You can execute a separate series of commands and try the Map Envelope Level command again.

- Transaction – select this parameter to associate with this command either a specific transaction or all the transactions:
 - Select a single outbound transaction to associate with the command.
 - ALL – associate all outbound transactions with the command. This is the default.

Map Level command

The Map Level command performs at one time all of the rules and mapping associated with the segments in a specified level. The Map Level command is enabled for inbound EDI-to-EDI mapping only.

Assign a level to one or more segments that you want to have process as a unit. This saves you from having to perform a write rule for each EDI element.

When you execute Map Level, all the segments and elements you associated with the specified level are mapped and any rule associated with an element is performed.

You can use the same level for a single output transaction or multiple output transactions.

Select Map Level from the Rule Command - New window. In the Parameters section, select the variables for each of the parameters of the Map Level command by clicking the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays. This command has these parameters:

- Level – the Levels window displays next to the Rule Command window. Double-click a level to automatically enter it in the Map Level text box.

If you have not yet created the level, you can do so on the Levels window by choosing New from the File menu. See “Adding, modifying and deleting levels,” in Chapter 15 “Creating a Map Flow” in the *ECMAP User Guide*.

- Transaction – select this parameter to associate with this command either a specific transaction or all the transactions:
 - Select a single outbound transaction to associate with the command.
 - ALL – associate all outbound transactions with the command. This is the default.

Trade Partner Lookup command

The Trade Partner Lookup command allows you to find the trade partner for the outbound transaction. This may be the same trade partner you have for the inbound transaction or it may be a different trade partner. This is an optional command.

You can execute the Trade Partner Lookup command at any time and for any number of times. The Trade Partner Lookup command executes only after you execute the Map Envelope Level command. Thus, only the most recently executed instance of the command takes effect for the trade partner you want to lookup.

Table 2-2 summarizes how the Trade Partner Lookup command works for different EDI outbound transaction types.

Table 2-2: Trade Partner Lookup for different EDI transaction types

EDI outbound transaction type	Lookup performed
EDIFACT	When the Map Envelope Level command is executed for the UNA segment. If the UNA segment is not used, the lookup occurs when the command is executed on the UNB segment.
HL7	When the Map Envelope Level command is executed for the FHS segment for “long” HL7 or for the MSH segment for “short” HL7.
X12	When the Map Envelope Level command is executed for the ISA segment.

Select Trade Partner Lookup from the Rule Command - New window. In the Parameters section, select the variables for each of the parameters of the Trade Partner Lookup command by clicking the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays. This command has these parameters:

- Trade Partner ID – select an option that contains the Trade Partner ID:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Trade Partner ID text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the Trade Partner ID text box.
 - System Variable – the System Variables window displays next to the Rule Command window. Double-click a system variable to automatically enter it in the Trade Partner ID text box.

- All Trade Partner Default – parameter options are:
 - Yes – ECRTP uses the default ALL Trade Partner entry if no trade partner match can be found.
 - No – ECRTP does not use the default ALL Trade Partner entry if no trade partner match can be found.
- Lookup Trade Partner from ECRTP Log – this parameter is used for TA1 maps. This is an optional parameter. The parameter options are:
 - Yes – ECRTP queries the log for the information necessary to locate the trade partner entry.
 - No – ECRTP does not query the ECRTP log for the trade partner entry. This is the default option.
- Map Name Extension – select an option that contains the map name extension. This is an optional parameter you can select if you have used map name extensions to distinguish multiple maps for the same transaction within a single trade agreement. The options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Map Name Extension text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the Map Name Extension text box.
 - System Variable – the System Variables window displays next to the Rule Command window. Double-click a system variable to automatically enter it in the Map Name Extension text box.
 - Clear – select this option to remove the Memory Variable, Record/Field, or System Variable options that you selected previously for the Map Name Extension parameter.
- ST03 (X12) – select an option that contains the ST03 element. This is an optional parameter you can select if you have used the ST03 element to distinguish multiple X12 maps for the same transaction within a single trade agreement. The options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the ST03 (X12) text box.

- Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the ST03 (X12) text box.
- System Variable – the System Variables window displays next to the Rule Command window. Double-click a system variable to automatically enter it in the ST03 (X12) text box.
- Clear – select this option to remove the Memory Variable, Record/Field, or System Variable options that you selected previously for the ST03 (X12) parameter.
- Transaction – select this parameter to associate with this command either a specific transaction or all the transactions. The options are:
 - Select a single outbound transaction to associate with the command.
 - ALL – associate all outbound transactions with the command. This is the default.

Updated reports

Reports in ECMAP have been updated for the EDI-to-EDI mapping feature and the new and updated rule commands. These reports now include the outbound EDI-to-EDI transactions. Access them from the Reports menu of the main ECMAP window:

- Project.
- Application Record Cross-Reference.
- Additional Cross-Reference Reports.
- Flow Detail.
- Map Implementation Guide – for either inbound or outbound transactions.
- Map Quick Reference – for either inbound or outbound transactions.
- Map Segment Summary – for either inbound or outbound transactions.
- Map Detailed Definition – for either inbound or outbound transactions.
- Element Rule Summary.
- Year 2000 Date Cross-Reference reports.
- Year 2000 Date Listings reports.

Boolean logic [EDI 5.0]

ECMAP and EC RTP have been enhanced with Boolean logic to simplify the handling of the complex logic in rules.

By using the enhanced Boolean logic, you can reduce the number of nested If Then statements to only one. Maps are easier to build, maintain, and debug; and the processing speed of ECMAP is improved.

The enhanced Boolean logic includes a new datatype, new commands, and updates to current commands.

Boolean datatype

A Boolean datatype has been added to ECMAP. The allowed values for this datatype are True or False.

You can select the Boolean datatype for a field or for a memory variable:

- Boolean datatype for fields – in the Field Properties window, select B - Boolean from the Field Type drop-down list.
- Boolean datatype for memory variables – in the Memory Variables Properties window, select B - Boolean from the Field Type drop-down list.

In the Field Properties and Memory Variables Properties windows, set the Field Length field value to 1 (one) as the Boolean datatype is defined with a length of one.

New Boolean rule commands

The Boolean Arithmetic and Boolean Assignment commands enable you to use the enhanced Boolean logic in ECMAP.

Boolean Arithmetic command

The Boolean Arithmetic command supports “And” and “Or” operators. You must use the Boolean datatype for all variables in this command.

Select Boolean Arithmetic from the Rule Command - New window. In the Parameters section, select the variables for each of the parameters of the Boolean Arithmetic command by clicking the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays.

- Left – options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Left text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the Left text box.
- Operator – options are:
 - And – click And to enter it in the Operator text box.
 - Or – click Or to enter it in the Operator text box.
- Right – options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Right text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the Right text box.
 - Boolean Constant – select True or False. The option you select is entered in the Right text box.
- Destination – options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Destination text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the Destination text box.

Boolean Assignment command

The Boolean Assignment command supports operations such as $A=B>C$ and $A=B<C$. This allows you to run logic over a series of commands instead of a series of If Then statements. The variable in the Destination parameter must be a Boolean datatype.

Select Boolean Assignment from the Rule Command - New window. In the Parameters section, select the variables for each of the parameters of the Boolean Assignment command by clicking the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays.

- Left – options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click or drag and drop a memory variable to enter it in the Left text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click or drag and drop a field to enter it in the Left text box.
 - System Variable – the System Variables window displays next to the Rule Command window. Double-click a system variable to automatically enter it in the Left text box.
- Operator – options are:
 - EQ – enters the Equal To operator in the Operator text box.
 - NE – enters the Not Equal To operator in the Operator text box.
 - GT – enters the Greater Than operator in the Operator text box.
 - GE – enters the Greater Than or Equal To operator in the Operator text box.
 - LT – enters the Less Than operator in the Operator text box.
 - LE – enters the Less Than or Equal To operator in the Operator text box.
- Right – options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable to automatically enter it in the Right text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click a field to automatically enter it in the Right text box.
 - System Variable – the System Variables window displays next to the Rule Command window. Double-click a system variable to automatically enter it in the Right text box.

- Constant – enter a value in the text box on the Constant Value window that displays and select OK. The value is entered in the Right text box.
- Destination – options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click or drag and drop a memory variable to enter it in the Destination text box.
 - Record/Field – the Records/Tables window displays next to the Rule Command window. Double-click or drag and drop a field to automatically enter it in the Destination text box.

Table 2-3 lists the possible expressions you can use with the Boolean Assignment command and the corresponding datatypes you can use for the Left and Right operands and the Destination.

Table 2-3: Datatypes used for Boolean Assignment command

Expression	Left operand	Right operand	Destination
B>C, B<C, B>=C, B<=C, B==C, B!=C	Alphanumeric	Alphanumeric	Boolean
B>C, B<C, B>=C, B<=C, B==C, B!=C	Date	Date	Boolean
B>C, B<C, B>=C, B<=C, B==C, B!=C	Time	Time	Boolean
B>C, B<C, B>=C, B<=C, B==C, B!=C	Decimal Explicit, Decimal Implicit, Packed Decimal, Real, Signed Decimal, Signed Integer, and Unsigned Integer	Decimal Explicit, Decimal Implicit, Packed Decimal, Real, Signed Decimal, Signed Integer, and Unsigned Integer	Boolean
B==C	Boolean	Boolean	Boolean
B!=C	Boolean	Boolean	Boolean

Updated rule commands

The rule commands that have been updated for the Boolean constant option are the Assignment, If Then, and Perform Rule commands.

Assignment command

Use the Assignment command to assign a constant value or the value of a field or variable to another field or variable.

❖ Using the Boolean constant in the Assignment command

- 1 In the Rule Command - New window, select the Assignment command.
- 2 In the Parameters section, select the Source parameter by clicking the up arrow to the right of the Source parameter text box. A menu displays the Memory Variable, Record/Field, System Variable, Constant, Parameter, and Boolean Constant options.
- 3 Select the Boolean Constant option.
- 4 Select the True or False option from the menu that displays when you click Boolean Constant.

If Condition command

Use the If Condition command to enter simple “If ... Then” and “If ... Then ... Else” logic into rules. This command also allows you to perform a specific rule if an error occurs which would be written to the log file or the trace file. After executing any If Condition command, the program always proceeds to the next command line in the rule.

❖ Using the Boolean constant in the If Condition command

- 1 In the Rule Command - New window, select the If Condition command.
- 2 In the Parameters section, select the Right parameter by clicking the up arrow to the right of the Right parameter text box. A menu displays the Memory Variable, Record/Field, System Variable, Constant, Error Condition, and Boolean Constant options.
- 3 Select the Boolean Constant option.
- 4 Select the True or False option from the menu that displays when you click Boolean Constant.

Perform Rule command

The Perform Rule command executes a specified rule number either one time or multiple times, based on a condition. Thus, by allowing one rule to call, or perform, another rule, you can reduce the number of commands that need to be entered.

❖ **Using the Boolean constant in the Perform Rule command**

- 1 In the Rule Command - New window, select the Perform Rule command.
- 2 In the Parameters section, select the Right parameter by clicking the up arrow to the right of the Right parameter text box. A menu displays the Memory Variable, Record/Field, System Variable, Constant, Error Condition, and Boolean Constant options.
- 3 Select the Boolean Constant option.
- 4 Select the True or False option from the menu that displays when you click Boolean Constant.

Global repository [EDI 5.0]

ECMAP can share code among all maps using a global repository.

With a global repository, you can:

- Use the same code among all maps without having to type in the same code for every map. This helps make the maps easier to maintain.
- Specify the same repository across all maps, or have multiple repositories.
- Save rules, records, or memory variables to a repository.
- Import rules, records, or memory variables from a repository.

A repository has several important properties:

- Although a repository can be associated with a map or multiple maps, it is not map specific.
- If you update a repository, it does not affect other maps that have used that repository.
- All maps can access updates made to a repository and use these updates.
- To update a map from a specific repository, enable that repository for the map and select the updates you want to import to your map.

Enabling the repository

You must enable a global repository for a map before you create or access a repository.

Enable the repository by providing the directory location for the repository in the Global Repository field on the Map Directories tab, in either the New Map Definition window, or the Map Definition window.

Note All export and import options using the repository are disabled if you do not enable it in the Map Definition window.

Importing and exporting rule definitions

Use the Rule Definitions window to import and export rule definitions using the global repository.

Note Enable the repository before importing or exporting rule definitions using the repository.

❖ Importing rule definitions from the global repository

The procedure for importing rule definitions from the global repository is similar to the procedure for importing rules from another map. See “Importing rules,” in Chapter 20, “Working with Rules” in the *ECMAP User Guide*.

- 1 In the Rule Definitions window, select Rule | Import | From Repository.
- 2 Select the rule you want to import from the list on the left.
- 3 In the Import Rule Options section of the window, select one of the following options:
 - Overwrite All Duplicate Rules
 - Do Not Overwrite Duplicate Rules
 - Overwrite or Renumber Duplicate Rules
 - Renumber All Rules
- 4 Select Import.

❖ **Exporting rule definitions to the global repository**

The procedure for exporting rule definitions to the global repository is similar to the import procedure.

- 1 In the Rule Definitions window, select Rule | Export to Repository. The destination map is automatically set to the global repository and cannot be modified. The Export Rules to Repository window displays.
- 2 Select the rule you want to export from the list of rules in the current map.
- 3 In the Export Rule Options section of the window, select one of the following options:
 - Overwrite All Duplicate Rules
 - Do Not Overwrite Duplicate Rules
 - Overwrite or Renumber Duplicate Rules
 - Renumber All Rules
- 4 Select Export.

Importing and exporting record definitions

With the global repository feature, you can export and import record definitions.

Note Enable the repository before importing or exporting record definitions using the repository.

❖ **Importing record definitions from the global repository**

The procedure for importing record definitions from the global repository is similar to the procedure for importing record definitions from a map. See “Importing a record definition from a map,” in Chapter 7, “Importing Definitions” in the *ECMAP User Guide*.

- 1 In the main ECMAP window, select Application Files | Import Definition | From Repository. The Records/Tables window and the Import Records from Repository window display side by side. All the records in the repository are listed in the Import Records window.
- 2 Drag a record that you want to import from the Import Records from Repository window and drop it in the Records/Tables window. The current map now includes the imported record definition.

❖ Exporting record definitions to the global repository

The procedure for exporting records definitions to the global repository is similar to the import procedure.

- 1 In the main ECMAP window, select Application Files | Export Definition | Repository. The Records/Tables window and the Export Records window display side by side. All the records in the map are listed in the Records/Tables window.
- 2 Drag a record that you want to export from the Records/Tables window and drop it in the Export Records window. The repository now includes the exported record definition.

Importing and exporting memory variables

The Memory Variables window includes two options to import and export memory variables using the global repository.

Note Enable the repository before importing or exporting memory variables using the repository.

❖ Importing memory variables from the global repository

The procedure for importing memory variables from the global repository is similar to the procedure for importing memory variables from a map. See “Importing memory variables from another map,” in Chapter 18, “Working with Memory Variables” in the *ECMAP User Guide*.

- 1 In the Memory Variables window, select Options | Import | From Repository.
- 2 Select a memory variable from the Import Memory Variables - Repository window and drag and drop it to the Memory Variables window. The current map now includes the imported memory variable.

❖ Exporting memory variables to the global repository

The procedure for exporting memory variables to the global repository is the reverse of the import procedure.

- 1 In the Memory Variables window, select Options | Export to Repository.

- 2 Select a memory variable from the Memory Variables window and drag and drop it to the Export Memory Variables - Repository window. The repository now includes the exported memory variable.

Managing the repositories

You can manage the rule, record, or memory variable repositories from the main ECMAP menu. Select View | Repository and then select Rules, Records, or Memory Variables to manage the individual repositories.

Note Enable the repositories before viewing and editing any of them. The View Repository options are disabled if you did not enable the repositories.

Within the Rule Repository, Records/Tables Repository, and Memory Variables Repositories windows, you can add, delete, and modify rules, records, and memory variables respectively but you cannot perform imports and exports. You can drag and drop rules, records, and memory variables between the current map and the repositories.

Save map options [EDI 5.0]

The Save and Save As options allow you to control which edits you want to save in the map you are working on. The Map Recovery option allows you to recover modifications to your map if there was an abnormal exit from ECMAP.

See “Adding new maps” and “Working with maps,” in Chapter 3, “Working with Maps” in the *ECMAP User Guide* for more details about the Map Definition window and creating and modifying maps.

Save option

The Save option allows you decide when you want to save changes you make when creating or modifying a map. With the Save option, when you create or modify a map, the changes you make are not automatically saved each time you make a change. Instead, select File | Save in the main ECMAP window to save your changes.

If you create or modify a map and do not select the File | Save option in the main ECMAP window, you are prompted to save the map when you exit ECMAP, when you reselect the same map, or when you select another map to work on. If you do not use the Save option all modifications since the last save are lost.

The Save option enables you to control when you want to save changes to maps and allows you to revert to the last saved version of the working map without having to recall all the edits you made since the last save and manually removing the unwanted edits.

Save As option

The Save As option allows you make a copy of the map you are working on, under a different map name. Select File | Save As in the main ECMAP window to use this option. Once you click Save As, the Map Definition - Save As window is displayed, containing the Map Properties, Map Directories, and Map DSN tabs allowing you to change the properties of the map. Change the directories listed in the Map Directories tab since you are creating a new map.

Map recovery option

The Map recovery option is executed if an abnormal exit of ECMAP occurs or if another user is attempting to modify the map you are working on. If these occur, you see this message:

```
Changes were found that may not have been saved. Either
an abnormal exit from ECMAP occurred or another user is
modifying the map.
```

- 1 Click OK on the message dialog box.
- 2 When you see “Would you like to save the changes to map **Project/Map?**” you can select Yes, No, or Cancel. This refers to the current working map.
- 3 If you click Yes, you see, “Would you like to create a backup of the previously saved map?”. Select Yes to create a backup of the last saved copy of the map.
- 4 If you click Yes, you are asked if you want to create a backup of the current map.

Password security [EDI 5.0]

Password security in EDI has been updated:

- Passwords in all windows and dialog boxes in ECMAP and EC Gateway display either the encrypted form of the password or asterisks.
- Maps and scripts store only encrypted passwords.
- ECRTP and EC Gateway runtime decrypt all encrypted passwords prior to their use.
- ECRTP does not write the DSN string and password to trace files.

Note Batch command files display asterisks in place of the DSN password.

The comment “DSN passwords are written as **** and must be entered with the correct password” will be added to the batch file.

Replace the asterisks in the batch file with the actual password before running the batch file.

You can now enter user IDs and passwords in separate User ID and Password fields instead of entering them as a plain text string in the Data Source Name field in ECMAP or EC Gateway.

Read_Uncommitted option [EBF 4.2.5]

The ECMAP GUI has been updated with a Read_Uncommitted option in the drop-down list of File Properties. This option enables you to specify a lower isolation level that reduces locking contention for reading from the database. It is used by ECRTP to reduce the locking contention for certain fields such as control numbers and run IDs.

TA1 acknowledgements [EBF 4.2.4]

This section describes enhancements to EDI Products that enable real-time processing of TA1 acknowledgements. The following maps have been added: the TA1XLG and LOGUPDATE maps.

TA1XLG map [EBF 4.2.4]

A TA1 map, named TA1XLG, has been added to ECMAP. This map is based on the original TA1 maps, TA1ODBCA and TA1ODBCT. TA1XLG reads log messages from an expanded text log instead of from an ODBC database. The other TA1 maps use ODBC. You set up the TA1XLG map as you would any other, through the Run Map screens in either ECMAP or EC Gateway.

This map was added to support faster real-time transaction processing by enabling the writing of log records to a memory buffer, instead of to disk. Writing to memory requires the records be written in expanded text log format. As a result, users can now run a TA1 map against these records stored in a memory buffer. As a final step, these log records should be written to disk by running the LOGUPDATE log persistence map.

LOGUPDATE map [EBF 4.2.4]

A log persistence map, named LOGUPDATE, has been built to take an in-memory ECRTP log, in expanded text log format, and write it to a log on disk, either in an ODBC database or an expanded text log file.

This map allows the logging to disk to be delayed until the real-time critical processes have been completed. These include running the HIPAA compliance map and optionally the TA1 acknowledgement map, the 997 acknowledgement map, business rules compliance map, translation map and, if a query/reply transaction, other maps associated with the reply process.

Table 2-4 describes the parameters for the LOGUPDATE map.

Table 2-4: Parameters for LOGUPDATE map

Parameter	Description
1	<p>If this parameter is set to Y, ECRTP writes the log to an expanded text log disk file. If this parameter is empty, the log is written to a database.</p> <p>The DSN used for the log is 997. Follow the same instructions associated with running the 997 Acknowledgement map to correctly run this map.</p>
2	<p>If this parameter is set to Y, ECRTP updates the log to indicate a 997 acknowledgement has been sent.</p>
3	<p>If this parameter is set to Y, ECRTP updates the log to indicate a TA1 acknowledgement has been sent.</p> <p>If the inbound file had an error in the Interchange Acknowledgement Code (ISA14) or the sender ID (ISA06) is unknown or invalid, the log is marked with an error indicating the TA1 could not be sent. An E appearing in the TR_ACK_TYP column in the TRLOG table reflects this error.</p>

Rules

Rules are commands that govern the way ECMAP processes your maps. Version 4.2 includes rules that allow you to specify the location of incoming binary data and import rules from other maps.

Binary Data Placement command [EDI 4.2]

The Binary Data Placement command allows you to override the default location for the binary data of an incoming BIN segment by specifying the full path for the data to be stored.

Select Binary Data Placement from the Rule Command - New window. In the Parameters section, select the variables for each of the parameters of the Binary Data Placement command by clicking the up arrow to the right of each Parameter text box and selecting one of the options in the menu that displays.

- Binary Data Location – options are:
 - Memory Variable – the Memory Variables window displays next to the Rule Command window. Double-click or drag and drop a memory variable to enter it in the Binary Data Location text box.

Note You must define the memory variable as a directory and file location.

- Select File and Location – the Directories (Mailboxes) window displays next to the Rule Command window. Drag and drop a file to enter it in the Binary Data Location text box.

Enhanced rules import [EDI 4.2]

The rule import feature has been enhanced to enable you to import multiple rules from another map and to specify how to handle duplicate rule numbers during the import. To use this feature, in the Rule Definitions window, select Rule | Import to open the Import Multiple Rules window. In the Import Rules Options section, you can:

- Overwrite all duplicate rules
- Skip all duplicate rules
- Renumber all incoming duplicate rules
- Overwrite, skip, or renumber rules on a rule-by-rule basis

For more information, see “Creating and using rules” in Chapter 20, “Working with Rules,” in the *ECMAP User Guide*.

RunMap options

RunMap options apply to ECRTP, ECMAP, and EC Gateway. Updates to the RunMap dialog box in ECMAP and EC Gateway reflect additional runtime parameters and switches provided by ECRTP. For additional information about these new options and parameters, see “RunMap options” on page 70. You can also see Chapter 4, “Working with the Address Book,” and Chapter 17, “Running Maps,” in the *ECMAP User Guide*.

Compliance map acknowledgement options [EDI 4.2]

Trading partner options allow you to specify how you want ECMAP to issue an acknowledgement if it cannot find a trade agreement match:

- Acknowledge based on trade agreement (default behavior)
- Acknowledge if no trade agreement match is made
- Always acknowledge

See “Adding a new trading partner,” in Chapter 4, “Working with the Address Book,” in the *ECMAP User Guide*.

Cross-reference table import utility [EDI 4.2]

A new utility allows you to import cross-reference tables from one map to another. This reduces the amount of work required to reuse cross-reference tables for multiple maps.

See “Importing cross-reference tables,” in Chapter 22, “Working with Cross-Reference Tables in the *ECMAP User Guide*.

Record size increase [EBF 4.2.1]

The default record size has been increased from 32,767 bytes to 99,999 bytes.

Applies to	ECMAP and ECRTTP
ECMAP version	4.2.1
Resolves	CR#314959

ECMAP Debugger

ECMAP Debugger is an add-on product to ECMAP that runs with EDI 5.0 and later on Windows.

ECMAP Debugger lets developers control the execution of maps, on a command by command basis, to monitor the data values of variables and record fields, the map flow, and the EDI input and output segments as the map runs.

ECMAP Debugger can help new map developers learn ECMAP and help experienced map developers deepen their understanding of the product.

Topic	Page
ECMAP Debugger features [EDI 5.1]	39
Accessing ECMAP Debugger	40
ECMAP Debugger main window [EDI 5.1]	42
Preparing to debug maps [EDI 5.1]	46
Running an ECMAP Debugger session	54

ECMAP Debugger features [EDI 5.1]

ECMAP Debugger features include:

- Runtime logic checking – set and remove breakpoints and step through the workflow built into maps.
- Real-time data view – track real-time data values in memory variables, system variables, and record/fields.
- Session control – create and manage multiple projects that retain information about the debugging sessions you run through ECMAP Debugger. You can also create and manage multiple sets of switches and breakpoints in each map debug session. This allows you to re-run debug sessions with the same or different settings the next time you use ECMAP Debugger to investigate different map configurations.

- Real-time map flow view – track the map flow as the map runs and understand how the map flow, and rules and commands are related.
- Viewing EDI data – view detailed information on the map you are debugging and real-time information on segments.

Note ECMAP Debugger works with maps that have one input and one output transaction. You cannot debug an EDI-to-EDI map, which has multiple output transactions. See “EDI-to-EDI mapping [EDI 5.0]” on page 6.

Accessing ECMAP Debugger

Access ECMAP Debugger by running the batch file or clicking the icons installed on a Windows platform. See the *EDI Products 5.1 Installation Guide* for instructions on installing ECMAP Debugger.

