

SYBASE®

User Guide

ECMap™

Version 4.2

[Windows]

DOCUMENT ID: DC00188-01-0420-01

LAST REVISED: November 2004

Copyright © 1999-2004 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, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DataWindow .NET, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Impact, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, iAnywhere, iAnywhere Application Alerts, iAnywhere Mobile Delivery, iAnywhere Mobile Document Viewer, iAnywhere Mobile Inspection, iAnywhere Mobile Marketing Channel, iAnywhere Mobile Pharma, iAnywhere Mobile Sales, iAnywhere Pylon, iAnywhere Pylon Application Server, iAnywhere Pylon Conduit, iAnywhere Pylon PIM Server, iAnywhere Pylon Pro, iAnywhere Solutions, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, Information Everywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, My iAnywhere, My iAnywhere Media Channel, My iAnywhere Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Orchestration Studio, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report-Execute, Report Workbench, Resource Manager, RW-DisplayLib, RW-Library, S-Designor, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILLS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, S.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TotalFix, TradeForce, Transact-SQL, Translation Toolkit, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 05/04

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

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

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	xiii	
CHAPTER 1	ECMap Overview	1
	Introduction	2
	What can ECMap do?	3
	Working with ECMap.....	4
	Starting ECMap.....	4
	About the workspace.....	5
	Task overview	8
	Creating a project	9
	Creating a map.....	9
	Setting up a Company ID	10
	Setting up your Trading Partner	10
	Adding a Trade Agreement to your Trading Partner	11
	Defining application data	11
	Creating a Transaction	12
	Building a map.....	12
	Generating a map.....	13
	Running a map	13
CHAPTER 2	Working with Projects	15
	Introduction	16
	Adding new projects.....	17
	Modifying projects	18
	Deleting projects	18
	Selecting projects.....	19
	Exporting and importing projects and maps.....	19
	Exporting a project with associated maps	19
	Importing a project with associated maps	20
CHAPTER 3	Working with Maps	23
	Introduction	24
	Adding new maps.....	25

	Defining map properties	27
	Defining map directories.....	31
	Defining map DSN settings	33
	Working with maps	34
	Modifying a map	34
	Deleting a map	35
	Copying a map	35
	Generating multiple maps in a project.....	38
	Archiving a map	38
	Automatic mapping of EDI to and from serialized NDO	39
	EDI ID data elements	45
CHAPTER 4	Working with the Address Book	47
	Overview	48
	Company ID	49
	Adding a new company profile	49
	Modifying a company profile.....	52
	Deleting a company profile.....	52
	Trading partner.....	53
	Adding a new trading partner	54
	Working with trading partners.....	64
	Setting up trade agreements.....	67
	Creating a new trade agreement.....	67
	Modifying a trade agreement.....	83
	Deleting a trade agreement.....	83
	Mailbox chart.....	84
	Good, In, and Out mailboxes.....	84
	Bad mailboxes.....	85
CHAPTER 5	User Administration	87
	Overview	88
	Getting started.....	88
	Administrative tasks	89
CHAPTER 6	Changing Passwords	91
	Introduction	92
	Changing passwords.....	92
CHAPTER 7	Importing Definitions.....	93
	Introduction	94
	Importing a COBOL record definition	94
	Importing an ODBC record definition	96

	Importing a record definition from an XML data file or XML schema	98
	Importing a record definition from an HTML form	100
	Importing a record definition from a map	102
	Importing a record definition from a file.....	102
CHAPTER 8	Exporting Definitions.....	105
	Introduction	106
	Creating ODBC tables.....	106
	Creating templates for an HTML form.....	107
	Customizing the HTML form.....	108
	Export schemas	114
	Exporting to e-Biz 2000	114
	Exporting to MSQI or e-Biz Integrator	115
CHAPTER 9	Working with Records and Tables	119
	Introduction	120
	Performing record-related activities	120
	Working with records and tables	121
	Performing field-related activities	123
	Working with fields	123
	Associating records and files	129
CHAPTER 10	Working with Files and Databases.....	131
	Introduction	132
	Performing file-related activities	132
	Creating a new file or database.....	133
	Administering files or databases	136
CHAPTER 11	Working with Directories and Mailboxes.....	139
	Introduction	140
	Performing directory-related activities	140
	Creating, renaming, and deleting directories.....	141
	Associating files and directories.....	142
CHAPTER 12	Creating Transactions	145
	Introduction	146
	Create transaction functions	146
	Administrative tasks for EDI messages.....	147
	Working with the standards library	152
	Administering transactions in the transaction table view	154
	Segments – transaction table view.....	156

	Elements – transaction table view	158
	Code lists – transaction table view	160
	The standards library – transaction tree view	162
	Transactions – transaction tree view	162
	Segments – transaction tree view	164
	Elements – transaction tree view	166
	Code lists – transaction tree view.....	168
CHAPTER 13	Mapping.....	173
	Introduction	174
	Actions from the Transaction Mapping window.....	174
	Actions from the Any-to-Any Map window.....	176
	Transaction mapping and any-to-any mapping.....	176
	Transaction mapping.....	180
	Working with inbound map segments	180
	Working with inbound map elements	185
	Working with inbound map application data.....	194
	Mapping EDI elements to application data.....	194
	Working with outbound map segments	201
	Working with outbound map elements	208
	Working with outbound map application data.....	216
	Mapping EDI data to application data	217
	Assigning levels	223
	Any-to-any mapping.....	225
CHAPTER 14	Conditional Mapping	229
	About conditional mapping	230
	Inbound conditional mapping	231
	Segment mapping based on trading partner	231
	Element mapping based on trading partner	232
	Element mapping based on data criteria.....	233
	Outbound conditional mapping	236
	Segment mapping based on trading partner	237
	Segment mapping based on data criteria.....	238
	Element mapping based on trading partner	239
	Element mapping based on presence of data.....	240
	Trading partner conditionals: inbound and outbound.....	242
	Mapping segments.....	242
	Mapping elements.....	243
	Trading partner-specific examples	244
	Inbound conditional segment mapping.....	244
	Inbound conditional element mapping.....	245
	Outbound conditional segment mapping.....	246

	Outbound conditional element mapping	246
CHAPTER 15	Creating a Map Flow	249
	Introduction	250
	Assigning levels to EDI segments	253
	Adding, modifying, and deleting levels	253
	Creating outbound map flow	254
	Automatically creating outbound map flow	254
	Manually creating outbound map flow	255
	Creating inbound map flow	263
	Automatic map flow creating	263
	Manually creating inbound map flow	263
	Creating an any-to-any map flow	270
	Manually creating and any-to-any map flow	270
CHAPTER 16	Generating Maps	281
	Introduction	282
	Generating maps	282
	Reviewing log file entries	283
CHAPTER 17	Running Maps	285
	Introduction	286
	Running an inbound map	287
	Required tab	287
	Option 1 tab	291
	Option 2 tab	294
	File Alias tab	303
	ODBC Alias tab	305
	Parameters tab	306
	I/O Redirect tab	308
	Web Script tab	310
	Running an inbound map	311
	Building a functional acknowledgement	314
	Required tab	314
	Option 1 tab	316
	Option 2 tab	317
	File Alias tab	318
	ODBC Alias tab	319
	Parameters tab	320
	I/O Redirect tab	321
	Running a functional acknowledgement	322
	Working with TA1 interchange acknowledgements	324

Setting up an interchange acknowledgement	325
Requesting a TA1 acknowledgement for outbound EDI maps	328
Building a reconciliation map	328
Running an outbound map	330
Required tab	330
Option 1 tab	334
Option 2 tab	336
File Alias tab	339
ODBC Alias tab	340
Parameters tab	341
I/O Redirect tab	342
Web Script tab	343
Running an outbound map	344
Running an any-to-any map	347
Required tab	347
Option 1 tab	349
Option 2 tab	351
File Alias tab	353
ODBC Alias tab	354
Parameters tab	355
I/O Redirect tab	356
Web Script tab	357
Running an any-to-any map	358

CHAPTER 18	Working with Memory Variables	361
	Introduction	362
	Using the Memory Variables window	363
	Adding memory variables	363
	Modifying memory variables	367
	Deleting memory variables	367
	Importing memory variables from another map	367
	Searching for memory variables	368
	Using the Memory Variable pane	368

CHAPTER 19	Working with System Variables	369
	Introduction	370
	Working with system variables	370
	Viewing system variables	370
	Using a system variable in a rule command	371
	Using a system variable on the Any-to-Any Map window	375
	Alphabetical listing of EMap system variables	376

CHAPTER 20	Working with Rules.....	397
	Introduction	398
	Overview of rule commands.....	399
	Creating and using rules	403
	Working with the Rule Definitions window.....	404
	Working with rule commands	409
	Rule commands reference	411
	Abort transaction	411
	Arithmetic	413
	Assignment.....	415
	Binary Data Placement.....	416
	Check Point	416
	Clear	418
	Comment.....	418
	Concatenate Strings.....	418
	Do Nothing	421
	File Management.....	421
	HTML I/O.....	422
	If Condition	423
	Keyed Record I/O.....	426
	Map Level.....	427
	NDO File Command	428
	Perform Rule	429
	Put File to Queue	431
	Sequential I/O.....	432
	SQL	432
	Stop Run	444
	String Operations	444
	Substring	452
	Table Conversion	454
	User Exit.....	455
	Write Log.....	459
	XML I/O	462
CHAPTER 21	Using the View Menu	465
	Introduction	466
	Viewing standards.....	466
	Viewing work space	467
	Viewing toolbars.....	467
	Viewing small or large icons.....	468
CHAPTER 22	Working with Cross-Reference Tables	469
	About cross-reference tables	470

	Accessing cross-reference tables from Tools	471
	Cross-reference tables for transaction maps	471
	Cross-reference tables for any-to-any maps	478
	Accessing cross-reference tables from Mapping	483
	Importing Cross-reference tables	486
CHAPTER 23	Copying Maps	489
	Introduction	490
	Working with the copy map utility	490
CHAPTER 24	Importing the SEF Standard	497
	About the SEF Import utility	498
	Working with the SEF Import utility	498
CHAPTER 25	Working with Print Maps.....	501
	About print maps	502
	Working with print maps	502
	Viewing log files	503
	Associating a print map with a trading partner	505
	Running a print map	505
	Modifying and regenerating a print map.....	506
CHAPTER 26	Working with Compliance Maps.....	509
	The EDI environment	510
	Typical EDI transactions	510
	Compliance checking and data mapping	511
	Creating and running a compliance map	514
	Defining a compliance map	514
	Creating the transaction to be checked by the compliance map..	515
	515	
	Associating the compliance map with a trading partner	515
	Generating a compliance map	517
	Running the compliance map	517
	Modifying and regenerating the compliance map.....	519
	Running a compliance map with a batch command file	519
CHAPTER 27	EDI to XML Generate Map Utility	523
	Introduction	524
	Working with the EDI to XML utility	524
CHAPTER 28	Update Database Structures.....	529

	About your databases	530
	Updating the database	530
	Upward-compatibility	531
	Backing up maps.....	531
CHAPTER 29	Compact Databases.....	533
	About the Compact Database utility.....	534
	Working with the utility	534
CHAPTER 30	Update Dates	535
	About the date utility.....	536
	Working with the date utility	536
CHAPTER 31	Generating Reports	539
	Alphabetic listing of all EMap reports	540
	Transaction map reports	543
	Accessing transaction map reports from the Reports menu .	543
	Accessing transaction map reports from the View menu	545
	Any-to-any map reports.....	547
	List of any-to-any map reports.....	547
	Accessing any-to-any reports from the Reports menu	549
Index		551

About This Book

This document is a user guide for EMap™, a tool for building and understanding structured information messages.

Audience

While you certainly do not need to be a programmer to use EMap, it is helpful to be familiar with certain technical concepts, such as the following:

- ECRT™ (process engine)
- Electronic Data Interchange (EDI) and HIPAA concepts
- HIPAA transaction formats and usage

How to use this book

This guide describes how to use EMap. It is organized into the following chapters:

- Chapter 1, “EMap Overview” presents a brief summary of EMap features and emphasizes additions to product functionality.
- Chapter 2, “Working with Projects” discusses how to add, modify, delete and select projects, as well as import and export projects to associated maps.
- Chapter 3, “Working with Maps” discusses how to add, select, work with, archive, and generate multiple maps in a project.
- Chapter 4, “Working with the Address Book” discusses EMap’s address book, which contains information about your company and its trading partners.
- Chapter 5, “User Administration” discusses administrative tasks, such as adding users and changing user passwords.
- Chapter 6, “Changing Passwords” discusses EMap’s Change Password function.
- Chapter 7, “Importing Definitions” discusses how to import record definitions in EMap.
- Chapter 8, “Exporting Definitions” discusses how to export record definitions in EMap.

-
- Chapter 9, “Working with Records and Tables” discusses how to work with records and tables.
 - Chapter 10, “Working with Files and Databases” discusses how to perform file-related activities in EMap.
 - Chapter 11, “Working with Directories and Mailboxes” discusses how to perform directory-related activities in EMap.
 - Chapter 12, “Creating Transactions” discusses how to create transactions in EMap and how to work with the standards library.
 - Chapter 13, “Mapping” discusses how the EMap mapping function works.
 - Chapter 14, “Conditional Mapping” discusses inbound and outbound conditional mapping, including trading partner conditions.
 - Chapter 15, “Creating a Map Flow” discusses how to work with map flows, such as assigning levels to EDI segments, creating outbound and inbound map flows, and manually creating map flows in any-to-any maps.
 - Chapter 16, “Generating Maps” discusses how to compile and generate maps.
 - Chapter 17, “Running Maps” discusses how to build functional acknowledgements and run inbound, outbound, and any-to-any maps.
 - Chapter 18, “Working with Memory Variables” discusses how to use memory variables in EMap.
 - Chapter 19, “Working with System Variables” discusses how to work with EMap’s system variables.
 - Chapter 20, “Working with Rules” discusses how to create and use rules.
 - Chapter 21, “Using the View Menu” discusses EMap’s view menu.
 - Chapter 22, “Working with Cross-Reference Tables” discusses how to create and use cross-reference tables.
 - Chapter 23, “Copying Maps” discusses how to use EMap’s copy map utility.
 - Chapter 24, “Importing the SEF Standard” discusses the SEF (Standard Exchange Format) Import utility, which works in conjunction with EMap.
 - Chapter 25, “Working with Print Maps” discusses how to use the print maps utility.

- Chapter 26, “Working with Compliance Maps” discusses how to create and run compliance maps.
- Chapter 27, “EDI to XML Generate Map Utility” discusses the EDI to XML Generate Map utility.
- Chapter 28, “Update Database Structures” discusses how to update database structures.
- Chapter 29, “Compact Databases” discusses the Compact Databases utility.
- Chapter 30, “Update Dates” discusses an EMap utility that updates dates.
- Chapter 31, “Generating Reports” discusses how to run reports in EMap.

Related documents

The following documents ship with EMap:

- *EMap New Features Guide*
- *EMap Installation Guide*
- *Release Bulletin for EMap*
- *EMap Reference Guide*
- *EMap User Guide*
- *EMap Getting Started*

Additional documents are referred to in the EMap documentation to supply you with specific information that supports this product:

- *ECRTP Reference Guide* to use the data transformation engine

Documentation that supports EMap can be found on the Sybase Product Manuals web site. Go to Product Manuals at <http://www.sybase.com/support/manuals>, select EMap from the drop-down list, and click Go!

Other sources of information

Use the Sybase Getting Started CD, the SyBooks Bookshelf 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 Bookshelf 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 Bookshelf CD is included with your software. It contains product manuals in a platform-independent bookshelf that contains fully searchable, HTML-based documentation.

Some documentation is provided in PDF format, which you can access through the PDF directory on the SyBooks Bookshelf CD. To view the PDF files, you need Adobe Acrobat Reader.

Refer to the *README.txt* file on the SyBooks Bookshelf CD for instructions on installing and starting SyBooks.

- The Sybase Product Manuals Web site is the online version of the SyBooks Bookshelf 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 Product Manuals Web site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

EDI Standards resources

You can access information about the EDI Standards from the following organizations:

Standard	Organization	URL
X12	Accredited Standards Committee, X12	http://www.x12.org
EDIFACT	United Nations Directories for Electronic Data Interchange for Administration, Commerce and Transport	http://www.unece.org/trade/untdid/welcome.htm
HL7	Health Level 7	http://www.hl7.org
NCPDP	National Council for Prescription Drug Programs	http://www.ncdp.org

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

v Finding the latest information on product certifications

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Select Products from the navigation bar on the left.
- 3 Select a product name from the product list and click Go.
- 4 Select the Certification Report filter, specify a time frame, and click Go.
- 5 Click a Certification Report title to display the report.

v **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

v **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.

Typographic conventions

This documentation uses the following typographic conventions:

Item	Description
Code	SQL and program code displays in a mono-spaced (fixed-width) font.
User entry	Text entered by the user is shown in bold serif type.
<i>emphasis</i>	Emphasized words are shown in italic.
<i>file names</i>	File names are shown in italic.
database objects	Names of database objects, such as tables and procedures, are shown in sans serif type in print, and in italic online.

Item	Description
<i>sybase\bin</i>	A backward slash (“\”) indicates cross-platform directory information. A forward slash (“/”) applies to UNIX-specific information. Directory names appearing in text display in lowercase unless the system is case sensitive.
File > Save	Menu names and menu items are displayed in plain text. The angle bracket indicates how to navigate menu selections, such as from the File menu to the Save option.
parse put get	The vertical (pipe) bar indicates <ul style="list-style-type: none"> • Options available within code • Delimiter within message examples
segment	Bold text indicates a glossary term.

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.

ECMap Overview

About this chapter

This chapter discusses the basics of ECMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	2
What can ECMap do?	3
Working with ECMap	4
Task overview	8

Introduction

ECMap is a powerful mapping tool provides support for creating and processing the structured information messages.

Target user

ECMap is oriented toward the analyst rather than the programmer. ECMap does not force its mappers to use programming code to do complex integration, but instead has powerful built-in capabilities, such as rules and flow:

- Flow – used in ECMap to mean the context in which data is interpreted, while rules are machine instructions for processing. Flow provides the means to control the sequencing of data operations — unlike other mapping products that force the user to output the data as it occurs.
- Rules – used to insert sophisticated processing logic into maps, it enables ECMap to handle a wide range of mapping tasks.

In addition, the combination of rules and flow provides ECMap with content-specific routing, which allows data to be directed into different files or to different locations based on the data content

ECMap components

ECMap has two distinct components:

- Workbench for developing maps
- Runtime program (RTP) for performing the data transformations specified in the map

Format classes

ECMap can handle three classes of formats:

- Traditional formats of flat files and databases
- EDI Standards: X12, EDIFACT, NCPDP, and HL7
- HTML or XML data

In addition, users can also import other standards with the EDI Standard Exchange Format (SEF) Import utility. Once imported, these standards can be used like the other standards that are bundled with the product. ECMap allows you to perform application-to-application integration, as well as Web-enabling XML data and XML-enabling databases.

What you can create

ECMap allows you to create:

- Transaction maps that perform data conversions between application data and EDI standards
- Any-to-any maps that convert data directly from one application to another

- Any-to-any maps that perform data conversions between HTML/XML data and application data
- Any-to-any maps that perform data conversions between HTML or XML data and EDI standards.

What can ECMap do?

ECMap has the ability to:

- Automatically generate a set of record and field definitions from a COBOL copybook, an ODBC file, a map or a special file.
- Automatically generate an ODBC database table from a set of record and field definitions.
- Use a wide variety of rule commands, including conditional logic, to embed business rules and proprietary compliance checks in your map.
- Use a wide range of SQL commands from within an SQL rule command, allowing direct integration with any ODBC-compliant database.
- Access functions outside of the product through User Exit rule commands.
- Perform dynamic map switching at runtime.
- Perform interactive, “real-time” transactions.
- Perform “any-to-any” mapping, directly integrating diverse application databases.
- Process large volumes at high speeds, without having to break up and reassemble the data.
- Translate data independent of file size.
- Perform data conversions between HTML or XML data and EDI standards.
- Automatically generate a set of record and field definitions from an HTML form.
- Automatically create the template for an HTML form from a set of record and field definitions.
- Dynamically build and populate an HTML form based on retrieved data and an HTML form template.

- Perform dynamic interactions between an HTML form and the values in a database.
- Map XML data to a proprietary application.
- Import customized standards (such as standards based on Implementation Guides) and use them just as you use the major standards included with the tool—X12, EDIFACT, and HL7.
- Automatically create a field from an EDI element during map development.
- Perform conditional mapping, by associating particular mapping commands to a specific trading partner, to the presence or absence of data, etc.
- Quickly and easily perform element to field mapping for any standard supported by EMap.
- The ability to map directly from one proprietary application to another.
- Create and run special “compliance” maps to verify that the data being processed by a map is compliant with the standard being used.
- Automatically generate a batch file that includes all of the switches needed to run the map from a command line.
- Use the runtime capability of substituting either a new input file or a new output file for the file specified in the map, regardless of whether the files are flat files or ODBC databases.
- Pass parameters into the map at runtime.
- Verify compliance with the HIPAA mandate.

Working with EMap

Starting EMap

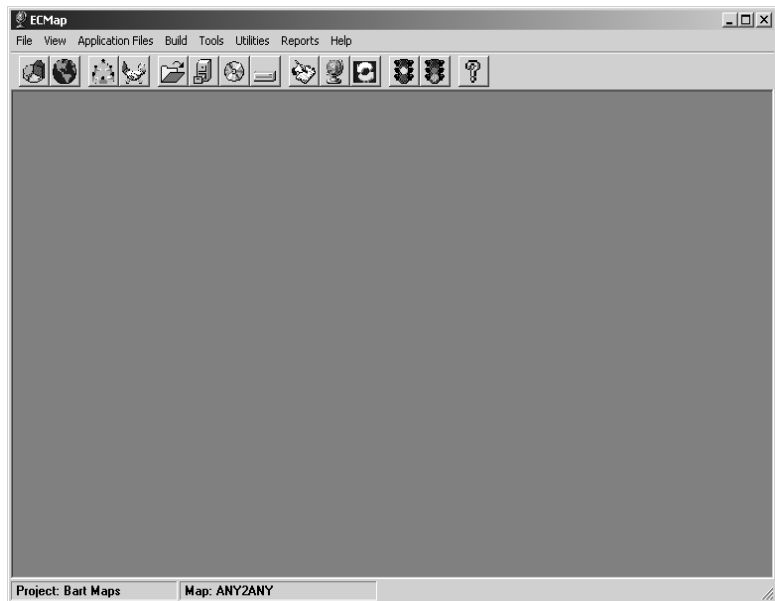
- ❖ **Starting EMap**
 - 1 Double-click the EMap icon on your desktop.
 - 2 Keep the user name as `Admin`.