Accessing ECMAP Debugger in Windows

- 1 Before you access ECMAP Debugger, modify these three variables in the *mapdebug.bat* batch file that was installed together with ECMAP Debugger for Windows:
 - JAVA_HOME – directory where JRE or JDK is installed.

Note Sybase recommends that you use Java 2 SDK version 1.5 or later.

 - ECRTP_HOME – directory where ECRTP shared library is installed.
 - MAPDEBUG_HOME – directory where ECMAP Debugger is installed.
- 2 You can access ECMAP Debugger in Windows in different ways:
 - Double-click the ECMAP Debugger icon on your desktop.
 - Select the ECMAP Debugger icon from the ECMAP folder in the program menu on your desktop.

- Run the batch file installed in Windows:
`%MAPDEBUG_HOME%\bin\mapdebug.bat`
where `%MAPDEBUG_HOME%` is the directory where ECMAP Debugger is installed.

Warning! When you double-click the ECMAP Debugger icon or run *mapdebugger.bat* from a command prompt, a Command Prompt window opens first, followed by ECMAP Debugger.

Do not close the Command Prompt window while ECMAP Debugger is open. The Command Prompt window closes when you exit ECMAP Debugger.

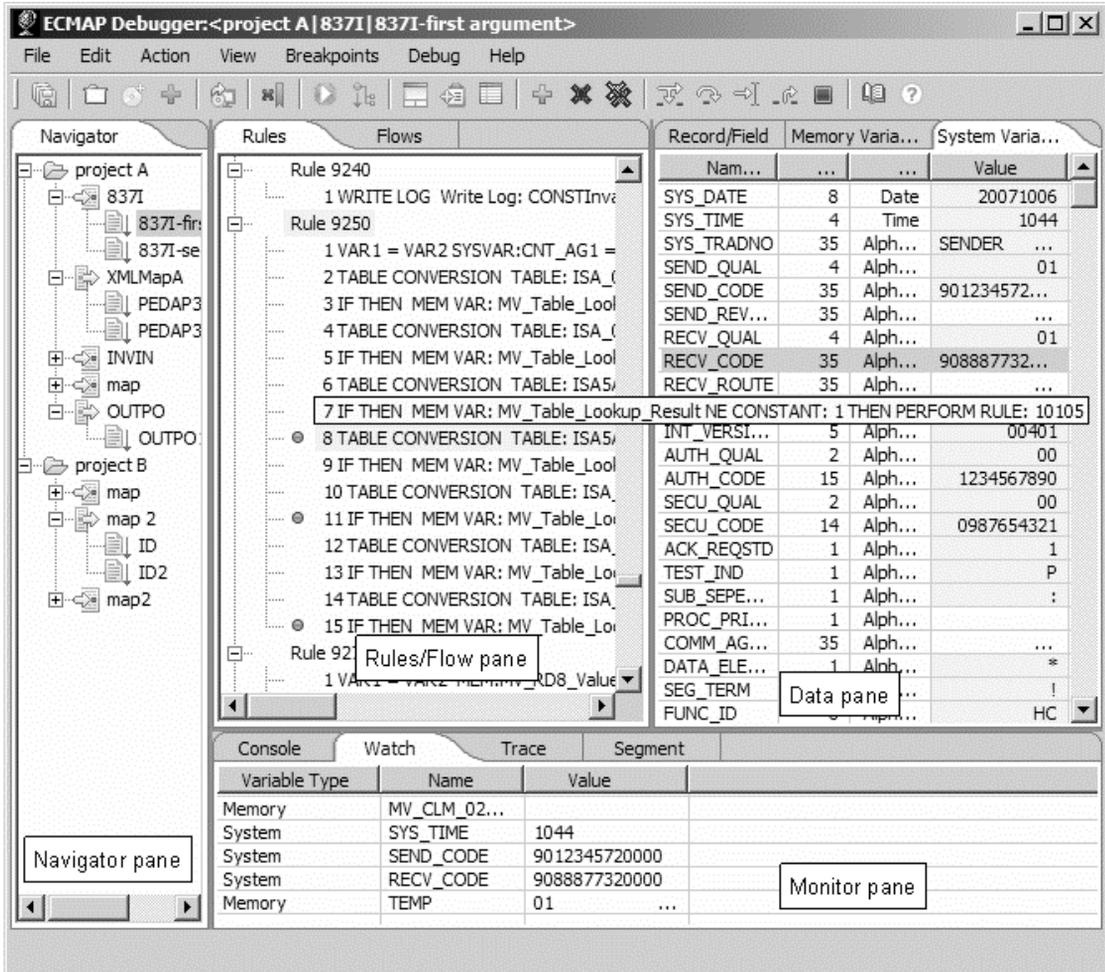
The Command Prompt window displays messages as you perform actions in ECMAP Debugger. You can save these messages to a text file to help you to troubleshoot any problems with ECMAP Debugger, by using the redirection option in DOS when you start ECMAP Debugger. For example, to start ECMAP Debugger and save Command Prompt window messages to the *console.txt* file, enter:

```
%MAPDEBUG_HOME%\bin\mapdebug.bat > console.txt
```

ECMAP Debugger main window [EDI 5.1]

The ECMAP Debugger main window has a main menu and four panes to track data values, and control and monitor the debug process. You can also use the buttons in the toolbar to run main menu commands.

Figure 3-1: ECMAP Debugger main window



The Navigator pane in the left of the ECMAP Debugger window displays in a tree structure—the projects, map debugs sessions, and map switch sets that you create:

- Projects – display as folders at the root level of the Navigator pane. Projects contain the map debug sessions that you run.
- Map debug sessions – display as icons indented one level inside project folders. An inbound map debug session icon has an arrow coming in from the left and an outbound map debug session icon has an arrow going out to the right. Map debug sessions correspond to maps you create in ECMAP and can contain multiple sets of map switches that you can debug.
- Map switch sets – display as document icons with a down arrow. Each map switch set contains the map arguments that you debug in each map debug session.

The title bar displays the names of the current project, map debug session, and map switch set loaded into ECMAP Debugger.

The Rules/Flow pane in the center of the ECMAP Debugger window and to the right of the Navigator pane, displays the rules, commands, and flow of the map you debug:

- Rules tab – the map rules display in the Rules tab. You can see the rules listed with a “+” beside each rule, which you can double-click to display the commands that belong to the rule. Rule descriptions do not display.

Commands display in the Rules tab under the rules the commands belong to.

When you set breakpoints in the map debug session, a blue dot displays to the left of the rule or command where an individual breakpoint is set.

When you clear the breakpoint, the blue dot is removed.

When you run a debug session, you can follow the progress of the map run, as the specific command that ECMAP Debugger is executing is highlighted in yellow.

- Flow tab – the map flow displays in the Flow tab.

When you run a debug session, you can follow the progress of the map run through the map flow, as the specific flow level that ECMAP Debugger is processing at any moment, is highlighted in yellow.

- Flow properties – the properties of the current flow level or the flow level that you select in the Flow tab, display at the bottom of the Flow tab. You can view the properties of inbound or outbound map flows. See Chapter 15, “Creating a Map Flow” in the *ECMAP User Guide* for more information.

Inbound:

- Current Level
- Segment
- Before Rule
- After Rule
- Next Level

Outbound:

- Current Level
- I/O Rule
- Before Rule
- After Rule
- Next Level
- Break Level

The Data pane is the upper-right pane of the ECMAP Debugger window. When you load a map into ECMAP Debugger, map data displays in the three tabs of the pane:

- Record/Field tab – lists records and fields within each record. Expand the record names to display the fields within the record and the name, length, datatype, and value of each field.
- Memory Variable tab – lists the name, length, datatype, and real-time value of each memory variable in tabular form.
- System Variable tab – lists the name, length, datatype, and real-time value of each system variable in tabular form.

You can select multiple system variables, memory variables, records and fields, and add the items you select to the Watch tab of the Monitor pane to display their real-time values during a debug session.

The Monitor pane is the lower right pane at the bottom of the ECMAP Debugger window. The Monitor pane displays messages generated when you perform any action in ECMAP Debugger, messages generated during a debug session, and real-time values of variables during a debug session. The Monitor pane has these tabs:

- Console tab – displays the different types of messages generated during a debug session:

- Debug information – debug session operational messages, such as when and where a breakpoint has been set, or when you add a memory variable to the Watch tab.
- Debug error messages – debug session error messages, such as when the ECRTP library does not load successfully.
- Runtime information – runtime trace messages, such as when ECRTP starts, pauses, and exits, or when a command with a breakpoint set on it executes successfully.
- Runtime error messages – runtime error messages, such as when ECRTP encounters an error.
- Watch tab – displays the real-time values of system variables, memory variables, and records/fields as the debug session progresses, in tabular form. The Watch tab table has columns for name, datatype, length, and real-time value.

You can select single or multiple system variables, memory variables, and records/fields from the tabs in the Data pane to add to the Watch tab for monitoring.

- Trace tab – displays the trace file that generates during any map run. The trace file in ECMAP Debugger displays messages related to the debug session, such as when breaks have occurred during map execution.
- Segment tab – displays the current segment information during any map run:
 - Map Name – the name of the map you are debugging.
 - File Path – the full path and file name of the map file. This map file is the input EDI file for inbound maps and the output EDI file for outbound maps. Nothing displays for an any-to-any map run as there is no output EDI file generated.
 - Segment Location – the position in the map file of the current segment. Nothing displays for an any-to-any map run as there is no output EDI file generated.
 - Segment Value – the value of the current segment that is being processed. Nothing displays for an any-to-any map run as there is no output EDI file generated.

Preparing to debug maps [EDI 5.1]

You can run a new debugging session or retrieve any map debug session you created and saved in ECMAP Debugger 5.1 and later, make changes to the project settings, map debug session settings, map switch sets, and breakpoints, and run debugging sessions with the changes.

Obtaining map command line switches

Before you start a debug session, provide the map switches used by the map you want to debug. The map switches are the command line arguments that ECMAP passes to ECRTP during a map run.

❖ Obtaining map command line switches

- 1 In the ECMAP Run Inbound Map or Run Outbound Map window, select all the options you want to run with the map.
- 2 In the Run Inbound Map or Run Outbound Map window, select the Option 2 tab.
- 3 Select the Create Batch Command File button.
- 4 In the Batch File Name window, enter the file name and the full path to the batch command file.
- 5 Click OK to create the batch file. The batch file includes remarks and the runmap executable followed by the switches that correspond to the options you select in the tabs of the Run Inbound Map or Run Outbound Map window.

See “Managing map switch sets [EDI 5.1]” on page 49 for instructions to copy map switches to ECMAP Debugger and define map switch sets.

Managing debugging sessions [EDI 5.1]

You can use projects, map debug sessions, and map switch sets to create, save, and change the settings for the maps you debug.

Managing projects [EDI 5.1]

In ECMAP Debugger 5.1 and later, you can create projects to manage the map debug sessions that you create and run. You must assign a map debug session to a project.

ECMAP Debugger retains the project and map debug session information that you save within a project even when you exit the application.

❖ **Creating projects**

You can create multiple projects to manage your map debug sessions.

- 1 Select File | New | Project from the menu bar to display the New Project window. You can also select the New Project button from the toolbar.
- 2 Provide a unique project name and a project description.
- 3 Select OK to save the new project.

❖ **Editing projects**

- 1 Select a project you want to edit.
- 2 Select Edit | Properties to display the Edit Project window. You can also select the Properties button from the toolbar.
- 3 Edit the project description.
- 4 Select OK to save your changes.

❖ **Deleting projects**

Warning! When you delete a project, ECMAP Debugger also deletes all map debug sessions and map switch sets that the project contains.

- 1 Select a project you want to delete.
- 2 Select Edit | Delete or the Delete button to display the Delete Project window and click OK to delete the project.

Managing map debug sessions [EDI 5.1]

In ECMAP Debugger 5.1 and later, you can create and save multiple map debug sessions in ECMAP Debugger to correspond to the actual maps you create in ECMAP that you want to debug in ECMAP Debugger. You can run the map debug sessions with different command line switches and breakpoints to see how maps run with different configurations and to solve map problems.

❖ **Creating map debug sessions**

You can create multiple map debug sessions in one project.

- 1 Select a project to create a map debug session under that project. If you select another map debug session, ECMAP Debugger creates the new map debug session within the same project as the map debug session you selected.
- 2 Select File | New | Map Debug Session. The Project Name field displays the project that the new map debug session belongs to.
- 3 Provide a unique name for the map debug session and a description. You cannot create another map debug session with the same name within one project.
- 4 Select either Inbound Map or Outbound Map to match the map direction of the map debug session you want to create.

Note You must select the correct map direction option—inbound or outbound—to match the switches you copied to ECMAP Debugger. ECMAP Debugger terminates if the map direction you select and the command line and switches that you copy do not match. Check the trace file for errors. See “Viewing the trace file” on page 55 for more information.

- 5 Select OK to save the new map debug session.

❖ **Editing map debug sessions**

- 1 Select a map you want to edit.
- 2 Select Edit | Properties to display the Edit Map Debug Session window.
- 3 Edit the map description and change the map direction if you want to. The map direction must match the direction of the map you want to debug and from which you copy the map command line switches.
- 4 Select OK to save your changes.

❖ **Deleting map debug sessions**

- 1 Select a map debug session you want to delete.
- 2 Select Edit | Delete to display the Delete Map Debug Session window and click OK to delete the map debug session.

Managing map switch sets [EDI 5.1]

In ECMAP Debugger 5.1 and later, you can create multiple sets of map command line switches for each map debug session you create. You can also edit the map command line switches you copy. This allows you to run map debug sessions with different combinations of command line switches and breakpoints.

You must assign a unique map switch name to identify each set of map command line switches.

❖ **Creating map switch sets in ECMAP Debugger**

You can create a set of the map switches for any map you want to debug by copying the map command line switches to a map debug session in ECMAP Debugger. You can create multiple sets of switches, each with a unique name, containing the map command line switches and arguments.

- 1 Select a map debug session where you want to add map command line switches.
- 2 Select File | New | Map Switch Set.
- 3 Provide a unique name for the new map switch set.
- 4 From the batch file you created in ECMAP to save map command line switches, copy everything in the command line, including all the arguments to the right of the executables. Do not copy the UNIX `mapinrun` or `rmapout` executables, or the Windows `wrmi32` or `wrmo32` executables.
- 5 Paste the command line arguments you have copied to the Map Switches field. You can edit the arguments to investigate their actions as part of the debugging process.

Note Verify that the map arguments you copy or edit match the map direction you specify—inbound or outbound—when you create the map debug session in ECMAP Debugger. ECMAP Debugger terminates if the map direction you specify and the command line arguments you copy or edit do not match. Check the trace file for errors. See “Viewing the trace file” on page 55 for more information.

- 6 Provide a description for the map switch set.

- 7 Click OK to save your settings. In the Console tab at the lower right of the ECMAP Debugger window, informational messages display, acknowledging ECMAP Debugger received the map arguments.

Note The Set Switches option is disabled once the map switches are set for a debug session. Only one debug session can run at a time. When a debug session completes or is manually stopped, the Set Switches option is enabled.

❖ **Editing map switch sets**

- 1 Select a map switch set you want to edit.
- 2 Select Edit | Properties to display the Edit Map Switch Set window.
- 3 Select OK to save your changes.

❖ **Deleting map switch sets**

- 1 Select the map switch set you want to delete.
- 2 Select Edit | Delete to display the Delete Map Switch Set and click OK to delete the map switch set.

Starting ECRTP and loading map information

After you create or modify a map switch set, start ECRTP and load the map information, and load and display the map information in ECMAP Debugger, before you can set breakpoints and run the debug session.

❖ **Starting ECRTP and loading map information**

- 1 Select a map switch set.
- 2 Select Action | Run, or click the Run button to invoke an instance of ECRTP and to pass all the map information you copied into ECMAP Debugger, such as the command line switches, to the ECRTP instance.
- 3 Select Action | Debug Process from the menu, or click the Debug Process button to obtain the file location of the map, load the map data into ECMAP Debugger, and display the map data in the ECMAP Debugger window.

The map information displays in the Rules/Flow pane, the Data pane, and Monitor pane. You can set breakpoints at the rules and commands that display in the Data pane.

Setting and clearing breakpoints

Setting and clearing breakpoints in a debug session allows you to control map execution during a debug session:

- You can set breakpoints and *conditional breakpoints* in the map you load into ECMAP Debugger to allow you to pause ECRTP execution of a map at the points you set.
- You can set and clear breakpoints in the map at any point during the debug session.

When ECMAP Debugger pauses ECRTP execution at a breakpoint, you can observe the data values of records/fields, memory variables, or system variables at that point and monitor the change in data values as ECRTP execution progresses.

❖ Setting breakpoints

- 1 Select a command or a rule in the map where you want to set a breakpoint.
- 2 Select Breakpoints | Set from the menu bar, press Ctrl+B, or click the Set Breakpoints button to set the breakpoint at the highlighted command or rule. A blue dot appears to the left of the command or rule where you have set the breakpoint.

Note If you set a breakpoint for a rule, all commands under that rule are set with breakpoints.

❖ Setting conditional breakpoints

You can set conditional breakpoints, which allow you to pause ECRTP execution at a breakpoint when a variable has a specific value.

- 1 In ECMAP, add a new placeholder rule to your map to check the value of a specified variable. You can do this by using the If Condition command in the placeholder rule and inserting the Do Nothing command as the only outcome in this placeholder rule.

An example of a placeholder rule can be represented in this form:

If *variable=value* Then Perform Rule *rule number*

where *variable* is a memory variable, system variable, or record/field, *value* is the value of *variable*, and *rule number* is the specific rule that contains the Do Nothing command.

The placeholder rule stops map execution when it is encountered because of the Do Nothing command. You can insert any other command besides the Do Nothing command if you want specific actions to occur.

- 2 In the map, add the placeholder rule to an existing rule where you want the value to be checked.
- 3 Regenerate the map.
- 4 Obtain the map command line switches from ECMAP, create a map debug session in ECMAP Debugger, copy the map command line switches to the new debug session and load the map in ECMAP Debugger.
- 5 Set a breakpoint in the map on the placeholder you created in step 1.
- 6 Run the debug session. When map execution reaches the existing rule whose value you want to check, it pauses if the value in *variable* matches the value you specified in the placeholder rule.

❖ **Clearing selected breakpoints**

- 1 Select the command or rule in the map where you want to clear a breakpoint.
- 2 Select Breakpoints | Clear from the menu bar, press Ctrl+C, or click the Clear Breakpoint button to remove the breakpoint at the command or rule that you highlight. ECMAP Debugger removes the blue dot to the left of the command or rule.

Note If you clear a breakpoint for a rule, all commands under that rule have their breakpoints removed.

❖ **Clearing all breakpoints**

- Select Breakpoints | Clear All from the main menu, press Ctrl+A, or click the Clear All Breakpoints button to remove all breakpoints that have been set in the map. ECMAP Debugger removes the blue dots to the left of commands or rules that have breakpoints.

Viewing specific real-time data values

ECMAP Debugger allows you to view specific data values and avoid scrolling through the other items in the Data pane.

❖ Viewing specific data values in the Watch tab

- 1 In the Data pane, select the memory variable, system variable, or record/field you want to observe.
- 2 Right-click the highlighted item and select Add Watch.

The highlighted item is added to the Watch tab of the Monitor pane. When you run a debug session, the data values in the Value column of the Watch tab that have changed since the last break are highlighted in blue.

Saving projects and debug sessions [EDI 5.1]

You can save the projects, maps, switches, breakpoints, and watched variables list you have created and changed. Select Project | Save All to save your changes for all projects. You cannot save information while a debug session is running.

When you start ECMAP Debugger again, you can run the debug sessions you configured and saved the previous time you used ECMAP Debugger.

Note Breakpoints and watched variables that you save are associated with the map debug session they belong to. All map switch sets that belong to a map debug session also inherit all the breakpoints and watched variables that you save under that map debug session. Breakpoints that you change for a particular map switch set are also changed for all map switch sets that belong to the same map debug session.

ECMAP Debugger stores the project, map debug session, map switch set, breakpoints, and watched variables information you save in the *.debugproject* file in the `%MAPDEBUG_HOME%\bin` directory where `%MAPDEBUG_HOME%` is the directory where ECMAP Debugger is installed.

Sybase recommends that you do not edit this file. You can use an XML reader to view the file contents. ECMAP Debugger cannot retrieve any information on your project if you delete the file or change its location.

Running an ECMAP Debugger session

You can initiate a debug session after you set the map switches and load the map information into ECMAP Debugger. During a debug session, ECMAP Debugger provides options to control the debug session and view data values as they change during a map run.

Starting and controlling a debug session

Use the options in the Debug menu to start, control and exit a debug session.

Note You can follow the debug process, as the command being executed is highlighted in yellow.

Select a map switch set of the map debug session you want to debug. Select Debug from the main menu to display options for running the debug session in different modes. You can also click the buttons for each option from the toolbar:

- Run Step by Step – the debug process pauses at every command or breakpoint it encounters even if no breakpoint has been set at a command. Continue to click the step-by-step option to advance the debug process to the next command or breakpoint until the map run is completed. You can also press Ctrl+T to perform a step-by-step debug session.
- Run to Next Breakpoint – the debug process skips the next breakpoint or command and pauses at the following breakpoint that it encounters. If no breakpoint is encountered, the debug process runs to map completion. You can also press Ctrl+O to perform a step-over debug session.
- Run To Cursor – the debug process runs and pauses at any command you highlight. It pauses at any breakpoints it encounters that occur before the highlighted command. You can also press Ctrl+R to perform a to cursor debug session.
- Run To the End – the debug process runs to the end of the map without pausing at any command or breakpoint. You can also press Ctrl+D to perform a to the end debug session.
- Stop Debug Session – stops maps execution, which stops the debug process.

ECMAP Debugger provides flexibility in how you want to run and control debug sessions:

- You can choose to change the debug run mode whenever control is returned during a debug session.
- You can choose to set and clear breakpoints when control is returned to you during a debug session.

Monitoring real-time data values

During a debug session, you can monitor the real-time data values of memory variables, system variables, and records/fields in the Data pane. Data values that have changed are highlighted in blue.

Monitoring the map run progress [5.1]

During a debug session, you can monitor the progress of the map run, map flow, and flow levels of the map that you are debugging. ECMAP Debugger highlights the rule, command, and flow level that it is currently processing.

Viewing the trace file

The Trace tab of the Monitor pane displays the trace file generated by ECRTF during the map run.

The trace file in ECMAP Debugger displays debug session information and shows when breaks have occurred during a map run.

For example, if the ECMAP Debugger pauses at the first command or rule 20, the trace file displays:

```
Debug: pause at the 1 command of rule 20
```

You can use the data in the trace file to determine where ECMAP Debugger is in the map run process, what is occurring, or what has happened since the last break was encountered.

Completing the debug session

When the debug session ends, a message box displays with a return value.

- If no errors were encountered during the debug session, a message box titled “Map Debug Session Finished” displays with a return value of 0 (zero).

A return value of 1 or 2 indicates that minor errors were encountered. You can check the trace file for the errors.

- If more serious errors were encountered during the debug session, a message box titled “Map Debug Session Finished With Errors” displays with a return value of more than 2. Check the trace file for the errors, solve them, and run the map through ECMAP Debugger again to verify that no serious errors remain.

Once you have completed a map run in ECMAP Debugger, you can load, run, and debug another map, or you can exit ECMAP Debugger by selecting File | Exit.

EC Gateway

EC Gateway is an enterprise-wide message management software, designed to support inter- and intra-company electronic messages. EC Gateway has three main functions: process management, communications, and scheduling. It includes extensive reporting facilities, a trading partner interface, a single ODBC-compliant log database, and archive and restore functions.

Updates to EC Gateway include support for reusing e-FTP scripts, added security for files created by pfs/Async, new search functions, wildcard support for e-mail attachments, script support for command line parameters, and new RunMap options.

Topic	Page
Reusing e-FTP scripts for multiple systems [EBF 4.2.3]	58
Added security for files created by pfs/Async [EBF 4.2.3]	58
Find and replace functions [EDI 4.2]	58
Copying process scripts to other systems [EDI 4.2]	59
Wildcard e-mail attachments [EDI 4.2]	60
Passing input to the EC Gateway executable [EDI 4.2]	61
Passing input to and getting output from the EC Gateway DLL shared library [EDI 4.2]	62
RunMap options	64
Return command [EDI 4.2]	65
Exit command [EDI 4.2]	65
StringLength command [EBF 4.2.1]	65

Reusing e-FTP scripts for multiple systems [EBF 4.2.3]

The e-FTP module can now obtain the script file name, server name, user ID, and password from the communications channel. With this added functionality, you can reuse the same e-FTP scripts for multiple systems by referencing different communication channels. Previously, only the script file name came from the communications channel; you could obtain the other information only from the script file itself.

Applies to	EC Gateway
EC Gateway version	4.2.3
Resolves	CR#382553

Added security for files created by pfs/Async [EBF 4.2.3]

The user ID for the person running pfs/Async is now recorded in the USER_IDENT field of the log file, *trlog*. Also, the files archived by the pfs/Async host will have the RunID of the pfs/Async host appended to the end of the file name. This ensures that the archived files have unique names, even if they are archived into common directories.

Applies to	EC Gateway
EC Gateway version	4.2.3
Resolves	CR#369152

Find and replace functions [EDI 4.2]

New Find and Replace options allow you to search for and replace strings in specific modules. Find and Replace options are available in the Comm Channel, Mailbox, Trade Partner, Company ID, and Process modules.

❖ Searching for strings

- 1 Select Edit | Find.
- 2 In the Find field, enter the string you want to find.