- 3 Enter the password, `emap` and select OK.

Note The password is case sensitive. Enter `emap` in all lowercase.

About the workspace

The EMap workspace is a standard application window consisting of a main menu, a toolbar, and a status bar.



Like many Windows-based applications, there are several methods of accessing information and performing tasks. You can use the:

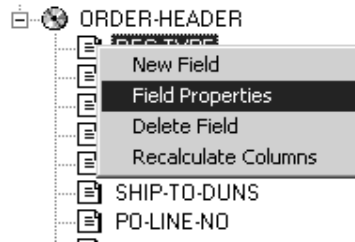
- Main menu

File View Application Files Build Tools Utilities Reports Help

- Toolbar

















- Context menu – when you right-click on an item you a context menu displays available tasks that can be performed on the current item



Toolbar icons

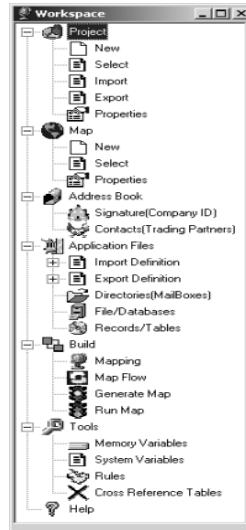
The EMap toolbar offers quick access to commonly used tasks.

Icon	Description
	Opens the Project window. Use to create, delete, and maintain projects. For more information see Chapter 2, “Working with Projects.”
	Opens the Map window. Use to create, delete, and maintain maps. For more information see Chapter 3, “Working with Maps.”
	Opens the Company ID window. Use to set up and maintain the company ID used in data exchange and communications with trading partners. For more information see Chapter 4, “Working with the Address Book.”
	Opens the Trading Partner window. Use to set up and maintain trading partner information and associate trade agreements with specific partners. For more information see Chapter 4, “Working with the Address Book.”
	Opens the Directories (Mailboxes) window. Use to set up and maintain application directories. For more information see Chapter 11, “Working with Directories and Mailboxes.”
	Opens the Files/Databases window. Use to set up and maintain application files or databases. For more information see Chapter 10, “Working with Files and Databases.”
	Opens the Records/Tables window. Use to set up and maintain application records, tables, and fields. For more information see Chapter 9, “Working with Records and Tables.”
	Opens the Memory Variables window. Use to create and maintain memory variables. For more information see Chapter 18, “Working with Memory Variables.”

Icon	Description
	Opens the Rule Definitions window. Used to create and maintain map rules For more information see Chapter 20, “Working with Rules.”
	Opens the Mapping window. Use to create mapping, set conditions, and flow levels. For more information see Chapter 13, “Mapping.”
	Opens the Flow window. Use to create and maintain application flow, which controls the way application data is processed. For more information see Chapter 15, “Creating a Map Flow.”
	Opens the Generate Map window. Use to compile a map for execution. For more information see Chapter 16, “Generating Maps.”
	Opens the Run Map window. Use to execute a map to process application data. For more information see Chapter 17, “Running Maps.”
	Help opens the online help. <hr/> For latest information The online help has not been updated for ECMap 4.2. Refer to the <i>ECMap User Guide</i> , Release Bulletin, and New Features guide up-to-date information. <hr/>

Workspace tool

In addition to the toolbar you can display the Workspace tool which offers the convenience of the toolbar from within the EMap main window.



Task overview

The steps for creating a new map in EMap can vary from map to map, depending on the EDI standard you use, the type and direction of the transaction, and the complexity of your application. The following section covers the general procedures required to create and run a map.

- ❖ **To create, compile, and run a new map**
 - 1 Create a project
 - 2 Create a map
 - 3 Set up a Company ID
 - 4 Set up your Trading Partner
 - 5 Add a Trade Agreement to your Trading Partner
 - 6 Define application data by creating and associating:
 - Records and fields

- Files and databases
 - Directories
- 7 Create a Transaction
 - 8 Build the map
 - a Define levels and flow
 - b Map records and segments
 - c Build map logic
 - d Define flow
 - 9 Generate the map
 - 10 Run the map

Creating a project

The first step in creating a map is to create a project that will contain the map and map-related information.

Specify the following information for the project:

- Name
- Description
- Directory
- Contact
- Phone Number

For more information See Chapter 2, “Working with Projects.”

Creating a map

When create a map, you specify the following information

- Map Properties – including map name, standard used, application type, direction
- Map Directories – directory locations for map files, trading partner information and applicable EDI standards

- Map DSN – ODBC connections to trading partner database and log files

For more information See Chapter 3, “Working with Maps.”

Setting up a Company ID

Before you can exchange data with other organizations, you must set identify your company to EMap. You do this by setting up a company ID, where you specify how your company will be identified during EDI transactions, including:

- Profile number
- Company name
- Default EDI envelope information to be included in all transactions for the appropriate standard

For more information See Chapter 4, “Working with the Address Book.”

Setting up your Trading Partner

Once you’ve set up a Company ID for yourself, you need to identify the trading partners with whom you wish to exchange data. You do this by specifying information that identifies your trading partner to EMap, including:

- Internal ID
- Trading partner name
- Location messages
- Contact information
- EDI default envelope information to identify the trading partner in
- Any file delimiters specific to this trading partner

For more information See Chapter 4, “Working with the Address Book.”

Adding a Trade Agreement to your Trading Partner

Part of the trading partner definition process is to add a trade agreement to specific trade partners which associate specific standards, transactions and maps with the trading partner, including

- The EDI standard used for this trade agreement
- The map, transaction, direction for the application flow
- Any overrides specific to this transaction and trade partner

For more information See Chapter 4, “Working with the Address Book.”

Defining application data

Creating records

Having set up your project, map, company and trading partners, you can now populate your map with application data. You can do this two ways:

- Import records from an outside source
- Create records manually

Depending on your application, you may be working with records and files (if your application data is stored in flat files) or tables and databases (if your application data is stored in a database).

For more information See Chapter 9, “Working with Records and Tables.”

Creating files

Once you have created your application records, you must associate them with a file that ECMap can use to locate the application data

For more information See Chapter 10, “Working with Files and Databases.”

Creating directory

Finally, you must associate your application records and file with a directory that ECMap can use to reference the application data

For more information See Chapter 11, “Working with Directories and Mailboxes.”

Creating a Transaction

If your map uses one of the EDI standards, you must create a transaction so specify to EMap the segment and elements will make up the data you are exchanging with your trading partner.

For more information See Chapter 12, “Creating Transactions.”

Building a map

Once you have selected the segments and elements you will use to exchange data with your trading partner, you can proceed to build the map. You build your map by:

- Defining levels and flow
- Mapping records and segments
- Building map logic

Defining map levels and flow

When you define a map levels, you specify the structure of the application data and group EDI elements for processing:

- *For outbound or any-to-any maps*, flow is based on when application records are to be read
- *For inbound maps*, flow is based on how EDI segments are to be processed and application data written

Mapping records and segments

After you’re created application levels, you can map the application data to the corresponding EDI elements

Building map logic

You can build logic into your maps by using rules, conditions, memory variables, system variables, cross-reference tables to control the mapping data

For more information See

- Chapter 13, “Mapping.”

- Chapter 15, “Creating a Map Flow.”
- Chapter 14, “Conditional Mapping.”
- Chapter 20, “Working with Rules.”
- Chapter 18, “Working with Memory Variables.”
- Chapter 19, “Working with System Variables.”
- Chapter 22, “Working with Cross-Reference Tables.”

Generating a map

Once your map has been completed, you need to generate your map to create a version of it that ECMap can run efficiently. During the generation process, you may encounter problems that you will be able to debug and fix in order to have a create a usable map.

For more information See Chapter 16, “Generating Maps.”

Running a map

The final step is to run your map. If your map has been created successfully, the EDI and application data should be exchanged as you intended. If there are problems running the map, or problems with the data that is exchanged, you will be able to generate trace logs to help you debug and fix your processing.

For more information See Chapter 17, “Running Maps.”

Working with Projects

About this chapter

This chapter discusses projects in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	16
Adding new projects	17
Modifying projects	18
Deleting projects	18
Selecting projects	19
Exporting a project with associated maps	19
Importing a project with associated maps	20

Introduction

ECMap uses projects to organize maps so they are easy to locate and use. Projects can be set up in any way that suits your business purposes, such as creating projects based on trading partners or business applications.

You can add a new project, modify a project, delete a project, select a project, export an entire project with all of its associated maps, or import an entire project and its maps.

Where data is stored	When you set up your first project, the system creates an Access database file called projects.mdb. This master database file consists of three tables—mproject, mtable, and mxref—which contain information about all of the projects, maps, and cross-reference tables in ECMap. The information in these tables is updated each time that you add, modify, or delete a project, map, or cross-reference table.
mproject table	<p>The mproject table contains the names of all of the projects in your system. It contains all of the information entered in the following five text boxes on the Project window:</p> <ul style="list-style-type: none">• Project Name• Project Description• (Project) Directory• Contact• Phone Number
mtable table	mtable contains the names of all of the projects and maps in your system. The project name links this table to the mproject table, and the project name/map name combination links this table to the mxref table. Each record in the table contains the name of a project, a map within that project, and the related trading partner information. The mtable table contains all of the information entered on the four tabs of the Map Definition window.
mxref table	<p>The mxref table contains the names of all the project, maps, and cross-reference tables in your system. Each record in the table contains the name of a project, a map within that project, and a cross-reference table used in that map. The project name links this table to the mproject table, and the project name/map name combination links this table to the mtable table. Since the same cross-reference tables can be used in more than one map, there may be multiple records that have the same cross-reference table name.</p> <p>The mxref table contains all of the information entered on the Cross-Reference Table window.</p>

When you export a project, EMap creates a copy of these same three tables, but they contain information only for the project being exported. When you import the project, EMap uses the exported project-specific information in these three tables to update the master projects.mdb database in the EMap installation to which the project is being imported.

Adding new projects

You add new projects through EMap's New Project window.

❖ Creating a new project

- 1 Choose File>Project>New from the EMap menu or the Project window menu.

or

Right-click anywhere on the Project window, then select New>Project from the context menu. The New Project window displays.

- 2 Enter the following information:
 - Project Name – a short descriptive title that allows you to quickly recognize the contents of the project. It is required.
 - Project Description (optional) – a brief textual explanation of the project.
 - Directory – the folder, or directory, where the project is physically stored on the computer. It is required and must include the entire path to the directory. Select Browse to search for the directory on the Select a Directory ... window. Double-click a directory and it is entered automatically in the Directory text box.
 - Contact (optional) – the name of the individual who is responsible for the maps in this project.
 - Phone Number (optional) – the telephone number of the individual who is responsible for the maps in this project.
- 3 Click OK to return to the Project window.

Modifying projects

Projects are modified on the Project Properties window.

❖ Modifying a project

- 1 Choose File>Project>Properties from the main menu

or

Highlight the project that you want to modify in the Project window, and choose Properties from the Edit menu.

or

Right-click the project that you want to modify on the Project window, and choose Properties from the context menu.

The Project Properties window displays the current information for the project.

- 2 Modify the following properties, as needed:

- Project Description – a brief textual explanation of the project. It is optional.
- Contact (optional) – the name of the individual who is responsible for the maps in this project.
- Phone Number (optional) – the telephone number of the individual who is responsible for the maps in this project.

You cannot modify a Project Name or Directory. To change either, create a new project.

- 3 Select OK to return to the Project window.

Deleting projects

❖ Deleting a project:

- Do one of the following:
 - Highlight the project that you want to delete on the Project window, and choose Delete from the Edit menu.

- Right-click the project that you want to delete on the Project window, and choose Delete from the dropdown menu.

In each case, you are asked to confirm that you want to delete the project before it is removed permanently.

Selecting projects

❖ Selecting a project:

- Do one of the following:
 - Choose File>Project>Select from the main menu.
 - Highlight a project on the Project window, and choose File>Select from the Project window menu.
 - Double-click a project on the Project window to list the maps for the project, then double-click a map to select the project and map.
 - Right-click a project on the Project window and choose Select from the context menu.

The name of the project you selected displays on the status bar.

Exporting and importing projects and maps

You can copy an entire project from one system to another using the export and import features. This allows groups of maps to be easily shared between co-workers. Projects which have been exported from one system can be imported into another system.

Exporting a project with associated maps

What gets exported

When you export a project, you create a copy of the *projects.mdb* database that contains all of the map and cross-reference information for the project. This includes the *mproject*, *mtable*, and *mxref* tables discussed in “Where data is stored” on page 16.

Collecting additional data for export

Since the Export Project function only creates the database and tables, you must place these tables together with all of the other map-related files for this project and make them available for importing.

For example, if you store all of the maps in the project in subdirectories under the main project directory, you can export the `projects.mdb` database into the main project directory, then compress the directory transfer it to another system.

❖ Exporting a project

- 1 Choose File>Project>Export.

The Export Project Definition window displays.

- 2 Select the up arrow to browse for a Project to export.

The Select Project window displays.

- 3 Double-click the project you want to export.

- 4 Enter the full path of the directory where you want to export the project.

or

Select Browse to select a directory in the Select a Directory window.

- 5 Select Run to export the selected project.

You do not receive a message saying that the project definition has been successfully exported, but you can verify success by looking in the directory.

If you exported the project database into the main project directory, that directory is available for importing. If you exported the database into a different directory, you must put the tables and the project files together in one directory.

Importing a project with associated maps

The Import Project function allows you to copy into your system an entire project that has been exported from another system. During the export process, EMap creates a `projects.mdb` file for the project being exported. Then, when the exported project is being imported, EMap uses the information in this `projects.mdb` file (created during export) to update the master `projects.mdb` file for the importing system.

In addition to updating the projects.mdb file, the Import Project function also adds all of the map-related files for the exported project to the importing system. The exporting user must place all of the map-related files for the exported project in the same directory as the exported projects.mdb file. The importing user can then import all of the necessary files and tables from this directory during the Import Project function.

❖ **Importing a project**

- 1 Choose File>Project>Import.

The Import Project Definition window displays.

- 2 Enter the full path of the directory from which to import the project.

or

Select Browse to choose a directory in the Select a Directory window.

- 3 Select Run to import the project in the selected directory.

The Project window displays with the name of the project you are importing.

- 4 Select OK to accept the project name that is displayed or enter a different project name and select OK.

If the project already exists, a Warning dialog box displays.

If you select No, you return to the Import Project Definition window. If you select Yes to rename the project, the Project dialog box displays again, allowing you to rename the project. After you have successfully renamed the project, the Project Directory window displays with the directory (full path) of the imported project

- 5 Select OK to accept the directory path that is displayed or enter a different directory path and then select OK.

The program imports the project. You do not receive a message saying that the project has been imported, but you can verify it by checking to see if it is listed On the Project window.

Working with Maps

About this chapter

This chapter discusses how EMap produces maps.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	24
Adding new maps	25
Selecting a map	34
Working with maps	34
Generating multiple maps in a project	38
Archiving a map	38
Automatic mapping of EDI to and from serialized NDO	39

Introduction

A map is a set of instructions used to transform data from one format to another. EMap can create:

- **Transaction maps** use EDI standards data in the mapping process—either as input or output.
- **Any-to-any maps** use application data as both the input and the output.

An any-to-any map can use EDI data, but it treats the EDI data as application data, rather than standards data.

Map direction

Transaction maps are said to be either:

- Inbound
- Outbound

An *inbound map* uses an EDI message to create application data. The EDI message is the input to the map, and it is being sent in to your company from your trading partner.

An *outbound map* uses application data to create an EDI message. The EDI message is the output of the map, and it is generally being sent from your company out to your trading partner.

EMap also uses two specialized maps, *compliance maps* and *print maps*, which can also be said to have an inbound direction. They are also said to have direction. They are discussed in Chapter 25, “Working with Print Maps,” and Chapter 26, “Working with Compliance Maps.”

In EMap, each transaction map is associated with a specific EDI standard—X12, EDIFACT, NCPDP, or HL7—and a specific transaction within that standard.

Where maps are stored

Maps are stored in projects. You must have a separate map for each business message that you exchange with a trading partner. For EDI transactions, this means one map per message type, application system, and map direction. The name given to a map often reflects the message type and the direction. For example, an X12 map that uses your company’s internal application data to produce an EDI invoice that is sent to a trading partner might be named 810OUT.

What you can do with maps using EMap

In EMap, you can add a new map, select a map, modify a map, delete a map, or copy a map. You can also generate all of the maps in one project, generate all of the maps in all of the projects, or view all of the maps in a project.

Starting point

You perform all map-related actions from the Map window.

❖ Accessing the Map window

- Select the Map icon on the main EMap toolbar

or

Choose File>Map>Select from the main menu

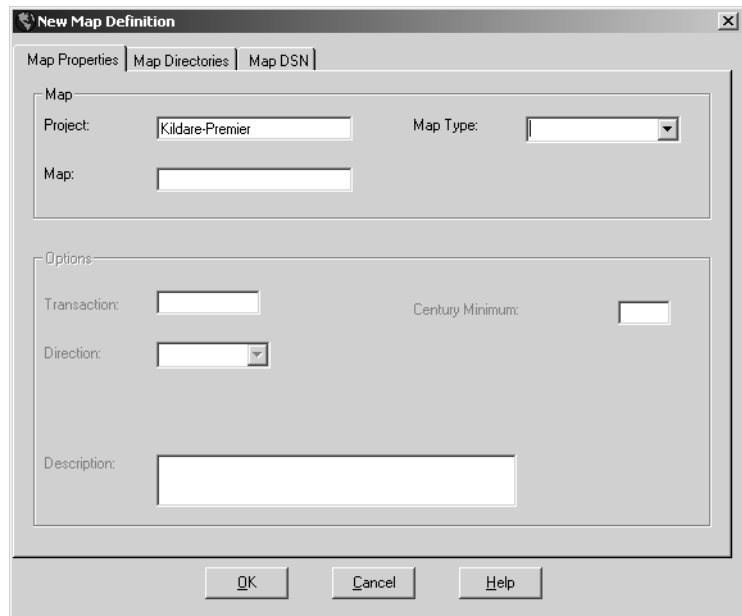
Adding new maps

New maps are defined on the New Map Definition window.

❖ Creating a new map

- 1 Choose File>Map>New from the main menu or the Project Window menu.

The New Map Definition window displays.



- 2 Enter the map information on three tabs:

- Map Properties

Map Properties | Map Directories | Map DSN

Map

Project: Map Type:

Map:

Options

Transaction: Century Minimum:

Direction: 8 Digit Date in X12 Envelope:

Version:

Description:

- Map Directories

Map Properties | Map Directories | Map DSN

Transaction Maps:

Application File Description:

Level Names:

Loop Flow Table:

Trading Partner:

Company Identification:

Rule Definition:

Generated Tables:

Cross Reference Tables:

Report Generation:

EDI Standard Tables:

- Map DSN

The screenshot shows a dialog box titled 'Map DSN' with three tabs: 'Map Properties', 'Map Directories', and 'Map DSN'. The 'Map DSN' tab is active. It contains two sections: 'Trade Partner' and 'Log'. Each section has a 'Data Source Name' field, a 'Configure Data Source' button, a text area for the driver connect string, and buttons for 'Clear [Section] Fields' and 'Test [Section] Connection'.

Defining map properties

The tabs have a different appearance on the New Map Definition window, depending on whether the map is a transaction map or any-to-any map.

Transaction maps require more information than any-to-any maps. They require all of the standards-related options – transaction, direction, version, and date information for envelopes – as well as the location of the standards directory. Maps that are being copied from another system should be defined just as they were when they were created in the other system. (Maps that are imported with a project are automatically defined during the import process.)

Any-to-any maps require less information than transaction maps, since they do not need any standards-related information—transaction, direction, version, date information for envelopes, or the location of the standards directory.

X12 standard

Enter these properties on the properties tab:

- Project Name – displays the name of the current project.
- Map Name – type a name for the map. It should have a maximum of 60 characters.

- Map Type –select X12 from the drop-down list.
- Transaction – type the name of the X12-specific identifier for the EDI message being mapped.
- Direction – select the direction for EDI data from the drop-down list. The choices are:
 - **IN** indicates that EDI data is the source (input) for the map transformation
 - **OUT** indicates that EDI data is the result (output) of the map transformation
- Version – select the version of X12 used for the current map from the drop-down list.
- Century Minimum – type the year you want to use as global century minimum for this map.

When converting 6-digit dates (with 2-digit years) to 8-digit dates (with 4-digit years), years higher than the Century Minimum are presumed to be in the current century; years lower and including the Century Minimum are presumed to be in the previous century.

- 8-Digit Date in X12 Envelope – automatically populated to comply with the EDI version selected for the map.
- Description – a brief description for the map. (Optional).

HL7 properties

The information on the Map Properties tab defines the map:

- Project Name – displays the name of the current project.
- Map Name – type a name for the map. It should have a maximum of 60 characters.
- Map Type – select HL7 from the drop-down list.
- Message/Type – type the name of the HL7-specific identifier for the EDI message being mapped.
- Direction – select the direction for EDI data from the drop-down list. The choices are:
 - **IN** indicates that EDI data is the source (input) for the map transformation

- **OUT** indicates that EDI data is the result (output) of the map transformation
- Version – select the version of HL7 used for the current map from the drop-down list.
- Century Minimum – type the year you want to use as global century minimum for this map.

When converting 6-digit dates (with 2-digit years) to 8-digit dates (with 4-digit years), years higher than the Century Minimum are presumed to be in the current century; years lower and including the Century Minimum are presumed to be in the previous century.

- Description – a brief description for the map. (Optional).

EDIFACT properties

The information on the Map Properties tab defines the map:

- Project Name – displays the name of the current project.
- Map Name – type a name for the map. It should have a maximum of 60 characters.
- Map Type – select EDIFACT from the drop-down list.
- Message – type the name of the EDIFACT-specific identifier for the EDI message you are mapping.
- Direction – select the direction for EDI data from the drop-down list. The choices are:
 - **IN** indicates that EDI data is the source (input) for the map transformation
 - **OUT** indicates that EDI data is the result (output) of the map transformation
- Version – select the EDIFACT version used for the current map from the drop-down list.
- Century Minimum – type the year you want to use as global century minimum for this map.