3 Click Find Next.

❖ **Searching and replacing strings**

1 Select Edit | Replace.

2 In the Find field, enter the string you want to find, then type the replacement string in the Replace With field.

3 Select one of these buttons:

- Replace – replaces the current instance of the search string with the replacement string.
- Replace All – replaces all instances of the search string with the replacement string.
- Find Next – finds the next occurrence of the search string.
- Details – identifies the location of the search string.

Copying process scripts to other systems [EDI 4.2]

A Copy Process option in the Process module enables you to copy a process (or script) from the current DSN (specified by the current system) to another DSN (may be specified by any system). This lets you run the same process against multiple trading partner databases.

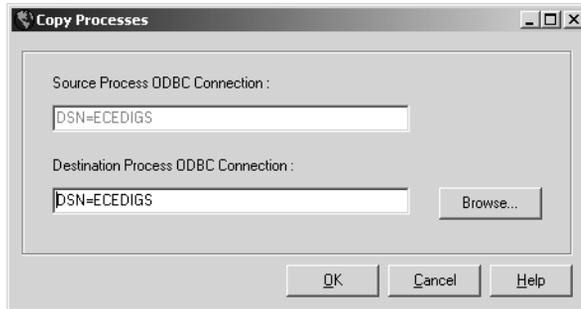
❖ **Copying process scripts**

1 Right-click a process name and choose Copy Process.

Alternatively, click a process name, select Utility, then select Copy Processes.

A dialog box prompts you for the DSN of the destination database. The destination DSN box displays the source DSN by default.

Figure 4-1: Copy Processes window in EC Gateway



- 2 In Destination Process ODBC Connection, enter the current DSN you want to use.

Note In Windows, add the prefix “Copy (n) of...” to the name of any copy you make with the Copy Processes option. To change the name of the process, right-click the name of the copy, choose Properties, and change the name in the Process Name box.

Wildcard e-mail attachments [EDI 4.2]

E-mail parameters on the EC Gateway Process dialog box in Windows include an Other Options field, which allows you to include e-mail attachments. In versions of EC Gateway earlier than 4.2, you had to explicitly identify each file to include as an e-mail attachment with a separate `-attach` command for each binary file, and a separate `-attacht` command for each text file. You can now use wildcards to attach multiple files.

To attach multiple binary files enter:

`-attach path/filename`

For example, to attach all binary executables in the *attachments* directory that start with the letter “a”, enter:

```
-attach c:\attachments\a*.exe
```

This example attaches all tar files in the *attachments* directory:

```
-attach c:\attachments\*.tar
```

To attach multiple text files, enter:

```
-attach path/filename
```

This example attaches all text files in the *attachments* directory:

```
-attach c:\attachments\*.txt
```

Note The UNIX version of *wsprocs* can functionally process the `-uencode` or `-mime` options with file names containing asterisks into multiple file names to be attached to an e-mail message.

Passing input to the EC Gateway executable [EDI 4.2]

You can pass input parameters to the EC Gateway executable on the command line. The switches for these are `-p1` through `-p20`. The Assign command simplifies passing parameter values from the command line to message variables, parameter variables, or shared memory variables.

When you use the Assign command, you must select Destination and Source parameters for the assignment.

- The Destination parameter identifies the variable to initialize. You can choose a message variable, parameter variable, or shared memory variable for the destination.
- The Source parameter identifies the command line parameter to assign to the Destination parameter. To pass a value from the command line, point to `CMD_Line_Parameter` and then choose the parameter you want.

For the *wsprocs* executable, the command line syntax is as follows:

```
wsprocs script file name [-t directory for the wsprocs trace file]
[-r the run id for this wsprocs run]
[-td maximum number for the threads]
[-p1 command parameter 1]
[-p2 command parameter 2] ... [-p20 command parameter 20]
```

In this example, the MQ manager and the MQ queue names are not explicitly defined in *testscript.pfs*. Instead, the script should assign `CMD_Line_Parameter-1` to the parameter with the MQ manager name and `CMD_Line_Parameter-2` to the parameter with the MQ queue name.

```
wsprocs c:\ecedigs\testscript.pfs -p1 mq_mgr1 -p2
mq_queue1
```

Type the command on a single continuous command line and insert a space after each option/parameter pair.

Passing input to and getting output from the EC Gateway DLL shared library [EDI 4.2]

The EC Gateway DLL shared library includes the ability to pass input into EC Gateway from the calling program and return output to the calling program. This is accomplished through an array of data structures called DATAIOSTRUCT where each structure contains a tag name, a pointer to the value, the length of the value, and the length of the buffer containing the value. Each structure can be referenced by its tag name or by its index, which is its position in the array structure.

With this capability, EC Gateway can dynamically increase the size of the array that it returns to the calling program. For example, if EC Gateway were to return an output for each item in the data, it would not know in advance, nor would the calling program know, how many items are to be returned. However, the array can be increased in size, one structure for each item found in the data.

The syntax for calling the DLL shared library is:

```
__declspec( dllexport ) int __stdcall wsprocesDataIO (LPSTR
    lpszCmdLine, DATAIOSTRUCT ** pDataIO)
```

where `lpszCmdLine` is the command line for the `wsproces` executable defined in section 3.6. Type the entries for this syntax on a single continuous command line with a space before “(LPSTR ... pDataIO).”

The structure `DATAIOSTRUCT` is defined as:

```
typedef struct {
    char *tag;
    char **pdata;
    long *pdatalen;
    long *pbuflen;
} DATAIOSTRUCT;
```

This structure has four fields, which are:

- `tag` – is for a pointer to the name of the data that can be referenced within a script.
- `pdata` – is for a pointer to the buffer for the data.
- `pdatalen` – is for a pointer to the length of the data.
- `pbuflen` – is for a pointer to the length of the storage buffer allocated for the data.

The array is terminated by an extra structure with its `tag` pointer set to `NULL`.

With this API, if there is no input data or output data to be passed in memory, the array of data I/O structures is not needed and the calling application should set the second parameter to `NULL`. The calling application must define a command line, which must include the script file name. For upward compatibility, for example if converting from the executable to the DLL, you can still enter the input parameters using the command line syntax of `-p1` through `-p20` and you can still assign them using the `Assign` command.

Use the `AssignInput` and `AssignOutput` commands to move data between the array of data structures and EC Gateway message variables, parameter variables, or shared memory variables.

When using the `AssignInput` command, you must provide values for the `Destination`, `Source By`, and `Source Value` parameters.

- The `Destination` parameter identifies the variable to initialize. You can choose a Message variable, a Parameter variable, or a Shared Memory variable for the destination.

- Source By is a string, either Tag or Index.
- Source Value depends on the value for Source By:
 - If the Source By is Tag, then the value should be a string with the tag name of one of the structures in the array.
 - If the Source By is Index, then the value should be an integer with the index of a structure in the array.

The first structure in the array corresponds to index zero (0).

When you use the AssignOutput command, you must provide the values for Destination By, Destination Value, Source Data, and Overwrite/Append parameters.

- Destination By and Destination Value define which structure in the array you want to set. Destination By is a string, either Tag or Index. Destination Value depends on the Destination By value.
 - If the Destination By value is Tag, then the value should be a string. If that value matches a tag name in the original structure, that structure is updated. If that value does not match any tag name in the structure, EC Gateway automatically adds another structure to the array with that tag name.
 - If the Destination By value is Index, then the value must be an integer in the range of the already existing structures in the array. If the Index is not in range, it does not add another structure to the array.
- The Source Data identifies the variable with the data to be used to update the structure. You can choose a Message variable, a Parameter variable, or a Shared Memory variable for the destination.
- The Overwrite/Append indicates whether the data should overwrite the data already in the buffer or append to it.

RunMap options

RunMap options apply to ECRTP, ECMAP, and EC Gateway. Updates to the RunMap dialog box in ECMAP and EC Gateway reflect additional runtime parameters and switches provided by ECRTP. For more about these new options and parameters, see “RunMap options” on page 70.

Return command [EDI 4.2]

The Return command allows you to include an optional return code, which is passed to the program that called it. You can use the Return command in the main script of the executable and the DLL shared library versions.

You can use Counter or Constant type variables as return codes. In the case of the DLL shared library version, if no return code value is assigned, a value of zero (0) is returned. The Return command itself, if executed inside a subscript called by Do, DoWhile, WhileDo, or RunThread, terminates the processing of the subscript. There is no value in specifying a return code in this instance because it is ignored. If you execute the Return command in the main script, the main script terminates.

Exit command [EDI 4.2]

The Exit command allows you to use Counter and Constant type variables as a return code. Setting the code is optional, and if not specified, is set to zero (0) for the DLL version and to 260 for the executable version for backward compatibility. The Exit command terminates execution of the entire script, including any subscripts, in the same thread or not. In general, do not use the Exit command in subscripts; doing so generates a warning during compilation.

StringLength command [EBF 4.2.1]

You can determine the length of a message at runtime. The StringLength command allows you to specify a Message variable as a string type, which returns the length of the message specified in the variable.

Applies to	EC Gateway, wsproces
EC Gateway version	4.2.1
Resolves	CR#372914

ECRTP is both a data transformation engine that analyzes, transforms, and routes messages, and a compliance-verification engine that checks EDI data against the corresponding EDI standard. ECRTP includes map files, a trading partner database, and log files.

- Map files contain the business rules and logic that define the relationships between incoming and outgoing data.
- The trading partner database contains information that routes messages between trading partners and that selects the specific map that should be run.
- Log files maintain an audit trail of the transaction processing.

Changes in ECRTP include support for new features, support for handling larger file sizes, new RunMap options, and new command line switches.

Topic	Page
ECRTP Web services API [EDI 5.1]	67
Large file handling [EDI 5.1]	68
Log bulk insert [EBF 4.2.4]	68
JNI wrapper for ECRTP [EBF 4.2.3]	69
PADEDI environment variable [EDI 4.2]	70
RunMap options	70
Command line switches [EDI 4.2]	84
ODBC switches [EDI 4.2]	87

ECRTP Web services API [EDI 5.1]

ECRTP version 5.1 and later, on Windows only, includes the ECRTP Web services API that enable you to create Web applications that call ECRTP.

For example, a payer can create a Web application with the API for providers to use for real-time eligibility verification. See “Installing and starting the ECRTTP Web service,” in Chapter 8, “Installing ECRTTP for Windows” in the *EDI Products 5.1 Installation Guide* to install and deploy the Web service.

Large file handling [EDI 5.1]

With ECRTTP 5.1 and later, on Windows only, ECRTTP can process EDI files that are larger than 2GB.

Log bulk insert [EBF 4.2.4]

With version 4.2.4, ECRTTP provides a Windows feature, log bulk insert, that improves performance when using an ODBC database for the log, TRLOG. With this feature, ECRTTP writes multiple log records at one time, instead of writing them one at a time, as was done prior to this enhancement.

An environment variable, TRLOG_BLKSIZE, allows users to specify the maximum number of records to be inserted into the TRLOG database at one time. The allowable range is 0 to 500 records. The default value is 100.

ECRTTP automatically detects if the TRLOG database supports bulk inserts. If the TRLOG database does not support bulk inserts or if the TRLOG_BLOCKSIZE environment variable is set to 0 or 1, the program inserts records into TRLOG one at a time, as it does in EDI 4.2.3 and earlier releases.

If ECRTTP cannot get sufficient memory for the bulk insert at the specified block size, it retries the insert with a block size of 50. If there is insufficient memory for the bulk inserts at a block size of 50, ECRTTP inserts records one at a time.

In addition, with version 4.2.4, a *transNNN.log* file is written to disk if either the single TRLOG insert fails or the bulk insert fails. The *transNNN.log* file is a dump of the LOG records that were buffered in memory but could not be written to the ODBC TRLOG table.

The *transNNN.log* file is stored in one of the following directories:

- The default map directory, indicated by the `-dg` switch. This is a required switch for all ECRTP runs.
- The directory specified for the text log file with the `-ed` switch. This is an optional switch used for text log files.

If the insert to the ODBC TRLOG table fails, and the *transNNN.log* file is written to disk, the following error message is written to the TRACE file, *TRNNN.DAT*.

```
(1042) FATAL: Failed To Write TRLOG - dumping log to  
full_path_directory\transNNN.log
```

See Chapter 2, “Running ECRTP,” of the *ECRTP Reference Guide* for information about other ECRTP environment variables, and for the short and expanded formats for writing log records to disk. The expanded format matches the record layout of the TRLOG table.

JNI wrapper for ECRTP [EBF 4.2.3]

ECRTP 4.2.3 and later contains a Java Native Interface (JNI) wrapper for the ECRTP API, implemented in EDI 4.2. This wrapper enables Java developers to not only call ECRTP, a C/C++ based program, but also take advantage of the new memory I/O features of the API.

The JNI wrapper consists of one package, `com.sybase.ecrtp`, which contains the following classes:

- RTP
- RTPException
- UtilCountMapResult

For more information, see the *README.txt* file, Javadoc files, and demo files, which are provided as part of the ECRTP installation.

PADEDI environment variable [EDI 4.2]

PADEDI is a previously undocumented environment variable, which pads numeric fields with leading zeros or alphanumeric fields with trailing spaces at runtime to ensure that data meets minimal field length requirements.

ECRTP includes a `-pe` switch, which pads EDI data to minimum length in compliance maps. This switch is an optional replacement for the PADEDI environment variable. See “Padding EDI data to minimum length [EDI 4.2]” on page 83 for more information.

Windows

Select Control Panel | System | Advanced | Environment Variables | System Variables. Set the Variable Name to PADEDI. Set the Variable Value to Y.

UNIX

In UNIX, open a shell and run the following commands relevant to your shell:

- ksh:

```
set PADEDI=YES ; export PADEDI
```

- sh:

```
set PADEDI=1 ; export PADEDI
```

- csh:

```
setenv PADEDI 1
```

Resolves

CR#297413

RunMap options

RunMap options apply to ECRTP, ECMAP, and EC Gateway. Updates to the RunMap dialog in ECMAP and EC Gateway reflect additional runtime parameters and switches provided by ECRTP.

Character checks for compliance maps [EBF 4.2.5]

Two switches, `-al` and `-vs`, have been added to the Run Inbound Map windows to handle character checks for compliance maps. The `-al` switch disallows the use of high-order ASCII characters between 128 – 255 in EDI data, while the `-vs` switch disallows using either the `^` (circumflex) or the `'` (accent grave) characters as data characters if the X12 version is before 4040. Prior to version 4040, even though these two characters were not permitted to be used as data characters, they were permitted to be used as delimiters.

Inbound map runs

The `-al` optional switch disallows the use of the extended ASCII characters 128 – 255. These characters are generally used for Western European languages only.

Applies to	Inbound maps.
ECRTP version	EBF 4.2.5
Resolves	CR#409591 and 409592
Command line syntax	The syntax for this switch is:

-al

❖ **Allowing only ASCII characters for inbound maps in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 1 tab.
- 3 Select Allow only ASCII characters.

Inbound map runs

The `-vs` switch disallows the use of the `^` (circumflex) or `'` (accent grave) as data characters in X12 versions prior to version 4040. An error is generated if one of these is used in this way. The `-vs` switch allows the use of these characters as delimiters, even before version 4040.

Applies to	Inbound maps.
ECRTP version	EBF 4.2.5
Resolves	CR#409591 and 409592
Command line syntax	The syntax for this switch is:

-vs

❖ **Performing version-specific character checks for inbound maps in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 1 tab.
- 3 Select Perform Version Specific Character Checks.

Setting maximum waiting time for ECRTP to get a database connection [EBF 4.2.5]

To reduce the probability of an ECRTP failure due to a temporary inability to establish a database connection, you can now set a maximum waiting time for ECRTP which specifies the maximum time period that ECRTP waits for a successful database connection.

The optional switch, `-tc NN`, where `NN` is a number in milliseconds, allows you to specify a maximum waiting time for ECRTP to establish a database connection. If a connection is or becomes available during that time period, ECRTP immediately stops waiting and starts using the connection. If no connection becomes available during that time period, ECRTP gives up waiting and returns with an error that says no database connection is available.

Inbound map runs

The option for the `-tc NN` switch is called “DBMS Connection Retry Time (msec)” and it appears on the Option 1 tab for inbound processing.

Applies to	Inbound maps.
ECRTP version	EBF 4.2.5
Resolves	CR#295784, 407159 and 407160
Command line syntax	The syntax for this switch is:

`-tc NN`

❖ **Setting a maximum waiting time for inbound map runs in Windows**

- 1 Open the Run Inbound Map acknowledgement window, or the Run Inbound Map window.
- 2 Click the Option 1 tab.

-
- 3 Enter a value in DBMS Connection Retry Time (msec).
-

Note The default *NN* value for this switch is 1000 msec, where msec refers to milliseconds. It can take a range of values between 0 and 21600000 milliseconds (6 hours).

Outbound map runs

The option for the `-tc NN` switch is called “DBMS Connection Retry Time (msec)” and it appears on the Option 1 tab for outbound processing.

Applies to	Outbound maps.
ECRTP version	EBF 4.2.5
Resolves	CR#295784, 407159 and 407160
Command line syntax	The syntax for this switch is:

-tc NN

❖ Setting a maximum waiting time for outbound maps in Windows

- 1 Open the Run Outbound Map window.
 - 2 Click the Option 1 tab.
 - 3 Enter a value in DBMS Connection Retry Time (msec).
-

Note The default *NN* value for this switch is 1000 msec, where msec refers to milliseconds. It can take a range of values between 0 and 21600000 milliseconds (6 hours).

Controlling updates and lookups for control numbers [EBF 4.2.4]

The `-cn` switch allows you to control updates and lookups for control numbers when running maps. This switch works differently for inbound and outbound map runs. Using this option improves performance for those customers who do not care about control numbers.

Inbound map runs

The `-cn` switch disables control number lookup and updates in the trading partner database during inbound map runs.

Applies to	Inbound maps.
ECRTP version	EBF 4.2.4
Resolves	CR#388842
Command line syntax	The syntax for this switch is:

`-cn`

❖ **Disabling control number updates for inbound maps in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 2 tab.
- 3 Select No Control Number Updates.

The Validate Control Number Sequence option in the Option 2 tab of the ECMAP Run Inbound Map window is disabled when you select the No Control Number Updates option, as no validation can occur if no control numbers are being updated, and vice versa.

The same holds true for the corresponding command line switches, `-sc` and `-cn`. The switches cannot be used concurrently. See “Validating the Interchange (ISA) and Group (GS) control number sequence [EDI 4.2]” on page 75.

If you do not use `-cn` and `-sc` on the command line or if you do not select No Control Number Updates and Validate Control Number Sequence in Windows, the control numbers are updated to the database but are not validated.

Outbound map runs

The `-cn` *control number* switch disables control number lookups and updates in the trading partner database during outbound map runs. The *control number* option is provided for the switch to accept a value for *control number* as input from the calling program or from the user. This overrides the default value from ECRTP. The value for *control number* is used for the outbound ISA.

Applies to	Outbound maps.
ECRTP version	EBF 4.2.4
Resolves	CR#388842
Command line syntax	The syntax for this switch is:

-cn control number❖ **Disabling control number updates and specifying a control number in Windows**

- 1 Open the Run Outbound Map window, or open the Run Outbound Map acknowledgement window.
- 2 Click the Option 1 tab.
- 3 Select Enter Control Number Override. A text box appears, where you may enter a value for the control number.

For both Windows and the command line:

- The control number must be a valid integer and can be up to 9 digits.
- If a value for control number with less than 9 digits is entered, ECRTP fills up the field with zeros as it has a fixed length of 9 digits.
- If the option is selected in Windows and the switch is used in the command line but no entry is made in the text box, the default value of 000000001 (1) is passed as the control number.
- If multiple interchanges occur, the interchange control number is incremented by 1 from the user-supplied value.
- The group control number starts at 1 for each interchange encountered and increment accordingly.

Validating the Interchange (ISA) and Group (GS) control number sequence [EDI 4.2]

The **-sc** switch validates the sequence of the Interchange (ISA) and Group (GS) control numbers.

If you use this switch, ECRTP checks the current control number against the trading partner database to validate the entry. If the entry in the file is not the next sequential entry, ECRTP reports an error.

Applies to	Inbound compliance maps.
ECRTP Version	4.2
Resolves	CR#322012
Command line syntax	The syntax for this switch is: -sc

❖ **Validating ISA and GS control sequence numbers in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 2 tab.
- 3 Select Validate Control Number Sequence.

The Validate Control Number Sequence option in the Option 2 tab of the ECMAP Run Inbound Map window is disabled when you select the No Control Number Updates option, as no validation can occur if no control numbers are being updated, and vice versa.

The same holds true for the corresponding command line switches, `-sc` and `-cn`. The `-sc` switch cannot be used concurrently with the `-cn` switch. See “Controlling updates and lookups for control numbers [EBF 4.2.4]” on page 73.

If you do not use `-cn` and `-sc` on the command line or if you do not select No Control Number Updates and Validate Control Number Sequence in Windows, the control numbers are updated to the database but are not validated.

Override for compliance map results [EDI 4.2]

There are RunMap options that allow you to override the default location for good and bad compliance map data. You can now specify the full file paths for storing compliance map results. These options correspond to the `-ab` and `-ag` ECRTP command line switches. See also Chapter 4, “Working with the Address Book,” and Chapter 17, “Running Maps,” in the *ECMAP User Guide*.

Specifying a file to use in place of a BAD EDI file [EDI 4.2]

The `-ab` switch allows you to specify which file to use in place of the BAD EDI file. You must specify a full path and file name.

- If you specify the `-ab` switch from the command line, ECRTP ignores the original BAD EDI destination, and writes the BAD EDI file to the path and file name you specify as parameters on the command line.
- If the file specified by the `-ab` switch is passed in the MemIOStruct, ECRTP writes to the memory buffer instead of disk.
- If you specify a file name without a path, ECRTP writes the BAD file to the current working directory for disk I/O, which may or may not be the map directory.

Applies to	Inbound maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is:

-ab *path/filename*

❖ **Substituting a BAD file name in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 2 tab.
- 3 Click Browse next to the Substitute Bad Filename field and choose an appropriate destination directory and file name.

Specifying a file to use in place of a GOOD EDI file [EDI 4.2]

In ECRTP version 4.2, for inbound maps only, the **-ag** switch allows you to specify which file to use in place of the GOOD EDI file. You must specify a full path and file name.

- If you pass the **-ag** switch from the command line, ECRTP ignores the original GOOD EDI destination, and writes the GOOD EDI file to the path and file name you specify as parameters on the command line.
- If the user specifies a file name without a path, then ECRTP writes the GOOD file to the current working directory for disk I/O, which may or may not be the map directory.

Applies to	Inbound maps.
ECRTP Version	4.2
Command line syntax	The syntax for this switch is:

-ag *path/filename*

❖ **Substituting a GOOD file name in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 2 tab.
- 3 Click Browse next to the Substitute Good Filename field and choose an appropriate destination directory and file name.

Specifying map name extensions [EDI 4.2]

In ECRTP version 4.2, for inbound and outbound maps, the `-mn` switch passes the map name extension as part of a command line argument. A Map Name Extension option has been added to the RunMap and Trade Agreement windows.

The map name extension works with other map lookup fields to find a correct map by allowing you to specify an additional identifier to distinguish multiple maps for the same transaction within a single trade agreement. This supports the need for multiple 287 and 837 HIPAA transactions.

For map functions with multiple parameters, such as `LOADMAP`, find the correct map name by using current map lookup fields and the map name extension before calling the functions.

Applies to Inbound and outbound maps.

ECRTP Version 4.2

Command line syntax The syntax for this switch is:

`-mn map name extension`

❖ Specifying a map name extension in Windows

- 1 Open the Run Inbound Map or Run Outbound Map window.
- 2 Click the Required tab.
- 3 In the Map Name Extension box, type the appropriate map name extension.

-mns switch removed for ECRTP 4.2 [EDI 4.2]

The `-mns` switch has been removed as of version 4.2.

Return codes [EDI 4.2]

In ECRTP 4.2, for inbound compliance maps only, the `-nret` switch adds return codes that provide additional information to the calling program. Based on the return value, you can determine the next step in the process. The return codes reflect the following information:

- At least one interchange is in error, implying a TA1 map should be run.

- At least one group or transaction is in error, implying a 997 map should be run.
- At least one good transaction is present, implying a translation map should be run.

ECRTP returns a code based on these conditions, and may trigger other processes. A given return code may, for example, indicate that TA1, 997, and transformation maps should be run.

Applies to	Inbound compliance maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is:

-nret

❖ **Adding the return codes in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 1 tab.
- 3 Select Enhanced Return Codes.

Examples	This list specifies the return codes and identifies the codes that should trigger the TA1, 997 or translation maps:
----------	---

- Return code 16 – ECRTP aborted processing.
- Return code 8 – a map abort occurred.
- Return codes 4, 5, 6, or 7 – indicates a TA1 acknowledgement map should be run because there was at least one interchange level error.
- Return codes 2, 3, 6, or 7 – indicates a 997 acknowledgement map should be run because there was at least one group or transaction level error.
- Return codes 1, 3, 5, or 7 – indicates a translation map should be run because there was at least one good transaction.

NCPDP file processing [EDI 4.2]

Changes to NCPDP processing include switches that identify NCPDP batch and telecommunications files.

NCPDP batch files [EDI 4.2]

The `-ncb` switch indicates that the EDI file you want to process is an NCPDP batch file. Required for all inbound NCPDP files unless `-nct` is active.

Applies to	Inbound maps.
ECRTP version	4.2
Resolves	CR#322012
Command line syntax	The syntax for this switch is:

`-ncb`

❖ Processing NCPDP batch files in Windows

- 1 Open the Run Inbound Map window.
- 2 Click the Required tab.
- 3 Select NCPDP Batch.