When converting 6-digit dates (with 2-digit years) to 8-digit dates (with 4-digit years), years higher than the Century Minimum are presumed to be in the current century; years lower and including the Century Minimum are presumed to be in the previous century.

- Description – a brief description for the map. (Optional).

NCPDP properties

Enter these properties on the Map Properties tab:

- The Project name is inserted by the program.
- Map Name – type a name for the map. It should have a maximum of 60 characters.
- Map Type – select NCPDP from the drop-down list.
- Transaction – type the name of an NCPDP-specific identifier for the EDI message you are mapping. For example, you can type B1 for a billing transactions or RB1 for a response to a billing transaction.

Note All response transactions are in this format: R + *transaction name*.

- Direction – select the direction for EDI data from the drop-down list. The choices are:
 - **IN** indicates that EDI data is the source (input) for the map transformation
 - **OUT** indicates that EDI data is the result (output) of the map transformation
- Version – select the NCPDP version used in the current map from the drop-down list.
- Century Minimum – type the year you want to use as global century minimum for this map.

When converting 6-digit dates (with 2-digit years) to 8-digit dates (with 4-digit years), years higher than the Century Minimum are presumed to be in the current century; years lower and including the Century Minimum are presumed to be in the previous century.

- Description – a brief description for the map. (Optional).

Any-to-any properties

Enter these properties on the Map Properties tab;

- Project Name – displays the name of the current project.

- Map Name – type a name for the map. It should have a maximum of 60 characters.
- Map Type – select ANY2ANY from the drop-down list.
- Century Minimum – type the year you want to use as global century minimum for this map.

When converting 6-digit dates (with 2-digit years) to 8-digit dates (with 4-digit years), years higher than the Century Minimum are presumed to be in the current century; years lower and including the Century Minimum are presumed to be in the previous century.

- Description – a brief description for the map. (Optional).

Defining map directories

Map information

The information entered on the Map Directories tab specifies the directories that contain data associated with the current map. This information includes:

- Transaction Maps – contains the segment/element mapping files that define the EDI-to-application relationship in the current map.
- Application File Description – contains the object (database, table, directory, file, and record) definitions of the application data in the current map.
- Level Names – contains the numbers and descriptions of the levels that define the map flow in the current map.
- Loop Flow Table – contains context (semantic) information used in the current map.
- Trading Partner – contains information that is used on the EDI envelopes to identify the trading partner in the current map and information that identifies the maps associated with the trading partner.
- Company Identification – contains the default sender information, including the information that is used on the EDI envelopes to identify the company in the current map.
- Rule Definition – contains all the rules used in the current map.
- Generated Tables – contains the compiled.map file created when the map is generated and the trace files created when the map is run
- Cross Reference Tables – contains the files that are used to convert codes in the data being mapped.

- Report Generation – was used in earlier versions of the software and can be ignored.
- EDI Standards Tables – contains the EDI standard used to build the current map.

Default directory structure

The EDI Standards Tables directory is a common directory shared by all maps. If you chose the default directory during installation, the EDI standards are stored in *c:\Program Files\Sybase\Standards\<version>*.

The default location for all other map-related data is the *<project>\<map>* directory specified on the New Project window when the project was created. If you want multiple maps to share the same directories, your directory structure deviates from this default directory structure. This is permitted, but you must specify the correct directories on this window.

Changing map directories

If you're working with a transaction map, the first time you access the Map Directories tab, you'll see that the EDI Standards Tables location is blank. All the other information is pointing to the default location of *<project>\<map>*. You can change the directory for any of the map information, but you must identify the location of the EDI standard table for your map.

Using the Change/Protect button

The Change/Protect buttons to the right of each directory text box are toggle buttons that allow you to change the current directory location. If the button is set to Protect, the directory can not be changes. If the toggle is set to Change, you can change the directory.

The Protect feature is very useful for preventing a shared directory, such as the EDI Standards Tables directory, from being changed when a global directory change is made.

The Change/Protect button works in conjunction with the Change All and Invert buttons at the bottom of the tab page:

- Change All – performs global directory changes to all unprotected map directories.
- Invert – changes all protected directories to unprotected status and all unprotected directories to protected status.

When you select Change All,

❖ Specifying a map directory

- 1 Unprotect the directories you wish to change with the Change/Protect or the Invert buttons.
- 2 Select the Change All button.
The Select a Directory window displays.
- 3 Select the directory you want to use to store the map information and Select OK to save the changes.

You return to the Map Directories tab. All text boxes with a Change status now contain the new directory path; text boxes that have a Protect status retain the original directory path.

Defining map DSN settings

Use the Map DSN tab when the trading partner and/or the log information are stored in databases that are accessed through ODBC. On this tab, enter the information that tells the program how to access the databases.

Note If you also use EC Gateway Server, you must use a different DSN for the trading partner and the log, even if they both point to the same database. Otherwise, you may experience problems with some of your Gateway reports.

❖ Defining DSN settings

Perform these steps for both the trading partner DSN and the log DSN

- 1 Select the Data Source Name text box up arrow
The Data Source Name window displays.

- 2 Double-click on a DSN to select it.

The DSN is entered in the Data Source Name text box and the driver connect string is automatically created.

If you have not created the DSN, select Configure Data Source to open the the ODBC configuration window. Once you have created your DSNs, proceed with the procedure.

Note For certain databases, such as MS SQL Server and Oracle, you may need to manually edit the string in the text box to add a user ID and password.

- 3 Select Test Trade Partner Connection or Test Log File Connection to test the connections.

If you select Clear TP Fields, the program removes all of the information entered in the text boxes on the Trade Partner section of this tab. If you select Clear Log Fields, the program removes all of the information entered in the text boxes on the Log section of this tab.

- 4 Select OK. You return to the Project/Map window.

Working with maps

Once you have created a map, you can modify, delete, and copy maps with EMap. In order to perform any of those tasks you must first make a map the active map. You do this by selecting it

Selecting a map

There are three ways to select a map from the Map window.

- Highlight a map and choose File>Select from menu.
- Right-click a map and choose Select from the context menu.
- Double-click a map.

For all three methods, the map is selected and you return to the main EMap window. The project name, map name, and — if the map is a transaction map — transaction, direction, and version display on the status bar.

Modifying a map

Maps are modified on the Map Definition Properties window. You can access the Map Definition Properties window in several ways:

- Choose Map from the File menu on the main EMap window and then choose New, Select, or Properties from the Map submenu.
- Highlight a map in the Map window, and choose Properties from the Edit menu.
- Right-click a map on the Map window, and choose Properties from the drop-down list. The Map Definition Properties window displays.

You can modify all the information on the three tabs except the project name.

Deleting a map

There are two ways to delete a map:

- Select the map and choose Edit>Delete from the menu.
- Right-click the map and choose Delete from the context menu.

Copying a map

EMap gives you the capability to copy maps between projects and to and from directories

❖ Copying a map

- 1 Open the map window.
- 2 Select Options>Map Copy from the Map window menu.

The Copy Map window displays the name of the current map and project. If you want to copy a different map:

- Select the up arrow next to the Project Name choose another map from the Select Project and Map window
- 3 Choose the Trade Partner Option you want. The options are:
 - Include Trade Partner Tables – copy the trading partner database to the specified directory.
 - Exclude Trade Partner Tables – do not copy the trading partner database to the specified directory.
 - 4 Select a map direction. There are four options:

- This Map to Map – copy this map to a new map in the current project or to new map in a new project.
- This Map from Map – copy this map from a map.
- This Map to Directory – copy this map to a new directory location.
- This Map from Directory – copy this map from a map in a directory.

Depending on which option you choose, appropriate text boxes appear in which you must type the names of the project, map, or directory. This Map to Map is the default selection and is checked when the Copy Map window opens.

This Map to Map option

❖ Copying this Map to Map

- 1 Type the names of the project and map to which you want to copy the map or select the up arrow next to the Map Name text box to search for the names.

The Select Project and Map window displays.

- 2 Double-click the project and map

You return to the Copy Map window and the program enters the project and map names in the text boxes in the Copy Map To section.

This Map from Map option

❖ Copying this Map from Map

- 1 Type the names of the project and map from which you want to copy the map or select the up arrow next to the Map Name text box to search for the names.

The Select Project and Map window displays.

- 2 Double-click the project and map

You return to the Copy Map window and the program enters the project and map names in the text boxes in the Copy Map From section.

This Map to Directory option

❖ Copying This Map to Directory

The map you specify in the text boxes at the top of the window is copied to a directory which you specify in the text boxes on the top right of the window.

- 1 Under Copy Map To, type the name of the directory to which you want to copy the map or Browse for it. If you select Browse, the Select a Directory ... window displays.
- 2 Highlight a directory and select OK. You return to the Copy Map window, and the program automatically enters the directory name in the text box in the Copy Map To section. The new map has the same name as the map entered in the Map Name text box at the top of the window.

This Map From Directory option

❖ Copying this Map from Directory

The map you specify in the text boxes at the top of the window is copied from a directory which you specify in the text boxes on the top right of the window.

- 1 Under Copy Map From, type the name of the directory from which you want to copy the map or Browse for it. The Select a Directory ... window displays.
- 2 Double-click the directory, and you return to the Copy Map window. The program automatically enters the directory name in the text box in the Copy Map From section.
- 3 The Create This Map Definition from Directory Copy button is available when you select the option to copy This Map from Directory. This button allows you to define a new map as it is being copied. EMap automatically creates the definition of the map created by the copy process. This is particularly useful for sharing maps with other developers.

Note When using this button, make sure that the map definition files are all stored together in a single default directory by selecting Change All on the Map Directories tab of the New Map Definition window.

For each copy option, the program displays a Confirm dialog box that restates the details of the copy map option you have chosen. The program asks for your confirmation before the map is actually copied.

Generating multiple maps in a project

ECMap includes a feature that allows you to generate all the maps in a specific project or all the maps in all projects, rather than generating maps one at a time. The ability to generate multiple maps simultaneously can save you a great deal of time since you can normally generate only one map at a time from the Build menu on the main ECMap window. This capability is particularly useful when you are upgrading maps from one version of the software to another.

❖ Generating multiple maps in a project

- 1 Choose Options>Generate Maps>All Projects from the Map window menu.

The Generate Map window displays.

- 2 Choose All Projects from the Generate Maps submenu.

The Generate Map window displays. No project name appears on the window, but an Options section appears.

- 3 Select Run to generate the maps.

Select View Log and the Generate Log window displays where you can see the results of the map run.

See Chapter 16, “Generating Maps,” for more information about generating maps.

Archiving a map

ECMap gives you the capability to archive maps. Each time you archive a map, the program creates and stores a backup copy of the map. ECMap automatically creates an archive directory in two instances – when a new project is set up and when the Archive command is invoked – unless the archive directory already exists. In each case, ECMap first checks to see whether the archive directory already exists and if not, creates it. Then, when you archive a map, ECMap creates a map directory under the archive directory. Again, ECMap first checks to see whether the map directory already exists. The archived map itself is named using the time at which the archiving occurred, as follows:

Map name = MMDDYYYYHHMM

❖ Archiving a map

- 1 Highlight a map on the Map window, and choose Archive from the Options menu. The system creates a copy of the map and stores it with the following location and naming scheme:

<project directory>\archive\<map name>\MMDDYYYYHHMM

If the file was archived, the system displays a dialog box showing the name of the archive file.

- 2 Select OK to return to the Map window.

If a file by this name already exists, the system asks whether you want to overwrite the file.

- 3 Select Yes to replace the archive file with the new one.

Select No to return to the Map window without creating the archived file.

Because of the naming scheme, which includes the time at which the file was created for the archived files, you receive the overwrite message infrequently.

Automatic mapping of EDI to and from serialized NDO

In the version 2.8.3 of the Sybase Adapter for EDI, EMap can export a schema to the repository for e-Biz Integrator, e-Biz 2000 or MQSI. ECRTTP (the runtime program that executes the maps developed in EMap) is able to Put serialized NDO onto a queue and Read serialized NDO from a queue via calls to the ADK version 3.3. A map must be created manually in EMap, to define the “application” record layout, map the data, and create the flow specifying the Read NDO or Write NDO and Put NDO commands. The NDO is then mapped in e-Biz Integrator, e-Biz 2000 or MQSI to the destination NDO format. In this version, you had to do mapping in two products (for example, EMap and either MQSI, e-Biz Integrator or e-Biz 2000) with two different GUIs.

To resolve this double-mapping issue, ECTMap 2.9 can automatically generate a map to either read or write serialized NDO from/to a queue and export the schema to the repository. In this case, mapping in ECTMap consists of a set of “configuration” activities versus mapping activities. ECTMap now automatically generates the mapping, create the Write NDO and Put NDO or Read NDO commands and exports the schema. To use ECTMap in this manner, the schema must be exported from ECTMap to the repository regardless whether the data is going into or coming out of the Queue.

This mapping solution should work automatically for most transactions. In the cases where the HL (Hierarchical Level) segment is present:

- If the native X12 transaction is used, then the map that automatically created has to be modified manually because the structure of the transaction is not defined in the transaction set.
- If an Implementation Guide (rather than the generic standard) is used to create the standard definition for these transactions, then this solution correctly handles the HL segments 100 percent automatically.

Options are provided for controlling the automatic message mapping utility:

- The default is to put data on the queue one transaction at a time. Only the data between an ST and SE, inclusive, is placed on the queue. A transaction is considered a unit of work.
- There is an option to include the Outer Envelopes with each transaction. Since there can be multiple transactions in an interchange and the largest unit of work the PUT NDO command can handle is a single transaction, the beginning outer envelopes is repeated on every transaction and the trailing outer envelopes does not appear in any serialized NDO data.
- There is an option for breaking out a single transaction into multiple units of work. This option is used in situations where the size of the transaction could exceed the maximum queue size for a transaction.

Note This option slows down performance in ECRTP.

When ECRTP is creating EDI data, each transaction placed on the queue corresponds to a single EDI transaction as defined in the standards. There are no merge or combine functionality to create a single EDI transaction for multiple transactions on the queue. The current functionality in ECRTP group transactions that are destined to the same trading partner in a single interchange providing the data is placed on the queue ordered in this manner.

This design assumes that both the inbound and outbound inbound EDI data are processed through a compliance map to validate that the EDI data complies with the appropriate standard definition.

This section uses X12 terminology but you can automatically map EDIFACT and HL7.

The automatically generated map can be imported into:

- e-Biz Integrator 3.1.1,3.2.2
- MQSI 2, 2.0.1, 2.0.2, 2.1
- e-Biz 2000

The automatically generated map is compatible with HIPAA Toolkit 4.1 or later.

❖ **Automatically creating a map**

1 Install EMap.

2 Run EMap. In EMap(AV) the default first window is in “Adapter View” with a limited toolbar as shown below.

The limited toolbar is all that you need to generate a map automatically. Later, you will learn how to display the “Full View” if you need to make changes to the map.

3 Create a project.

If you do not remember how create a project follow the steps in Chapter 3, “Projects”.

4 Create a map.

Fill out the Map Properties window. The Properties window is filled out for an inbound 997 Map.

The Map definition specifies type of map – X12, EDIFACT and HL7, transaction, version/release, direction (EDI into or out of EMap/ECRTP), Y2K windowing logic and Description. In the Map DSN tab, the database version of the Trading Partner File and Log can be specified (This does not affect the production options). If the direction is IN, the source is EDI and the destination is NDO. If the direction is OUT, the source is NDO and the destination is EDI.

5 Select the Map Directories tab.

If your Standards are in a different directory than the default value, then change this path. Otherwise, no changes need to be made to this window.

- 6 No changes need to be made to the Map DSN windows.
- 7 Select OK to close this window.
- 8 From the View menu, select Transaction Tree.

The EDI segments for the transaction set must be selected. The default view appears with only the mandatory segments selected (checked). To select additional segments, left-click on the segment. In a typical EDI implementation only a subset of the segments defined in the transaction set are needed.

If the standard database was created from an implementation guide then all the segments are checked, since an implementation guide is a specific definition of the transaction subset that is to be used.

In EMap, there are two views for the transaction set, the Table view and the Tree View. Below is a sample of the Tree view.

After the segments are selected, close this window.

- 9 From the Build menu, select the Generate Map option. The Generate Integration Broker Map window appears. The Generate Map option automatically creates the EMap map with the appropriate I/O commands and the schema is stored in the storage.mdb database for the schema export process. When you create an automatic map for an inbound map, you are converting EDI to an Integration Broker Map. When you create an automatic map for an outbound map, you are converting an Integration Broker Map to EDI.

If the Create Full EDI Envelope Structure check box is selected, then the header EDI Outer Envelopes are included in the NDO schema, otherwise only the transaction header appears in the schema. For example:

If Create Full EDI Envelope Structure is checked, then the following appears at the beginning of the schema and data.

ISA

GS

ST

Otherwise only the ST appears at the beginning of the schema and data.

- 10 Select on Run.

During the Generate Process the status is displayed and upon completion, a Generate Complete message appears.

When you select Run, and this is the First generation of the map, a confirmation message appears.

This message describes what the default option is for the placement of the PUT NDO command. The default placement of the PUT NDO command is at the end of every transaction, which for X12 is the SE segment.

- If Yes is selected, then the generate process continues to create the map and saves the schema for export.
 - If No, is selected, then the following window is displayed so that a segment can be selected to place the PUT NDO command. This is so large transactions like, Health Care Claims (837), Consolidated Service Invoice/Statement (811), Advanced Ship Notice (856), etc., can be broken out into smaller units of work so that the size limitation of MQ Series is not exceeded. The following window is displayed to select the segment that when the map either returns to or is about to process data at a lower numbered level.
- 11 To highlight a segment, left-click on the desired segment, and then select the OK button to continue. Only one segment can be selected in this window. If the Cancel button is selected, the generate process is stopped and the user is returned to the EMap Desktop.
 - 12 If this is not the first time you generated a map, a confirmation message appears to overwrite the information or not.
 - If Yes is selected, then the map is regenerated from scratch.
 - If No is selected, then the changes made are incorporated into the new version of the map being generated.

After you select Run, ECRTP begins processing the application data. If you do not receive a message on the window telling that your map has run successfully, you can select View Trace at the bottom of the window to verify the outcome. Usually the Auto-Mapping features generates the map automatically. Otherwise you receive errors. If you receive errors, select the View Trace button and see the Informational Messages chapter in the *EMap Reference Guide* for steps on how to remove the error.

- 13 From the Utilities Menu, select Export Schema and select the formatter you use: e-Biz 2000, e-Biz Integrator, or MQSI.
- 14 Once the repository type is selected, then the user is queried to use an exiting configuration file or create a new configuration file.
 - If Yes is selected, the configuration selection window appears and an existing configuration must be selected.

- If No is selected,:
 - a Create a new configuration file, then select OK for the next window.
 - b Select OK and the Configuration File Options window appears.

If With MQSI or e-Biz Integrator Options is selected and OK is selected, then the Prefix keeps the schemas unique in the repository. Clicking Next populates the configuration file and displays it in the Create Configuration File window so it can be edited.

If the Manually option is selected, then the Create Configuration File window is open for the configuration file to be manually keyed in.
- 15 If at any time you select Cancel, the schema export process is exited and nothing is saved. When Save is selected, a confirmation prompt appears.
 - If you select No, control is returned to the Create Configuration File window. If Yes is selected, the configuration file is automatically saved into the ECMap installation directory, for example: *c:\program files\sybase\ecmap*. For the next step, go to step 15 below.
 - If you select Yes on step 7, then the configuration selection window appears and an existing configuration must be selected.
- 16 Once a configuration file is selected, then the Schema Name Selection window is displayed:

At this point, a schema name is displayed. You can override it by selecting on the Add button or select the Finish button. The Finish button causes the schema to be generated and exported into the repository. A file definition is automatically created so the map can be tested before data is written to the repository.
- 17 After you select Finish, the Add Schema Name dialog appears. Fill out the Schema Name and a description.
- 18 After the information has been added, select OK and you are returned back to the Schema Name Selection window.

EDI ID data elements

As part of the schema creation process, a delimited file is created that contains all of the EDI identifier values for reference when mapping in MQSI, e-Biz Integrator or e-Biz 2000.

If the schema is being created from a standard distributed in ECMap, the file contains two record types and there is only one table defined for each element number regardless of how many times it is used in a transaction:

```
"A", "Element Number"
"C", "Value", "Description"
```

which looks like:

```
"A", "355"
"C", "01", "Actual Pounds"
"C", "02", "Statute Mile"
"C", "03", "Seconds"
"C", "04", "Small Spray"
...
```

If the schema is being created from a standard created from an implementation guide, the file contains two records, and there are many occurrences of the element number based on which segment the element appears in.

Implementation Guides are very specific on which values are valid in specific instances of a segment. A global list provides no added value in this circumstance.

```
"B", "Element Number", "Record Type"
"C", "Value", "Description"
```

which looks like:

```
"B", "355", "MEA_19"
"C", "01", "Actual Pounds"
"C", "50", "Actual Kilograms"
"C", "51", "Actual Tonnes"
...
```


Working with the Address Book

About this chapter

The address book in ECMap contains information about your company and its trading partners that is required for transaction maps, but not for any-to-any maps or Web maps.

Topics

This chapter includes the following topics:

Topic	Page
Overview	48
Company ID	49
Trading partner	53
Setting up trade agreements	67
Mailbox chart	84

Overview

The information in the address book:

- Holds information that you might need in an EDI message, but that is not included in the original application data.
- Contains information that is used to dynamically switch maps at runtime, based on data being sent and received.
- Provides a method for routing data based on EDI envelopes.
- Serves as the audit and control point relative to keeping track of control numbers for the last EDI messages sent and received.
- Provides a cross-reference between the internal application number for a trading partner and the identifiers for that trading partner on the EDI envelope.

The address book provides the sender and receiver information that allows EDI messages to be correctly identified and exchanged. Company data provides the default sender information for outbound maps and the default receiver information for inbound maps. Trading partner data provides the default receiver information for outbound maps and the default sender information for inbound maps. Trade agreements that link trading partner to specific maps can contain additional sender and receiver information that can be used to override the default information – for that specific trading partner/map combination.

This chapter is organized into three sections—company, trading partners, and trade agreements. In EMap, you can add a new company profile, delete a company profile, or modify a company profile. You can add a new trading partner, delete a trading partner, modify a trading partner, copy a trading partner, or copy the trading partner and trade agreement tables for a trading partner. Finally, you can create a new trade agreement linking a trading partner to a specific map, delete a trade agreement, or modify a trade agreement.

Company ID

The information you enter about your company is the electronic signature on your outgoing EDI messages, telling the recipient who sent the message. The company profile is used as the sender address on both the interchange and group level envelopes that hold the EDI message. Because the requirements for exchanging business messages often differ from trading partner to trading partner, ECMAP allows you to set up multiple profiles for your company - allowing you to tailor your message exchange information for each trading partnership. On incoming EDI messages, the company ID identifies your company as the recipient of the message.

Where to begin

You perform all company-related actions on the Company – ID window.

❖ Accessing the Company ID window

- Select File>Address Book>Signature (Company ID) from the main menu. The Company ID window displays.

The following procedures are initiated from the Company ID window.

Adding a new company profile

❖ Adding a new company profile

- 1 Choose File>New.

The Company ID – New window displays.

- 2 Type information to create your company’s electronic address on the envelopes that enclose your EDI messages.

The codes and qualifiers you type are those that you and your trading partner have agreed is used for the interchange and group level addresses for the messages you are exchanging. They are usually explicitly specified in an implementation guide.

- Profile Number – a unique user-assigned code that allows multiple profiles to be created for the same company. Since the requirements of trading partners vary greatly, this feature enables one company to be recognized as different “senders” by different trading partners.
- Name – the name of your company.

- 3 Type the information that identifies your company as the sender of outbound messages on one of the following Outbound Sender Default Envelope Values tabs:
 - X12, EDIFACT, & HL7
 - NCPDP
- 4 Select OK. Exit all open windows to return to main EMap window.

X12, EDIFACT, and HL7 tab

Default outbound sender value mapping

You can enter envelope values for these parameters:

- The Interchange code and qualifier is the default sender code and qualifier on outbound messages.

For outbound X12 messages, the code and qualifier are the Interchange Sender Code/Qualifier on the ISA segment.

For outbound EDIFACT messages, the code is the Interchange Sender Identification on the UNB address.

For outbound HL7 messages, the code is the Sending Facility on the MSH segment, the File Sending Facility on the FHS segment, or the Batch Sending Facility on the BHS segment.

- The Interchange Internal ID and Interchange Internal Sub-ID codes (EDIFACT only).
- The Group code and qualifier is used on EDI group-level inner envelopes as the default sender code on outbound messages.

For outbound X12 messages, this is the Application Sender code on the GS segment.

For outbound EDIFACT messages, this is the Application Sender Identification on the UNG segment.

For outbound HL7, it is the Sending Application on the MSH segment, the File Sending Application on the FHS segment, or the Batch Sending Application on the BHS segment.

- The Authorization code and qualifier (X12 and EDIFACT only) is used for authentication purposes, such as user IDs.

For outbound X12 messages, the code and qualifier are the Authorization Information and Authorization Information Qualifier on the ISA segment.

For outbound EDIFACT messages, the code is the Application Password on the UNG segment.

- The Security code and qualifier is used for additional authentication purposes, such as passwords. T

For outbound X12 messages, the code and qualifier are the Security Information and Security Information Qualifier on the ISA segment.

For outbound EDIFACT messages, they are the Recipient Reference/Password and Recipient Reference/Password Qualifier on the UNB segment.

For outbound HL7 messages, they are the Security on the MSH segment, the File Security on the FHS segment, or the Batch Security on the BHS segment.

Entering outbound sender default values

Interchange, Authorization, and Security each have a Qualifier, Description, and Code. The Qualifier identifies what type of Code is used, the Description is a short explanation for that type of code, and the Code is the actual code. To select these values:

❖ **Selecting outbound sender default values**

- 1 Right-click the Interchange Qualifier text box

The INTERCHANGE SELECTION window displays, listing all the possible interchange qualifiers.

- 2 Choose a qualifier and select OK.

- 3 Right-click the Authorization Qualifier text box

The AUTHORITY SELECTION window displays

- 4 Choose a qualifier and select OK.

- 5 Right-click the Security Qualifier text box.

The SECURITY SELECTION window displays, with two (2) options:

- 00 No Security Information Present
- 01 Present

- 6 Choose a qualifier and select OK.

NCPDP tab

Enter the envelope values for these parameters:

- Batch Sender ID is the identification number of the sender of the file. (Batch only.)
- Bin Number is a number assigned by ANSI or NCPDP.
- Processor Control Number is assigned by the processor and identifies who the claim is for.
- Service Provider ID Qualifier
- Service Provider ID/Pharmacy Number
- Software Vendor/Certification ID

Modifying a company profile

❖ **Modify a company profile**

- 1 Highlight a company profile and choose Edit>Properties.
The Company ID – Properties: window displays.
- 2 Modify the Company ID as needed.

Deleting a company profile

❖ **Delete a company profile**

- Highlight a company profile and choose Edit>Delete.
You are asked to confirm that you want to delete the company profile record before it is permanently removed.

Trading partner

Trading partners are the entities with whom your company exchanges messages. The trading partner files contain demographic information about your trading partners and the data needed to process the electronic messages you exchange – identifying information used in EDI envelopes (sender and receiver codes), routing data (mailbox name), and information required to read the data (delimiter, release, and repeat characters).

The information that you type about your trading partners is used as the electronic contact information for your EDI messages, telling the system who should receive the messages that are being sent or who sent the messages that are being received. On outbound transaction maps, the trading partner information is used to populate the receiver address on both the interchange and group level envelopes that hold the EDI message. On inbound transaction maps, the trading partner information identifies who sent the message and determines which map runs.

ALL trading partner

ECMap provides a default trading partner called the ALL Trade Partner that is helpful in the following ways:

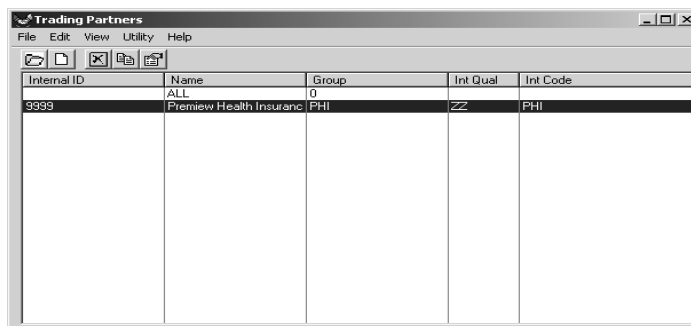
- If you are running a map and ECMap does not find the trading partner in the trading partner database or does not find a trade agreement linking the current map and trading partner, ECMap can run the map using the ALL Trade Partner default. Without this useful feature, the map run would be cancelled.
- ECMap automatically defaults to the All Trade Partner to map all segments that do not have trading-partner-specific mapping. The ALL Trade Partner cannot be removed or modified, nor can the ID, Name, or Group reserved for the ALL Trade Partner be used for any other trading partner. (The ID is thirteen spaces, the Name is ALL, and the Group is 0 (zero). In the trading partner database, the CUST_NO for the All Trade Partner is 35 spaces, and the GSID is 0 followed by 34 trailing spaces. No other trading partner is allowed to have these values.)

Where to begin

You perform all trading-partner-related actions on the Trading Partners window.

❖ **Accessing the Trading Partner window**

- Select File>Address Book>Contacts (Trading Partner) from the main menu.



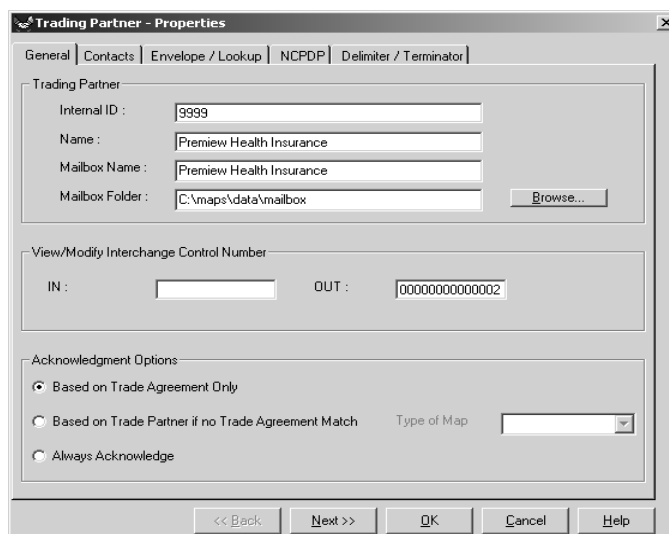
The Trading Partners window displays.

The following procedures are initiated from the Trading Partners window.

Adding a new trading partner

❖ **Add a new trading partner**

- Choose File>New.



The Trading Partner – New window displays. The window has five tabs

- General
- Contacts
- Envelope/Lookup
- NCPDP
- Delimiter/Terminator

Use the Back and Next buttons at the bottom of the window to move from tab to tab.

General tab

Enter the following information on the General tab:

- Internal ID – the internal application number you use for the trading partner. This number links the EDI envelopes to the application information. On outbound maps, this number is used during the trading partner look-up. On inbound maps, this information is recorded in the log file.
- Name – the name you use internally for the trading partner. It is mostly informational and is not used by the program.
- Mailbox Name – used for reports and display.
- Mailbox Folder – the directory, or folder, where the runtime engine (RTP) drops off EDI data produced by outbound maps and compliance runs and places inbound EDI data being passed through. It is attached to the trading partner and is used for both inbound and outbound routing. Either type in the mailbox Folder or select Browse to search for it on the Select a Directory... window.
- You can view or change the Interchange Control Number for both IN(bound) and OUT(bound) messages. The Interchange Control Number is used in both the beginning and ending segments of the interchange-level envelopes.

If you run a compliance map with the Validate Control Numbers option, and envelope and all its contents are rejected if these numbers do not match.

For outbound maps, the control count is automatically incremented by the program each time a message is sent to this trading partner.

For inbound maps, each time a message is received, the current control count is compared with the last control count to confirm that no messages have been lost.

You normally do not change these numbers, but there are instances when you may need to do so. For example, if a transaction is aborted after the control number is incremented, you might want to reset the control number to account for this.

- Acknowledgement options – Specify how you want to ECTMap to issue an acknowledgement in the event that it is unable to find a trade agreement match:
 - Acknowledge based on Trade Agreement entry – (Default). An acknowledgement will be sent based on what is specified in the trade agreement for this partner. If no trade agreement is found, no acknowledgement will be sent.
 - Acknowledge if no Trade Agreement match – If there is no trade agreement match for this partner, an acknowledgement will be sent for the type of map run specified in the Type of Map Run field.
 - Always acknowledge – Always send an acknowledgement for the type of map run specified in the Type of Map Run field, regardless of what is specified in the trade agreement.

There must be a Trade Partner match

The acknowledgement options are based on what is specified in a particular Trading Partner. If a Trading Partner is not found for the map, no acknowledgement will be sent, no matter what is specified in the Acknowledgement options.

- Type of Map Run – Select the type of map run you want the Acknowledgement options to be applicable to:
 - Compliance – Apply the Acknowledgement options to compliance map runs.
 - Inbound – Apply the Acknowledgement options to both inbound map runs.
 - Both – Apply the Acknowledgement options to both inbound and compliance map runs.

Contacts tab

Enter the address and contact information on the Contacts tab (optional).

The following fields hold comments and/or data that may be required by a trading partner (such as a customer number or vendor number specific to the trading partner), but that is not maintained in the application system or used in the map. This information is stored in system variables so that it can be accessed within the map and used in mapping if there is a need to do so.

- The Address information consists of all the traditional parts of a company address:
 - Street
 - City
 - State
 - Postal Code
 - Country
- The Personnel information consists of the Name and Phone number of two individuals who work for the trading partner and are involved with the project.
- The Ship and Bill section contains electronic addressing information about the trading partner's:
 - Ship To address
 - Bill To address

For both the shipping and billing address, you can enter a qualifier and the appropriate code designated by the qualifier. You can select from the same codes used in the Interchange section on the General tab of this window. It does not matter what you use as the code – what is important is that the code be unique. For this reason, a Dun and Bradstreet number or a telephone number with the area code is often used.

Envelope/Lookup tab

The Envelope/Lookup tab of the Trading Partner – New window contains identifying information for the trading partner that is used when creating outbound envelopes and can be used when selecting the map that runs during inbound processing. On this tab, you must enter the default receiver information for outbound envelopes, but you can also enter information that overrides the outbound sender information that you entered on the Company ID window. The sender and receiver information that you enter on this tab can also be used as part of the trading partner lookup that selects the map to run on inbound processing. If you use EC Gateway Server, ECTMap's powerful message routing software, this override feature is very useful for tailoring the Gateway to your specific requirements.

In the top section of the window, you enter Outbound Receiver Default Envelope Values and Inbound Sender Lookup Values. (The information that you enter is just like the information that you entered on the Company ID window.) The information that you enter here is used on the interchange and group envelopes to identify your trading partner as the receiver of outbound messages, can be used as part of the trading partner lookup during inbound processing, and includes information used for authorization and security.

When the program creates the outbound envelopes for your messages, it uses some or all of this information (depending on the EDI standard being used) on the interchange and group envelopes. You enter qualifiers and codes, and the program automatically enters the descriptions associated with the qualifiers you select. The qualifier defines the type of code that is used as an identifier. Qualifiers can be values such as telephone numbers or Dun & Bradstreet numbers. Possible qualifiers are found in the code lists associated with each standard. There is a more detailed explanation of qualifiers on the next page, following the instructions for entering information in the text boxes on this tab.

General Contacts Envelope / Lookup NCPDP Delimiter / Terminator			
Outbound Receiver Default Envelope Values and Inbound Sender Lookup Values			
Qualifier	Description	Code	
Interchange :	ZZ	Mutually Defined	PHI
Interchange Internal ID (EDIFACT Only) :			
Interchange Internal Sub-ID (EDIFACT Only) :			
Group :			PHI
Authorization :	00	No Authorization Present	
Security :	00	No Security Information Present	
Outbound Sender Override Envelope Values and Inbound Receiver Lookup Values			
Qualifier	Description	Code	
Interchange :			
Interchange Internal ID (EDIFACT Only) :			
Interchange Internal Sub-ID (EDIFACT Only) :			
Group :			

- The Interchange code and qualifier are used on EDI interchange-level outer envelopes – as the default receiver code and qualifier on outbound messages. (They can be overridden by entries in the Outbound Receiver Override Envelope Values > Interchange > Qualifier/Code text boxes on the Overrides tab of the Trade Agreement window.) For outbound X12 messages, the code and qualifier are the Interchange Receiver Qualifier/Code on the ISA segment. For outbound EDIFACT messages, the code is the Interchange Receiver Identification on the UNB segment. For outbound HL7 messages, the code is the Receiving Facility on the MSH segment, the File Receiving Facility on the FHS segment, or the Batch Receiving Facility on the BHS segment.
- The Interchange Internal ID and Interchange Internal Sub-ID codes are used only on the UNB segment of EDIFACT messages.

- The Group code and qualifier are used on EDI group-level inner envelopes – as the default receiver code on outbound messages. (They can be overridden by entries in the Outbound Receiver Override Envelope Values > Group > Qualifier/Code text boxes on the Overrides tab of the Trade Agreement window.) For outbound X12 messages, this is the Application Receiver code on the GS segment. For outbound EDIFACT messages, this is the Application Receiver Identification on the UNG segment. For outbound HL7, it is the Receiving Application on the MSH segment, the File Receiving Application on the FHS segment, or the Batch Receiving Application on the BHS segment.
- The Authorization code and qualifier are used for authentication purposes, such as user IDs. (They can be overridden by entries in the Outbound Receiver Override Envelope Values > Authorization > Qualifier/Code text boxes on the Overrides tab of the Trade Agreement window.) For outbound X12 messages, the code and qualifier are the Authorization Information and Authorization Information Qualifier on the ISA segment. For outbound EDIFACT messages, the code is the Application Password on the UNG segment. These values are not used on HL7 messages.
- The Security code and qualifier are used for additional authentication purposes, such as passwords. (They can be overridden by entries in the Outbound Receiver Override Envelope Values > Security > Qualifier/Code text boxes on the Overrides tab of the Trade Agreement window.) For outbound X12 messages, the code and qualifier are the Security Information and Security Information Qualifier on the ISA segment. For outbound EDIFACT messages, they are the Recipient Reference/Password and Recipient Reference/Password Qualifier on the UNB segment. For outbound HL7 messages, they are the Security on the MSH segment, the File Security on the FHS segment, or the Batch Security on the BHS segment.

Interchange, Authorization, and Security each have a Qualifier, Description, and Code. The Qualifier identifies what type of Code is used, the Description is a short textual explanation for that type of code, and the Code is the actual code. In X12, for example, a qualifier of 01 is described as D-U-N-S Number, Dun & Bradstreet. The code is the actual Dun & Bradstreet number. If this were chosen as the type of interchange code, the company's Dun & Bradstreet number would be entered in the Code field. Each set of standards that uses qualifiers has a list of available qualifiers and their associated descriptions.

- 1 Right-click the Interchange Qualifier text box, and the INTERCHANGE SELECTION window displays, listing all the possible interchange qualifiers.

- 2 Right-click the Authorization Qualifier text box, and an AUTHORITY SELECTION window displays, listing all the possible authorization qualifiers.
- 3 Right-click the Security Qualifier text box, and a SECURITY SELECTION window displays, with two (2) options:
 - 00 No Security Information Present
 - 01 Present.

You can scroll down the list on the INTERCHANGE SELECTION (or AUTHORITY SELECTION) window and select the qualifier you want to use by double-clicking it or by highlighting the selection and selecting OK.

NCPDP tab

Select the NCPDP tab to enter envelope information.

The NCPDP tab provides default receiver/sender IDs and override values for outbound NCPDP maps. Display the NCPDP tab of the Trading Partner window.

You can enter envelope override values for these parameters:

- Batch Receiver ID – identifies the receiver of the file or the inbound sender of the file.
- Bin Number – number assigned by ANSI or NCPDP.
- Process Control Number – assigned by the processor and identifies who the claim is for.
- Service Provider ID Qualifier
- Service Provider ID/Pharmacy Number
- Software Vendor/Certification ID

Delimiter/Terminator tab

Select Delimiter/Terminator tab to set file delimiter characters

Sometimes receivers dictate the delimiters. The Delimiter/Terminator tab of the Trading Partners – New window allows you to enter information to override the default delimiters, release characters, and repeat characters normally used in X12 and EDIFACT—for specific maps for specific trading partners. For X12 maps, the program uses the default transaction delimiters defined in X12 version 0200. For EDIFACT maps, the program uses the standard EDIFACT message delimiters. For HL7 maps, you must enter delimiter values on this window because there are no default values for HL7 message delimiters.

The delimiters and release characters are stored in the trading partner database with their 3-digit ASCII equivalents. These overrides apply only to outbound maps. On inbound maps, the program knows where to look in the incoming data to find the characters that are being used for delimiters, repeat characters, and release characters. You also enter information on this tab indicating whether a period or comma is used as the decimal indicator for EDIFACT messages.

- Delimiters are special characters that are used to separate segments, elements, subelements, fields, components, and subcomponents. When the program is reading EDI data and encounters a delimiter, it knows that the current segment/element/etc. has ended and a new one has begun.
- A release or escape character is a special character that is placed before a delimiter to signal to the program that it should not treat the following character as a delimiter. When the program is reading EDI data and encounters a release/escape character, it simply reads the following character rather than using it as a delimiter.
- A repeat or repetition character is a special character used between repeating elements. Certain EDI standards allow multiple occurrences of an element and its subelements (or components and subcomponents)—within the definition of a single element/component. When EMap encounters repeating elements/components, the program recognizes a single element/component because the repeating elements/components are bounded on either end with element/component delimiters. However, the repeat characters that separate the elements/components within the element/component signal to the program that these are the next instance of the repeating elements/components—within the single delimited element/component.

If you right-click in any of the text boxes on the window, EMap displays a drop-down list of available delimiter overrides. You must choose the delimiter from this list.

For X12, you can override delimiters for:

- Segment
- Element
- Subelement
- Repeat Character

Note The ability to have repeating elements was added to X12 in version 4020. Prior to that version, there were no repeating elements in X12 data.

For UN/EDIFACT, you can override delimiters for:

- Segment
- Element
- Subelement
- Release Character
- Repeat Character
- Decimal Indicator

For HL7, you can override delimiters for:

- Segment

Note The HL7 segment terminator is defined as <CR> and technically cannot be changed. However, since some trading partners mandate that it be different, ECMap allows you to override this value.

- Field
- Component
- Subcomponent
- Escape Character
- Repetition Character

Note All NCPDP delimiters are fixed and cannot be altered. The Batch has a fixed-length header and trailer. The batch detail record is a variable length.

At the bottom of the tab, there is a Data Encoding section. From the Packed Decimal Character drop-down list, select the character that indicates that a signed packed decimal is positive. The legal values include B, C, and D. When you specify that a field is a “Packed Decimal” field type on the New Field or Field Properties window, you enter information that tells the program whether or not the field is signed, as well as the number of non-decimal and decimal numbers the field contains. When the program encounters a “Packed Decimal” field that is signed, it looks at the value selected from this drop-down list to determine whether the number in the field is positive.

Working with trading partners

Once you have set up a trading partner, you can modify, copy, or delete it as needed.

Modifying a trading partner

❖ Modifying a trading partner

- 1 Highlight the trading partner and choose Properties from the Edit menu on the Trading Partner window. The Trading Partner – Properties window displays.
- 2 On the Trading Partner – Properties window, you can modify all of the information that you entered on all the tabs of the Trading Partner window. Use the Back and Next buttons at the bottom of the window to move from tab to tab.

Deleting a trading partner

❖ Deleting a trading partner

- Highlight the trading partner that you want to remove and choose Delete from the Edit menu on the Trading Partner window.

You are asked to confirm that you want to delete the trading partner record before it is permanently removed. You are not allowed to delete the ALL Trade Partner.

Copying a trading partner

❖ Copying a trading partner

- 1 Highlight the trading partner and choose Copy from the Edit menu on the Trading Partner window. The Trading Partner – Copy window displays.

The Trading Partner – Copy window has the same four tabs as the Trading Partner – New window. A few text boxes on the tabs are blank, but the other text boxes contain the information for the trading partner being copied. Use the Back and Next buttons at the bottom of the window to move from tab to tab.

- 2 Enter the following information for the new trading partner (to be created by the copy process):

On the General tab:

- Internal ID – the internal application number you use for the trading partner
- Name – the internal name you use for the trading partner
- Copy this Trading Partner’s data when ‘Copy Tables’ utility is used check box

On the Envelope/Lookup tab:

- Group in the Outbound Receiver Default Envelope Values and Inbound Sender Lookup Values section – the part of the trading partner’s electronic address that is used on the group-level EDI inner envelope

You may keep or change the remaining information that was copied to the other text boxes and tabs.