NCPDP telecommunications files [EDI 4.2]

The `-nct` switch indicates that the EDI file you want to process is an NCPDP telecommunications file. Required for all inbound NCPDP files unless `-ncb` is active.

Applies to	Inbound maps.
ECRTP Version	4.2
Resolves	CR#322012
Command line syntax	The syntax for this switch is:

`-nct`

❖ Processing NCPDP telecommunication files in Windows

- 1 Open the Run Inbound Map window.
- 2 Click the Required tab.
- 3 Select NCPDP Telecommunications.

Literal numeric data mapping [EDI 4.2]

The `-nz` switch maps numeric data (including blank fields) exactly as it appears on the map. Earlier versions of EC RTP padded numeric values with leading zeros are based on field length.

Applies to	Inbound maps.
EC RTP version	4.2
Command line syntax	The syntax for this switch is:

`-nz`

❖ Adding literal mapping options in Windows

- 1 Open the Run Inbound Map window.
- 2 Click the Option 2 tab.
- 3 In the Zero Handling Options group, select No Zero Fill on Null or Non-Null Numbers.

Examples	<p>Different switch options provide different results. These examples assume a numeric field size of 5, and the following input values: '123', '0', '' (empty).</p> <ul style="list-style-type: none"> • Default (no switch) option, the output is: '123', '0' and '0'. • <code>-z</code> switch option, the output is: '00123', '00000' and '' (empty). • <code>-nz</code> switch option, the output is: '123', '0' and '' (empty).
----------	---

Leading zeros in HIPAA compliance maps [EDI 4.2]

The `-clz` switch flags leading zeros in numeric X12 fields as an error on HIPAA compliance maps.

EC RTP flags an error if leading zeros are not used to meet the minimum length requirement of that particular element. If the leading zeros are necessary to meet the minimum length of the element, no error is generated. This switch does not check leading zeros on noncompliance maps.

Applies to	Inbound maps.
EC RTP version	4.2
Command line syntax	The syntax for this switch is:

`-clz`

❖ **Flagging leading zeros in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 2 tab.
- 3 Select Generate Error for Invalid Leading Zeros.

ST control number validation

This section describes control number validation options for inbound compliance maps.

Check for ST control number sequence [EDI 4.2]

The `-as` switch ensures that each ST Transaction Control Number within a GS to GE group is greater than the previous ST Transaction Control Number.

Applies to	Inbound maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is: -as

❖ **Validating ST control numbers in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 2 tab.
- 3 In the Transaction Control Number Check group, select Increasing Control Numbers.

Check for unique ST control numbers [EDI 4.2]

The `-cu` switch checks for unique control numbers within a transaction. ST Transaction Control Numbers can now occur in any sequence, as long as they are unique.

If duplicate control numbers are found, a 6054 error is written to the log.

Applies to	Inbound maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is:

-cu❖ **Validating control numbers in Windows**

- 1 Open the Run Inbound Map window.
- 2 Click the Option 2 tab.
- 3 In the Transaction Control Number Check group, select Unique Control Numbers.

Padding EDI data to minimum length [EDI 4.2]

The `-pe` switch pads EDI data to minimum length in compliance maps, which ensures that data meets minimal field length requirements. This switch is an optional replacement for the PADEDI environment variable.

Using this switch pads the data without changing the value of the data. The `-pe` switch pads alphanumeric fields with trailing spaces if those spaces are required to meet the minimum length of the element. Numeric fields are padded with leading 0s (zeros) if those 0s are required to meet the minimum length of the element.

Applies to	Outbound maps.
ECRTP version	4.2
Command line syntax	The syntax for this switch is:

-pe❖ **Padding EDI data to minimum length in Windows**

- 1 Open the Run Outbound Map window.
- 2 Click the Option 1 tab.
- 3 Select Pad EDI Data to Minimum Length.

Outbound trading partner lookup option [EDI 4.2]

The `-ol` switch triggers a series of lookups against the trading partner database when the ECMAP/EC Gateway Log is used as input. If a trading partner match is found, outbound processing uses the entry to populate the EDI envelope.

Applies to	Outbound maps.
ECRTP version	4.2

Command line syntax The syntax for this switch is:

-ol

❖ **Adding the outbound trading partner lookup option in Windows**

- 1 Open the Run Outbound Map window.
- 2 Click the Option 1 tab.
- 3 Select Lookup Trading Partner from ECRTP Log.

Specifying maximum number of cached ODBC connections [EDI 4.2]

The **-sdb** switch allows you to specify the maximum number of cached ODBC connections. The default value 0 (zero) indicates no ODBC connection caching.

Applies to Inbound and outbound maps.

ECRTP version 4.2

Command line syntax The syntax for this switch is:

-sdb *number of cached connections*

❖ **Specifying connection caches in Windows**

- 1 Open the Run Inbound Map or Run Outbound Map window.
- 2 Click the Option 1 tab.
- 3 In the Number of ODBC Cache box, type the number of cache connections.

Command line switches [EDI 4.2]

Table 5-1 describes the available switches for ODBC and non-ODBC trading partner files for EDI version 4.2 and later. These switches correspond to various RunMap options in ECRTP, EC Gateway, and ECMAP, and apply to both Windows and UNIX platforms.

Table 5-1: EDI 4.2 switches for ODBC and non-ODBC trading partners

Parameter	Description	Inbound/ outbound	Required/ optional	EDI version
-ab	Specifies a new full path and file name to be used in place of a BAD EDI file.	Inbound	Optional	4.2
-ag	Specifies a new full path and file name to be used in place of a GOOD EDI file.	Inbound	Optional	4.2
-al	Handles character checks for compliance maps. Only valid ASCII characters with values 0 — 127 are allowed, and not the extended ASCII characters 128 — 255.	Inbound	Optional	4.2.5
-as	Checks that each ST Transaction Control Number in a GS to GE is greater than the previous ST Transaction Control Number. Validation assumes that control numbers appear in ascending order.	Inbound	Optional	4.2
-clz	Flags leading zeros in numeric X12 fields as an error on HIPAA compliance maps. An error is flagged if leading zeros are not used to meet the minimum length requirement of that particular element. If the leading zeros are necessary to meet the minimum length of the element, no error is generated. This option does not check leading zeros on noncompliance maps.	Inbound	Optional	4.2
Inbound: -cn Outbound: -cn <i>control number</i>	Disables control number lookups and updates when running inbound and outbound maps. For inbound processing, -cn prevents the code from doing any control number updates. Use either -cn or -sc. For outbound processing, -cn <i>control number</i> disables control number lookups and updates when running an outbound map and the value provided for <i>control number</i> is used instead as the control number for the outbound ISA.	Inbound and outbound	Optional	4.2.4
-cu	Checks for unique control numbers within a transaction. Control numbers can now occur in any sequence, as long as they are unique. If duplicate control numbers are found, ECRTP logs a 6054 error. Use only one of -as or -cu for any map run. If both switches are present, ECRTP defaults to the last switch encountered.	Inbound	Optional	4.2
-mn	Passes the map name extension as part of a command line argument. The Map Name Extension works with other map lookup fields to find a correct map.	Inbound and outbound	Optional	4.2

Parameter	Description	Inbound/ outbound	Required/ optional	EDI version
-ncb	Indicates that the EDI file to be processed is an NCPDP batch file. Required for all inbound NCPDP files unless -nct is active.	Inbound	Required for inbound NCPDP batch files	4.2
-nct	Indicates that the EDI file to be processed is an NCPDP telecommunications file. Required for all inbound NCPDP files unless -ncb is active.	Inbound	Required for inbound NCPDP telecommunications files	4.2
-nret	Adds new return codes that provide additional information to the calling program. Based on the return value, the user can determine the next step in the process.	Inbound	Optional for compliance maps only	4.2
-nz	Maps numeric data exactly (including blank fields) as it appears on the map.	Inbound	Optional	4.2
-ol	Triggers a series of lookups against the trading partner database when the ECMAP/EC Gateway Log is used as input. If a trading partner match is found, ECRTP uses the entry to populate the EDI envelope.	Outbound	Optional	4.2
-pe	Pads EDI data to minimum length in compliance maps. This option ensures that data meets minimal field length requirements. This switch is an optional replacement for the PADEDI environment variable.	Outbound	Optional	4.2
-sc	Validates the sequence of the Interchange (ISA) and Group (GS) control numbers. If you use this switch, RTP checks the current control number against the trading partner database to validate the entry. If the entry in the file is not the next sequential entry, an error is reported.	Inbound	Optional	4.2
-sdb	Specifies the maximum number of cached ODBC connections. The default value '0' indicates no ODBC connection caching.	Inbound and outbound	Optional	4.2
-tc <i>NN</i>	Specifies a maximum waiting time, in milliseconds, for ECRTP to establish a database connection. The default <i>NN</i> value for this switch is 1000 milliseconds.	Inbound and outbound	Optional	4.2.5

Parameter	Description	Inbound/ outbound	Required/ optional	EDI version
-vs	Checks the validity of ASCII characters from 0 — 127 against the version of X12. If the character is ^ (circumflex) or ' (accent grave) and the X12 version is prior to 4040, the character is invalid and an error is generated.	Inbound	Optional	4.2.5

ODBC switches [EDI 4.2]

Table 5-2 identifies switches for EDI version 4.2 and later, that are specific to ODBC trading partners and which correspond to the NCPDP trading partner lookup values. You can select only one ODBC trading partner switch for a given map.

Table 5-2: EDI 4.2 switches for ODBC trading partners only

Parameter	Description	Inbound/ outbound	Required/ optional	EDI version
-e14	Performs the trading partner lookup based on Batch Sender. Validates interchange and group data against the trading partner database during inbound processing, and, if any information does not match, ECRTP writes an error to the log and returns an error in the TA1 acknowledgement.	Inbound	Optional	4.2
-e15	Performs the trading partner lookup based on Batch Receiver.	Inbound	Optional	4.2
-e16	Performs the trading partner lookup based on Batch Sender and Receiver.	Inbound	Optional	4.2
-e17	Performs the trading partner lookup based on Bin Number.	Inbound	Optional	4.2
-e18	Performs the trading partner lookup based on Processor Control Number.	Inbound	Optional	4.2
-e19	Performs the trading partner lookup based on Batch Sender and Bin Number.	Inbound	Optional	4.2
-e20	Performs the trading partner lookup based on Batch Sender and Processor Control Number.	Inbound	Optional	4.2
-e21	Performs the trading partner lookup based on Batch Receiver and Bin Number.	Inbound	Optional	4.2
-e22	Performs the trading partner lookup based on Batch Receiver and Processor Control Number.	Inbound	Optional	4.2

Runtime Parameters and Switches

This appendix contains tables that list all the runtime parameters and switches available in EDI Products.

Topic	Page
Introduction	89
Alphabetical listing of parameters and switches for ODBC and non-ODBC trading partner files	90
Parameters and switches for ODBC trading partners only	95
Required parameters	97
Inbound/outbound optional parameters	100
Inbound/outbound optional parameters for ODBC database users	103
Outbound-only optional parameters	103
Inbound-only optional parameters	104
Inbound-only optional trading partner lookup switches	106
Inbound-only optional trading partner lookup switches for ODBC database users	106

Introduction

The following tables contain a listing of the parameters and switches that can be used with the runtime program, ECRTP—whether you run it from a script or invoke it with a function call.

Some optional parameters can be used with both inbound and outbound runs, while other optional parameters can be used for only inbound or only outbound runs.

For each parameter and switch, there is a brief description, an indication of whether the parameter and switch is used for inbound or outbound messages, and an indication of whether it is required or optional.

Alphabetical listing of parameters and switches for ODBC and non-ODBC trading partner files

Table A-1 contains optional and required parameters and switches that are used with both non-ODBC and ODBC trading partner files. Each parameter is described in greater detail and grouped for easier reference in the later tables.

Table A-1: Parameters and switches for trading partner files

Parameter	Description	Inbound/ outbound	Required/ optional
-a	Updates the ISA Out control count field only in the ALL TradePartner record when the interchange envelopes are built.	Outbound	Optional
-ab	Specifies a new full path and file name to be used in place of the BAD EDI file.	Inbound and outbound	Optional
-af	Specifies a new full-path file name to be used in place of a file name embedded in the map file.	Inbound and outbound	Optional
-ag	Specifies a new full path and file name to be used in place of the GOOD EDI file.	Inbound and outbound	Optional
-al	Handles character checks for compliance maps.	Inbound	Optional
-as	Checks for ST control number sequence.	Inbound	Optional
-b	Does not save the rejected EDI transactions and messages into the <i>badtrans.nmt</i> file.	Inbound	Optional
-c	Closes the trace file after every write statement.	Inbound and outbound	Optional
-clz	Flags leading zeros in numeric X12 fields as an error on HIPAA compliance maps.	Inbound	Optional
Inbound: -cn Outbound: -cn <i>control number</i>	Disables control number lookups and updates when running inbound and outbound maps.	Inbound and outbound	Optional
-cu	Checks for unique control numbers within a transaction.	Inbound	Optional
-db	Does not delete outbound temporary files that are created when processing multiple files.	Outbound	Optional
-dg	Specifies the directory in which the map files are located.	Inbound and outbound	Required

Parameter	Description	Inbound/ outbound	Required/ optional
-dm	Specifies the directory in which the trading partner, map, and log files are located. Use -dm to replace -dg and -dt when the trading partner directory and the map directory are the same.	Inbound and outbound	Optional
-dt	Specifies the directory in which the trading partner files are located.	Inbound and outbound	Required for Non-ODBC
-du	Specifies the directory to be used in place of the application directories embedded in the map.	Inbound and outbound	Optional
-dw	Specifies the directory in which the company (<i>wixset.dat</i>) file is located.	Outbound	Optional
-ec	Does not create the transaction log file (<i>translog.in</i> , <i>translog.out</i> , or <i>trlog</i>) or the status file (<i>status.in</i> or <i>status.out</i>).	Inbound and outbound	Optional
-ed	Specifies the directory in which the transaction log file (<i>translog.in</i> , <i>translog.out</i> , or <i>trlog</i>), trace file (<i>incoming.err</i> , <i>outgoing.err</i> , or <i>trnn.dat</i>), and status file (<i>status.in</i> or <i>status.out</i>) are located.	Inbound and outbound	Optional/ Required for Tandem and Stratus
-ee	Ends processing of the EDI file after it processes the specified number of characters.	Inbound	Optional
-ef	Does not create the status file (<i>status.in</i> or <i>status.out</i>)	Inbound and outbound	Optional
-ei	Specifies the full-path file name to be used in place of the input file name embedded in the map.	Outbound	Optional
-el	Specifies the full-path file name to be used for the transaction log.	Inbound and outbound	Optional Required for Tandem and Stratus platforms
-eo	Specifies the full-path file name to be used in place of the output application file name embedded in the map.	Inbound	Optional
-er	Performs the trading partner lookup based on the group receiver. (basic reverse lookup)	Inbound	Optional
-es	Starts processing the EDI file after it has read a specified number of characters.	Inbound	Optional
-et	Specifies the directory in which the trace file, <i>incoming.err</i> or <i>outgoing.err</i> , is located.	Inbound and outbound	Optional

Alphabetical listing of parameters and switches for ODBC and non-ODBC trading partner files

Parameter	Description	Inbound/ outbound	Required/ optional
-eu	Specifies a string variable to be used in place of all but the first character in the application file name (not including the file extension) embedded in the map.	Inbound and outbound	Optional
-ev	Specifies a string variable to be placed in front of the application file name in the map.	Inbound and outbound	Optional/
-id	Specifies a run ID number to be used instead of having the program look it up.	Inbound	Optional
-it	Ignores the trading partner mailbox.	Inbound and outbound	Optional
-k	Sets compliance checking.	Inbound	Optional
-kf	Splits files into multiple files only once.	Outbound	Optional
-l	Writes long trace messages to an error file.	Inbound and outbound	Optional
-m	Runs a specific map (identified by the mapname, which is the file name of the map with no extension) without referencing company or trading partner files.	Inbound	Optional
-mi	Uses stdin in place of a file embedded in the map.	Inbound	Optional
-mm	Uses a temporary memory location in place of a file embedded in the map.	Inbound and outbound	Optional
-mn	Passes the map name extension as part of a command line argument.	Inbound and outbound	Optional
-mo	Uses stdout in place of a file embedded in the map.	Outbound	Optional
-mp	Uses a specific memory address in place of a file embedded in the map.	Inbound and outbound	Optional
-mx	Keeps a specified number of maps open in memory.	Inbound and outbound	Optional
-n	Uses the ALL TradePartner record if no trading partner match is found in the trading partner file.	Inbound and outbound	Optional
-ncb	Indicates that the EDI file to be processed is an NCPDP batch file. Required for all inbound NCPDP files unless -nct is active.	Inbound	Required for inbound NCPDP batch files

Parameter	Description	Inbound/ outbound	Required/ optional
-nct	Indicates that the EDI file to be processed is an NCPDP telecommunications file. Required for all inbound NCPDP files unless -ncb is active.	Inbound	Required for inbound NCPDP telecommunications files
-ne	Does not produce an outbound EDI file.	Outbound	Optional
-nret	Adds new return codes that provide more information to the calling program. Based on the return value, the user can determine the next step in the process.	Inbound for compliance maps only	Optional
-nt	Does not perform a trading partner lookup. Uses the map specified in the command line.	Outbound	Optional
-nz	Maps numeric data literally (including blank fields) as it appears on the map.	Inbound	Optional
-o	Does not use the trade agreement mailbox directory and file name.	Inbound and outbound	Optional
-ol	Triggers a series of look ups against the Trading Partner database when the ECMAP/EC Gateway Log is used as input.	Outbound	Optional
-pe	Pads alphanumeric fields with trailing spaces.	Outbound	Optional
-pf	Uses contents of the file as the command line parameters	Inbound and outbound	Optional
-r	Specifies the maximum number of cross-reference table entries that will be loaded into memory.	Inbound and outbound	Optional
-rb	Ignores the trade agreement mailbox and places routed EDI data in the trading partner BAD mailbox.	Inbound and outbound	Optional
-re	Ignores the trade agreement mailbox and places routed EDI data in the trading partner IN mailbox.	Inbound and outbound	Optional
-rg	Ignores the trade agreement mailbox and places routed EDI data in the trading partner GOOD mailbox.	Inbound and outbound	Optional
-ro	Ignores the trade agreement mailbox and places routed EDI data in the trading partner OUT mailbox.	Inbound and outbound	Optional
-rt	Ignores the trade agreement mailbox and places routed EDI data in the trading partner OTHER mailbox.	Inbound and outbound	Optional

Alphabetical listing of parameters and switches for ODBC and non-ODBC trading partner files

Parameter	Description	Inbound/ outbound	Required/ optional
-s	Does not produce a trace file.	Inbound and outbound	Optional
-s3	Processes the X12 ST03 element.	Inbound and outbound	Optional
-sc	Validates the sequence of the Interchange (ISA) and Group (GS) control numbers.	Inbound	Optional
-sdb	Specifies the maximum number of cached ODBC connections. The default value 0 (zero) indicates no ODBC connection caching.	Inbound and outbound	Optional
-t	Specifies the message/transaction set being mapped.	Outbound	Required
-tc <i>NN</i>	Specifies a maximum waiting time, in milliseconds, for ECRTP to establish a database connection. The default <i>NN</i> value for this switch is 1000 milliseconds.	Inbound and outbound	Optional
-td	Specifies the directory in which the bin files are located if the -du switch is not set. Specifies the directory in which the temporary split files are located.	Inbound and outbound	Optional Optional
-tm	Writes the elapsed runtime to the trace file.	Inbound and outbound	Optional Optional
-u	Does not write EDIFACT UNB and UNG segments.	Outbound	Optional
-vs	Checks the validity of ASCII characters from 0—127 against the version of X12.	Inbound	Optional
-w	Overwrites all application output files. The default is to append the application output files.	Inbound	Optional
-xf	Closes maps that have been left open in memory.	Inbound and outbound	Optional
-xl	Writes the text transaction log file in expanded field length format.	Inbound and outbound	Optional
-xmi	Uses stdin in place of the EDI file in the map.	Inbound	Optional
-xmo	Uses stdout in place of the EDI file in the map.	Outbound	Optional
-xmp	Uses a specific memory address in place of the EDI file in the map.	Inbound and outbound	Optional
-z	Zero-fills numeric fields that contain data.	Inbound and outbound	Optional

Parameters and switches for ODBC trading partners only

Table A-2 contains optional and required UNIX runtime parameters that are used only when the trading partner information is in an ODBC database.

Table A-2: ODBC trading partner parameters/switches

Parameter	Description	Inbound/ outbound	Required/ optional
-ad	Specifies a DSN connect string to be used in place of a specified DSN connect string embedded in the map file.	Inbound and outbound	Optional
-e1	Performs the trading partner lookup based on the group sender and receiver.	Inbound	Optional
-e2	Performs the trading partner lookup based on the group and interchange sender.	Inbound	Optional
-e3	Performs the trading partner lookup based on the group and interchange sender and receiver.	Inbound	Optional
-e4	Performs the trading partner lookup based on the group and interchange receiver.	Inbound	Optional
-e5	Performs the trading partner lookup based on interchange sender.	Inbound	Optional
-e6	Performs the trading partner lookup based on the interchange receiver.	Inbound	Optional
-e7	Performs the trading partner lookup based on the interchange sender and receiver.	Inbound	Optional
-e8	Performs a reverse trading partner lookup based on the interchange sender and receiver.	Inbound	Optional
-e9	Performs a reverse trading partner lookup based on the group and interchange sender and receiver.	Inbound	Optional
-e10	Performs a reverse trading partner lookup based on the interchange receiver.	Inbound	Optional
-e11	Performs a reverse trading partner lookup based on the interchange sender.	Inbound	Optional
-e12	Performs a reverse trading partner lookup based on the group and interchange sender.	Inbound	Optional
-e13	Performs a reverse trading partner lookup based on the group and interchange receiver.	Inbound	Optional
-e14	Performs the trading partner lookup based on Batch Sender. Validates interchange and group data against the Trading Partner database during inbound processing, and, if any information does not match, ECRTP writes an error to the log and returns an error in the TA1 acknowledgement.	Inbound	Optional

Parameters and switches for ODBC trading partners only

Parameter	Description	Inbound/ outbound	Required/ optional
-e15	Performs the trading partner lookup based on Batch Receiver.	Inbound	Optional
-e16	Performs the trading partner lookup based on Batch Sender and Receiver.	Inbound	Optional
-e17	Performs the trading partner lookup based on Bin Number.	Inbound	Optional
-e18	Performs the trading partner lookup based on Processor Control Number.	Inbound	Optional
-e19	Performs the trading partner lookup based on Batch Sender and Bin Number.	Inbound	Optional
-e20	Performs the trading partner lookup based on Batch Sender and Processor Control Number.	Inbound	Optional
-e21	Performs the trading partner lookup based on Batch Receiver and Bin Number.	Inbound	Optional
-e22	Performs the trading partner lookup based on Batch Receiver and Processor Control Number.	Inbound	Optional
-et	Specifies the trace file directory <i>trnn.dat</i> .	Inbound and outbound	Optional
-sl	Specifies a DSN connect string to be used in place of the log database DSN connect string embedded in the map.	Inbound and outbound	Required for ODBC log
-st	Specifies a DSN connect string to be used in place of the trading partner database DSN connect string embedded in the map.	Inbound and outbound	Required for ODBC trading partner
-td	Specifies directory used to make sure information is correctly backed out with Backout Point and Check Point command types of the Check Point command.	Inbound and outbound	Optional
-wx	Uses the record in the WIXSET company ID table where the RECORD_NO field is the same as the specified number (in other words, selects a specific company profile).	Inbound and outbound	Required for ODBC trading partner

Required parameters

Some parameters are required for both inbound and outbound runs, while others are required only for outbound runs or only for inbound runs.

Note For clarity, in each of the syntax examples shown in the following sections, the entire command is divided into separate lines. Each parameter with its flag or switch, is shown on a separate line. You must type the entire command in a single continuous command line.

Outbound required parameters

The normal outbound run has the syntax with these parameters:

```
rmapout full-path EDI output filename
map
transaction/message code
-t transaction/message
-dt full-path trading partner directory
-dg full-path map directory
```

The first three parameters are always required and must be in the following order:

- 1 *full-path EDI output filename*
- 2 *map*
- 3 *transaction/message*

The next three required parameters do not have to be in any specific order, but they must be preceded by the appropriate “-letters” flag.

- **-t** *transaction* – specifies the code of the EDI transaction or message being mapped.
- **-dt** *directory* – specifies the directory that contains trading partner information.
- **-dg** *directory* – specifies the directory that contains the generated files—map files, cross-reference tables, and log files.

When the trading partner directory and the map directory are the same, the **-dm** switch can be used in place of both the **-dt** and **-dg** switches:

rmapout *full-path EDI output filename*
map
transaction/message code
-t *transaction/message*
-dm *full-path trading partner and map directory*

- **-dm** *directory* – specifies the directory that contains trading partner information and generated files; map files, cross-reference tables, and log files.

Outbound required parameters for ODBC database users

The outbound run for ODBC database users has this syntax:

rmapout *full-path EDI output filename*
map
transaction/message code
-t *transaction/message*
-st *trading partner DSN connect string*
-dg *full-path map directory*
-sl *log DSN connect string*

The syntax differs from a normal outbound run because when an ODBC trading partner database is used, this command:

-dt *trading partner directory*

is replaced by:

-st *"DSN=data source name;uid=user ID;pwd=password"*

and when an ODBC log database is used, the following switch must be included:

-sl *"DSN=data source name;uid=user ID;pwd=password"*

Note The DSN pointing to the log database is the only required parameter; uid and pwd are included only if they are required by the database.

- **-st** *trading partner DSN connect string* – specifies the data source name (DSN) connect string for the ODBC trading partner database that contains the trading partner and trade agreement tables. In addition to the DSN, this connect string must include the user ID and password if they are required.
- **-sl** *log DSN connect string* – specifies the data source name (DSN) connect string for the ODBC log database that contains the log tables. In addition to the DSN, this connect string must include the user ID and password if they are required.

Inbound required parameters

The normal inbound run uses this syntax with these parameters:

```
mapinrun full-path EDI input filename
-dt full-path trading partner directory
-dg full-path map directory
```

When the trading partner and the map directory are the same, the command line can be shortened by using the `-dm` directory in their place, as shown below:

```
mapinrun full-path EDI input filename
-dm full-path trading partner and map directory
```

full-path EDI input filename – this parameter is required for an inbound run and must be first in order.

The three other required parameters do not have to be in any specific order, but they must be preceded by the appropriate “-letters” flag.

- `-dt` *directory* – specifies the directory that contains trading partner information.
- `-dg` *directory* – specifies the directory that contains the generated files which are the map files, cross reference tables, and log files.

When the trading partner directory and the map directory are the same, the `-dm` switch can be used in place of both the `-dt` and `-dg` switches.

- `-dm` *directory* – specifies the directory that contains trading partner information and generated files; map files, cross-reference tables, and log files.

Inbound required parameters for ODBC database users

The inbound run for ODBC database users has this syntax:

```
rmapinrun full-path EDI input filename
-st trading partner DSN connect string
-dg full-path map directory
-sl log DSN connect string
```

The syntax differs from a normal inbound run because when an ODBC trading partner database is used:

```
-dt trading partner directory
```

is replaced by:

```
-st “DSN=data source name;uid=user ID;pwd=password”
```

and when an ODBC log database is used, the following switch must be included:

-sl "DSN=*data source name*;uid=*user ID*;pwd=*password*"

Note The DSN pointing to the log database is the only required parameter; uid and pwd are included only if they are required by the database.

- **-st** *trading partner DSN connect string* – specifies the data source name (DSN) connect string for the ODBC trading partner database that contains the trading partner and trade agreement tables. In addition to the DSN, this connect string must include the user ID and password if they are required.
- **-sl** *log DSN connect string* – specifies the data source name (DSN) connect string for the ODBC log database that contains the log tables. In addition to the DSN, this connect string must include the user ID and password if they are required.

Inbound/outbound optional parameters

Table A-3 contains optional parameters that can be used with both inbound and outbound runs.

Table A-3: Inbound/outbound optional parameters

Parameter	Description
-at <i>full-path map filename</i> <i>full-path new filename</i>	Uses the named "new file name" in place of the named "map file name" embedded in the map.
-c	Closes the trace file after every write statement. This is used to ensure that the last trace message is written to disk. This flag impedes processing and should not be used unless a serious problem is encountered and must be debugged.
Inbound: -cn Outbound: -cn <i>control number</i>	Disables control number lookups and updates when running inbound and outbound maps. For inbound processing, -cn prevents the code from doing any control number updates. Use either -cn or -sc. For outbound processing, -cn <i>control number</i> disables control number lookups and updates when running an outbound map and the value provided for <i>control number</i> is used instead as the control number for the outbound ISA.
-du <i>directory</i>	Uses the named directory in place of the application directories embedded in the map.
-ec	Does not create the transaction log file (<i>translog.in</i> , <i>translog.out</i> , or <i>trlog</i>) or the status file (<i>status.in</i> or <i>status.out</i>).

Parameter	Description
-ed <i>directory</i>	Writes the transaction log file (<i>translog.in</i> or <i>translog.out</i>), status file (<i>status.in</i> or <i>status.out</i>), or trace file (<i>incoming.err</i> , <i>outgoing.err</i> , or <i>trnn.dat</i>), to the named directory. Required for Tandem and Stratus; optional for all other versions.
-ef	Does not create the status file (<i>status.in</i> or <i>status.out</i>).
-el	Uses the named full path file name as the transaction log file name. Required for Tandem and Stratus; optional for all other versions.
-el <i>full-path filename</i>	Uses the named directory for the trace file (<i>incoming.err</i> or <i>outgoing.err</i>). If this switch is not used, the trace file is placed in the map directory.
-et <i>directory</i>	Replaces all except the first character in the application file name in the map, excluding the file extension, with the named string variable. This switch is required when multiple copies of the executable are run simultaneously.
-eu <i>string variable</i> (up to 7 characters)	Replaces all except the first character in the application file name in the map, excluding the file extension, with the named string variable. This switch is required when multiple copies of the executable are run simultaneously.
-ev <i>string variable</i> (up to 8 characters)	Places the named string variable in front of all application file names in the map.
-id <i>run ID number</i>	Uses the specified run ID number instead of looking it up. The run ID number can have from one to eight digits. If there is a log database, the program normally looks up the run ID number in the run ID table in the log database.
-it	Ignores the trading partner mailbox. If this switch is not set for inbound maps, a copy of the inbound EDI file is placed in the trading partner IN mailbox and no rules are processed unless the trade agreement records have the EDI_OUT flag set. If this switch is not set for outbound maps, the outbound EDI file is placed in the trading partner OUT mailbox. If a trade agreement mailbox exists and this switch is not set, the trade agreement mailbox overrides the trading partner mailbox.
-l	Turns the long trace on, causing trace messages to be written to an error file. This switch is used for debugging.
-mm <i>full-path filename</i>	Uses a temporary memory location in place of the named file imbedded in the map.
-mn	Passes the map name extension as part of a command line argument. The Map Name Extension works with other map lookup fields to find a correct map. For map functions with multiple parameters, such as LOADMAP, the correct map name should be found by using current map lookup fields with the Map Name Extension before calling the functions.
-mp <i>full-path filename</i> <i>pointer to memory</i> <i>address</i> <i>pointer to # of bytes</i> <i>pointer to size of memory</i> <i>buffer</i>	Uses a specified memory location instead of a specified file during map execution.
-mx <i>number</i>	Keeps a specified number of maps open in memory.

Parameter	Description
-n	Uses the ALL TradePartner if no trading partner match is found in the trading partner file.
-o	Ignores the trade agreement mailbox file name that was set up for routing EDI data.
-pf	Uses the map switches in the named batch file on the command line.
-r	Specifies the maximum number of cross-reference table entries to be loaded into memory. Anything over the maximum must be accessed from the database.
-rb	Ignores the trade agreement mailbox and places the EDI data in the trading partner BAD mailbox on inbound maps and on outbound maps with "Route Bad" selected.
-rg	Ignores the trade agreement mailbox and places the EDI data in the trading partner GOOD mailbox on inbound maps and on outbound maps with "Route Good" selected.
-ri	Ignores the trade agreement mailbox and places the EDI data in the trading partner IN mailbox on inbound maps and on outbound maps with "Route In" selected.
-ro	Ignores the trade agreement mailbox and places the EDI output in the trading partner OUT mailbox on outbound maps and on inbound maps with "Route Out" selected.
-rt	Ignores the trade agreement mailbox and places the EDI data in the trading partner OTHER mailbox on inbound maps and on outbound maps with "Route Other" selected.
-s	Does not produce a trace file.
-tc <i>NN</i>	Specifies a maximum waiting time, in milliseconds, for ECRTTP to establish a database connection. The default <i>NN</i> value for this switch is 1000 milliseconds.
-tm	Writes the elapsed time of execution for the entire run of this command line to the trace file (<i>incoming.err</i> file for inbound maps or <i>outgoing.err</i> file for outbound maps).
-xf	Closes all open maps.
-xf <i>mapname</i>	Closes the specified map.
-xl	Writes the text transaction log file in the expanded field-length format. See "Text transaction log files," in Chapter 7, "Non-ODBC Database and File Formats" in the <i>ECRTTP Reference Guide</i> for the format of the non-ODBC expanded text transaction log file.
-xmp <i>pointer to memory address</i> <i>pointer to # of bytes</i> <i>pointer to size of memory buffer</i>	Indicates that a specified memory location should be used in place of the EDI file during map execution.
-z	Zero-fills numeric fields that contain data. For inbound maps, numeric fields are zero-filled if the EDI element contains data, and left blank if there is no data. For outbound maps, a zero is produced if a zero is contained in the data field.

Inbound/outbound optional parameters for ODBC database users

Table A-4 contains optional parameters for ODBC database users that can be used with both inbound and outbound runs.

Table A-4: Inbound/outbound optional parameters for ODBC database users

Parameter	Description
<i>-ad map DSN connect string new DSN connect string</i>	Specifies the named <i>new DSN connect string</i> to be used in place of the named <i>map DSN connect string</i> in the map file. Connect strings include the data source name and any other required connection information, such as user ID and password.
<i>-et directory</i>	Uses the named directory for the trace file. The trace file name is in the format “ <i>trnn.dat</i> ” where <i>nn</i> is a nonzero run ID number. If this switch is not used, the trace file is placed in the current directory. The run ID number can have from one to eight digits.
<i>-sdb number</i>	Specifies the maximum number of cached ODBC connections. The default value 0 (zero) indicates no ODBC connection caching.
<i>-td directory</i>	Specifies the named directory to be used with the Backout Point and Check Point command types of the Check Point command to make sure that information is correctly backed out. The <i>-td</i> switch has two other uses; one only for inbound and one only for outbound.
<i>-wx number</i>	Uses the record in the <i>wixset</i> company table where the <i>RECORD_NO</i> field is the same as the specified number. Allows a specific company profile to be selected.

Outbound-only optional parameters

Table A-5 contains optional parameters can be used for outbound runs only.

Table A-5: Outbound-only optional parameters

Parameter	Description
<i>-a</i>	Updates the ISA Out control count field only in the All TradePartner record (customer number = 0) when the interchange envelopes are built.
<i>-db</i>	Does not delete outbound temporary files created when processing multiple files (in other words, when the map is set by the user to separate one input file into several input files). These files are not deleted and can be used to diagnose map flow problems. Use this switch when debugging.
<i>-dw directory</i>	Uses the named directory as the location of the <i>wixset.dat</i> company file.

Parameter	Description
-ei <i>full-path filename</i>	Uses the named file name instead of the input application file name embedded in the map.
-kf	Splits files into multiple files only once when processing multiple files (when the map is set by the user to separate one input file into several input files).
-mo <i>full-path filename</i>	Uses stdout in place of the named file imbedded in the map.
-ne	Does not produce an outbound EDI file.
-nt	Does not perform trading partner lookup—Uses the map passed in on the command line.
-ol	Triggers a series of lookups against the trading partner database when the ECMAP/EC Gateway log is used as input. If a trading partner match is found, ECRTTP uses the entry to populate the EDI envelope.
-pe	Pads alphanumeric fields with trailing spaces if those spaces are required to meet the minimum length of the element. Numeric fields are padded with leading 0s if those 0s are required to meet the minimum length of the element. This switch is an optional replacement for PADEDI environmental variable.
-s3 <i>value</i>	Uses the specified value to create the optional ST03 element on outbound X12 EDI transactions.
-td <i>directory</i>	Writes temporary split files to the named directory. Temporary split files (<i>pfs.*</i>) are created when the Multiple Files option is selected, when the map is set by the user to separate one input file into several input files. If this switch is not set, the temporary files are written to the current directory. In earlier versions, temporary files were written to the root directory.
-u	Does not write EDIFACT UNB and UNG segments.
-xmo	Uses stdout in place of the EDI file during map execution.

Inbound-only optional parameters

Table A-6 contains optional parameters can be used for inbound runs only.

Table A-6: Inbound-only optional parameters

Parameter	Description
-ab	Specifies a new full path and file name to be used in place of the BAD EDI file.
-ag	Specifies a new full path and file name to be used in place of the GOOD EDI file.
-al	Handles character checks for compliance maps. Only valid ASCII characters with values 0 — 127 are allowed, and not the extended ASCII characters 128 — 255.
-as	Checks that each ST Transaction Control Number in a GS to GE is greater than the previous ST Transaction Control Number. Validation assumes control numbers appear in ascending order.

Parameter	Description
-b	Does not save the rejected EDI messages/transactions into the <i>badtrans.nmt</i> file.
-clz	Flags leading zeros in numeric X12 fields as an error on HIPAA compliance maps. An error is flagged if leading zeros are not used to meet the minimum length requirement of that particular element. If the leading zeros are necessary to meet the minimum length of the element, no error is generated. This option does not check leading zeros on non-compliance maps.
-cu	Checks for unique control numbers within a transaction. Control numbers can now occur in any sequence, as long as they are unique. If duplicate control numbers are found, ECRTP logs a 6054 error. Use only one of the switches (-as or -cu) for any map run. If both switches are present, ECRTP defaults to the last switch encountered.
-ee <i>number of characters</i>	Ends processing of the EDI file after it processes the specified number of characters. Useful when processing is done by VANS who charge by the byte.
-eo <i>full path filename</i>	Uses the named full path file name in place of the output application file name embedded in the map.
-es <i>number of characters</i>	Starts processing the EDI file after it has read the specified number of characters. Useful when processing is done by VANS who charge by the byte.
-k	Sets compliance checking for: <ul style="list-style-type: none"> • Missing mandatory segments • Exceeding loop counts • Exceeding segment counts • Segments out of sequence • Exceeding standards definition for number of elements in a segment
-m <i>mapname</i>	Runs the map with the specified map name without referencing company or trading partner files. <i>mapname</i> is the file name of the map without the extension. For example, 837IN is the <i>mapname</i> for the map file <i>837IN.map</i> .
-mi <i>full path filename</i>	Uses stdin in place of the named file imbedded in the map.
-ncb	Indicates that the EDI file to be processed is an NCPDP batch file. Required for all inbound NCPDP files unless -nct is active.
-nct	Indicates that the EDI file to be processed is an NCPDP telecommunications file. Required for all inbound NCPDP files unless -ncb is active.
-nret	Adds new return codes that provide more information to the calling program. Based on the return value, the user can determine the next step in the process. The return codes reflect the following information: <ul style="list-style-type: none"> • At least one interchange or group is in error implying a TA1 map should be run. • At least one group or transaction is in error, implying a 997 map should be run. • At least one good transaction is present, implying a translation map should be run.
-nt	Does not perform trading partner lookup—Uses the map passed in on the command line.

Parameter	Description
-nz	Maps numeric data literally (including blank fields) as it appears on the map. Earlier versions of ECRTP pad numeric values with leading zeros based on field length.
-s3	Processes the optional 12 ST03 element in incoming EDI data.
-sc	Validates the sequence of the Interchange (ISA) and Group (GS) control numbers. If you use this command, RTP checks the current control number against the trading partner database to validate the entry. If the entry in the file is not the next sequential entry, an error is reported. If you run an inbound compliance map, choose the Validate Control Number Sequence option to add an -sc switch to the command line options.
-td <i>directory</i>	Writes the bin files to the named directory if the -du switch is not set. All bin files start with <i>bin</i> .
-vs	Checks the validity of ASCII characters from 0 — 127 against the version of X12. If the character is ^ (circumflex) or ' (accent grave) and the X12 version is prior to 4040, the character is invalid and an error is generated.
-w	Overwrites all output map files. The default is to append the map files.
-xmi	Uses stdin in place of the named EDI file during map execution.

Inbound-only optional trading partner lookup switches

Table A-7 contains optional trading partner switches that can be used for inbound runs only.

Table A-7: Trading partner lookup switches – inbound-only

Parameter	Description
-er	Performs a reverse trading partner lookup based on the group receiver code.
-ol	Triggers a series of lookups against the trading partner database when the ECMAP/EC Gateway log is used as input. If a trading partner match is found, ECRTP uses the entry to populate the EDI envelope.

Inbound-only optional trading partner lookup switches for ODBC database users

Table A-8 contains optional trading partner switches for ODBC database users that can be used for inbound runs only.

Table A-8: ODBC trading partner lookup switches – inbound-only

Parameter	Description
-e1	Performs the trading partner lookup based on the group sender and receiver codes.
-e2	Performs the trading partner lookup based on the group sender code and the interchange sender code and qualifier.
-e3	Performs the trading partner lookup based on the group sender and receiver codes and the interchange sender and receiver codes and qualifiers.
-e4	Performs the trading partner lookup based on the group receiver code and the interchange receiver code and qualifier.
-e5	Performs the trading partner lookup based on the interchange sender code and qualifier.
-e6	Performs the trading partner lookup based on the interchange receiver code and qualifier.
-e7	Performs the trading partner lookup based on the interchange sender and receiver codes and qualifiers.
-e8	Performs a reverse trading partner lookup based on the interchange sender and receiver codes and qualifiers.
-e9	Performs a reverse trading partner lookup based on the group sender and receiver codes and the interchange sender and receiver codes and qualifiers.
-e10	Performs a reverse trading partner lookup based on the interchange receiver code and qualifier.
-e11	Performs a reverse trading partner lookup based on the interchange sender code and qualifier.
-e12	Performs a reverse trading partner lookup based on the group sender code and the interchange sender code and qualifier.
-e13	Performs a reverse trading partner lookup based on the group receiver code and the interchange receiver code and qualifier.
-e14	Performs the trading partner lookup based on Batch Sender. Validates interchange and group data against the trading partner database during inbound processing, and, if any information does not match, ECRTP writes an error to the log and returns an error in the TA1 acknowledgement.
-e15	Performs the trading partner lookup based on Batch Receiver.
-e16	Performs the trading partner lookup based on Batch Sender and Receiver.
-e17	Performs the trading partner lookup based on Bin Number.
-e18	Performs the trading partner lookup based on Processor Control Number.
-e19	Performs the trading partner lookup based on Batch Sender and Bin Number.
-e20	Performs the trading partner lookup based on Batch Sender and Processor Control Number.
-e21	Performs the trading partner lookup based on Batch Receiver and Bin Number.
-e22	Performs the trading partner lookup based on Batch Receiver and Processor Control Number.

System Variables

This appendix contains a table of all the system variables available in ECMAP 5.0 and later.

Alphabetical listing of ECMAP system variables

System variables with the suffix “EDIOUT” are the system variables that support the EDI-to-EDI mapping feature introduced with EDI 5.0. See “System variables for outbound EDI-to-EDI transactions” on page 13 for more information.

Table B-1: ECMAP system variables

Name	Size	Type	Description
ABORT_GROUP	1	AN	Flag that specifies that an entire functional group should be aborted, on an outbound map.
ABORT_GROUP_EDIOUT	1	AN	Flag that specifies that an entire functional group should be aborted, on an EDI-to-EDI map.
ABORT_INTERCHANGE	1	AN	Flag that specifies that an entire interchange should be aborted, on an outbound map.
ABORT_INTERCHANGE_EDIOUT	1	AN	Flag that specifies that an entire interchange should be aborted, on an EDI-to-EDI map.
ACK_INTCH	1	AN	<p>Flag that specifies whether an EDIFACT interchange-level acknowledgement is expected from the receiver of an outbound map:</p> <ul style="list-style-type: none"> • 1 = Acknowledgement expected. • (blank) = Acknowledgement not expected. <p>When ACK_INTCH is 1, the program populates the field in the outgoing EDIFACT UNB envelope that requests the receiver to return an interchange-level acknowledgement.</p> <p>Loaded from on outbound: tradstat/ACK_RQSTD.</p> <p>Loaded to on outbound: EDIFACT: UNB S005 0031.</p>

Name	Size	Type	Description
ACK_INTCH_EDIOUS	1	AN	<p>Flag that specifies whether an EDIFACT interchange-level acknowledgement is expected from the receiver of an outbound map:</p> <ul style="list-style-type: none"> • 1 = Acknowledgement expected. • (blank) = Acknowledgement not expected. <p>When ACK_INTCH is 1, the program populates the field in the outgoing EDIFACT UNB envelope that requests the receiver to return an interchange-level acknowledgement.</p> <p>Loaded from an EDI-to-EDI map: tradstat/ACK_RQSTD.</p> <p>Loaded to an EDI-to-EDI map: EDIFACT: UNB S005 0031.</p>
ACK_MSG	1	AN	<p>Flag that specifies whether an EDIFACT message-level acknowledgement is expected from the receiver of an outbound map:</p> <ul style="list-style-type: none"> • 1 = Acknowledgement expected. • (blank) = Acknowledgement not expected.
ACK_MSG_EDIOUS	1	AN	<p>Flag that specifies whether an EDIFACT message-level acknowledgement is expected from the receiver of an EDI-to-EDI map:</p> <ul style="list-style-type: none"> • 1 = Acknowledgement expected. • (blank) = Acknowledgement not expected.
ACK_REQSTD	1	AN	<p>Flag that specifies whether an X12 TA1 interchange-level acknowledgement is expected from the receiver of an outbound map:</p> <ul style="list-style-type: none"> • 1 = TA1 acknowledgement expected. • (blank) = TA1 acknowledgement not expected. <p>When ACK_REQSTD is 1, the program populates the field in the outgoing X12 ISA envelope that requests the receiver to return an interchange-level acknowledgement.</p> <p>Loaded from on outbound: tradstat/ACK_RQSTD.</p> <p>Loaded to on outbound: X12: ISA 14.</p>

Name	Size	Type	Description
ACK_REQSTD_EDIOUS	1	AN	<p>Flag that specifies whether an X12 TA1 interchange-level acknowledgement is expected from the receiver of an outbound map:</p> <ul style="list-style-type: none"> • 1 = TA1 acknowledgement expected. • (blank) = TA1 acknowledgement not expected. <p>When ACK_REQSTD is 1, the program populates the field in the outgoing X12 ISA envelope that requests the receiver to return an interchange-level acknowledgement.</p> <p>Loaded from an EDI-to-EDI map: tradstat/ACK_RQSTD.</p> <p>Loaded to an EDI-to-EDI map: X12: ISA 14.</p>
APP_PASSWORD	14	AN	<p>Password on the group-level envelope.</p> <p>Loaded to on outbound/loaded from on inbound: EDIFACT: UNB S005 0022.</p>
APP_PASSWORD_EDIOUS	14	AN	<p>Password on the group-level envelope.</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNB S005 0022.</p>
APP_RECV_CODE	35	AN	<p>Receiver ID code on the group-level envelope.</p> <p>Loaded from on outbound: tp/GSID (if blank, use tradstat/RCV_GSID).</p> <p>Loaded to on outbound/loaded from on inbound: X12: GS 03.</p> <p>EDIFACT: UNB S003 0046. UNG S007 0044.</p> <p>HL7: MSH 0006. FHS 0072. BHS 0086.</p>
APP_RECV_CODE_EDIOUS	35	AN	<p>Receiver ID code on the group-level envelope.</p> <p>Loaded from an EDI-to-EDI map: tp/GSID (if blank, use tradstat/RCV_GSID).</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: X12: GS 03.</p> <p>EDIFACT: UNB S003 0046. UNG S007 0044.</p> <p>HL7: MSH 0006. FHS 0072. BHS 0086.</p>
APP_RECV_QUAL	4	AN	<p>Receiver ID code qualifier on the group-level envelope.</p> <p>Loaded to on outbound/loaded from on inbound: EDIFACT: UNG S007 0007.</p>
APP_RECV_QUAL_EDIOUS	4	AN	<p>Receiver ID code qualifier on the group-level envelope.</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNG S007 0007.</p>

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
APP_SEND_CODE	35	AN	Sender ID code on the group-level envelope. Loaded from on outbound: tp/SND_GSID (if blank, use wixset/WIX_GSID). Loaded to on outbound/loaded from on inbound: X12: GS 02. EDIFACT: UNB S002 0042. UNG S006 0040. HL7: MSH 0004. FHS 0070. BHS 0084.
APP_SEND_CODE_EDIOUT	35	AN	Sender ID code on the group-level envelope. Loaded from on EDI-to-EDI outbound: tp/SND_GSID (if blank, use wixset/WIX_GSID). Loaded to an EDI-to-EDI map/loaded from on inbound: X12: GS 02. EDIFACT: UNB S002 0042. UNG S006 0040. HL7: MSH 0004. FHS 0070. BHS 0084.
APP_SEND_QUAL	4	AN	Sender ID code qualifier on the group-level envelope. Loaded to on outbound/loaded from on inbound: EDIFACT: UNG S006 0007.
APP_SEND_QUAL_EDIOUT	4	AN	Sender ID code qualifier on the group-level envelope. Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNG S006 0007.
ASSOC_CODE	6	AN	Associate assigned code.
ASSOC_CODE_EDIOUT	6	AN	Associate assigned code.
AUTH_CODE	15	AN	Trading partner authorization ID code. Loaded from on outbound: tp/AUTH_CODE. Loaded to on outbound/loaded from on inbound: X12: ISA 02.
AUTH_CODE_EDIOUT	15	AN	Trading partner authorization ID code. Loaded from an EDI-to-EDI map: tp/AUTH_CODE. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 02.
AUTH_QUAL	2	AN	Trading partner authorization ID code qualifier. Loaded from on outbound: tp/AUTH_QUAL (if blank, use "00" for X12 and null for EDIFACT). Loaded to on outbound/loaded from on inbound: X12: ISA 01.