Copying the trading partner database tables

ECMap allows you to copy all or selected tables in the trading partner database. (The trading partner database includes the company, trading partner, and trade agreement tables.)

There must be an ODBC connection to the trading partner databases in order to use this copy function. If you are using Dbase or Paradox tables, you may not have a DSN that points to the database and must create one. Since both DBase and Paradox are ODBC-compliant, you can select the ODBC Configure ... button and use Microsoft Access to set up a DSN.

❖ **Copying one or more tables**

- 1 Highlight a Trade Partner and choose Utilities>Copy Tables from Trading Partners window.

The Copy Trading Partners Tables window displays. The DSN for the current Trade Partner ODBC connection is the source for the table copy.

- 2 Enter the DSN connect string for the target database of the trading partner copy,

or

Click Browse and select a data source from the ODBC Data Source Name window.

- 3 Select one of the following Trade Agreement copy options:
 - All - Copy all Trade Agreements – Copy all trade agreements in addition to trading partner tables
 - None - Copy no Trade Agreements – Copy trading partner tables and no trade agreement tables
 - Select - Copy selected Trade Agreements – Select the trade agreements you want to include with the trading partner tables

- 4 Click OK.

If you choose All or None, a prompt displays indicating that the copy is complete.

If you choose Select, the Trade Agreement window displays. Select the Trade Agreements you want to include in the copy and click OK.

Setting up trade agreements

A trade agreement associates a specific map with a specific trading partner and has the capability of overriding certain addressing information for this particular combination. Trade agreements provide power and flexibility by allowing you to associate multiple maps with one trading partner. They also add additional routing capabilities so that documents can be sent to the correct mailbox within your and the trading partner's organization - since the routing information associated with a trade agreement overrides routing information associated with a map, including a mailbox. For example, you can override the receiver interchange or group code for a specific map when it is used to translate data for a specific trading partner. Or, you can override the mailbox name and folder where the program stores the trading partner data for that specific exchange of business information.

The information requested on the tabs of the trade agreement window, as well as which standard-specific tab is active, depends on the standard that is used in the map.

Creating a new trade agreement

❖ Creating a trade agreement

- 1 Select Company on the main ECTMap window or choose Address Book from the File menu and then choose Contacts (Trading Partner) from the Address Book submenu.

The Trading Partners window displays.

Tran	ST03 (X:12)	Purpos	Stat	Version	Map	Map Name Ext	GS Control	ISA
837		OUT	T	004010	D_837_E		000000000	0040
837		OUT	P	004010	D_837_DB	V1	000000000	

- 2 On the Trading Partners window, highlight an entry and choose Trade Agreement from the File menu. The Trade Agreement with Trade Partner <trading partner> window displays.

All the maps for the trading partner are listed on the Trade Agreement with Trade Partner <trading partner> window. You can create a trade agreement in two ways. The easiest way is to

- 3 Select the map you want to link to the trading partner and choose Add Map from the File menu. A trade agreement is automatically created linking the trading partner and the map.

or

Choose New from the File menu. With this method, you must manually enter the information for the trade agreement.

The Trade Agreements window displays. It has six tabs:

Trade Agreement - New

General | Overrides | X12 | EDIFACT | HL7

Standard Used for Map :

Trade Agreement

Trading Partner Internal ID :

Trading Partner Name :

Group Code :

Map Information

Map Name : Map Name Extension :

Version/Release/Industry Identifier Code : Map Type :

Transaction Set Identifier Code : Test Indicator :

<< Back Next >> OK Cancel Help

- General
- Overrides
- X12
- EDIFACT
- NCDCP
- HL7.

How you fill out these tabs depends on the standard used for the trade agreement. Each tab is described below.

General Tab

The General tab of the Trade Agreements window contains information about the trading partner and the map that are linked by this trade agreement.

There are six choices for the Standard Used for Map:

- Any-to-Any
- X12
- EDIFACT – Syntax 3
- EDIFACT – Syntax 4
- HL7
- NCPDP

The Trade Agreement information in the top section of the window is populated with the information you entered on the Trading Partner window – Internal ID, Name, and Group Code.

- Trading Partner Internal ID is the internal customer number you use for this trading partner. (Internal ID entered on the General tab of the Trading Partner window)
- Trading Partner Name is the internal name you use for this trading partner. (Name entered on the General tab of the Trading Partner window)
- Group Code is the identifier code for the trading partner used on the inner group-level EDI envelope. (Group > Code entered on the Envelope/Lookup tab of the Trading Partner window)
- The information in the Map Information section on the lower half of the window depends on the EDI standard being used.

General tab for X12

When X12 is the EDI standard used in the map, the General tab includes the following information:

- The Map Name is the name you gave to the map when you set it up. If you want to choose a different map, you can right-click in the Map Name text box to display a drop-down list of all the maps defined in the current project. You can select a map by double-clicking on it, and the program automatically populates the text boxes on the bottom half of the window with the information associated with the map you selected.

- The Version/Release/Industry Identifier Code is the version of X12 used in the map linked with this trade agreement.
- The Transaction Set Identifier Code is the identifier for the X12 transaction set used in the map.
- Map Name Extension is a unique name that is used to distinguish this map from any other maps using the same transaction.
- The Map Type indicates either the direction of the map (IN or OUT) or whether it is a print (PRT) or compliance (CMP) map. You specify the same map type on the Run Map window when you are ready to run your map.
- Test Indicator indicates whether this is test (T), production (P), or informational (I). For X12 maps that use version 4020 and later, the choices for Status include T, P, and I. For X12 maps prior to version 4020, the choices include only T and P.
- The Implementation Convention Reference (ST03) is an optional element that was added in version 4030. When a value is entered here, the RTP uses ST03 as part of the trading partner lookup for X12 maps. The default is not to use ST03.

General tab for
EDIFACT

General tab – Trade Agreements window – EDIFACT (Syntax 3 and 4) standard. When EDIFACT is the EDI standard used in the map, the General tab of the Trade Agreement window includes the following information:

- The Map Name is the name you gave to the map when you set it up. If you want to choose a different map, you can right-click in the Map Name text box to display a drop-down list of all the maps defined in the current project. You can select a map by double-clicking on it, and the program automatically populates the text boxes on the bottom half of the window with the information associated with the map you selected.
- The Message Version/Release Number is the version of EDIFACT used in the map linked with this trade agreement.
- The Message Type is the identifier for the EDIFACT message used in the map:
- The Map Type indicates either the direction of the map (IN or OUT) or whether it is a print (PRT) or compliance (CMP) map. You specify the same map type on the Run Map window when you are ready to run your map.

- Test Indicator indicates whether this is used for test or production. Valid choices include 0 – 9 and N. N indicates a production run, and 0 – 9 indicate varying degrees of test runs.
- Map Number

General tab for HL7

When HL7 is the EDI standard used in the map, the General tab includes the following information:

- The Map Name is the name you gave to the map when you set it up. If you want to choose a different map, you can right-click in the Map Name text box to display a drop-down list of all the maps defined in the current project. You can select a map by double-clicking on it, and the program automatically populates the text boxes on the bottom half of the window with the information associated with the map you selected.
- The Message Version/Release Number is the version of HL7 used in the map linked with this trade agreement
- Map Name Extension is a unique name that is used to distinguish this map from any other maps using the same transaction.
- The Message Type is the identifier for the EDIFACT message used in the map.
- The Map Type indicates either the direction of the map (IN or OUT) or whether it is a print (PRT) or compliance (CMP) map. You specify the same map type on the Run Map window when you are ready to run your map.
- Test Indicator indicates whether this is used for test or production. Valid choices include P (production), T (test), and D (debug).

General Tab for NCPDP

For NCPDP, you can enter values for these parameters:

- Standard Used for Map identifies the map standard you are using. Choose NCPDP from the drop-down list.
The X12, EDIFACT, and HL7 tabs are now inactive.
- Trading Partner Internal ID is the internal customer number you use for this trading partner. It is entered by EMap.
- Trading Partner Name is the internal name you use for this trading partner. It is entered by EMap.
- Group code is entered by EMap.
- Map Name is the name you gave to the map when you created it.

- Version/Release/Industry Identifier Code identifies the NCPDP version supported for this map. Choose 51 (for version 5.1) from the drop-down list.
- Transaction Set Identifier Code is the identifier for the NCPDP transaction set used in the map.
- NCPDP Map Type identifies the map as telecommunications (interactive) or batch. Choose Telecom v5.1 for interactive or Batchv1.1, v5.1 Telecom for batch.
- Map Name Extension is a unique name that is used to distinguish this map from any other maps using the same transaction.
- Map Type specifies whether the direction of the map is IN or OUT.
Print (PRT) and compliance (CMP) map types are not supported for NCPDP.
- Test Indicator specifies whether this trade agreement is test (T) or production (P).

General Tab for Any-to-Any

For and Any-to-Any map, you can enter values for these parameters

- Standard Used for Map identifies the map standard you are using. Choose NCPDP from the drop-down list.
The X12, EDIFACT, and HL7 tabs are now inactive.
- Trading Partner Internal ID is the internal customer number you use for this trading partner. It is entered by EMap.
- Trading Partner Name is the internal name you use for this trading partner. It is entered by EMap.
- Group code is entered by EMap.
- Map Name is the name you gave to the map when you created it.
- Map Name Extension is a unique name that is used to distinguish this map from any other maps using the same transaction.

Overrides Tab

The Overrides tab of the Trade Agreements window differs for inbound and outbound maps. For inbound maps, the window contains mailbox override information. For outbound maps, it contains both mailbox override information and override information for the receiver on outbound envelopes. For outbound maps, the Overrides tab is the same for X12 and HL7, but slightly different for EDIFACT.

Mailboxes are used for routing. They allow one pass of mapping to create multiple files. The ability to send data to multiple mailboxes eliminates the need for multiple passes over the data. Mailboxes allow you to send different messages for the same trading partner to different mailboxes. This is a capability that is often required by trading partners who have different functions in multiple locations. It is also used to route (pass through) inbound messages to different internal mailboxes. For example, you might use this feature to keep test maps or compliance maps separate from production maps.

Mailbox information entered on the Trade Agreement window overrides information entered on the Trading Partner window. (You can also override the use of this mailbox or the trading partner mailbox at runtime by selecting the Ignore Trading Partner Mailbox or Ignore Trade Agreement Mailbox check boxes on the Option 1 tab of the Run Inbound Map window.)

Outbound receiver information entered on the Trade Agreement window overrides information entered on the Trading Partner window. This information is used to populate the outbound EDI envelopes.

Overrides tab – Trade Agreements window – inbound maps. The Overrides tab of the Trade Agreements window for inbound maps contains the following mailbox override information:

- The Mailbox Name is used only for reports and display. (An entry in this text box overrides the value you entered on the General tab of the Trading Partner window.)
- The Mailbox Folder is the directory, or folder, where the runtime engine (RTP) drops off EDI data produced by outbound maps and compliance runs and places inbound EDI data being passed through. It is used for both inbound and outbound routing. Either type in the mailbox Folder or select Browse to search for it. (An entry in this text box overrides the value you entered on the General tab of the Trading Partner window.)
- In the File Name text box, you can specify the name of an X12, EDIFACT, or HL7 output file for routed data. This file name is used in the IN, OUT, GOOD, BAD, and OTHER mailboxes.

- The Map and route inbound EDI to Trade Agreement mailbox check box allows both the pass-through (routing) of data and the running of maps if it is selected. This must be checked to generate bad EDI and place it in the trading partner's BAD folder. This is always used when running compliance maps.

Overrides tab – Trade Agreements window – outbound X12 and HL7 maps. The Overrides tab of the Trade Agreements window for outbound X12 and HL7 maps contains the following mailbox override information and override information for the receiver on outbound envelopes:

In the Mailbox Override Values section:

- The Mailbox Name is used only for reports and display. (An entry in this text box overrides the value you entered on the General tab of the Trading Partner window.)
- The Mailbox Folder is the directory, or folder, where the runtime engine drops off EDI data produced by outbound maps and compliance runs and places inbound EDI data being passed through. It is used for both inbound and outbound routing. Either type in the mailbox Folder or select Browse to search for it. (An entry in this text box overrides the value you entered on the General tab of the Trading Partner window.)
- In the File Name text box, you can specify the name of an X12 or HL7 file for routed data. This file name is used in the IN, OUT, GOOD, BAD, and OTHER mailboxes.
- The Map and route inbound EDI to Trade Agreement mailbox check box allows both the pass-through (routing) of data and the running of maps if it is selected. This must be checked to generate bad EDI and place it in the trading partner's BAD folder. This is always used when running compliance maps.

In the Outbound Receiver Override Envelope Values section:

- The Interchange>Qualifier and Code override the values entered in the Interchange> Qualifier and Code text boxes in the Outbound Receiver Default Envelope Values and Inbound Sender Lookup Values section of the General tab of the Trading Partner window. For outbound X12 messages, the code and qualifier are the Interchange Receiver ID Code/Qualifier on the ISA segment. For outbound HL7 messages, it is the Receiving Facility on the MSH segment, the File Receiving Facility on the FHS segment, or the Batch Receiving Facility on the BHS segment.

- The Group>Code overrides the value entered in the Group>Code text box in the Outbound Receiver Default Envelope Values and Inbound Sender Lookup Values section of the General tab of the Trading Partner window. For outbound X12 messages, this is the Application Receiver's Code on the GS segment. For outbound HL7 messages, this is the Receiving Application on the MSH segment, the File Receiving Application on the FHS segment, or the Batch Receiving Application on the BHS segment.

Overrides tab – Trade Agreements window – outbound EDIFACT maps. The Overrides tab of the Trade Agreements window for outbound EDIFACT maps contains the following mailbox override information and override information for the receiver on outbound envelopes:

In the Mailbox Override Values section:

- The Mailbox Name is used only for reports and display. (An entry in this text box overrides the value you entered on the General tab of the Trading Partner window.)
- The Mailbox Folder is the directory, or folder, where the runtime engine (RTP) drops off EDI data produced by outbound maps and compliance runs and places inbound EDI data being passed through. It is used for both inbound and outbound routing. Either type in the mailbox Folder or select Browse to search for it. (An entry in this text box overrides the value you entered on the General tab of the Trading Partner window.)
- In the File Name text box, you can specify the name of an EDIFACT output file for routed data. This file name is used in the IN, OUT, GOOD, BAD, and OTHER mailboxes.
- The Map and route inbound EDI to Trade Agreement mailbox check box allows both the pass-through (routing) of data and the running of maps if it is selected. This must be checked to generate bad EDI and place it in the trading partner's BAD folder. This is always used when running compliance maps.

In the Outbound Receiver Override Envelope Values section:

- The Interchange>Qualifier and Code override the values entered in the Interchange> Qualifier and Code text boxes in the Outbound Receiver Default Envelope Values and Inbound Sender Lookup Values section of the General tab of the Trading Partner window. For outbound EDIFACT messages, these are the Interchange Receiver Identification and Identification Code Qualifier on the UNB segment.

- The Group>Qualifier and Code overrides the value entered in the Group>Qualifier and Code text boxes in the Outbound Receiver Default Envelope Values and Inbound Sender Lookup Values section of the General tab of the Trading Partner window. For outbound EDIFACT messages, these are the Application Recipient Identification and Identification Code Qualifier on the UNG segment.

X12, EDIFACT, and HL7 tabs – Trade Agreement window. The Trade Agreements window has a specific tab for each of the standards. Only one standard-specific tab is active at a time, and it is the tab for the standard used in the map linked to the trade agreement. The standard-specific tabs differ for inbound and outbound maps for each standard. Each tab contains information that is used only in that standard or in the case of EDIFACT, in that Syntax version of the standard.

X12 Tab

The X12 tab of the Trade Agreements window contains information that is used only for X12 transaction sets. The EDIFACT tab of the Trade Agreements window contains information that is used only for EDIFACT messages. The HL7 tab of the Trade Agreements window contains information that is used only for HL7 messages.

X12 tab – Trade Agreements window – inbound maps. The X12 tab for inbound maps allows you to enter the version of the X12 standard that is being used in the trade agreement map and to modify the inbound group control number maintained by the system.

- The Interchange Control Version Number is the version of the X12 standard used in the map. This value is used to populate ISA 12.
- An entry in the Group Control Number text box allows you to override the group count that is automatically incremented each time the transaction is run. Ordinarily, you would not change this number.
- The Expect FA 997 Functional Acknowledgement is used to indicate whether a functional acknowledgement should be returned to the sender of this transaction.

X12 tab – Trade Agreements window – outbound maps. The X12 tab for outbound maps allows you to enter the version of the X12 standard that is being used in the trade agreement map and to modify the group control number used on the outbound group-level X12 envelope. It also allows the user to specify whether a TA1 interchange acknowledgement should be requested or whether a FA 997 functional acknowledgement is expected – and the time within which the 997 should be received.

A TA1 interchange acknowledgement does not guarantee anything about the correctness or completeness of the contents of the ISA envelope – it merely acknowledges that it was received.

A FA 997 functional acknowledgement actually performs a syntactical check. The program captures certain information in the log during mapping that is used in the creation of a functional acknowledgement. When the user indicates on this tab that a 997 is expected for an outbound X12 document, it causes the program to capture certain information in the log during mapping that enables it to process the functional acknowledgement.

The Envelope Information section contains the following information:

- The Interchange Control Version Number is the version of the X12 standard used in the map. This value is used to populate ISA 12.
- An entry in the Group Control Number text box allows you to override the group count that is automatically incremented each time the transaction is run. Ordinarily, you would not change this number.

The Notification section contains the following information:

- The Request TA1 Interchange Acknowledgement (outbound maps only) is used to indicate whether there is an agreement between the sender and receiver that the receiver of this EDI message sends a response back to the sender affirming that the ISA envelope was received. If so, the system should expect to receive this acknowledgement.
- The Expect FA 997 Functional Acknowledgement is used to indicate whether a functional acknowledgement is expected back from the receiver of this transaction set. If so, it causes certain information to be written to the log during mapping that enables it to process the functional acknowledgement.

- In the two text boxes following Consider unacknowledged if not received after, the user enters values that tell the program the period of time in which you expect to receive a functional acknowledgement. If an acknowledgement is received after this time period, it is considered overdue. (If you use EC Gateway, this allows Gateway to report on overdue acknowledgements.)
 - Time Units specifies the unit of measure (DAYS, HOURS, MINUTES, or SECONDS) for the time period during which an acknowledgement to your outgoing message must be received.
 - In the text box following Time Units, you select the actual number of units of time after which an acknowledgement is considered overdue. (The drop-down list includes the valid values 1 – 99.)

EDIFACT tab

EDIFACT tab – Trade Agreements window – inbound maps. The EDIFACT tab is the same for inbound and outbound maps, but it includes more information for Syntax 4 than for Syntax 3. On this tab, the user can specify whether a CONTRL message has been requested or is expected, and they enter EDIFACT-specific information.

EDIFACT (Syntax 3) tab

EDIFACT (Syntax 3) tab – Trade Agreements window – inbound and outbound maps. The EDIFACT tab for Syntax 3 contains the CONTRL Message Requested/Expected. It is used to indicate whether a CONTRL message is expected back from the receiver of this message. If a CONTRL message is expected, the program writes certain information to the log during mapping that enables it to process the CONTRL message. You should choose:

- Interchange Level if you and your trading partner have agreed in an Implementation Agreement that the receiver of an EDIFACT message returns a CONTRL message containing a UCI segment indicating whether the interchange-level UNB segment was received and whether it was syntactically acceptable.
- Message Level if you and your trading partner have agreed in an Implementation Agreement that the receiver of an EDIFACT message returns a CONTRL message containing a UCM segment indicating whether a message-level UNH segment was received and whether it was syntactically acceptable.

In the two text boxes following Consider unacknowledged if not received after, the user enters values that tell the program the period of time in which you expect to receive a CONTRL message. If the CONTRL message is received after this time period, it is considered overdue. (If you use EC Gateway, this allows Gateway to report on overdue acknowledgements.)

- Time Units specifies the unit of measure (DAYS, HOURS, MINUTES, or SECONDS) for the time period during which an acknowledgement to your outgoing message must be received.
- In the text box following Time Units, you select the actual number of units of time after which an acknowledgement is considered overdue. (The drop-down list includes the valid values 1 – 99.)

The information entered in the following text boxes is used to populate various things:

- Application Reference – populates the S005 0026 field on the UNB segment. This is the name of the EDI messages that are contained in the UNB envelope. (for example, “PAYMUL”)
- Interchange Agreement ID – populate the S005 0032 field on the UNB segment.
- Association Assigned Code – populates the S008 0057 field on the UNG segment and on the S009 0057 field on the UNH segment.
- Syntax ID – populates the S001 0001 field on the UNB segment.
- Processing Priority Code – populates the S005 0029 field on the UNB segment.
- Application Password – populates the S008 0058 field on the UNG segment.
- Controlling Agency – populates the S004 0051 field on the UNG segment and on the S009 0051 field on the UNH segment.
- Group Reference Number – populates the S005 0038 field on the UNG segment and on the S008 0048 field on the UNE segment.

EDIFACT (Syntax 4) tab

EDIFACT (Syntax 4) tab – Trade Agreements window – inbound and outbound maps. The CONTRL Message Requested/Expected is used to indicate whether a CONTRL message is expected back from the receiver of this message. If a CONTRL message is expected, the program writes certain information to the log during mapping that enables it to process the CONTRL message. You should choose:

- Interchange Level – if you and your trading partner have agreed in an Implementation Agreement that the receiver of an EDIFACT message returns a CONTRL message containing a UCI segment indicating whether the interchange-level UNB segment was received and whether it was syntactically acceptable.
- Message Level – if you and your trading partner have agreed in an Implementation Agreement that the receiver of an EDIFACT message returns a CONTRL message containing a UCM segment indicating whether a message-level UNH segment was received and whether it was syntactically acceptable.

In the two text boxes following Consider unacknowledged if not received after, the user enters values that tell the program the period of time in which you expect to receive a CONTRL message. If the CONTRL message is received after this time period, it is considered overdue. (If you use EC Gateway, this allows Gateway to report on overdue acknowledgements.)

- Time Units specifies the unit of measure (DAYS, HOURS, MINUTES, or SECONDS) for the time period during which an acknowledgement to your outgoing message must be received.
- In the text box following Time Units, you select the actual number of units of time after which an acknowledgement is considered overdue. (The drop-down list includes the valid values 1 – 99.)