Name	Size	Type	Description
AUTH_QUAL_EDIOUT	2	AN	Trading partner authorization ID code qualifier. Loaded from an EDI-to-EDI map: tp/AUTH_QUAL (if blank, use "00" for X12 and null for EDIFACT). Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 01.
B_RECV_ID	24	AN	Batch receiver ID.
B_RECV_ID_EDIOUT	24	AN	Batch receiver ID.
B_SEND_ID	24	AN	Batch sender ID.
B_SEND_ID_EDIOUT	24	AN	Batch sender ID.
B_VERS	2	AN	Batch version.
B_VERS_EDIOUT	2	AN	Batch version.
BIN_NUMB	6	AN	Bin number.
BIN_NUMB_EDIOUT	6	AN	Bin number.
CHANGE_LEVEL	4	UI	Change outbound flow level. Assigned by user on outbound. When the user assigns a new value to CHANGE_LEVEL in a rule command, the program first processes the rule, then goes to the new level specified in CHANGE_LEVEL, and finally resets the value of CHANGE_LEVEL to spaces. (CHANGE_LEVEL and COMBINED_BREAK are used in rule commands to control map flow processing on outbound maps.)
CHANGE_LEVEL_EDIOUT	4	UI	Change EDI-to-EDI map flow level. Assigned by user on outbound. When the user assigns a new value to CHANGE_LEVEL in a rule command, the program first processes the rule, then goes to the new level specified in CHANGE_LEVEL, and finally resets the value of CHANGE_LEVEL to spaces. (CHANGE_LEVEL and COMBINED_BREAK are used in rule commands to control map flow processing on EDI-to-EDI maps.)
CLIST_VER	6	AN	Code List Directory Version Number. EDIFACT: UNH S009 0110.
CLIST_VER_EDIOUT	6	AN	Code List Directory Version Number. EDIFACT: UNH S009 0110.
CNT_AG1	3	AN	Controlling Agency, Coded. EDIFACT: UNH S009 0051.
CNT_AG1_EDIOUT	3	AN	Controlling Agency, Coded. EDIFACT: UNH S009 0051.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
CNT_AG2	3	AN	Controlling Agency, Coded. EDIFACT: UNH S016 0051.
CNT_AG2_EDIOUS	3	AN	Controlling Agency, Coded. EDIFACT: UNH S016 0051.
CNT_AG3	3	AN	Controlling Agency, Coded. EDIFACT: UNH S017 0051.
CNT_AG3_EDIOUS	3	AN	Controlling Agency, Coded. EDIFACT: UNH S017 0051.
CNT_AG4	3	AN	Controlling Agency, Coded. EDIFACT: UNH S018 0051.
CNT_AG4_EDIOUS	3	AN	Controlling Agency, Coded. EDIFACT: UNH S018 0051.
COMBINED_BREAK	1	UI	<p>End-of-file (EOF) indicator.</p> <p>Used in the master level I/O rule to do EOF processing if COMBINED_BREAK = 1 and (with CHANGE_LEVEL) to help control map flow processing when a parent level has multiple child levels that repeat in an undetermined order.</p> <p>Generated by the program on outbound: After a successful read, COMBINED_BREAK is set to blank. After an unsuccessful read (record type changes when reading a file with multiple record types), COMBINED_BREAK is set to 1.</p> <p>User-assigned value on inbound: When this variable is set to 1 and EOF is reached for an inbound EDI file, the user can create a rule and attach it to the end-of-outer-envelope flow level to clean up processing.</p>

Name	Size	Type	Description
COMBINED_BREAK_EDIOUT	1	UI	<p>End-of-file (EOF) indicator.</p> <p>Used in the master level I/O rule to do EOF processing if COMBINED_BREAK = 1 and (with CHANGE_LEVEL) to help control map flow processing when a parent level has multiple child levels that repeat in an undetermined order.</p> <p>Generated by the program on EDI-to-EDI map: After a successful read, COMBINED_BREAK is set to blank. After an unsuccessful read (record type changes when reading a file with multiple record types), COMBINED_BREAK is set to 1.</p> <p>User-assigned value on inbound: When this variable is set to 1 and EOF is reached for an inbound EDI file, the user can create a rule and attach it to the end-of-outer-envelope flow level to clean up processing.</p>
COMM_AC_REF	35	AN	<p>Common access reference.</p> <p>User-assigned value for outbound use.</p> <p>Loaded to on outbound/loaded from on inbound: EDIFACT: UNB S005 0026.</p>
COMM_AC_REF_EDIOUT	35	AN	<p>Common access reference.</p> <p>User-assigned value for EDI-to-EDI map use.</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNB S005 0026.</p>
COMM_AGM_ID	35	AN	<p>Communication agreement ID.</p> <p>User-assigned value for outbound use.</p> <p>Loaded to on outbound/loaded from on inbound: EDIFACT: UNB S005 0032.</p>
COMM_AGM_ID_EDIOUT	35	AN	<p>Communication agreement ID.</p> <p>User-assigned value for EDI-to-EDI outbound use.</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNB S005 0032.</p>
CONTACT1	35	AN	<p>Name of trading partner contact #1.</p> <p>Loaded from on inbound: tp/CONTACT1.</p>
CONTACT1_EDIOUT	35	AN	<p>Name of trading partner contact #1.</p> <p>Loaded from on inbound: tp/CONTACT1.</p>
CONTACT2	35	AN	<p>Name of trading partner contact #2.</p> <p>Loaded from on inbound: tp/CONTACT2.</p>
CONTACT2_EDIOUT	35	AN	<p>Name of trading partner contact #2.</p> <p>Loaded from on inbound: tp/CONTACT2.</p>

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
DATA_ELEM_SEP	1	AN	<p>X12 element delimiter.</p> <p>Loaded from on outbound: tp/ELE_DELIMT (if blank, the program uses the default).</p> <p>Loaded from on inbound: ISA Column 4.</p>
DATA_ELEM_SEP_EDIOUS	1	AN	<p>X12 element delimiter.</p> <p>Loaded from an EDI-to-EDI map: tp/ELE_DELIMT (if blank, the program uses the default).</p> <p>Loaded from on inbound: ISA Column 4.</p>
EDFT_A_COMP_SEP	1	AN	<p>EDIFACT composite element delimiter – syntax A. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from on outbound/loaded to on inbound: tp/EDIF_SUBDL (if blank, the program uses the default).</p> <p>Loaded to on outbound/loaded from on inbound: UNA Column 4.</p> <p>Changing this value affects the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.</p>
EDFT_A_COMP_SEP_EDIOUS	1	AN	<p>EDIFACT composite element delimiter – syntax A. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from an EDI-to-EDI map/loaded to on inbound: tp/EDIF_SUBDL (if blank, the program uses the default).</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 4.</p> <p>Changing this value affects the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.</p>
EDFT_A_DEC_IND	1	AN	<p>EDIFACT decimal indicator – syntax A. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from on outbound/loaded to on inbound: “.” (decimal point).</p> <p>Loaded to on outbound/loaded from on inbound: UNA Column 6.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>

Name	Size	Type	Description
EDFT_A_DEC_IND_EDIOUS	1	AN	<p>EDIFACT decimal indicator – syntax A. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from an EDI-to-EDI map/loaded to on inbound: “.” (decimal point).</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 6.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_A_ELEM_SEP	1	AN	<p>EDIFACT element delimiter – syntax A. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from on outbound/loaded to on inbound: tp/EDIF_ELEDL (if blank, the program uses the default).</p> <p>Loaded to on outbound/loaded from on inbound: UNA Column 5.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_A_ELEM_SEP_EDIOUS	1	AN	<p>EDIFACT element delimiter – syntax A. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from an EDI-to-EDI map/loaded to on inbound: tp/EDIF_ELEDL (if blank, the program uses the default).</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 5.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>

Name	Size	Type	Description
EDFT_A_REL_IND	1	AN	<p>EDIFACT release character – syntax A. (The syntax identifier is in UNB S001 0002.) Loaded from on outbound/loaded to on inbound: tp/EDIF_RELCHL (if blank, the program uses the default). Loaded to on outbound/loaded from on inbound: UNA Column 7. (Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_A_REL_IND_EDIOUS	1	AN	<p>EDIFACT release character – syntax A. (The syntax identifier is in UNB S001 0002.) Loaded from an EDI-to-EDI map/loaded to on inbound: tp/EDIF_RELCHL (if blank, the program uses the default). Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 7. (Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_A_SEG_TERM	1	AN	<p>EDIFACT segment delimiter – syntax A. (The syntax identifier is in UNB S001 0002.) Loaded from on outbound/loaded to on inbound: tp/EDIF_SEGDL (if blank, the program uses the default). Loaded to on outbound/loaded from on inbound: UNA Column 8. (Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>

Name	Size	Type	Description
EDFT_A_SEG_TERM_EDIOUT	1	AN	<p>EDIFACT segment delimiter – syntax A. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from an EDI-to-EDI map/loaded to on inbound: tp/EDIF_SEGDL (if blank, the program uses the default). Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 8.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_B_COMP_SEP	1	AN	<p>EDIFACT composite element delimiter – syntax B. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from on outbound/loaded to on inbound: tp/EDIF_SUBDL (if blank, the program uses the default). Loaded to on outbound/loaded from on inbound: UNA Column 4.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_B_COMP_SEP_EDIOUT	1	AN	<p>EDIFACT composite element delimiter – syntax B. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from an EDI-to-EDI map/loaded to on inbound: tp/EDIF_SUBDL (if blank, the program uses the default). Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 4.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>

Name	Size	Type	Description
EDFT_B_DEC_IND	1	AN	<p>EDIFACT decimal indicator – syntax B. (The syntax identifier is in UNB S001 0002.) Loaded from on outbound/loaded to on inbound: “.” (decimal point). Loaded to on outbound/loaded from on inbound: UNA Column 6. (Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_B_DEC_IND_EDIOUS	1	AN	<p>EDIFACT decimal indicator – syntax B. (The syntax identifier is in UNB S001 0002.) Loaded from an EDI-to-EDI map/loaded to on inbound: “.” (decimal point). Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 6. (Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_B_ELEM_SEP	1	AN	<p>EDIFACT element delimiter – syntax B. (The syntax identifier is in UNB S001 0002.) Loaded from on outbound/loaded to on inbound: tp/EDIF_ELEDL (if blank, the program uses the default). Loaded to on outbound/loaded from on inbound: UNA Column 5. (Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_B_ELEM_SEP_EDIOUS	1	AN	<p>EDIFACT element delimiter – syntax B. (The syntax identifier is in UNB S001 0002.) Loaded from an EDI-to-EDI map/loaded to on inbound: tp/EDIF_ELEDL (if blank, the program uses the default). Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 5. (Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>

Name	Size	Type	Description
EDFT_B_REL_IND	1	AN	<p>EDIFACT release character – syntax B. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from on outbound/loaded to on inbound: tp/EDIF_RELCHL (if blank, the program uses the default).</p> <p>Loaded to on outbound/loaded from on inbound: UNA Column 7.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_B_REL_IND_EDIOUT	1	AN	<p>EDIFACT release character – syntax B. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from an EDI-to-EDI map/loaded to on inbound: tp/EDIF_RELCHL (if blank, the program uses the default).</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 7.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDFT_B_SEG_TERM	1	AN	<p>EDIFACT segment delimiter – syntax B. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from on outbound/loaded to on inbound: tp/EDIF_SEGDL (if blank, the program uses the default).</p> <p>Loaded to on outbound/loaded from on inbound: UNA Column 8.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>

Name	Size	Type	Description
EDFT_B_SEG_TERM_EDIOUT	1	AN	<p>EDIFACT segment delimiter – syntax B. (The syntax identifier is in UNB S001 0002.)</p> <p>Loaded from an EDI-to-EDI map/loaded to on inbound: tp/EDIF_SEGDL (if blank, the program uses the default). Loaded to an EDI-to-EDI map/loaded from on inbound: UNA Column 8.</p> <p>(Changing this value does not affect on the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)</p>
EDI_OUT_FILENAME	80	AN	<p>Read only. EDI output file name.</p> <p>On inbound: The file to which EDI data that is passed through is written and which is generated by the program using the run ID number.</p> <p>On outbound: The file to which the mapped EDI data is written. (This is parameter 1 on the command line or the entry in the Output EDI File text box on the Required tab of the Run Outbound Map window.)</p>
EDI_OUT_FILENAME_EDIOUT	80	AN	<p>Read only. EDI output file name.</p> <p>On inbound: The file to which EDI data that is passed through is written and which is generated by the program using the run ID number.</p> <p>On EDI-to-EDI map: The file to which the mapped EDI data is written. (This is parameter 1 on the command line or the entry in the Output EDI File text box on the Required tab of the Run Outbound Map window.)</p>
EDI_START_FILENAME	160	AN	<p>Input/output EDI file name.</p> <p>On inbound: The file from which the EDI input data is read. (This is the name of the EDI input file specified in the Input EDI File text box on the Required tab of the Run Inbound Map window or parameter 1on the command line.)</p> <p>On outbound: The file to which the mapped EDI data is written. (This is the name of the EDI input file specified in the Output EDI File text box on the Required tab of the Run Any-to-Any Map window or parameter 1on the command line.)</p>

Name	Size	Type	Description
EDI_START_FILENAME_EDIOUT	160	AN	<p>Input/ output EDI-to-EDI file name.</p> <p>On inbound: The file from which the EDI input data is read. (This is the name of the EDI input file specified in the Input EDI File text box on the Required tab of the Run Inbound Map window or parameter 1 on the command line.)</p> <p>On EDI-to-EDI map: The file to which the mapped EDI data is written. (This is the name of the EDI input file specified in the Output EDI File text box on the Required tab of the Run Any-to-Any Map window or parameter 1 on the command line.)</p>
ERR_OCCURRED	1	AN	<p>Flag to indicate if transaction errors occurred:</p> <ul style="list-style-type: none"> • Y = errors occurred • N = no errors occurred <p>Generated by the program.</p> <p>Users can perform an action based on the value.</p>
ERR_OCCURRED_EDIOUT	1	AN	<p>Flag to indicate if transaction errors occurred:</p> <ul style="list-style-type: none"> • Y = errors occurred • N = no errors occurred <p>Generated by the program.</p> <p>Users can perform an action based on the value.</p>
FIRST_LAST_TRAN	1	AN	<p>First and last transfers.</p> <p>Loaded to on outbound/loaded from on inbound: EDIFACT: S010 0073.</p>
FIRST_LAST_TRAN_EDIOUT	1	AN	<p>First and last transfers.</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: S010 0073.</p>
FORM_FEED	1	AN	<p>Form feed character – Ctrl+L.</p> <p>Loaded by the program.</p> <p>This unprintable character is available for users when they create reports, such as writing out the records in a file or the fields in a record. They can assign this system variable to an appropriate field in a record to execute form-feed action.</p>
FUNC_GP_DATE	8	Date	<p>Date on group-level envelopes.</p> <p>Loaded with on outbound: Computer date.</p> <p>Loaded to an outbound/loaded from on inbound: X12: GS 04.</p> <p>EDIFACT: UNG S004 0017.</p>

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
FUNC_GP_DATE_EDIOUT	8	Date	Date on group-level envelopes. Loaded with an EDI-to-EDI map: Computer date. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: GS 04. EDIFACT: UNG S004 0017.
FUNC_GP_NUM	35	AN	Control/reference number on group-level envelopes. Loaded from on outbound: Tradstat/GS_NO. Loaded to on outbound/loaded from on inbound: X12: GS 06. EDIFACT: UNG S004 0048.
FUNC_GP_NUM_EDIOUT	35	AN	Control/reference number on group-level envelopes. Loaded from an EDI-to-EDI map: Tradstat/GS_NO. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: GS 06. EDIFACT: UNG S004 0048.
FUNC_GP_TIME	6	Time	Time on group-level envelopes. Loaded with on outbound: Computer time. Loaded to on outbound/loaded from on inbound: X12: GS 05. EDIFACT: UNG S004 0019.
FUNC_GP_TIME_EDIOUT	6	Time	Time on group-level envelopes. Loaded with an EDI-to-EDI map: Computer time. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: GS 05. EDIFACT: UNG S004 0019.
FUNC_ID	6	AN	Functional identifier code. Loaded to on outbound/loaded from on inbound: X12: GS 01.
FUNC_ID_EDIOUT	6	AN	Functional identifier code. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: GS 01.
FUNC_ST_CHG	35	AN	Indicator that the transaction has changed. Generated by the program to: Trigger a trading partner lookup and potentially create a new ISA or GS envelope (used with FUNC_ST_REF).

Name	Size	Type	Description
FUNC_ST_REF	35	AN	Value of ST 03. Supplied by the user on outbound as the <i>value</i> in the -s3 switch. Loaded to on outbound/loaded from on inbound: X12: ST 03.
FUNC_ST_REF_EDIOUT	35	AN	Value of ST 03. Supplied by the user an EDI-to-EDI map as the <i>value</i> in the -s3 switch. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ST 03.
INB_CURR_SEG	3	AN	Inbound current segment. Captured by the program each time it does an inbound read. Users can use the value of this system variable when they are performing additional compliance checks.
INB_PRIOR_SEG	3	AN	Inbound prior segment. Captured by the program each time it does an inbound read. Users can use the value of this system variable when they are performing additional compliance checks.
INT_FORMAT	4	AN	This is the Interchange Header Segment format. This is the value in ISA11. It is used for inbound purposes.
INT_FORMAT_EDIOUT	4	AN	This is the Interchange Header Segment format. This is the value in ISA11. It is used for inbound purposes.
INT_HEAD_DATE	8	Date	Date on interchange-level envelopes. Loaded with on outbound: Computer date. Loaded to on outbound/loaded from on inbound: X12: ISA 09. EDIFACT: UNB S004 0017.
INT_HEAD_DATE_EDIOUT	8	Date	Date on interchange-level envelopes. Loaded with an EDI-to-EDI map: Computer date. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 09. EDIFACT: UNB S004 0017.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
INT_HEAD_NUM	35	AN	Control/reference number on interchange-level envelopes. Loaded from on outbound: Tradstat/GS_NO. Loaded to on outbound/loaded from on inbound: X12: ISA 13. EDIFACT: UNB S004 0048.
INT_HEAD_NUM EDIOUT	35	AN	Control/reference number on interchange-level envelopes. Loaded from an EDI-to-EDI map: Tradstat/GS_NO. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 13. EDIFACT: UNB S004 0048.
INT_HEAD_TIME	6	Time	Time on interchange-level envelopes. Loaded with on outbound: Computer time. Loaded to on outbound/loaded from on inbound: X12: ISA 10. EDIFACT: UNB S004 0019.
INT_HEAD_TIME EDIOUT	6	Time	Time on interchange-level envelopes. Loaded with an EDI-to-EDI map: Computer time. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 10, EDIFACT: UNB S004 0019.
INT_VERSION	5	AN	EDI version number. Loaded from on inbound: X12: ISA 12. EDIFACT: UNB S001 0001 and 0002. Loaded to on inbound: tradstat/VERS (first 5 characters).
INT_VERSION EDIOUT	5	AN	EDI version number. Loaded from on inbound: X12: ISA 12. EDIFACT: UNB S001 0001 and 0002. Loaded to on inbound: tradstat/VERS (first 5 characters).
ISA_TYPE	5	AN	EDI Standard used in the map: <ul style="list-style-type: none"> • (blank) = X12 standard. • 70 = EDIFACT standard. Loaded from on outbound and inbound: tradstat/ISA_TYPE.

Name	Size	Type	Description
ISA_TYPE_EDIOUT	5	AN	EDI Standard used in the map: <ul style="list-style-type: none"> (blank) = X12 standard. 70 = EDIFACT standard. Loaded from on outbound and an inbound: tradstat/ISA_TYPE.
LOG_BYTES	5	UI	Size of the message/transaction. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_BYTES_EDIOUT	5	UI	Size of the message/transaction. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_ELEMENT	2	UI	EDI element name. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_ELEMENT_EDIOUT	2	UI	EDI element name. Loaded from log file on inbound and outbound. Available for interrogation by users
LOG_ERRS	5	UI	Number of errors in the message/transaction. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_ERRS_EDIOUT	5	UI	Number of errors in the message/transaction. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_FIELDNAME	15	AN	Field name. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_FIELDNAME_EDIOUT	15	AN	Field name. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_LEVEL	5	UI	Flow level number. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_LEVEL_EDIOUT	5	UI	Flow level number. Loaded from log file on inbound and outbound. Available for interrogation by users.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
LOG_MSG_NO	5	UI	Error message number. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_MSG_TXT	45	AN	Text of the error message. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_READ_CNT	9	UI	Number of records read. Loaded from log file on inbound and outbound. Available for interrogation by users. Inbound: Number of EDI file records read. Outbound: Number of records read for record type.
LOG_READ_CNT_EDIOUT	9	UI	Number of records read. Loaded from log file on inbound and EDI-to-EDI outbound. Available for interrogation by users. Inbound: Number of EDI file records read. EDI-to-EDI map: Number of records read for record type.
LOG_RECNAME	10	AN	Record field name. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_RECNAME_EDIOUT	10	AN	Record field name. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_RPT_NO	2	UI	Log repeat number.
LOG_RPT_NO_EDIOUT	2	UI	Log repeat number.
LOG_SEGMENT	3	AN	EDI segment name. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_SEGMENT_EDIOUT	3	AN	EDI segment name. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_STATUS	1	AN	Log status code. Loaded from log file on inbound and outbound. Available for interrogation by users.

Name	Size	Type	Description
LOG_STATUS_EDIOUS	1	AN	Log status code. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_SUBELEMENT	2	UI	EDI sub-element number. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_SUBELEMENT_EDIOUS	2	UI	EDI sub-element number. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_VALUE	30	AN	Value of record/field or EDI element when an error occurred. Loaded from log file on inbound and outbound. Available for interrogation by users.
LOG_VALUE_EDIOUS	30	AN	Value of record/field or EDI element when an error occurred. Loaded from log file on inbound and outbound. Available for interrogation by users.
MSG_IMPID	14	AN	Message Implementation Guide Identification. EDIFACT: UNH S017 0121.
MSG_IMPID_EDIOUS	14	AN	Message Implementation Guide Identification. EDIFACT: UNH S017 0121.
MSG_IMPREL	3	AN	Message Implementation Guide Release Number. EDIFACT: UNH S017 0124.
MSG_IMPREL_EDIOUS	3	AN	Message Implementation Guide Release Number. EDIFACT: UNH S017 0124.
MSG_IMPVER	3	AN	Message Implementation Guide Version Number. EDIFACT: UNH S017 0122.
MSG_IMPVER_EDIOUS	3	AN	Message Implementation Guide Version Number. EDIFACT: UNH S017 0122.
MSG_SUBID	14	AN	Message Subset Identification. EDIFACT: UNH S016 0115.
MSG_SUBID_EDIOUS	14	AN	Message Subset Identification. EDIFACT: UNH S016 0115.
MSG_SUBREL	3	AN	Message Subset Release Number. EDIFACT: UNH S016 0118.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
MSG_SUBREL_EDIOUS	3	AN	Message Subset Release Number. EDIFACT: UNH S016 0118.
MSG_SUBVER	3	AN	Message Subset Version Number. EDIFACT: UNH S016 0116.
MSG_SUBVER_EDIOUS	3	AN	Message Subset Version Number. EDIFACT: UNH S016 0116.
MSG_TYPE	6	AN	Message Type Subfunction. EDIFACT: UNH S009 0113.
MSG_TYPE_EDIOUS	6	AN	Message Type Subfunction. EDIFACT: UNH S009 0113.
NCPDP_BATCH_MSG	35	AN	Batch trailer message.
NCPDP_BATCH_MSG_EDIOUS	35	AN	Batch trailer message.
NEW_COMPANY_ID	35	AN	Set to create a new ISA.
NEXT_XML_TAG	40	AN	Value of the XML start tag following the last successfully read XML start tag. Captured by the program. Since users must build all XML read statements manually, this value allows users to know how to process the current start tag. (In other words, it allows users to know whether the following tag continues down the XPath, stays at the same place on the XPath, or goes back up the XPath.)
NEXT_XML_TAG_EDIOUS	40	AN	Value of the XML start tag following the last successfully read XML start tag. Captured by the program. Since users must build all XML read statements manually, this value allows users to know how to process the current start tag. (In other words, it allows users to know whether the following tag continues down the XPath, stays at the same place on the XPath, or goes back up the XPath.)
OUTPUT_FILENAME	60	AN	File name opened for Write. Captured by the program. Inbound: Map or Runtime override.
OUTPUT_FILENAME_EDIOUS	60	AN	File name opened for Write. Captured by the program. Inbound: Map or Runtime override.
PROC_NUMB	10	AN	Processing control number.