The information entered in the following text boxes is used to populate various things:

- Application Reference text box is used to populate the S005 0026 field on the UNB segment. This is the name of the EDI messages that is contained in the UNB envelope. (for example, “PAYMUL”)
- Interchange Agreement ID text box is used to populate the S005 0032 field on the UNB segment.
- Association Assigned Code – populates the S008 0057 field on the UNG segment and on the S009 0057 field on the UNH segment.

- Syntax ID text box – populates the S001 0001 field on the UNB segment.
- Service Code List Directory Version Number – populates the S001 0080 field on the UNB segment.
- Processing Priority Code – populates the S005 0029 field on the UNB segment.
- Application Password – populates the S008 0058 field on the UNG segment.
- Controlling Agency – populates the S004 0051 field on the UNG segment and on the S009 0051 field on the UNH segment.
- Code List Directory Version Number – populates the S009 0110 field on the UNH segment
- Message Type Sub-Function – populates the S009 0113 field on the UNH segment
- Group Reference Number – populates the S005 0038 field on the UNG segment and on the S008 0048 field on the UNE segment.

In the Syntax 4 Items section at the bottom of the window, you must enter four pieces of information – ID, Version, Release, and Controlling Agency – in three categories –Message Subset, Message Implementation Guide, and Scenario:

Under Message Subset, the information entered in the following text boxes is used to populate various things:

- ID – populates the S016 0115 field on the UNH segment.
- Version – populates the S016 0116 field on the UNH segment.
- Release – populates the S016 0118 field on the UNH segment.
- Controlling Agency – populates the S016 0051 field on the UNH segment.

Under Message Imp Guide, the information entered in the following text boxes is used to populate various things:

- ID – populates the S017 0121 field on the UNH segment.
- Version – populates the S017 0122 field on the UNH segment.
- Release – populates the S017 0124 field on the UNH segment.
- Controlling Agency – populates the S017 0051 field on the UNH segment.

Under Scenario, the information entered in the following text boxes is used to populate various things:

- ID – populates the S018 0127 field on the UNH segment.
- Version – populates the S018 0128 field on the UNH segment.
- Release – populates the S018 0130 field on the UNH segment.
- Controlling Agency – populates the S018 0051 field on the UNH segment.

HL7 Tab

HL7 tab – Trade Agreements window – inbound and outbound maps. The HL7 tab for inbound and outbound maps is the same. In the top section of this tab, you specify whether to generate only an MSH message, or whether to also generate the FHS and BHS messages. In the bottom section of the tab, you can specify whether you expect to receive an original acknowledgement or an enhanced acknowledgement, as well as the time period during which the acknowledgment should be received.

The Options section contains the following information:

- For the Envelope Generation Option, there are two choices on the drop-down list:

0072 – FHS, BHS

0073 – MSH Only

The Notification section contains the following information:

- Original Acknowledgement – used to indicate whether there is an agreement between the sender and receiver that the receiver of this EDI message sends a response back to the sender, acknowledging receipt of the message. If so, the system should expect to receive this acknowledgement.
- Enhanced Acknowledgement – used to indicate whether there is an agreement between the sender and receiver that the receiver of this EDI message sends an enhanced response back to the sender, acknowledging receipt of the message. If so, the system should expect to receive this acknowledgement.

In the two text boxes following Consider unacknowledged if not received after, the user enters values that tell the program the period of time in which you expect to receive a functional acknowledgement. If an acknowledgement is received after this time period, it is considered overdue. (If you use EC Gateway, this allows Gateway to report on overdue acknowledgements.)

- Time Units specifies the unit of measure (DAYS, HOURS, MINUTES, or SECONDS) for the time period during which an acknowledgement to your outgoing message must be received.
- In the text box following Time Units, you select the actual number of units of time after which an acknowledgement is considered overdue. (The drop-down list includes the valid values 1 – 99.)

Closing the Trade Agreement Window

To move from tab to tab on the Trade Agreements window, you can either select the tab label at the top of each tab or select Back and Next at the bottom of the window. When you have entered all the required information on all the tabs of the Trade Agreements window, select OK. Exit all open windows to return to main EMap window.

Modifying a trade agreement

❖ Modifying a trade agreement

- Highlight a map on the Trade Agreement with Trade Partner <trading partner> window and choose Properties from the Edit menu. The Trade Agreements - Properties window displays.

The Trade Agreements - Properties window has the same tabs as the Trade Agreements - New window that you used to create a new trade agreement – General, Overrides, X12, EDIFACT, and HL7. You can modify the same information that you entered on that window. You can also use the Back and Next buttons at the bottom of the window to move from tab to tab.

Deleting a trade agreement

❖ Deleting a trade agreement

- Highlight the trade agreement on the Trade Agreement with Trade Partner <trading partner> window and choose Delete from the Edit menu.

You are asked to confirm your decision before the trade agreement is permanently removed.

Mailbox chart

Mailboxes are used for routing. They eliminate the need for multiple passes over the data used for routing because they allow one pass of mapping to send separate data files to multiple mailboxes.

The mailbox is used by the trading partner, both as the sender and the receiver of data. When the trading partner is the sender, the mailbox is the folder where the input data resides before it is run through the map. When the trading partner is the receiver, the mailbox is the folder where the output data created by a map is stored so that it can be picked up by the trading partner.

- GOOD mailboxes – used by Compliance maps.
- BAD mailboxes – used by Compliance maps and Inbound maps.
- IN mailboxes – used both by Inbound maps and by Outbound maps with the Route In switch set.
- OUT mailboxes – used both by Outbound maps and by Inbound maps with the Route Out switch set.
- OTHER mailboxes – used internally for data that moves between IN or GOOD folders into the OTHER folder for processing.

Good, In, and Out mailboxes

If Trade Agreement Mailbox Exists (regardless of whether you “Ignore Trading Partner Mailbox”):

- If Trade Agreement Filename Exists:

Trade Agreement Mailbox\GOOD\Filename	ODBC/NonODBC
Trade Agreement Mailbox\IN\Filename	ODBC/NonODBC
Trade Agreement Mailbox\OUT\Filename	ODBC/NonODBC

- If Trade Agreement Filename Does Not Exist:

Trade Agreement Mailbox\GOOD\goodNNNN.edi or good0000.edi	ODBC/NonODBC
Trade Agreement Mailbox\IN\ NNNN.edi or 0000.edi	ODBC/NonODBC
Trade Agreement Mailbox\OUT\ NNNN.edi or 0000.edi	ODBC/NonODBC

- If you do not ignore Trading Partner Mailbox:

Trading Partner Mailbox\GOOD\goodNNNN.edi or good0000.edi	ODBC/NonODBC
Trading Partner Mailbox\IN\ NNNN.edi or 0000.edi	ODBC/NonODBC
Trading Partner Mailbox\OUT\ NNNN.edi or 0000.edi	ODBC/NonODBC

- If you ignore Trading Partner Mailbox:
No GOOD data is written.
- If you want to substitute a new file for the Good mailbox, use the -ag switch:
-ag full_path_and_filename
or set the Good Mailbox override on the run map screen.

Bad mailboxes

If the ALL TradePartner Mailbox Exists:

- If you do not ignore Trading Partner Mailbox:

All TradePartner Mailbox\BAD\badNNNN.edi or bad0000.edi	ODBC/NonODBC
---	--------------
- If you ignore TradePartner Mailbox:

-dt Trading Partner Directory\BAD\badNNNN.edi or bad0000.edi	NonODBC
Map Directory\BAD\badNNNN.edi	ODBC

If the ALL TradePartner Mailbox Does Not Exist:

- If you do not ignore Trading Partner Mailbox:

-dt Trading Partner Directory\BAD\badNNNN.edi or bad0000.edi	NonODBC
Map Directory\BAD\badNNNN.edi	NonODBC
- If you ignore Trading Partner Mailbox:

-dt Trading Partner Directory\BAD\badNNNN.edi or bad0000.edi	NonODBC
Map Directory\BAD\badNNNN.edi	ODBC

- If you want to substitute a new file for the Bad mailbox, use the -ab switch:
-ab full_path_and_filename
or set the Bad Mailbox override on the run map screen.

Note Where there are two mailbox file name options, ending with NNNN.edi or 0000.edi:

- NNNN.edi is used if this is an ODBC Log run or if this is a nonODBC Log run and there is a Run ID. (WinMap uses the Run ID number as NNNN.)
 - 0000.edi is used if this is not an ODBC Log run and no -id run number switch was passed in.
-

About this chapter

This chapter discusses user administration for EMap.

Topics

This chapter includes the following topics:

Topic	Page
Overview	88
Administrative tasks	89
Deleting a user	89
Changing the password of a user	90

Overview

Each time EMap is opened, a Login dialog box appears. It contains two text boxes: User Name and Password.

Getting started

To use EMap, you must first enter a valid user name and password in these text boxes. If an invalid entry is made, entry into EMap is denied.

The Login window allows the Administrator to control who has access to EMap because only the Administrator can add names and passwords to the list of authorized users. The Administrator uses the User Administration function in EMap to enter the names and passwords of authorized users, delete the names of users who are no longer authorized to use EMap, and change the passwords of users.

The first password that the Administrator should change is the administrative password itself. When EMap is installed, the Administrator is set up as a user with the user name "Admin" and the default password "emap". The Admin user name cannot be deleted, but the default password for the Administrator should be changed as soon as possible. This should be done for security purposes since anyone can log on as the Administrator using the default password until the administrative password is changed.

The first time you use EMap, Admin appears in the User Name text box and the Password text box is blank.

Enter the default password, emap, and select OK. Passwords are case-sensitive, so you must enter emap in all lower-case letters. The main EMap window displays. The Administrator should immediately change the administrative password using either the Change Password function that allows individual users to change their own passwords or the User Administration function available only to the Administrator.

Once the administrative password has been changed, the Administrator can then perform user administration activity, such as adding user names and passwords for authorized EMap users.

Administrative tasks

User administration window

All activity related to user administration is performed on the User Administration window.

❖ Accessing the User Administration window

- Select File>User Administration from the main menu.

The User Administration window appears.

The User Administration window has three tabs

- User – used to create, modify, and delete user accounts
- Login – used to create or modify login information for a user
- Private Directory – used to create a directory for a user which will hold temporary files and configuration information. Private directories are used to store user-specific information and are particularly important when there are multiple users of ECMap.

❖ Adding a new user

- 1 Select New on the Users tab

Login tab becomes the active tab.

- 2 Enter the name and password for user you are adding and select OK.

An alert box displays the message: `Private Directory Required`.

Note This password should be considered temporary. Once the user logs in to ECMap they should change the password to assure the security of their login information.

- 3 Select OK

The Private Directory tab becomes active.

- 4 Enter the full path of the private directory

or

Select Browse to search for the private directory.

- 5 Select OK to return to the User Administration window.

❖ Deleting a user

- 1 Select a user on the Users tab.

- 2 Select Delete.

The system asks for confirmation before the user is permanently removed from the system.

❖ **Changing the password of a user**

- 1 Select a user name and select Properties.

The Login tab displays the login information for the user.

- 2 Enter the new password in the Password and Confirmation fields and select OK.

Changing Passwords

About this chapter

This chapter discusses the Change Password function of EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	92
Changing passwords	92

Introduction

The Change Password function allows you to change your own password. This is a different function from the User Administration function that allows an Administrator to change the passwords of other users. However, the Change Password function can also be used by the Administrator to change the default Admin password created during installation. (This is often the first thing that an Administrator does, for security reasons.)

When the Administrator adds a new user to the list of authorized users, he or she gives the new user a user name and a temporary password. When that user opens EMap for the first time, he or she must use the user name and temporary password assigned by the Administrator. However, the user can then use the Change Password function to change the temporary password. (The user name itself cannot be changed. In order for a user name to be changed, the Administrator would have to delete the user name and add a new one.)

Changing passwords

❖ Changing passwords

- 1 Choose File>Change Password on the main menu.
The Change Password window displays.
- 2 Enter your current password in the Old Password text box.
- 3 Enter your new password in the New Password and Confirmation text boxes.
- 4 Select OK.

Importing Definitions

About this chapter

This chapter discusses how to import record definitions in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	94
Importing a COBOL record definition	94
Importing an ODBC record definition	96
Importing a record definition from an XML data file or XML schema	98
Importing a record definition from an HTML form	100
Importing a record definition from a map	102
Importing a record definition from a file	102

Introduction

A record is the logical definition of how data is organized. To interpret application data, EMap needs to know the definitions of the records in the application data file. These record definitions describe the fields in each record so that the program can understand and use the data in the fields.

In EMap, a record definition must include:

- Name of each data field in the record.
- Length and starting position of each field in a fixed-length record.
- Maximum length of each field in a delimited record and the delimiter character used to separate fields.
- Web parameter associated with each field if that field is used in an HTML form.

The record definition may also include other information, such as a description of each data field in the record, but such information is optional and not required by EMap.

Although you can manually enter the file layout, importing record definitions is more efficient.

You can import them from:

- A COBOL copybook
- An ODBC database
- An HTML form
- An XML file or XML schema
- A map
- A file

Importing a COBOL record definition

❖ Importing a record definition from a COBOL copybook

- 1 Choose Application Files>Import Definitions>COBOL from the main menu.

The COBOL Record Definition Import window displays.

- 2 Browse to the COBOL file you will use as the source for your record definitions.
- 3 Select one of the following options for the import:
 - Add Application Record Only – only the record names and definitions are imported from the COBOL file.
 - Add Application Record and File Name – the names and definitions of the records are imported from the COBOL file. They are associated with the file listed in the File Name text box.
 - Add Application Record, File Name, and Directory – the names and definitions of the records are imported from the COBOL file and are associated with both the file and directory listed in File Name and Directory text boxes.

Note The File Name and Directory text boxes list file and directory from which you are importing the record definitions. You may use these names, or you can change them.

- 4 Select Continue on the COBOL Record Definition Import window.
If the file name does not exist, the New File window displays.
 - Enter the File Type (and optional Description) and select OK. You return to the COBOL Record Definition Import window.

If you have errors, various error messages display in dialog boxes. Make the necessary corrections and select Continue on the COBOL Record Definition Import window.

The Select COBOL Records window displays.
- 5 Select the COBOL record you want to import and select Load.

When the program has finished importing the record definitions and performing the file and directory associations (if you chose either of those options), a View Log message box appears saying “Run Complete”.
- 6 Select Yes on the View Log message box, and the COBOL Import Log window displays. It lists the records from which definitions were extracted and imported and displays an import error status.
- 7 Select OK, and the excobol window displays saying “Extract Run Completed Successfully”.

- 8 Select OK on the excobol window to return to the COBOL Record Definition Import window.

Importing an ODBC record definition

When you import a record definition from an ODBC table, ECMAP uses an algorithm to find the closest match for the field being imported relative to the specific type of database used.

ECMap checks for the best match, but this is not always a straight one-to-one conversion.

Each vendor's database is interpreted on a case-by-case basis and is not always interpreted the same way. After a database is converted into ECMap, you should analyze how the database was converted and make adjustments if needed. In some cases, there are some consistent interpretations. For example, an Oracle Date-Time field is converted to a DT field with a time stamp attribute and a TM field.

❖ Importing record definitions from an ODBC database

- 1 Choose Application Files>Import Definitions>ODBC from the main menu.

The ODBC Record Definition Import window displays.

- 2 Enter the Data Source Name.

or

Select the up arrow and choose a Data Source from the Data Source Names window.

If you have not yet configured the DSN or to reconfigure a previously configured DSN, select Configure Data Source to access the ODBC Data Source Administrator window.

- 3 Modify the Driver Connect String, if necessary.

ECMap creates a connect string based on the Data Source Name. Depending upon the database you use, you may need to modify the driver connect string to include a username and password. If you add them, the format of the driver connect string becomes:

```
DSN=<data source name>;UID=<username>;PWD=<userpassword>.
```

- 4 Select Continue at the bottom of the window.
The Table Selection window displays.
- 5 Select a record definition import option:
 - Add Application Record Only – the imported record definitions are unassociated data objects and are not linked with a file or a directory.
 - Add Application Record and File Name – the imported record definitions are associated with the File Name on the same line in the bottom section of the window. The program prompts you to define the file before the record definitions are imported. You must:
 - Define the file type as SQL
 - Enter a DSN in the Data Source Name text box.
 - Add Application Record, File Name, and Location – the imported record definitions are associated with the File Name on the same line in the bottom section of the window, and the File Name is associated with the Application File Directory. You must
 - Enter the name of the application file directory associated with the file name and record definitions
 - Define the file type as Sequential (Fixed-Length) or Delimited
 - Enter a directory path in the Application File Directory text box.
- 6 On the top right section of the Table Selection window, the following information is entered:
 - The program automatically enters the Data Source Name, or DSN, of the ODBC database from which the record definitions are being imported. This name cannot be changed.
 - The Application File Directory specifies the location of the application data that is interpreted using the imported record definition. It is the name of the directory that is associated with the File Name and Record Name when you choose the option to perform this association.
- 7 Select each record (Table Name) that contains the definition you want to import.
- 8 When you have finished selecting the records, select OK to return to the main EMap window. A message box appears, saying “Record Definition Imported”.

Importing a record definition from an XML data file or XML schema

You can import a record definition from either an XML data file or XML schema. When you import a record definition from XML data, you can also create the flow at the same time.

❖ Importing a record definition from an XML schema

- 1 Choose Application Files>Import Definitions>XML>Schema from the main menu.

The XML Schema Import window displays.

- 2 Enter the full path of the XML schema file in the File to Load text box on the Schema Import tab

or

Browse for it.

- 3 Select Create Records.

The imported record display on the Schema Import tab.

- 4 Exit from the window to return to the main EMap window.

❖ Importing a record definition from an XML data file

- 1 Choose Application Files>Import Definitions>XML>Data File from the main menu.

The XML Schema Import window displays.

Enter the name of the XML data file in the File to Load text box on the Schema Import tab or select the button at the end of the text box to search for it. After you have entered or browsed for and selected the XML data file, select Load.

The program creates record definitions from the XML data file and displays them on the Schema Import tab. The Create Flow button is now active.

- 2 Enter the full path of the XML file in the File to Load text box on the Schema Import tab

or

Browse for it.

- 3 Select Load to import the records.

The imported record display on the Schema Import tab.

You can now return to ECTMap or continue working on the Schema Import tab to create a map flow based on the XML data.

- 4 To create a map flow, select Create Flow.

A dialog box displays the message: Is XML the primary input for the map?

Note You can only create a flow for an outbound XML map. If you try to create flow for an inbound XML map, a dialog box displays stating:

Cannot create XML Flow on an 'Inbound' Map.

- 5 Select Yes.

If flow already exists for this map, the following information displays:

Flow already exists! Do you want to overwrite?

- 6 Do one of the following:

- To keep the flow, select No.
- To create new flow from the XML data file, select Yes.

Next, the levels are checked and the following information displays:

Levels already exist! Do you want to overwrite?

- To keep the levels, select No.
- To create new flow levels from the XML data file, select Yes.

If rules have already been created for this map, you are given the option to choose which rules to keep:

- Delete No Rules
- Delete All Rules
- Delete Rule Range

Make your selection and select OK. The program creates the flow. A message displays telling you that the flow was successfully created.

- 7 Exit from the window to return to the main ECTMap window.

Importing a record definition from an HTML form

ECMap can import many fields on an HTML form. It can import:

- Text boxes
- Drop-down lists
- Radio buttons
- Check boxes

It cannot import from text areas.

❖ Importing a record definition from an HTML form

- 1 Choose Application Files>Import Definitions>HTML from the main menu.

The HTML Import window displays.

The appearance of the HTML Import window changes depending on the choices you make.

- 2 Do one of the following:

- Select Dir/File/Record to import a record definition and associate it with a file and a directory.

Enter the following information:

- The Input File is the name of the file from which the record definition is imported. You can enter the file name or Browse for it. If you select Browse, the Browse – HTML/XML File window displays. Double-click a file or enter it and select Open. You return to the HTML Import window, and the file you selected is entered in the Input File text box.
- The Directory is the name of the directory with which the file is associated.
- The File is the name of the file with which the record is associated.
- The Record is the internal ECMap name for the record which is associated with the directory and/or file above and which is interpreted using the imported record definition. ECMap assigns Record1 as the default record name for the first imported record definition, Record2 as the record name for the second imported record definition, etc. You can change the record name.

- Select File/Record to import a record definition and associate it with a file.

Enter the following information:

- The Input File is the name of the file from which the record definition is imported. You can enter the file name or Browse for it. If you select Browse, the Browse – HTML/XML File window displays. Double-click a file or enter it and select Open. You return to the HTML Import window, and the file you selected is entered in the Input File text box.
 - The File is the name of the file with which the record is associated.
 - The Record is the internal EMap name for the record which is associated with the file above and which is interpreted using the imported record definition. EMap assigns Record1 as the record name for the first imported record definition, Record2 as the record name for the second imported record definition, and so on
- Select Record Only to import a record definition only and do not associate it with a file or a directory.

Enter the following information:

- The Input File is the name of the file from which the record definition is imported. You can enter the file name or Browse for it. If you select Browse, the Browse – HTML File window displays. Double-click a file or enter it and select Open. You return to the HTML Import window, and the file you selected is entered in the Input File text box.
 - The Record is the internal EMap name for the record which is interpreted using the imported record definition. EMap assigns Record1 as the record name for the first imported record definition, Record2 as the record name for the second imported record definition, and so on.
- 3 Select View HTML, to view the HTML in your default Browser.

Importing a record definition from a map

❖ Importing a record definition from another map

- 1 Choose Application Files>Import Definitions>FROM MAP from the main menu.

The Select Project and Map window displays with a list of all current project/maps.

- 2 Double-click the map from which you want to import record definitions or

Highlight it and choose Select from the File menu.

The Records/Tables window and the Import Records window display side by side. All of the records associated with the selected map display on the Import Records window

- 3 Drag a record from the Import Records window and drop it on a record on the Records/Tables window.

If a record by that name already exists, you are asked to confirm that you want to overwrite the record.

- 4 When you have finished adding record definitions, close the windows and return to the main window.

Importing a record definition from a file

ECMap gives you the ability to import record definitions from a fixed-length flat file in the format shown below:

Table 7-1: Record definitions from fixed-length flat file

Field	Start column	End column
Record Name	1	40
Field Name	41	70
Field Type	71	72
Field Length	73	81
Number Decimals	82	86
Date Format	87	97
Century Type	98	98
Century Minimum	99	100

The record name can be up to 40 characters, but is truncated if it exceeds 40.

The last four fields may or may not be required based on the field type.

The century type should be one of the following for data fields:

- 1 = use field century minimum
- 2 = use global century minimum
- 3 = use no century minimum

❖ **Importing a record definition from a file in the format shown on the previous page**

- 1 Choose Application Files>Import Definitions>FROM FILE from the main menu.

The Import Records From File window displays.

- 2 Enter the name of the file from which you want to import a record definition

or

Browse for it.

- 3 Select Run to import the record definitions.

A message displays saying that your records were imported.

Sample import file

A sample EMap import file is shown below:

```

ImportRecord MEMBER# AN 12
ImportRecord SRV# AN 20
ImportRecord YMD_BIRTH DT 8 YYYYMMDD 250
ImportRecord AMTALLOW-P DI 8 2
ImportRecord AMTCHARGE DE 8 2
ImportRecord AMTPAY DE 8 2

```

Importing a record definition from a file

ImportRecord	AMTCOPAY	DE	8	2
ImportRecord	AMTCO2NS	DE	8	2
ImportRecord	AMTRISK	UI	8	
ImportRecord	AMTPP	UI	8	
ImportRecord	AMTDISCOUNT	UI	8	
ImportRecord	M-PAY-APPLY	DI	8	2
ImportRecord	FFS-EQUIV	DI	8	2
ImportRecord	AUTHORIZATION#	AN	12	
ImportRecord	CDR-REGION	AN	2	
ImportRecord	CLAIM-TYPE	AN	2	
ImportRecord	DIAG#	AN	6	
ImportRecord	DIAG2#	AN	6	
ImportRecord	DIAG3#	AN	6	
ImportRecord	DIAG4#	AN	6	
ImportRecord	DIAG5#	AN	6	
ImportRecord	DIVISION#	AN	10	
ImportRecord	DRG#	AN	4	
ImportRecord	INPATDAYS	DI	8	2
ImportRecord	DOS	DI	8	2
ImportRecord	MEDICARE-FLAG	AN	2	
ImportRecord	PCP-AFF#	AN	16	
ImportRecord	PCP-IPA	AN	2	
ImportRecord	PCP-OFFICE#	AN	4	
ImportRecord	POS-AFF#	AN	16	
ImportRecord	POS-IPA	AN	2	
ImportRecord	POS-OFFICE#	AN	4	
ImportRecord	POS-PRAC#	AN	4	
ImportRecord	PROC#	AN	6	
ImportRecord	PROC2#	AN	6	
ImportRecord	PROC3#	AN	6	
ImportRecord	PROG#	AN	2	
ImportRecord	PT-SVC-OPT	AN	2	
ImportRecord	SEX	AN	2	
ImportRecord	VC-CNT	DI	8	2
ImportRecord	YMDEFF	DT	8	MM/DD/YY 150
ImportRecord	YMDEND	DT	11	YYYY MON DD350
ImportRecord	YMDPAIDDT	DT	7	JULIAN 350

Exporting Definitions

About this chapter

This chapter discusses how to export record definitions in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	106
Creating ODBC tables	106
Creating templates for an HTML form	107
Export schemas	114
Exporting a schema to e-Biz 2000	114
Exporting to MSQI or e-Biz Integrator	115

Introduction

In addition to extracting and importing record definitions into EMap, you can also use current EMap record definitions to create an:

- ODBC table
- HTML fragment
- NDO schema for export to the NDO repository

When you create a new table, you use the layout of an internal EMap ODBC database table to create a new table in another database. The layout of the new table is exactly the same as the layout of the original table - both tables have the same record definitions.

When you create HTML fragments, you use some or all of the fields defined in a current EMap record to construct the pieces of an HTML form that can display the information in the record on a Web browser.

When you create an NDO schema and export it to the NDO repository, it can be used to read and write NDO records and place NDO files on a queue.

Creating ODBC tables

❖ **To create a new table using an ODBC record definition**

- 1 Select Application Files>Export Definition>ODBC from the main menu.

The ODBC Table Create window displays.

- 2 Enter the Data Source Name.

or

Select the up arrow and choose a Data Source from the Data Source Names window.

If you have not yet configured the DSN or to reconfigure a previously configured DSN, select Configure Data Source to access the ODBC Data Source Administrator window.

Note To export your record or field to MS Excel, do not select MS Excel. Instead select MS Access Database. After the record is saved in .mdb format, import the file into MS Excel.

- 3 Modify the Driver Connect String, if necessary.

ECMap creates a connect string based on the Data Source Name.

Depending upon the database you use, you may need to modify the driver connect string to include a username and password. If you add them, the format of the driver connect string becomes:

```
DSN=<data source name>;UID=<username>;PWD=<userpassword>.
```

- 4 Select the application record from which you use as the format for the new ODBC table from the drop-down list.

The Table Name text box is populated with the table in which the record is located. You can change the table name.

Note In the current version of the software, if there is a space in a record or field name, the table is not created. This restriction applies only to exporting an ODBC record definition. You can successfully import an ODBC record definition with a space in a field name or a record name.

- 5 Select Continue, and the program creates the table. You return to the main ECMap window.

Creating templates for an HTML form

If you want to create the template for an HTML form from a record definition, choose Export Definition from the Application Files menu on the main ECMap window and then choose HTML from the Export Definition submenu.

❖ To create an HTML form

- 1 Select Application Files>Export Definition>HTML from the main menu.

The HTML Form Template Creation window displays.

- 2 Enter the full path for the HTML file to use as the form template

or

Browse to a location for the HTML file.

- 3 Select Next.

The HTML Form Purpose window displays.

- 4 Select one of the following options and click Next:

- Display – the form will display the field labels and data.
- Collection – the form accepts information as input for submitting. On the window, there are field labels with associated blank text boxes/fields. The user is able to enter information and select a Submit button. (Data entry occurs.)
- Display and Collection – the form accept information as input and submits it. On the window, there are field labels with associated text boxes/fields that contain current information. The user is able to modify certain information and select a Submit button. (Data entry occurs.)

The Select Records for Form Template window displays.

- 5 Highlight the record to be used as the template for the HTML form, and select Next.

The HTML Form Fields window displays.

- 6 Enter the information for your form as described in “Customizing the HTML form,” and select Next.

The HTML Form Template Completion Window displays.

- 7 Select Finish to create the HTML template.

You can preview the HTML file by selecting View HTML.

Customizing the HTML form

If you have chosen to create a form that is used for display only, you can customize the form by:

- Modifying the web parameter and field label
- Attaching a style sheet to the HTML page

If you are creating a form that is used to collect information from the user, you can also:

- Specify field types to use in the form
- Specify the input method for the form
- Add action buttons

You begin all of these tasks on the HTML Form Fields window.

Modifying the Web parameter or field label associated with a field

❖ To change a field label or web parameter:

- 1 Choose the field you want to alter and select Modify.

The Modify Field: <fieldname> window displays.

The current values the *Web Parameter* and the *Field Label* are automatically entered in the text boxes.

- 2 Make the necessary changes and Select OK.

Selecting a style sheet to apply to the HTML form

You can control the appearance of your HTML form by applying a Cascading Style Sheet (CSS) to it.

❖ Applying a style sheet to the HTML form

- 1 Select Style Sheet on the HTML Form Fields window.

The Style Sheets window displays.

- 2 Select New on the Style Sheets window.

The Style Sheets window displays.

- 3 Enter the Web path to the style sheet (for example, *./styles/myforms.css*). Select OK.

The Web path you entered displays on the window, unless the path you entered did not contain a “/”. In that case, the window remains blank, and you receive no message. You must select New again and reenter a Web path to the style sheet that has a valid format.

- 4 Select Close to return to the HTML Form Fields window.

❖ Removing an style sheet from an HTML form

- 1 Select Style Sheet on the HTML Form Fields window.

The Style Sheets window displays.

- 2 Highlight the style sheet you wish to remove from the HTML form, and select Remove.

- 3 Select Close to return to the HTML Form Fields window.

Specifying the type of fields to be used on the form

❖ Specifying a field type to use on an HTML form

- 1 Highlight the field you want to format Options.

The Field Options: <fieldname> window displays.

There are four field types you can use to display your field information on an HTML form.

- Text box
- Drop-down list
- Radio button
- Check box

- 2 Enter the information for each field type as described below.

Text box fields

❖ Creating a text box field

- 1 Choose Text and select Configure.

The Text Box Value window displays.

- 2 Enter the value that is returned to the server for this field when the HTML form is submitted (as <field label>=<text box value>).

- 3 Select OK to return to the Field Options window.

Drop-down lists

❖ Creating a drop-down list

- 1 Choose Drop Down List and select Configure.

The Drop Down List window displays.

- 2 Select New.

The New Drop Down List window displays.

- 3 Enter the label and value information information for the drop-down list.

- The *Label* is the text that displays in the drop-down list.

- The *Value* is the value that is returned to the server when the corresponding Label is selected. This information is optional; if you do not enter a value, the program automatically uses the Label as the Value.

The information displays in the drop-down list in the order you enter it

- 4 Select OK to return to the Drop Down List window. The new entries appear on the window.
- 5 Select OK to close the Field Options window.

Modifying a drop-down list

You can modify and delete items on a dropdown list.

❖ **Modifying a drop-down list**

- 1 Choose Drop Down List and select Configure.

The Drop Down List window displays the items on the drop down list.

- 2 Select the list item you want to change and select Modify.

The Modify Drop Down List window displays the current label and value information for the list item.

- 3 Modify the label and value as needed, and select OK to return to the Drop Down List Window
- 4 Select OK to close the Field Options window.

❖ **Deleting a drop-down list**

- 1 Choose Drop Down List and select Configure.

The Drop Down List window displays.

- 2 Select the list item you want to delete and select Remove.

The list item is removed from the drop-down list.

- 3 Select OK to return to the Drop Down List Window
- 4 Select OK to close the Field Options window.

Radio buttons

❖ **Adding a new radio button**

- 1 Choose Radio Button and select Configure.

The Radio Button window displays.

- 2 Select New.
The New Radio Button window displays.
- 3 Enter the text that displays as a radio button and the label that is associated with that information.
 - The *Label* is the text that displays as a radio button.
 - The *Value* is the value that is returned to the server when the corresponding Label is selected.
- 4 Select OK to return to the Radio Button window. The new radio button information appears on the window.
- 5 Select OK to close the Field Options window.

❖ **Modifying a radio button**

- 1 Choose Radio Button and select Configure.
The Radio Button window displays.
- 2 Highlight a label/value pair you wish to alter and select Modify.
The Modify Radio Button window displays
- 3 Modify the label and value information as needed.
- 4 Select OK to return to the Radio Button window. The modified radio button information appears on the window.
- 5 Select OK to close the Field Options window.

❖ **Deleting a radio button**

- 1 Choose Radio Button and select Configure.
The Radio Button window displays.
- 2 Highlight a label/value pair and select Remove.
The label/value pair is removed from the list.
- 3 Select OK to return to the Field Options: <fieldname> window.
- 4 Select OK to close the Field Options window.

Check boxes

❖ **Adding a new check box**

- 1 Choose Check Box on the Field Options window and select Configure.

The Check Box Value window displays.

- 2 Enter the value that is returned to the server for this field when the HTML form is submitted
- 3 Select OK to return to the Field Options window.
- 4 Select OK to close the Field Options window.

Adding action buttons to your HTML form template

You can add submit and reset buttons to your HTML form.

When a submit button is selected, it sends all the information entered in an HTML form to the server for processing.

When a reset button is selected, all the information entered in an HTML form is cleared and returned to the default display state.

❖ Adding an action button to an HTML form

- 1 Select Buttons on the HTML Form Fields window.

The Add Buttons to HTML Form window displays.

- 2 Select the type of button you want to add to your form.

For each button you've selected enter a name and value and enter the following information:

- The name attribute is used to identify the button. If it is used with a submit button, the name parameter is passed to the server with the form data.
- The value attribute is used as the display name for the button. For example, you would typically enter a value of Clear for the reset button, and a value of OK or Enter for the submit button.

Adding the form method and action to your HTML form template

❖ To specify the method and action attributes for the HTML form,

- 1 Select Form Tag on the HTML Form Fields window.

The Form Tag Information window displays.

- 2 Enter the following information:

- The *Method* attribute specifies how the browser sends data to the server for processing. There are two choices for sending the data:

- POST – This is a two step transmission. The browser contacts the server in the first transmission. After contact has been made, the browser sends the data to the server in a second transmission.
- GET – The server is contacted and the data sent at the same time during one transmission.
- The *Form Action* attribute specifies the URL of the application that will process the form data that is submitted to the server.

Export schemas

You can export schemas to three products:

- e-Biz 2000
- e-Biz Integrator
- MQSI

For each product, you first create a schema name, associate it with record definition(s), and then export it.

Exporting to e-Biz 2000

To export a schema to e-Biz 2000, you first create a schema name and associate it with record definition(s) and then export it to e-Biz 2000.

❖ Exporting a schema to e-Biz 2000

- 1 Choose Application Files>Export Definition>Schema>e-Biz 2000 from the main menu.

The first time that you choose this option, this prompt displays:

Would you like to select a Configuration File for schema mode?

- 2 *If you select Yes*, the Choose Configuration File window displays where you can select a configuration file.

If you select No, the Configuration Information window displays.

In this window you can create a configuration file. The configuration file will be in Schema mode.

Exporting to MSQI or e-Biz Integrator

To export a schema to the MQSI or e-Biz Integrator, first create a schema name and associate it with record definition(s) and then export it to the e-Biz Integrator, e-Biz 2000, or MQSI.

❖ **Exporting a schema to MQSI or e-Biz Integrator**

- 1 Choose Application Files>Export Definition>Schema>e-Biz Integrator

or

Application Files>Export Definition>Schema>MSQI from the main menu.

- 2 The SchemaExport window displays, asking if you would like to select a configuration File for schema mode.

If you select Yes, the Choose Configuration File window displays. Select a configuration file.

If you select No, the Configuration File Options window appears.

- 3 Select With MQSI or e-Biz Integrator Options, select OK.

The Configuration Information window displays.

- 4 Enter the following information on the Configuration Information window:

- Server Name – the name of the computer where the database used by the Rules and Formatter is installed.
- User Name – used to log onto the computer where the database used by the Rules and Formatter is installed.
- Password – used to log in to the computer where the database used by the Rules and Formatter is installed.
- Database – the name of the database that Rules and Formatter set up as a schema repository and to which you will export this schema.
- Formatter Version – the database (including version) that is loaded on the computer where Rules and Formatter is installed. Select from the following:
 - Version 4.11 – Microsoft SQL Server
 - Version 4.11 – Oracle 8.x Server
 - Version 4.11 – DB2 5.x Server

- Version 4.11 – Sybase 11.x Server
 - Version 5.1 – Microsoft SQL Server
 - Version 5.1 – Oracle 8.x Server
 - Version 5.1 – DB2 5.x Server
 - Version 5.1 – Sybase 11.x Server
- Prefix – a unique identifier that is placed at the beginning of all of the schema names, and which help locate the schemas in the Rules and Formatter schema repository. The prefix cannot be longer than 5 characters.

5 Select Next.

The schema configuration file appears. Sample configuration files are in the *ECMap Reference Guide*. Make any modifications needed to the configuration file and then select Save. The Schema Export – Schema Name Selection window displays.

6 Create the name of the schema that you are exporting to the Rules and Formatter repository, and a description of the schema.

7 Select the Add button.

The Add Schema Name window displays.

8 Enter the following on the Add Schema Name window:

- In the Schema Name text box, enter a name for the schema you are exporting to the Rules and Formatter schema repository.
- In the Description text box, enter a brief description of the schema.

9 Select OK to return to the Schema Export – Schema Name Selection window.

The names and descriptions of all schemas associated with this map display.

- To delete a schema, highlight the schema and select Remove.
- To add another schema, select Add and follow the instructions above.

10 When you have finished adding schemas, select Next at the bottom of the window.

The Schema Export - Record Selection window displays.

All of the records that are defined in the current map display on the Schema Export - Record Selection window.

- 11 Select the check boxes next to the record definitions that you want to export to the Rules and Formatter schema repository.

- 12 Select Next.

The Schema Export – Schema Creation window displays.

- 13 Drag and drop associate records with a file and to create parent-child relationships between the records.

The record(s) selected for export display in the Records pane on the left side of the window. The schema name(s) you created display in the pane on the right side of the window.

For each schema, drag the parent record that you want associated with a schema and drop it on the schema name. The record(s) now display under the schema name(s) in the pane on the right side of the window.

- 14 Select Next and the Schema Export – Export < schema name > window displays.

On this window, you have three options. You may select any combination of these options. If you choose:

- Export Schema – the program exports the schema(s) you created, and places them in the Rules and Formatter schema repository.
- Create Flow – the program automatically creates the outbound map flow table associated with this schema. The flow table includes the flow relationships and the rules to read the NDO records. (This option is not available for inbound maps.)
- Create File Definition – the program creates a file with the name of the schema/message, assigns NDO as the file type, and creates the association between the file and the records you selected for export.

- 15 Select Finish.

The window closes, and the program performs the option(s) you selected. You receive messages telling you that the operations were successfully performed.

- 16 Select OK on each message box.

Working with Records and Tables

About this chapter

This chapter discusses records and tables in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	120
Working with records and tables	121
Performing field-related activities	123
Associating records and files	129

Introduction

Application data is stored in flat files or in databases. Flat files are made up of records, and the records are made up of fields that hold data. Databases are made up of tables, and the tables are made up of column entries, or fields, that hold data.

A record, or table, is a collection of data items arranged for processing by a program. A record can have a fixed length or a variable length and typically contains the length information.

To interpret application data, EMap needs to know the definitions of all the records in a file. Record definitions describe the fields in each kind of record so that the program can understand and use the data in the fields.

Record definitions can be entered manually, or imported from an external source as described in Chapter 7, “Importing Definitions”.

Performing record-related activities

In EMap, you can:

- Create a new record or table
- Delete a record or table
- Modify a record or table
- Create a copy of a record, including all the fields in the record

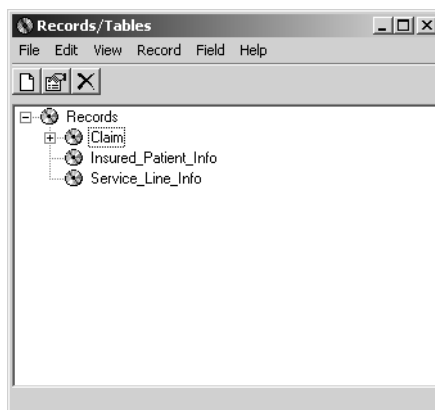
When you create or modify a record or table, you can add, delete, modify, or delete a field. EMap can also automatically recalculate the starting positions of the fields in a record.

Where to begin

All record- and table-related actions are initiated from the Records/Tables window.

- ❖ **To access the Records/Tables window**
 - Choose Application Files>Records/ Tables from the main menu.

The Records/Tables window displays.



Working with records and tables

This section discuss creating new records, adding new fields to records, deleting records, modifying records, and viewing record details.

❖ Creating a new record

- 1 Choose File>New>Record from the Records/Tables window menu.

The New Record window displays. There are two tabs pages: Required and Optional.

- 2 Enter the following information on the Required tab:
 - Name – the name of the record.
 - Description – a brief description of the record.
 - Length – the actual length of the record if this is a fixed-length file or the maximum length of the record if it is a delimited file. If you do not know the length of the record, you can enter a zero (0), and EMap calculates the length for you as you add fields to the record.
 - End Character – Select the character used to indicate the end of the record from the drop-down list. Choices are:
 - CRLF
 - None
 - Null

- SP CH (special character) Select an End Character from a drop-down list of characters that can be used as record delimiters. For example, to put a quote (“) as your end character, enter 034.
- 3 Enter the following information on the Optional tab:
 - Record Type – select from the following choices on the drop-down list:
 - Source – the record is the input data to the map.
 - Destination – the record is the output data of the map.
 - Both – the record is both input data to and output data of the map.If you make no entry, the program uses Both as the default.
 - Record Tag – only used when writing XML data. It is the name of the XML tag used to enclose the data, as shown below:

```
<RecordTagName>data</RecordTagName>
```
 - Select Repeating if the record can repeat. (NDO files only.)
 - Select Mandatory if the record is required. (NDO files only.)
 - 4 Select OK.

The New Field window displays. There are two tabs on the New Field window – Fields and Options. On these two tabs, enter detailed information about the fields in the record. For more information see “Adding a new field to a record” on page 123.

❖ **Deleting a record**

- 1 Select a record from the Records/Tables window and choose Record>Delete.
You are prompted to confirm the deletion.
- 2 Select Yes to delete the record.

❖ **Modifying a record**

- 1 Select a record and choose Record>Properties.
The Record Properties window displays.
- 2 Make the necessary changes on the The Record Properties as described in “Creating a new record”.

❖ Viewing record details

- Select a record and choose View>Details from the View menu on the Records/Tables window.

The Details for Record window displays. On the window is a list of all the fields in the highlighted record with all of the attributes associated with each field.

❖ Copying a record

- 1 Select a record on the Records/Tables window and choose Record>Template

The Template Record window displays.

The record-related information is identical to the source file. If you need to change the template copy, enter the information as described in “Creating a new record”.

- 2 Enter a name for the new record.
- 3 Select OK.

Performing field-related activities

You can perform the following field activities:

- Add new fields to a record
- Delete fields from a record
- Modify fields

Working with fields

❖ Adding a new field to a record

- 1 Select File>New>Field.

The New Field window displays. There are two tabs pages: Fields and Options.

- 2 Enter the following in the Fields tab:
 - Field Name – the name of the field.

- Field Description – a brief textual description of the field. (If you do not enter a description, the field name is used to populate this field.)
- Start Position – the number of the column in which the field begins in a record in a fixed-length file. EMap automatically calculates this based on the previous field lengths.

Note In fixed-length files, multiple fields can have the same “columns”- two fields can share a location. In delimited files and table definitions, fields may NOT overlap column positions.

- Field Length – the size of the field in bytes. Fixed-length records are made up fixed-width fields. Variable-length records are made up of variable-width fields.
- Field Type – describes the type of data in the field. Enter the field type or select it from the drop-down list. Depending on the field type you select, additional text boxes appear in which information must be entered:
 - AN – Alphanumeric. Field where the characters consist of the letters A – Z, numbers 0 – 9, and special characters (-,/). (ex. ABC12-3D)
 - DE – Decimal Explicit. Numeric field containing an actual decimal point (ex. 123.45). In the No. Decimals text box, enter the number of places to the right of the actual decimal point.
 - DI – Decimal Implicit. Numeric field with the position of an implied decimal point defined by the number of characters following the decimal point. (ex. 12345 with 3 decimal places = 12.345)

In the No. Decimals text box, enter the number of places to the right of the implied (not actual) decimal point.