Name	Size	Type	Description
PROC_NUMB_EDIOUS	10	AN	Processing control number.
PROC_PRIO_CODE	1	AN	Processing priority code. Loaded to on outbound/loaded from on inbound: EDIFACT: UNB S005 0029.
PROC_PRIO_CODE_EDIOUS	1	AN	Processing priority code. Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNB S005 0029.
PROPRIETARY_LENGTH	3	AN	Length of proprietary header.
PROPRIETARY_LENGTH_EDIOUS	3	AN	Length of proprietary header.
PROPRIETARY_TEXT	250	AN	Proprietary header information.
PROPRIETARY_TEXT_EDIOUS	250	AN	Proprietary header information.
RCV_CODE	35	AN	Interchange/Outer Envelope Receiver ID code. Loaded from on outbound: tp/IDCODE (if blank, use tradstat/RCV_IDCODE). Loaded to on outbound/loaded from on inbound: X12: ISA 08. EDIFACT: UNB S003 0010.
RCV_CODE_EDIOUS	35	AN	Interchange/Outer Envelope Receiver ID code. Loaded from an EDI-to-EDI map: tp/IDCODE (if blank, use tradstat/RCV_IDCODE). Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 08. EDIFACT: UNB S003 0010.
RCV_QUAL	4	AN	Interchange/Outer Envelope Receiver ID code qualifier. Loaded from on outbound: tp/IDQUAL (if blank, use tradstat/RCV_IDQUAL) (Also if blank, use "00" for X12 and null for EDIFACT.) Loaded to on outbound/loaded from on inbound: X12: ISA 07 (max length 4). EDIFACT: UNB S003 0007.
RCV_QUAL_EDIOUS	4	AN	Interchange/Outer Envelope Receiver ID code qualifier Loaded from an EDI-to-EDI map: tp/IDQUAL (if blank, use tradstat/RCV_IDQUAL). (Also if blank, use "00" for X12 and null for EDIFACT.) Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 07 (max length 4). EDIFACT: UNB S003 0007.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
RECV_ROUTE	14	AN	Interchange/Outer Envelope Receiver Routing Code. Loaded to on outbound/loaded from on inbound: EDIFACT: UNB S003 0014.
RECV_ROUTE_EDIOUT	14	AN	Interchange/Outer Envelope Receiver Routing Code. Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNB S003 0014.
RECVR_SUBID	35	AN	Interchange Receiver Internal Sub-Identification. EDIFACT: UNB S003 0046.
RECVR_SUBID_EDIOUT	35	AN	Interchange Receiver Internal Sub-Identification. EDIFACT: UNB S003 0046.
RESP_AGENCY_CODE	1	AN	Responsible agency code. User-assigned value for outbound use (if blank, the system loads "X"). Loaded to on outbound/loaded from on inbound: X12: GS 07.
RESP_AGENCY_CODE_EDIOUT	1	AN	Responsible agency code. User-assigned value for EDI-to-EDI map use (if blank, the system loads "X"). Loaded to an EDI-to-EDI map/loaded from on inbound: X12: GS 07.
SCEN_ID	14	AN	Scenario Identification. EDIFACT: UNH S018 0127.
SCEN_ID_EDIOUT	14	AN	Scenario Identification. EDIFACT: UNH S018 0127.
SCEN_REL	3	AN	Scenario Release Number. EDIFACT: UNH S018 0130.
SCEN_REL_EDIOUT	3	AN	Scenario Release Number. EDIFACT: UNH S018 0130.
SCEN_VER	3	AN	Scenario Version Number. EDIFACT: UNH S018 0128.
SCEN_VER_EDIOUT	3	AN	Scenario Version Number. EDIFACT: UNH S018 0128.
SECU_CODE	14	AN	Trading partner security ID code. Loaded from on outbound: tp/SECU_CODE. Loaded to on outbound/loaded from on inbound: X12: ISA 04. EDIFACT: UNB S005 0022.

Name	Size	Type	Description
SECU_CODE_EDIOUT	14	AN	Trading partner security ID code. Loaded from an EDI-to-EDI map: tp/SECU_CODE. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 04. EDIFACT: UNB S005 0022.
SECU_QUAL	2	AN	Trading partner security ID code qualifier. Loaded from on outbound: tp/SECU_QUAL (if blank, use "00" for X12 and null for EDIFACT). Loaded to on outbound/loaded from on inbound: X12: ISA 03. EDIFACT: UNB S005 0025.
SECU_QUAL_EDIOUT	2	AN	Trading partner security ID code qualifier. Loaded from an EDI-to-EDI map: tp/SECU_QUAL (if blank, use "00" for X12 and null for EDIFACT). Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 03. EDIFACT: UNB S005 0025.
SEG_TERM	1	AN	X12 segment delimiter. Loaded from on outbound: tp/SEG_DELMT (if blank, use X12 default segment terminator). Loaded to on outbound/loaded from on inbound: X12: ISA 16.
SEG_TERM_EDIOUT	1	AN	X12 segment delimiter. Loaded from an EDI-to-EDI map: tp/SEG_DELMT (if blank, use X12 default segment terminator). Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 16.
SEGMENT_COUNT	9	UI	Number of segments in a transaction. Loaded from on outbound: tradstat/GS_NO. Loaded to on outbound/loaded from on inbound: <ul style="list-style-type: none"> • EDIFACT: UNT01 • X12: SE01
SEGMENT_COUNT_EDIOUT	9	UI	Number of segments in a transaction. Loaded from an EDI-to-EDI map: tradstat/GS_NO. Loaded to an EDI-to-EDI map/loaded from on inbound: <ul style="list-style-type: none"> • EDIFACT: UNT01 • X12: SE01

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
SEND_CODE	35	AN	Sender ID code on interchange-level envelopes. Loaded from on outbound: tp/SND_IDCODE (if blank, use wixset/WIX_IDCODE). Loaded to on outbound/loaded from on inbound: X12: ISA 06. EDIFACT: UNB S002 0004.
SEND_CODE_EDIOUT	35	AN	Sender ID code on interchange-level envelopes. Loaded from an EDI-to-EDI map: tp/SND_IDCODE (if blank, use wixset/WIX_IDCODE). Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 06. EDIFACT: UNB S002 0004.
SEND_QUAL	4	AN	Sender ID code qualifier on interchange-level envelopes. Loaded from on outbound: tp/SND_IDQUAL (if blank, use wixset/WIX_IDQUAL). Loaded to on outbound/loaded from on inbound: X12: ISA 05. EDIFACT: UNB S002 0007.
SEND_QUAL_EDIOUT	4	AN	Sender ID code qualifier on interchange-level envelopes. Loaded from an EDI-to-EDI map: tp/SND_IDQUAL (if blank, use wixset/WIX_IDQUAL). Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA 05. EDIFACT: UNB S002 0007.
SEND_REV_ROUTE	14	AN	Sender reverse routing code. Loaded to on outbound/loaded from on inbound: EDIFACT: UNB S002 0008.
SEND_REV_ROUTE_EDIOUT	14	AN	Sender reverse routing code. Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNB S002 0008.
SEQ_TRAN	1	AN	Sequence of transfers. Loaded to on outbound/loaded from on inbound: EDIFACT: UNH S010 0070.
SEQ_TRAN_EDIOUT	1	AN	Sequence of transfers. Loaded to an EDI-to-EDI map/loaded from on inbound: EDIFACT: UNH S010 0070.
SERV_CODE	6	AN	Service Code List Directory Version Number. EDIFACT: UNB S001 0080.

Name	Size	Type	Description
SERV_CODE_EDIOUT	6	AN	Service Code List Directory Version Number. EDIFACT: UNB S001 0080.
SERV_ID	15	AN	Service provider ID.
SERV_ID_EDIOUT	15	AN	Service provider ID.
SERV_QUAL	2	AN	Service provider ID qualifier.
SERV_QUAL_EDIOUT	2	AN	Service provider ID qualifier.
SKIP_SEGMENT_WRITE	1	AN	<p>Flag that specifies to not write this EDI segment. Generated by the program on outbound and inbound:</p> <ul style="list-style-type: none"> In outbound maps, sets the SKIP_SEGMENT_WRITE SYSVAR in a Before or After segment element EDIT rule to “Y”, then when the level containing this segment is generated, the rule turns it to “Y” and this segment is not written. SKIP_SEGMENT_WRITE is reset to “ ” (blank) after each segment is written. So if there are 10 segments at a level, you can have rules in each segment to determine whether to write to it or not. You can set it in a Before Rule in the flow. But then it affects only the first segment at that level. In inbound maps, SKIP_SEGMENT_WRITE can be used in Segment element EDIT rules, or in the Before Flow rule to affect whether the segment is placed in the EDI OUT file.

Name	Size	Type	Description
SKIP_SEGMENT_WRITE_EDIOUT	1	AN	<p>Flag that specifies to not write this EDI segment.</p> <p>Generated by the program on outbound and inbound:</p> <ul style="list-style-type: none"> For the outbound transactions of an EDI-to-EDI map, sets the SKIP_SEGMENT_WRITE.SYSVAR in a Before or After segment element EDIT rule to “Y”, then when the level which contains this segment is generated, the rule turns it to “Y” and this segment is not written. <p>SKIP_SEGMENT_WRITE is reset to “ ” (blank) after each segment is written. So if there are 10 segments at a level, you can have rules in each segment to determine whether to write to it or not.</p> <p>You could also set it in a Before Rule in the flow. But then it affects only the first segment at that level.</p> <ul style="list-style-type: none"> In inbound maps, SKIP_SEGMENT_WRITE can be used in Segment element EDIT rules, or in the Before Flow rule to affect whether the segment is placed in the EDI OUT file.
SNDR_SUBID	35	AN	<p>Interchange Sender Internal Sub-Identification.</p> <p>EDIFACT: UNB S002 0042.</p>
SNDR_SUBID_EDIOUT	35	AN	<p>Interchange Sender Internal Sub-Identification.</p> <p>EDIFACT: UNB S002 0042.</p>
SOFT_ID	10	AN	<p>Software/vendor ID.</p>
SOFT_ID_EDIOUT	10	AN	<p>Software/vendor ID.</p>
SQL_RETURN	2	UI	<p>SQL return code.</p> <p>Returns the status of SQL commands.</p> <p>Generated by the program and used to determine if the last FETCH command returned a result set. For example, you can use the SQL_RETURN system variable in an IF/THEN/ELSE command to determine if a result set has been returned.</p>
SQL_RETURN_EDIOUT	2	UI	<p>SQL return code.</p> <p>Returns the status of SQL commands.</p> <p>Generated by the program and used to determine if the last FETCH command returned a result set. For example, you can use the SQL_RETURN system variable in an IF/THEN/ELSE command to determine if a result set has been returned.</p>

Name	Size	Type	Description
SUB_SEPERATOR	1	AN	Sub-element delimiter for X12 and HL7 data. Loaded to on outbound/loaded from on inbound: X12: ISA Column 105. Loaded from outbound: tp/ELE_DELIMIT (if blank, use standard default).
SUB_SEPERATOR_EDIOUT	1	AN	Sub-element delimiter for X12 and HL7 data. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ISA Column 105. Loaded from an EDI-to-EDI map: tp/ELE_DELIMIT (if blank, use standard default).
SYS_ADDR1	35	AN	First line of trading partner street address. Loaded to on inbound: tp/ADDR1.
SYS_ADDR1_EDIOUT	35	AN	First line of trading partner street address. Loaded to on inbound: tp/ADDR1. Obtained from the outbound transactions trading partner entry.
SYS_ADDR2	35	AN	Second line of trading partner street. Loaded to on inbound: tp/ADDR2.
SYS_ADDR2_EDIOUT	35	AN	Second line of trading partner street. Loaded to on inbound: tp/ADDR2. Obtained from the outbound transactions trading partner entry.
SYS_BILLIDEN	15	AN	Trading partner Bill To ID code. Loaded to on inbound: tp/BILLIDEN.
SYS_BILLIDEN_EDIOUT	15	AN	Trading partner Bill To ID code. Loaded to on inbound: tp/BILLIDEN. Obtained from the outbound transactions trading partner entry.
SYS_BILLQUAL	2	AN	Trading partner Bill To ID code qualifier. Loaded to on inbound: tp/BILLQUAL.
SYS_BILLQUAL_EDIOUT	2	AN	Trading partner Bill To ID code qualifier. Loaded to on inbound: tp/BILLQUAL. Obtained from the outbound transactions trading partner entry.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
SYS_CHG_TRANS	6	AN	Indicator that transaction has changed. Generated by the program. On outbound, this triggers a trading partner lookup. (In other words, it is used by the program in conjunction with SYS_TRANS to change maps for different transactions when multiple transactions are in one file.) Each time the program writes a message/transaction, if SYS_CHG_TRANS is different from SYS_TRANS, the program, writes a new: <ul style="list-style-type: none"> • Outer envelope if the trading partner changed. • Inner envelope if the trading partner did not change.
SYS_CITY	30	AN	City where the trading partner is located. Loaded to on inbound: tp/CITY.
SYS_CITY_EDIOUT	30	AN	City where the trading partner is located. Loaded to on inbound: tp/CITY. Obtained from the outbound transactions trading partner entry.
SYS_COUNTRY	25	AN	Country where the trading partner is located. Loaded to on inbound: tp/COUNTRY.
SYS_COUNTRY_EDIOUT	25	AN	Country where the trading partner is located. Loaded to on inbound: tp/COUNTRY. Obtained from the outbound transactions trading partner entry.
SYS_DATE	8	Date	Runtime system date. Loaded with on inbound and outbound: Computer date.
SYS_DATE_EDIOUT	8	Date	Runtime system date. Loaded with on inbound and outbound: Computer date. Obtained from the outbound transactions trading partner entry.
SYS_GENGS	1	AN	Flag to generate multiple group envelopes. Generated by the program on Outbound to: Generate multiple group envelopes.
SYS_HHMMSS	6	Time	System time in format HHMMSS. Loaded with on inbound and outbound: Computer time.
SYS_HHMMSS_EDIOUT	6	Time	System time in format HHMMSS. Loaded with on inbound and outbound: Computer time. Obtained from the outbound transactions trading partner entry.

Name	Size	Type	Description
SYS_IDCODE	35	AN	Trading partner ID code. Not currently used.
SYS_IDCODE_EDIOUT	35	AN	Trading partner ID code. Not currently used. Obtained from the outbound transactions trading partner entry.
SYS_IDQUAL	2	AN	Trading partner ID code qualifier. Not currently used.
SYS_IDQUAL_EDIOUT	2	AN	Trading partner ID code qualifier. Not currently used. Obtained from the outbound transactions trading partner entry.
SYS_MAP_EXT	8	AN	Map name extension. User-defined at trade agreement creation to distinguish multiple map definitions for the same transaction, direction, and trade partner.
SYS_MAP_EXT_EDIOUT	8	AN	Map name extension. User-defined at trade agreement creation to distinguish multiple map definitions for the same transaction, direction, and trade partner.
SYS_MAP_EXT_CHG	8	AN	Used to change the map name extension of the current map (outbound only).
SYS_NAME	35	AN	Internal name for the trading partner. Loaded to on inbound: tp/NAME.
SYS_NAME_EDIOUT	35	AN	Internal name for the trading partner. Loaded to on inbound: tp/NAME. Obtained from the outbound transactions trading partner entry.
SYS_RUN_ID	9	UI	Log run ID number. For inbound and outbound, this value can be: <ul style="list-style-type: none"> • Passed in as a parameter on the -id switch. • Loaded from trlog/runid if using an ODBC log. • Loaded with 0000 if using a non-ODBC log. This number is used to create output file names for routed data.
SYS_SHIPIDEN	15	AN	Trading partner ship-to identification code. Loaded to on inbound: tp/SHIPIDEN.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
SYS_SHIPIDEN_EDIOUT	15	AN	Trading partner ship-to identification code. Loaded to on inbound: tp/SHIPIDEN. Obtained from the outbound transactions trading partner entry.
SYS_SHIPQUAL	2	AN	Trading partner ship-to identification code qualifier. Loaded to on inbound: tp/SHIPQUAL.
SYS_SHIPQUAL_EDIOUT	2	AN	Trading partner Ship To identification code qualifier. Loaded to on inbound: tp/SHIPQUAL. Obtained from the outbound transactions trading partner entry.
SYS_STATE	2	AN	State where the trading partner is located. Loaded to on inbound: tp/STATE.
SYS_STATE_EDIOUT	2	AN	State where the trading partner is located. Loaded to on inbound: tp/STATE. Obtained from the outbound transactions trading partner entry.
SYS_TIME	4	Time	System Time in format HHMM. Loaded with: Computer time.
SYS_TIME_EDIOUT	4	Time	System Time in format HHMM. Loaded with: Computer time.
SYS_TRADNO	35	AN	Internal ID number for trading partner generated by the trading partner lookup. Loaded from on inbound and outbound: tp/CUSTNO. Trading partner lookup on inbound: Using the trading partner search option criteria specified by the user, the program compares selected fields in the EDI envelopes with the appropriate fields in the trading partner database, to find the CUSTNO. Trading partner lookup on outbound: The program matches the trading partner ID field in the user file with the group receiver code in the trading partner database, to find the CUSTNO. Both lookups are affected by the presence or absence of override values and switches.

Name	Size	Type	Description
SYS_TRADNO_EDIOUS	35	AN	<p>Internal ID number for trading partner generated by the trading partner lookup.</p> <p>Loaded from on inbound and on an EDI-to-EDI map: tp/CUSTNO.</p> <p>Trading partner lookup on inbound: Using the trading partner search option criteria specified by the user, the program compares selected fields in the EDI envelopes with the appropriate fields in the trading partner database, to find the CUSTNO.</p> <p>Trading partner lookup on EDI-to-EDI map: The program matches the trading partner ID field in the user file with the group receiver code in the trading partner database, to find the CUSTNO.</p> <p>Both lookups are affected by the presence or absence of override values and switches.</p>
SYS_TRANS	6	AN	<p>Message/transaction set.</p> <p>Passed in on outbound as required parameter 4: -t <i>message/ transaction set</i>.</p> <p>Loaded to on outbound/loaded from on inbound: X12: ST 01.</p> <p>EDIFACT: UNH S009 0065.</p> <p>Loaded to on inbound and outbound: tradstat/MAP_TRAN.</p> <p>Changing this value does not affect the envelope or the actual transaction name used.</p>
SYS_TRANS_EDIOUS	6	AN	<p>Message/transaction set</p> <p>Passed in on EDI-to-EDI outbound as required parameter 4: -t <i>message/ transaction set</i>.</p> <p>Loaded to an EDI-to-EDI map/loaded from on inbound: X12: ST 01.</p> <p>EDIFACT: UNH S009 0065.</p> <p>Loaded to on inbound and on an EDI-to-EDI map: tradstat/MAP_TRAN.</p> <p>Changing this value does not affect the envelope or the actual transaction name used.</p>
SYS_TRCODE	2	AN	<p>Transaction code.</p> <p>Passed in on outbound as required parameter 3: <i>two-character transaction code</i></p>

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
SYS_TRCODE_EDIOUT	2	AN	Transaction code. Passed in an EDI-to-EDI map as required parameter 3: <i>two-character transaction code.</i>
SYS_USER_FIELD	35	AN	USER DEFINED FIELD.
SYS_USER_FIELD_EDIOUT	35	AN	USER DEFINED FIELD.
SYS_ZIP	9	AN	Postal code of trading partner location. Loaded to on inbound: tp/ZIP.
SYS_ZIP_EDIOUT	9	AN	Postal code of trading partner location. Loaded to on inbound: tp/ZIP. Obtained from the outbound transaction trading partner entry.
TELEPHONE1	22	AN	Trading partner telephone number #1. Loaded to on inbound: tp/TELEPHONE1.
TELEPHONE1_EDIOUT	22	AN	Trading partner telephone number #1. Loaded to on inbound: tp/TELEPHONE1. Obtained from the outbound transaction trading partner entry.
TELEPHONE2	22	AN	Trading partner telephone number #2. Loaded to on inbound: tp/TELEPHONE2.
TELEPHONE2_EDIOUT	22	AN	Trading partner telephone number #2. Loaded to on inbound: tp/TELEPHONE2. Obtained from the outbound transaction trading partner entry.
TEST_IND	1	AN	Transaction mode (T, P, or I). Loaded from on outbound: tradstat/STAT. Loaded to on outbound/loaded from on inbound: X12: ISA 15. EDIFACT: UN B S005 0035.
TEST_IND_EDIOUT	1	AN	Transaction mode (T, P, or I). Loaded from an EDI-to-EDI map: tradstat/STAT. Loaded to an EDI-to-EDI/EDI-to-EDI map/loaded from on inbound: X12: ISA 15. EDIFACT: UN B S005 0035.
TOT_ERRS	9	UI	Total number of errors during a map run. Loaded from log file on inbound and outbound. Available for interrogation by users.

Name	Size	Type	Description
TOT_ERRS_EDIOUT	9	UI	<p>Total number of errors during a map run.</p> <p>Loaded from log file on inbound and on an EDI-to-EDI map.</p> <p>Available for interrogation by users.</p>
TRANS_ACK_REQ	1	AN	<p>Specifies that a Functional Acknowledgement is expected from the receiver of an outbound map or has been requested by the sender of an inbound map:</p> <p>1 = Expected/requested. 0 = Not expected/requested.</p> <p>Loaded from on inbound and outbound: tradstat/ARK_RQSTD2.</p> <p>On outbound maps: When the value of TRANS_ACK_REQ is 1, the program saves additional information to the log. This information is used to match the Functional Acknowledgement that is sent back with the data that requested it and confirm that it was received within the time period specified.</p> <p>On inbound maps: When the value of TRANS_ACK_REQ is 1, the program “turns on” trlog/TR_ACK_TYPE and saves additional information to the log. This information is used as input to the map that creates the Functional Acknowledgement.</p> <p>On outbound Functional Acknowledgements: When an EDI transaction is received which the trading partners have agreed should generate a Functional Acknowledgement, the receiver must run one of the acknowledgement maps provided with the software. As input to the map, the program uses all of the data in trlog with TR_ACK_TYPE “turned on”. After creating the acknowledgements, the program “turns off” TR_ACK_TYPE to prevent sending duplicate acknowledgements.</p> <p>On inbound Functional Acknowledgements: When an Interchange Acknowledgement is received, the program compares it with the information entered in the log to match the acknowledgement with the correct requesting data it and to confirm that it was received within the time period specified. The program must “turn off” trlog/ACK_EXPECT after acknowledgements have been received to indicate that the request for an acknowledgement has been fulfilled.</p>

Name	Size	Type	Description
TRANS_ACK_REQ_EDIOUS	1	AN	<p>Specifies that a Functional Acknowledgement is expected from the receiver of an EDI-to-EDI outbound map or has been requested by the sender of an inbound map:</p> <p>1 = Expected/requested. 0 = Not expected/requested.</p> <p>Loaded from on inbound and EDI-to-EDI map: tradstat/ARK_RQSTD2.</p> <p>On EDI-to-EDI maps: When the value of TRANS_ACK_REQ is 1, the program saves additional information to the log. This information is used to match the Functional Acknowledgement that is sent back with the data that requested it and confirm that it was received within the time period specified.</p> <p>On inbound maps: When the value of TRANS_ACK_REQ is 1, the program “turns on” trlog/TR_ACK_TYPE and saves additional information to the log. This information is used as input to the map that creates the Functional Acknowledgement.</p> <p>On EDI-to-EDI outbound Functional Acknowledgements: When an EDI transaction is received which the trading partners have agreed should generate a Functional Acknowledgement, the receiver must run one of the acknowledgement maps provided with the software. As input to the map, the program uses all of the data in trlog with TR_ACK_TYPE “turned on”. After creating the acknowledgements, the program “turns off” TR_ACK_TYPE to prevent sending duplicate acknowledgements.</p> <p>On inbound Functional Acknowledgements: When an Interchange Acknowledgement is received, the program compares it with the information entered in the log to match the acknowledgement with the correct requesting data it and to confirm that it was received within the time period specified. The program must “turn off” trlog/ACK_EXPECT after acknowledgements have been received to indicate that the request for an acknowledgement has been fulfilled.</p>

Name	Size	Type	Description
TRANS_CTRL_NUM	35	AN	Transaction control number. Loaded from on outbound: (tradstat/GS_NO X 1000) + # of transactions. Loaded to on outbound/loaded from on inbound: X12 ST/SE 02. EDIFACT: UNH/UNT 0062.
TRANS_CTRL_NUM EDIOUT	35	AN	Transaction control number. Loaded from an EDI-to-EDI map: (tradstat/GS_NO X 1000) + # of transactions. Loaded to an EDI-to-EDI map/loaded from on inbound: X12 ST/SE 02. EDIFACT: UNH/UNT 0062.
TRANSMISSION_TYPE	2	AN	Batch header transmission type.
TRANSMISSION_TYPE EDIOUT	2	AN	Batch header transmission type.
USER_BHS0201_SCODE	15	AN	Batch sending application.
USER_BHS0301_R_ROUTE	15	AN	Batch sending facility.
USER_BHS0401_RCODE	15	AN	Batch receiving application.
USER_BHS0501_RCVCODE	15	AN	Batch receiving facility.
USER_BHS0601_DATE	8	AN	Batch creation date.
USER_BHS0602_TIME	4	AN	Batch creation time.
USER_BHS0701_PASSWORD	12	AN	Batch security.
USER_BHS1001_GS_CCTRL	14	AN	Batch control ID.
USER_FHS0201_SCODE	15	AN	File sending application.
USER_FHS0301_R_ROUTE	15	AN	File sending facility.
USER_FHS0401_RCODE	15	AN	File receiving application.
USER_FHS0501_RCVCODE	15	AN	File receiving facility.
USER_FHS0601_DATE	8	AN	File creation date.
USER_FHS0602_TIME	4	AN	File creation time.
USER_FHS0701_PASSWORD	12	AN	File security.
USER_FHS0801_FNAME	20	AN	File name/ID.
USER_FHS1001_GISABUF	35	AN	File control ID.
USER_GS01_SYS_TRCODE	2	AN	Functional identifier code.
USER_GS02_SCODE	15	AN	Application sender's code.
USER_GS03_RCODE	15	AN	Application receiver's code.
USER_GS04_SYS_DATE	8	AN	Date.
USER_GS05_SYS_TIME	8	AN	Time.
USER_GS06_FUNC_GP_NUM	9	AN	Group control number.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
USER_GS07_CH_AGENCY	2	AN	Responsible agency code.
USER_GS08_XVER	12	AN	Version/release/industry identifier code.
USER_ISA01_AUTH_QUAL	2	AN	Authorization information qualifier.
USER_ISA02_AUTH_CODE	15	AN	Authorization information.
USER_ISA03_SECU_QUAL	2	AN	Security information qualifier.
USER_ISA04_SECU_CODE	14	AN	Security information.
USER_ISA05_SEND_QUAL	4	AN	Interchange ID qualifier.
USER_ISA06_SEND_CODE	35	AN	Interchange sender ID.
USER_ISA07_RECV_QUAL	4	AN	Interchange ID qualifier.
USER_ISA08_RECV_CODE	35	AN	Interchange receiver ID.
USER_ISA09_DATE	6	AN	Interchange date.
USER_ISA10_TIME	4	AN	Interchange time.
USER_ISA11_PTSTD	1	AN	Interchange control standards identifier.
USER_ISA12_ISA_TYPE	5	AN	Interchange control version number.
USER_ISA13_GISABUF	9	AN	Interchange control number.
USER_ISA14_ACK_REQSTD	1	AN	Acknowledgement requested.
USER_ISA15_TEST_IND	1	AN	Usage indicator.
USER_ISA16_SUB_SEP	1	AN	Component element separator.
USER_OUT_FILENAME	160	AN	User-generated override for the EDI output file name. On inbound: The file to which EDI data that is passed through is written and which is generated by the program using the run ID number. On outbound: The file to which the mapped EDI data is written.
USER_OUT_FILENAME_EDIOUT	160	AN	User-generated override for the EDI output file name. On inbound: The file to which EDI data that is passed through is written and which is generated by the program using the run ID number. On an EDI-to-EDI map: The file to which the mapped EDI data is written.
USER_UNB0101_VERS	4	AN	Syntax identifier.
USER_UNB0102_VERS	1	AN	Syntax version number.
USER_UNB0103_SERV_CODE	6	AN	Syntax version.
USER_UNB0201_SCODE	35	AN	Sender identification.
USER_UNB0202_PSQUAL	4	AN	Partner identification code qualifier.
USER_UNB0203_SNDR_ROUTE	35	AN	Sender routing address.
USER_UNB0204_SNDR_SUBID	35	AN	Sender subID.

Name	Size	Type	Description
USER_UNB0301_RCODE	35	AN	Recipient identification.
USER_UNB0302_PRECV	4	AN	Partner identification code qualifier.
USER_UNB0303_RCVR_ROUTE	35	AN	Recipient routing address.
USER_UNB0304_RCVR_SUBID	35	AN	Recipient subID.
USER_UNB0401_DATE	8	AN	Date of preparation.
USER_UNB0402_TIME	4	AN	Time of preparation.
USER_UNB0501_GISABUF	35	AN	Interchange control reference.
USER_UNB0601_PASSWORD	14	AN	Recipient reference/password.
USER_UNB0602_PSECU	2	AN	Recipient reference/password.
USER_UNB0701_COMREF	14	AN	Application reference.
USER_UNB0801_PCODE	1	AN	Processing priority code.
USER_UNB0901_ACK_MSG	1	AN	Acknowledgement request.
USER_UNB1001_COMAG	35	AN	Communications agreement ID.
USER_UNB1101_PTIND	1	AN	Test indicator.
USER_UNG0101_F_IDCODE	6	AN	Functional group identification.
USER_UNG0201_SCODE	35	AN	Application sender identification.
USER_UNG0202_APSQUAL	4	AN	Partner identification code qualifier.
USER_UNG0301_RCODE	35	AN	Application recipient's identification.
USER_UNG0302_APRQUAL	4	AN	Partner identification code.
USER_UNG0401_DATE	8	AN	Date of preparation.
USER_UNG0402_TIME	4	AN	Time of preparation.
USER_UNG0501_GUNGBUF	14	AN	Functional group reference number.
USER_UNG0601_CNT_AG1	3	AN	Controlling agency.
USER_UNG0701_XVER	3	AN	Message type version number.
USER_UNG0702_XVER2	3	AN	Message type release number.
USER_UNG0703_ASSOC_CODE	6	AN	Association assigned code.
USER_UNG0801_AP_PASE	14	AN	Application password.
WIX_ADP_AD_PASS	8	AN	Company ADP authorization password.
WIX_ADP_AD_PASS_EDIOUT	8	AN	Company ADP authorization password.
WIX_ADP_ID	8	AN	Company ADP user identification.
WIX_ADP_ID_EDIOUT	8	AN	Company ADP user identification.
WIX_ADP_PASS	8	AN	Company ADP password.
WIX_ADP_PASS_EDIOUT	8	AN	Company ADP password.
WIX_APP_SND_QL	4	AN	UNG0202.
WIX_APP_SND_QL_EDIOUT	4	AN	UNG0202.
WIX_AUTH_CODE	10	AN	Company authorization code.
WIX_AUTH_CODE_EDIOUT	10	AN	Company authorization code.

Alphabetical listing of ECMAP system variables

Name	Size	Type	Description
WIX_AUTH_QUAL	2	AN	Company authorization code qualifier.
WIX_AUTH_QUAL_EDIOUS	2	AN	Company authorization code qualifier.
WIX_B_SEND_ID	24	AN	Company batch sender ID.
WIX_B_SEND_ID_EDIOUS	24	AN	Company batch sender ID.
WIX_BIN_NUMB	6	AN	Company bin number.
WIX_BIN_NUMB_EDIOUS	6	AN	Company bin number.
WIX_COMPANY_NAME	35	AN	Company name.
WIX_COMPANY_NAME_EDIOUS	35	AN	Company name.
WIX_GSID	35	AN	Company group/inner envelope sender ID code.
WIX_GSID_EDIOUS	35	AN	Company group/inner envelope sender ID code.
WIX_IDCODE	35	AN	Company interchange/outer envelope sender ID code.
WIX_IDCODE_EDIOUS	35	AN	Company interchange/outer envelope sender ID code.
WIX_IDQUAL	2	AN	Company interchange/outer envelope sender ID code qualifier.
WIX_IDQUAL_EDIOUS	2	AN	Company interchange/outer envelope sender ID code qualifier.
WIX_MODEM_BAUD	4	AN	Company modem baud rate.
WIX_MODEM_BAUD_EDIOUS	4	AN	Company modem baud rate.
WIX_MODEM_PORT	2	AN	Company modem port.
WIX_MODEM_PORT_EDIOUS	2	AN	Company modem port.
WIX_PHONE_TONE	1	AN	Company modem tone.
WIX_PHONE_TONE_EDIOUS	1	AN	Company modem tone.
WIX_PROC_NUMB	10	AN	Company process control number.
WIX_PROC_NUMB_EDIOUS	10	AN	Company process control number.
WIX_PRODUCER_ID	12	AN	Company producer ID code.
WIX_PRODUCER_ID_EDIOUS	12	AN	Company producer ID code.
WIX_SECU_CODE	10	AN	Company security code.
WIX_SECU_CODE_EDIOUS	10	AN	Company security code.
WIX_SECU_QUAL	2	AN	Company security code qualifier.
WIX_SECU_QUAL_EDIOUS	2	AN	Company security code qualifier.
WIX_SERV_ID	15	AN	Company service ID.
WIX_SERV_ID_EDIOUS	15	AN	Company service ID.
WIX_SERV_QUAL	2	AN	Company service ID qualifier.
WIX_SERV_QUAL_EDIOUS	2	AN	Company service ID qualifier.
WIX_SNDR_ROUTE	15	AN	Sender internal ID.
WIX_SNDR_ROUTE_EDIOUS	15	AN	Sender internal ID.
WIX_SNDR_SUBID	15	AN	Sender internal subID.

Name	Size	Type	Description
WIX_SNDR_SUBID_EDIOUS	15	AN	Sender internal subID.
WIX_SOFT_ID	10	AN	Company software ID.
WIX_SOFT_ID_EDIOUS	10	AN	Company software ID.
WIX_TEL_NUMBR	16	AN	Company telephone number.
WIX_TEL_NUMBR_EDIOUS	16	AN	Company telephone number.
X12_VERSION	12	AN	Version of EDI standard used in the map. Loaded from on outbound: tradstat/VERS. Loaded to on outbound/loaded from on inbound: X12: GS 08. EDIFACT: UNG S008 0052.
X12_VERSION_EDIOUS	12	AN	Version of EDI standard used in the map. Loaded from an EDI-to-EDI map: tradstat/VERS. Loaded to an EDI-to-EDI map/loaded from on inbound: X12: GS 08. EDIFACT: UNG S008 0052.

Index

Symbols

^ 71
' 71

A

-ab *path/filename* switch 76
abnormal exit 33
accent grave 71
Advance Shipment Notice. *See* ASN
-ag *path/filename* switch 77
-al switch 71
-as switch 82
ASN 7
Assign command 61
AssignInput command 63
Assignment command 27
AssignOutput command 63
asterisks in password security 34
attach command 60
attacht command 60
Availability and Certification Reports 2

B

BIN segment 36
Binary Data Placement command 36
Boolean Arithmetic command 23
Boolean Assignment command 24
boolean datatype 23
Boolean logic 23
 Assignment command 27
 Boolean Arithmetic command 23
 Boolean Assignment command 24
 Boolean Assignment command datatypes, table of
 26
 boolean datatype 23

If Condition command 27
Perform Rule command 27
rule commands, new 23
rule commands, updated 26

breakpoints
 clearing all, in ECMAP Debugger 52
 clearing selected, in ECMAP Debugger 52
 setting conditional, in ECMAP Debugger 51
 setting, in ECMAP Debugger 51
breakpoints, in ECMAP Debugger 51
breakpoints, marking, in ECMAP Debugger 43

C

character checks for compliance maps 71
circumflex character 71
clearing all breakpoints, in ECMAP Debugger 52
clearing selected breakpoints, in ECMAP Debugger
 52
-clz switch 81
-cn switch 73
command line switches 84
command line switches, obtaining, in ECMAP Debugger
 46
commands
 Assign 61
 AssignInput 63
 Assignment 27
 AssignOutput 63
 attach 60
 attacht 60
 Binary Data Placement 36
 Boolean Arithmetic 23
 Boolean Assignment 24
 Company Lookup 13
 EDI Filename 14
 EDI-to-EDI Run Switches 15
 Exit, in EC Gateway 65
 If Condition 27

- Map Envelope Level** 17
- Map Level** 19
- Perform Rule** 27
- Return**, in EC Gateway 65
- Stringlength**, in EC Gateway 65
- Trade Partner Lookup** 20
- commands for EDI-to-EDI 13
- Company Lookup** command 13
- compliance map acknowledgement options 38
- compliance maps, HIPAA 81
- Console tab, in ECMAP Debugger 44
- Constant type variable 65
- controllin debug sessions, in ECMAP Debugger 54
- controlling updates and lookups for control numbers 73
- conventions, document style x
- copy process option 59
- copying process scripts procedure, in EC Gateway 59
- copying process scripts to other systems, in EC Gateway 59
- Counter type variable 65
- creating EDI-to-EDI maps 7
- creating outbound EDI-to-EDI transactions, procedure for 8
- cross-reference table import utility 38
- cu** switch 82

D

- Data pane, in ECMAP Debugger 44
- databases supported 2
- DATAIOSTRUCT 63
 - pbuflen field 63
 - pdata field 63
 - pdatalen field 63
 - tag field 63
- DATAIOSTRUCT data structure 62
- datatype for boolean logic 23
- debug error messages, in ECMAP Debugger 45
- debug information, in ECMAP Debugger 45
- debug session options
 - step by step 54
 - stepover 54
 - stop debugging 54
 - to cursor 54
 - to the end 54

- debug session, completing, in ECMAP Debugger 56
- Defining the transaction for the outbound standard,
 - procedure for 8
- defining transactions for outbound standards 8
- developing EDI-to-EDI maps 8
- developing EDI-to-EDI maps, procedure for 9

E

- EC Gateway 2, 60
 - copying process scripts to other systems 59
 - copying process scripts, procedure for 59
 - e-FTP scripts 58
 - Exit** command 65
 - find and replace functions, modules available in 58
 - find and replace functions, searching and replacing strings procedure 59
 - find and replace functions, searching for strings procedure 58
 - passing input to and getting output from EC Gateway
 - DLL shared library 62
 - passing input to EC Gateway executable 61
 - pfs/Async files, added security for 58
 - Return** command 65
 - StringLength** command 65
- EC Gateway Operations Viewer 2
- ECMAP 2, 5
- ECMAP Debugger 2, 39
 - accessing 40
 - batch file 40
 - breakpoints 51
 - clearing all breakpoints 52
 - clearing selected breakpoints 52
 - completing debug session 56
 - Console tab 44
 - controlling debug sessions 54
 - Data pane 44
 - debug error messages 45
 - debug information 45
 - debug session options, step by step 54
 - debug session options, stepover 54
 - debug session options, stop debugging 54
 - debug session options, to cursor 54
 - debug session options, to the end 54
 - EC RTP_HOME variable 40

- error messages 56
- features 39
- file path display 45
- Flow properties 43
- flow properties, inbound map 44
- flow properties, outbound map 44
- Flow tab 43
- following progress of debug session 43
- icon 40
- JAVA_HOME variable 40
- loading map information 50
- main window 42
- managing debugging sessions 46
- map debug session icon 43
- map debug sessions, creating 48
- map debug sessions, deleting 48
- map debug sessions, editing 48
- map debug sessions, managing 47
- map name display 45
- map switch set icon 43
- map switch sets, creating 49
- map switch sets, deleting 50
- map switch sets, editing 50
- map switch sets, managing 49
- mapdebug.bat* file 40
- MAPDEBUG_HOME variable 40
- marking breakpoints 43
- Memory Variable tab 44
- Monitor pane 44
- monitoring map flow 55
- monitoring real-time data values 55
- Navigator pane 42
- obtaining command line switches 46
- preparing debugging sessions 46
- projects folder 43
- projects, creating 47
- projects, deleting 47
- projects, editing 47
- projects, managing 47
- real-time data view 39
- real-time map flow view 40
- Record/Field tab 44
- return values 56
- Rules tab 43
- Rules/Flow pane 43
- running ECMAP Debugger sessions 54
- runtime error messages 45
- runtime information 45
- runtime logic checking 39
- saving projects and debug sessions 53
- segment location 45
- Segment tab 45
- segment value 45
- session control 39
- setting breakpoints 51
- setting conditional breakpoints 51
- starting debug sessions 54
- starting ECRTP 50
- System Variable tab 44
- title bar, in main window 43
- Trace tab 45
- viewing EDI data 40
- viewing specific values 52
- viewing trace file 55
- Watch tab 45
- ECMAP Network Client 2
- ECRTP 2, 67
 - 2GB and larger files, processed by ECRTP 68
 - command line switches 84
 - ECRTP Web Services API 67
 - JNI wrapper 69
 - Log Bulk Insert 68
 - ODBC switches for EDI 4.2 87
 - PADEDI environment variable 70
 - RunMap options 70
- ECRTP database connection wait time 72
- ECRTP Web Services API 67
- ECRTP/TP 2
- ECRTP_HOME variable 40
- EDI Filename** command 14
- EDI product set 1
- EDI Products components 2
- EDI Server 2
- EDI standards, support for 5
- EDI-to-EDI 6
 - Advance Ship Notice (ASN) 7
 - claim splitting and rejoining 7
 - Company Lookup** command 13
 - converting data 6
 - creating maps 7
 - creating multiple outputs 7
 - creating outbound transactions 8

- defining transactions for outbound standards 8
- developing maps 8, 9
- EDI Filename** command 14
- EDI-to-EDI Run Switches** command 15
- enriching data 7
- features 6
- Map Envelope Level** command 17
- Map Level** command 19
- mapping window, screenshot of 11
- note on compliance maps 7
- outbound standard options, defining 8
- query and response scenario 7
- reports 22
- RFID 7
- rule commands 13
- selecting EDI-to-EDI map types 8
- system variables 13
- Trade Partner Lookup** command 20
- what you can do 6
- EDI-to-EDI Run Switches** command 15
- e-FTP 58
- e-FTP scripts, reusing in EC Gateway 58
- electronic data interchange. *See* EDI 1
- e-mail attachments, using wildcards, in EC Gateway 60
- enabling Global repository 29
- encrypted passwords 34
- error messages, in ECMAP Debugger 56
- Exit** command, in EC Gateway 65
- exporting memory variables, procedure for 31
- exporting record definitions, procedure for 31
- exporting rule definitions, procedure for 30

F

- features
 - ECMAP Debugger 39
- file path display, in ECMAP Debugger 45
- find and replace functions, in EC Gateway 58, 59
- flow properties
 - inbound maps in ECMAP Debugger 44
 - outbound maps in ECMAP Debugger 44
- Flow properties, in ECMAP Debugger 43
- Flow tab, in ECMAP Debugger 43

G

- Global repository 28
 - enabling 29
 - exporting memory variables, procedure for 31
 - exporting record definitions, procedure for 31
 - exporting rule definitions, procedure for 30
 - features 28
 - importing and exporting memory variables 31
 - importing and exporting record definitions 30
 - importing and exporting rule definitions 29
 - importing memory variables, procedure for 31
 - importing record definitions, procedure for 30
 - importing rule definitions, procedure for 29
 - managing 32
 - properties 28
- grace period 3

H

- HIPAA compliance maps, leading zeros in 81

I

- If Condition** command 27
- importing and exporting
 - memory variables 31
 - record definitions 30
 - rule definitions 29
- importing memory variables, procedure for 31
- importing record definitions, procedure for 30
- importing rule definitions, procedure for 29
- importing tables 38
- inbound maps
 - flow properties, in ECMAP Debugger 44
- inbound parameters
 - optional 104
 - optional trading partner lookup switches 106
 - optional trading partner lookup switches for ODBC
 - database users 106
 - required 99
 - required for ODBC database users 99
- inbound/outbound parameters
 - optional 100
 - optional for ODBC database users 103

increase in default record size 38

J

Java Native Interface. *See* JNI wrapper

JAVA_HOME variable 40

JNI wrapper, in ECRTTP 69

L

large file size handling, in ECRTTP 68

leading zeros in HIPAA compliance maps 81

license grace period 3

literal numeric data mapping 81

locking contention 34

Log Bulk Insert

memory for bulk insert 68

transNNN.log file 68

TRLOG database 68

TRLOG_BLKSIZE environment variable 68

Log Bulk Insert, in ECRTTP 68

log persistence map 35

LOGUPDATE map 35

LOGUPDATE map parameters 36

lower isolation level 34

lpszCmdline command line 63

M

managing debugging sessions, in ECMAP Debugger 46

managing Global repository 32

map command line switches, obtaining, in ECMAP Debugger 46

map debug session icon, in ECMAP Debugger 43

map debug sessions

creating, in ECMAP Debugger 48

deleting, in ECMAP Debugger 48

editing, in ECMAP Debugger 48

managing, in ECMAP Debugger 47

map debugging sessions, preparing 46

Map Envelope Level command 17

map flow, monitoring, in ECMAP Debugger 55

Map Level command 19

map name display, in ECMAP Debugger 45

map recovery option 33

map switch set icon, in ECMAP Debugger 43

map switch sets

creating, in ECMAP Debugger 49

deleting, in ECMAP Debugger 50

editing, in ECMAP Debugger 50

managing, in ECMAP Debugger 49

mapdebug.bat file 40

MAPDEBUG_HOME variable 40

mapping window for EDI-to-EDI 11

Memory Variable tab, in ECMAP Debugger 44

memory variables, importing and exporting 31

-mn *map name extension* switch 78

-mns switch removed for ECRTTP 4.2 78

Monitor pane, in ECMAP Debugger 44

multiple systems 58

N

Navigator pane, in ECMAP Debugger 42

-ncb switch 80

NCPDP file processing 79

NCPDP batch files 80

NCPDP telecommunications files 80

-nct switch 80

new feature tags 2

-nret switch 78

-nz switch 81

O

ODBC switches 87

-ol switch 83

other sources of information viii

outbound maps

flow properties, in ECMAP Debugger 44

outbound parameters

optional 103

required 97

required for ODBC database users 98

outbound standards options, for EDI-to-EDI 8

outbound trading partner lookup option 83

override for compliance map results 76, 77

P

padding EDI data to minimum length 83
 PADEDI environment variable, in EC RTP 70
 parameters and switches

- alphabetic list for ODBC and non-ODBC trading partner files 90
- inbound required parameters 99
- inbound required parameters for ODBC database users 99
- inbound/outbound optional parameters 100
- inbound/outbound optional parameters for ODBC database users 103
- inbound-only optional parameters 104
- inbound-only optional trading partner lookup switches 106
- inbound-only optional trading partner lookup switches for ODBC database users 106
- only for ODBC trading partner files 95
- outbound required parameters 97
- outbound required parameters for ODBC database users 98
- outbound-only optional parameters 103
- required parameters 97

 passing input to and getting output from EC Gateway DLL

- shared library 62
 - AssignInput** command 63
 - AssignOutput** command 63
 - DATAIOSTRUCT data structure 62
 - lpszCmdline command line 63
 - syntax for calling shared library 63

 passing input to EC Gateway executable 61

- Assign** command 61
- syntax using input parameters 61

 Password security 34

- asterisks 34
- encrypted passwords 34

 pbufflen field 63
 pdata field 63
 pdatalen field 63
-pe switch 70, 83
Perform Rule command 27
 pfs/Async files, in EC Gateway 58

placeholder rule 51
 platforms supported 2
 process module, copy process option 59
 product bundle 2
 product licensing 3
 progress of debug session, following, in ECMAP Debugger 43
 projects

- creating, in ECMAP Debugger 47
- deleting, in ECMAP Debugger 47
- editing, in ECMAP Debugger 47
- managing, in ECMAP Debugger 47

 projects folder, in ECMAP Debugger window 43

R

radio frequency ID. *See* RFID
 Read_Uncommitted file property 34
 real-time data values, viewing in ECMAP Debugger 52
 real-time data view, in ECMAP Debugger 39
 real-time map flow view, in ECMAP Debugger 40
 record definitions, importing and exporting 30
 record size increase 38
 Record/Field tab, in ECMAP Debugger 44
 related documents vii
 reports for EDI-to-EDI 22
 repository 28
 return codes

- examples 79

 return codes, adding 78
Return command, in EC Gateway 65
 return values, in ECMAP Debugger 56
 reusing e-FTP scripts 58
 RFID 7
 riewing EDI data, in ECMAP Debugger 40
 rule commands for EDI-to-EDI 13
 rule definitions, importing and exporting 29
 rule import from another map 37
 Rules

- Binary Data Placement** command 36

 rules 36

- rule import from another map 37

 Rules tab, in ECMAP Debugger 43
 Rules/Flow pane, in ECMAP Debugger 43

RunMap options

- ab** *path/filename* switch 76
- ag** *path/filename* switch 77
- al** switch 71
- as** switch 82
- character checks for compliance maps 71
- clz** switch 81
- cn** switch 73
- controlling updates and lookups for control numbers 73
- cu** switch 82
- leading zeros in HIPAA compliance maps 81
- literal numeric data mapping 81
- mn** *map name extension* switch 78
- mns** switch removed for ECRTP 4.2 78
- ncb** switch 80
- NCPDP file processing 79
- nct** switch 80
- nret** switch 78
- nz** switch 81
- ol** switch 83
- outbound trading partner lookup option 83
- override for compliance map results 76
- padding EDI data to minimum length 83
- pe** switch 70, 83
- return codes, adding 78
- sc** switch 75
- sdb** *number of cached connections* switch 84
- setting a maximum waiting time for ECRTP to get a database connection 72
- specifying map name extensions 78
- specifying the maximum number of cached ODBC connections 84
- ST control number validation 82
- tc** *NN* switch 72
- validating the Interchange (ISA) and Group (GS) control number sequence 75
- vs** switch 71
- z** switch 81
- RunMap options, in EDI 70
- running debugging session, in ECMAP Debugger 54
- runtime error messages, in ECMAP Debugger 45
- runtime information, in ECMAP Debugger 45
- runtime logic checking, in ECMAP Debugger 39

S

- save as option 33
- Save map options 32
 - abnormal exit 33
 - map recovery option 33
 - save as option 33
 - save option 32
- save option 32
- saving projects, in ECMAP Debugger 53
- sc** switch 75
- sdb** *number of cached connections* switch 84
- searching and replacing strings procedure, in EC Gateway 59
- searching for strings procedure, in EC Gateway 58
- security for pfs/Async files 58
- segment location, in ECMAP Debugger 45
- Segment tab, in ECMAP Debugger 45
- segment value, in ECMAP Debugger 45
- selecting EDI-to-EDI map types 8
- session control, in ECMAP Debugger 39
- setting a maximum waiting time for ECRTP to get a database connection 72
- setting breakpoints, in ECMAP Debugger 51
- setting conditional breakpoints, in ECMAP Debugger 51
- specifying map name extensions 78
- specifying the maximum number of cached ODBC connections 84
- ST control number validation 82
 - check for ST control number sequence 82
 - check for unique ST control numbers 82
- standards reports for EDI-to-EDI 22
- standards, support for 5
- starting debug sessions, in ECMAP Debugger 54
- starting ECRTP, in ECMAP Debugger 50
- step by step debugging, in ECMAP Debugger 54
- stepper debugging, in ECMAP Debugger 54
- stop debugging, in ECMAP Debugger 54
- StringLength** command, in EC Gateway 65
- substituting a BAD EDI file 76
- substituting a GOOD EDI file 77
- supported databases 2
- supported platforms 2
- switches *See* RunMap options
- Sybase Software Asset Management. *See* SySAM
- syntax conventions xi

Index

SySAM 3
SySAM concepts 3
SySAM implementation. *See Installation Guide*
system requirements 2
System Variable tab, in ECMAP Debugger 44
system variables
 alphabetical listing of all 109
 for EDI-to-EDI 13

T

TA1 acknowledgements 34
 log persistence map 35
 LOGUPDATE map 35
 LOGUPDATE map parameters 36
 TA1XLG map 35
TA1XLG map 35
table import 38
tag field 63
tagging new features 2
-tc NN switch 72
title bar, in ECMAP Debugger 43
to cursor debugging, in ECMAP Debugger 54
to the end debugging, in ECMAP Debugger 54
trace file, viewing, in ECMAP Debugger 55
Trace tab, in ECMAP Debugger 45
Trade Partner Lookup command 20
transNNN.log file 68
TRLOG database 68
TRLOG_BLKSIZE environment variable 68

V

validating the Interchange (ISA) and Group (GS) control
 number sequence 75
-vs switch 71

W

Watch tab, in ECMAP Debugger 45
Web services API for ECRTP 67
what is EDI 1
wildcard e-mail attachments

attach command 60
attacht command 60
wildcard e-mail attachments, attaching 60
wsproces 61, 63, 65

Z

-z switch 81