Note Before any calculation or assignment, a DI field type is changed to a DE field type internally by EMap.

- DT – Date. Numeric field containing the date in a specified format.
- For the Date Format, choose from the following entries in the drop-down list:
 - YYMMDD

- YYYYMMDD
 - MMDDYY
 - DMMYY
 - JULIAN – EMap supports 7-digit (4-digit year + 3-digit days) Julian dates.
 - MMDDYYYY
 - MM/DD/YY
 - DD/MM/YY
 - MON DD YYYY
 - DD MON YYYY
 - YYYY MON DD
- Century Minimum Type – defines the way in which 2-digit years in specified date fields are handled. Select one of the following three choices from the drop-down list:
 - Global century minimum – to change all 2-digit years to 4-digit years, use value specified when map was defined. (All years prior to global century year are treated as next century. All years including and after global century year are treated as current century.)
 - Field century minimum – specify a year to be used as global century year (overriding value specified when map was defined) to change all 2-digit years to 4-digit years. An additional text box appears. Enter the year you want to use as the global century year for this date field in the Field Century Minimum text box that appears. The Field Century Minimum is a numeric value between 0 and 99 used to determine the century when translating a 2-digit year to a 4-digit year. Every 2-digit year less than or equal to this value is assumed to have a century of 20 and every 2-digit year greater than this value is assumed to have a century of 19. Enter the numeric value in the text box.
 - No century minimum – specify that no century type logic is to be used. (2-digit years are not converted to 4-digit years.)
 - TM – time. Numeric field containing time in 24-hour format—HHMM (for example, 9 pm = 2100).
 - PD – packed decimal. Decimal field with compressed notation.

- R – real number. Numeric field that allows for a floating decimal.
- SD – signed decimal. Decimal field with the last character corresponding to an overpunch field (for COBOL-based systems).

In the No. Decimals text box, enter the number of places to the right of the implied decimal point:

- SI – signed integer.
 - UI – unsigned integer. Numeric field without a plus or minus sign.
- 3 The Web Parameter is used to link the data in a field to an HTML form, when the form is being read or written.
 - The Attributes and Edits text boxes contain the information that you enter on the Options tab of this window. The choices for attributes and edits are entered on the Options tab of the New Field window and are explained below.
 - The XML text box contains either an A or a D, based on the choice you made on the Options tab under XML Options—A for Attribute and D for Data.
- 3 Enter the following information under Attributes on the Options tab
 - Key Field – reserved for future use.
 - Trade Partner ID – checked if this field contains the value that identifies the business entity with which data is being exchanged. On outbound maps, you must have a field with this attribute in the master record because this field is used to search the trading partner database for the receiver code to put on the EDI envelope. (In addition to providing required information for the EDI envelope, the trading partner lookup is used for routing and dynamic map selection.) On inbound maps, this field is used to search the trading partner database for your application’s internal sender code for this trading partner.
 - Protected – specifies that this field is not cleared when the “clear unprotected” rule command is applied. This attribute is used primarily in the data looping structure when you need to clear records between loops received but you want some application fields to remain uncleared.

- Today's Date +/- n Days – used to indicate that this field contains today's date plus (+) or minus (-) the number of days indicated in the text box next to the check box. The date is derived from the system, not from data received. (The minus (-) appears in the text box, but the plus (+) does not appear.)
- Date Timestamp – used in many ODBC record definitions where the date column is a timestamp. Note that each database from a vendor (for example, Sybase) is interpreted on a case-by-case basis and is not always interpreted the same way. After a database is converted into EMap, you should analyze how the database was converted and make adjustments if needed. In some cases, there are some consistent interpretations. For example, an Oracle Date-Time field is converted to a DT field with a time stamp attribute and a TM field.
- Auto Increment – used when an SQL table has an auto increment column, and EMap should not update the field. The field is not passed back to the database when an insert or an update is performed. This allows the database to continue to automatically populate key fields. [TIP: This attribute can be set on any fields that should not be updated in an SQL Select and Update command.]
- Fixed String – lets you specify the default for this field as the value entered in the text box to the right. It is used on records that are written (output records).
- Record Type – specifies that this field contains the record type identifier entered in the text box to the right. This attribute is used for files that contain multiple record types which each have different record definitions.
- SQL Stored Procedure Parameter attribute indicates that this field is accessed during the rule execution in an SQL Stored Procedure Call command. If this box is checked, a text box appears to the right of it.

In the text box, choose from a drop-down list to specify whether the data in this field is used as:

- INPUT to the stored procedure
- OUTPUT of the stored procedure
- INPUT to and OUTPUT of the stored procedure

The choice you make here is used in conjunction with the Marked Field in Rec are Parameters check box in the Record Field Parameters section of the SQL Procedure Call window. (See the Rules chapter for a detailed explanation of how this works.)

Right-click the bottom text box, and a list of possible SQL field attributes displays. Double-click an attribute, and it is entered in the bottom text box.

- 4 Enter the following information under Edits on the Options tab:
 - Justification for a field:
 - L – left-justification. The data in the field begins in the leftmost column of the field.
 - R – Right-justification. The data in the field ends in the rightmost column of the field.
 - N – No justification. The program uses default justification rules for data in the field.
 - Mandatory fields must contain data or a warning is generated on inbound maps.
 - If Zero Fill is checked, the program fills all empty spaces with zeroes. This applies only to data with a field type of Decimal Implicit (DI) or Signed Decimal (SD). (These are the two primary types of fields used by COBOL-based environments, which is where zero-filling fields is most important.)
- 5 Enter the following information under XML Options on the Options tab:
 - Attribute – the field contains an XML attribute.
 - Data – the field contains actual XML data.

You must choose one of these two options if the information in the field needs to be read from or written to an XML document. Otherwise, this information is not required.

- 6 Select Next to enter information for another field.

If you have entered all the fields in the record, select OK to return to the New Record window.

❖ **Deleting a field in a record**

- 1 Select a and choose Field>Delete.

You are asked to confirm the deletion.

2 Select Yes to delete the field.

❖ **Modifying a field in a record**

1 Select a record and choose Field>Properties.

The Field Properties window displays.

2 Make the necessary changes on the The Field Properties as described in “Adding a new field to a record”.

❖ **Recalculating the starting column of fields in a record**

Use this function when a field length has changed or when a field is deleted or added.

- Select a field and choose Field>Recalculate Columns.

The start column is recalculated for the highlighted field and all subsequent fields.

Associating records and files

This section provides instructions for administering the association between records and files.

❖ **Creating an association between a record and a file**

1 Choose Application Files>Files/Databases from the main menu.

The Files/Databases window displays side-by-side with the Records/Tables window.

2 Drag a Record from the window and drop it onto a directory in the Files/Database window.

Once the link is created, all record associations with a file display when you double-click the file.

❖ **Removing an association between a record and a file**

- Highlight a record choose Record>Remove.

Working with Files and Databases

About this chapter

This chapter discusses files and databases for EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	132
Performing file-related activities	132
Administering files or databases	136

Introduction

ECMap stores application data in flat files or databases. Flat files, often referred to as ASCII or text files, are made up of records. Databases are made up of tables and can be either ODBC or non-ODBC compliant. In ECMap, databases are the equivalent of files, and tables are the equivalent of records.

In ECMap, directories, files, databases, records, and tables are separate data objects. This provides flexibility in defining file-based application data. One directory can have many files, one file can have many records, and one database can have many tables. On the other hand, a number of files can point to the same record, and many databases can include the same table.

File types

To interpret application data, ECMap must know the file type. ECMap recognizes seven file types:

- Delimited – flat files. Contains records with data in variable-width fields separated by special characters called delimiters.
- HTML/XML
- Keyed Dbase – databases. The only non-ODBC database that ECMap supports, but it is generally used only for maps created in earlier versions of ECMap.
- NDO
- Sequential-ASCII – flat files.
- Sequential-EBCDIC – flat files.
- SQL database – databases. Files reside in ODBC-compliant databases.

Performing file-related activities

In ECMap, you can:

- Create a new file/database
- Delete a file/database
- Change the name, type or description of a file/database
- You can create and remove associations between:
 - A directory and a file

- A file and a record
- A database and a table

Where to begin

All file and database-related actions are initiated from the Files/Databases window.

❖ **To access the Files/Databases window:**

- Choose Application Files>Files/Databases from the main menu.

The Files/Databases window displays.



Creating a new file or database

❖ **To create a new file or database**

- 1 File>New from the Files/Databases window menu.

The New File window displays. It has two tabs—Required and SQL Options.

- 2 Enter the following information on the Required tab page:

- Enter the name of the new file or Browse for it

Note Do not use blanks in file names.

- Select the file type. The choices are:
 - DELIMITED

- HTML/XML – used with Web-based maps
- KEYED DBASE – used to support maps created with old software versions
- NDO – not be selected when creating a new file
- SEQUENTIAL-ASCII
- SEQUENTIAL-EBCDIC
- SQL DATABASE – used with ODBC databases

Defining non-SQL file types

The appearance of the New File window depends upon the file type you select. The specific window and the information required for each file type are shown in the following section:

For HTML/XML, Sequential, or Keyed Dbase file types, enter:

- File Description (optional)
- Decimal Indicator – choose either Period or Comma from the drop-down list. (optional).

For a Delimited file type, enter:

- File Description (optional)
- Decimal Indicator – choose either Period or Comma from the drop-down list. (optional).
- Delimiter – Select the up arrow to open the Select Delimiter window. Scroll through the list of delimiter characters and double-click the one you want to use to separate the fields in the record.
- Quote – Select the up arrow to open the Select Quote Delimiter window. Scroll through the list of delimiter characters and double-click the one you want to use, or type 000 for no quote.

See the *ECMap Reference Guide* for a chart of ASCII characters.

Defining a SQL Database file type

If you choose SQL Database as the file type, the SQL Options tab becomes active. Enter additional information on the Required tab, and on the SQL Options tab.

On the Required tab, enter:

- Data Source Name – the DSN you have associated with this database.
- Driver Connect String – constructed by the program from the information you entered in the Data Source Name text box.

You may modify the driver connect string to include a username and password if they are required.

If you select Test SQL Connection, EMap tests whether it can talk to the database associated with the DSN via ODBC. A message appears saying whether or not the test was successful.

After you have finished entering information on the Required tab, select the SQL Options tab to make it active.

On the SQL Options tab, enter:

- When you select the Auto Commit check box, the program makes immediate updates to the database. As a result, the SQL Rollback function is disabled, and you cannot use the SQL Check point (and Backout Point) commands.
- When you select the Insert Alphanumerics as Null When Blank check box, the program inserts octal null characters into fields that have no data characters.
- When you select the Close and Drop Cursor After Insert arrow, you are given two choices:
 - Y = Yes. Perform automatic COMMITs
 - N = No. Do not perform automatic COMMITs

If you choose Y, you must provide two additional pieces of information:

- Choose one of following for the Translation Isolation Option:
 - Database Default List
 - Highest Allowed
 - Transaction Serializable
 - Transaction Repeatable Read
 - Transaction Read Committed
- Choose one of following for the Translation Commit Performed:
 - On Every EDI Message

- On Map Switches and end of Program
- No Preset Commit

Administering files or databases

This section provides instructions for administering files or databases.

❖ Removing a file or database

- Select the file and choose Application Files>Delete.

The program asks you to confirm that you want to delete the file before removing it permanently.

❖ Modifying an file or database

- 1 Select a file and Application Files>Properties.

The File Properties window displays.

- 2 Make the appropriate changes to the information as described in “Creating a new file or database” on page 133.

Associating files with directories and records

You can create and remove associations between:

- Files and directories
- Records and files

In order to make any of these associations, you must have the appropriate windows open in the EMap workspace. If you are associating a file with a directory, you must have the Files/Databases window and the Directories (Mailboxes) window open; if you are associating a file with a record, you must have the Files/Databases window and the Records/Tables window open.

❖ Creating an association between a file and a directory

- 1 Choose Application Files>Directories (Mailboxes) from the main menu.

The Directories (Mailboxes) window displays side-by-side with the File/Databases window.

- 2 Drag a file from the Files/Database window and drop it onto a directory in the Directories (Mailboxes) window.

Once the link is created, all files associations with a directory display when you double-click the directory.

❖ **Associating a record and a file**

- 1 Choose Record>Open from the Files/Databases window menu.

The Records/Tables window displays next to the Files/Databases window.

- 2 Drag a Record from the window and drop it onto a directory in the Files/Database window.

Once the link is created, all record associations with a file display when you double-click the file.

❖ **Removing an association between a file and a directory**

- 1 Select a file on the Directories (Mailboxes) window.
- 2 Choose Application Files>Remove.

❖ **Removing an association between a record and a file**

- 1 Highlight a record on the Files/Databases window.
- 2 Choose Remove from the Record menu.

Copying a file or database

You can create a copy of a file or database along with all its record associations by using the Template command.

❖ **Creating a copy of a file**

- 1 Select a file choose Application Files>Template.

The Template File window displays.

The file type and related information is identical to the source file. If you change the file type for the template copy, enter the file type related information as described in “Creating a new file or database” on page 133.

- 2 Enter the new file name in the File Name text box or Browse to the file.
- 3 Select OK to return to the Files/Databases window.

Working with Directories and Mailboxes

About this chapter

This chapter discusses directories in ECMaP.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	140
Performing directory-related activities	140

Introduction

In EMap, while information is logically organized in projects and maps, it is physically organized in a directory/file/record structure or a database/table structure. Files are located in directories and subdirectories. Directories are sometimes referred to as folders, and certain directories are called mailboxes. A mailbox is a directory that is used for the pickup and delivery of data that has been mapped or will be mapped.

In EMap, directories and files are separate data objects. A file does not need to be associated with a directory in order to exist. This provides great flexibility in defining file-based application data. One directory can have a number of files, and many directories can include the same file.

Performing directory-related activities

In EMap, you can

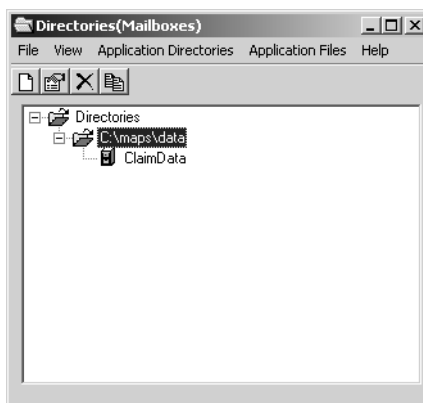
- Create a new directory
- Select a directory
- Rename a directory
- Delete a directory
- Copy of a directory
- Establish directory-file associations
- Remove directory-file associations

Where to begin

All directory-related actions are initiated from the Directories (Mailboxes) window.

- ❖ **To access the Directories (Mailboxes) window**
 - Choose Application Files>Directories (Mailboxes) from the main menu.

The Directories (Mailboxes) window displays.



Creating, renaming, and deleting directories

❖ Creating a new directory

- 1 Select File>New from the Directories (Mailboxes) window.

The New Directory window displays.

- 2 Enter the name of the new directory

or

Select Browse to browse to a directory in the Select a Directory... window

- 3 Select OK.

If you enter the name of a directory that does not exist, you are asked to confirm that you want to create it.

- 4 Select OK to return to the Directories (Mailboxes) window.

❖ Changing the name of a directory

- 1 Select a directory choose Application Directories>Properties.

The Directory Properties window appears.

- 2 Enter the new directory name in the Directory text box or Browse for it and Select OK.

- 3 On the Select a Directory ... window, either double-click a directory or enter a directory path in the Folder Name text box and select OK.

You are prompted to confirm the name change for all directory occurrences.

- 4 Select OK. You return to the Directories (Mailboxes) window and the new name of the directory displays.

❖ **Deleting a directory**

- Select a directory and choose Application Directories>Delete.

The program asks you to confirm that you want to delete the directory before it removes it permanently.

Associating files and directories

❖ **Creating an association between a directory and a file**

- 1 Select Application Files>Open from the Directories (Mailboxes) window.

The Files/Databases window displays side-by-side with the Directories (Mailboxes) window.

- 2 Drag a file from the Files/Database window and drop it onto a directory in the Directories (Mailboxes) window.

Once the link is created, all files associations with a directory display when you double-click the directory.

❖ **Removing an association between a file and a directory**

- Select the file and choose Application Files>Remove.

The file is removed from the directory list.

Copying a directory

You can create a copy of a directory along with all its file associations by using the Template command.

❖ **Creating a copy of a directory**

- 1 Select a directory choose Application Directories>Template.

The Template Directory window displays.

- 2 Enter the new directory name in the Directory text box or Browse for it and Select OK.

- 3 On the Select a Directory ... window, either double-click a directory or enter a directory path in the Folder Name text box and select OK.
You are prompted to confirm the name change for all directory occurrences.
- 4 Select OK. You return to the Directories (Mailboxes) window and the new name of the directory displays including all the file and record associations from the source directory.

Creating Transactions

About this chapter

This chapter describes how to create transactions in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	146
Create transaction functions	146
Working with the standards library	152
The standards library – transaction tree view	162

Introduction

When you create a transaction, you specify the segments to use when you exchange information with your trading partner.

You can create your transaction with the segments displayed in either a tree view (Transaction Tree) or a table view (Transaction Table). The default is the Transaction Table, but you can change the display by selecting View>Transaction Tree from the main menu.

You can use the Transaction Tree or Transaction Table to add or modify segments, elements, levels, or codes

Create transaction functions

When you define a transaction map, select the standard, version, and message on the Map Properties tab of the New Map Definition window. To create the transaction that is mapped:

- Add any new segments, elements, or codes that must be included in your EDI message, or transaction, and modify existing segments, elements or code, if necessary.
- Tell the program which segments to include in your EDI message, or transaction.
- “Make” the transaction.

If you choose:

- Transaction Table – the Create Transaction: *<transaction set>* window displays.
- Transaction Tree – both the Transaction and the *<transaction set>* windows display.

You can toggle between the two views by selecting the presentation that is not currently being used from the View menu on the current window. Both views have the same menu options, but they appear in different places on the two windows. In each view, the number and name of the current transaction is part of the caption at the top of the window. (The following explanations for adding and modifying segments, elements, and codes use the Transaction Table view.)

Administrative tasks for EDI messages

❖ Adding a new segment for use in your customized transaction

1 Activate the Segment tab.

2 Choose File>New.

The New Segment window displays.

If you're using the Transaction Tree view, choose Segment>New.

3 Enter the following on the New Segment window:

- Segment – Select the up arrow to select a segment from the Select Segment window
- Sequence Number – Enter the position of the new segment in the transaction.
- Description – a textual description of the new segment. It is auto-populated by the program from the segment code selected. You can modify this field. Changes made to this field will be saved with this segment within this transaction.
- Requirement has three options:
 - Mandatory segment – must occur at this point in the sequence.
 - Optional segment – may occur.
 - Conditional segment – required only if a specific condition exists.
- Max Seg Occurrence – the maximum number of times that the new segment can occur at this point if it does not start a loop.

The following information pertains when this segment starts a loop:

- Loop ID – the identifier for the loop started by the new segment.
- Level – the nested depth of this loop within this message and within the loops of this message.
- Max Loop Occurrence – the maximum number of times this loop can occur.

Use the following for implementation guides only (for compliance checking in particular) and not for full X12 standards:

- Element – identifies the element position within this segment whose value (entered in Value) can cause this segment to start this loop.
 - Sub-Element – identifies the sub-element within this element whose value (entered in Value) can cause this segment to start a loop.
 - Value – the value in the Element/Sub-Element specified above that can cause this segment to start a loop.
- 4 Select Next to add another segment
- or
- OK if you are finished adding segments.

❖ **Modifying a segment to use in your customized transaction**

- Highlight the segment and choose Edit>Properties. from the Edit menu. (n the Transaction Tree view, highlight the segment on the <transaction set> window and choose Properties from the Edit menu.

The Segment Properties window displays.

On the Segment Properties window, you can change the same information you entered on the New Segment window.

❖ **Adding a new element**

- 1 Highlight the segment on the Segments tab of the Create Transaction: <transaction set> window.
- 2 Make the Elements tab active and choose New from the File menu. In the Transaction Tree view, highlight the segment and choose New from the Element menu on the <transaction set> window. The New Element window displays.
- 3 Enter the following information on the New Element window:
 - Segment – the identifier for the segment to which you are adding an element and is auto-populated by the program from the currently selected segment.
 - Segment Description – a brief textual description of the segment associated with the segment identifier and is auto-populated by the program from the currently selected segment.
 - Element – the position of the new element within the list of elements in the segment.

- Sub-Element – an additional element identifier used to create a composite, or compound, element.
- Element Number – the numeric identifier of the EDI element. You can enter a valid element number or select the up arrow to select from a list. The Select Element window displays. Double-click an element to select it from the list on the window.

Double-click an element and the program automatically populates the Element Number, Element Type, Maximum Repetitions, Minimum Length, and Requirement text boxes. You can change the values in the individual text boxes.

- Element Type – displayed by the program, based on the element number selected.
 - Element Description – a brief textual description of the element associated with the element identifier and is populated by the program from the currently selected element.
 - Maximum Repetitions is the maximum number of times this element can appear in this segment.
 - Minimum Length – the minimum length of this element and is displayed by the program, based on the element number selected.
 - Maximum Length – the maximum length of this element and is displayed by the program, based on the element number selected.
 - Requirement has three options:
 - A Mandatory element must contain data.
 - An Optional element may or may not contain data.
 - A Conditional element is required only if a specific condition exists.
- 4 Once you have entered all the necessary information, select Next to add another element or OK to return to the Create Transaction: *<transaction set>* window.

❖ **Modifying an element**

- 1 Highlight the element on the Elements tab of the Create Transaction: *<transaction set>* window and choose Properties from the Edit menu. In the Transaction Tree view, highlight the element on the *<transaction set>* window and choose Properties from the Element menu. The Element Properties window displays.

- 2 On the Element Properties window, you can change the same information you entered on the New Element window. Select Next to modify another element or OK to return to the Create Transaction: *<transaction set>* window.

❖ **Adding a new code to the full code or implementation code list**

- 1 Highlight the element on the Elements tab of the Create Transaction: *<transaction set>* window.
- 2 Make the Full Code List or Implementation Code List tab active and choose New from the File menu.

In the Transaction Tree view, highlight the element on the *<transaction set>* window and choose Full Code List or Implementation Code List from the Code List menu.

Either the Full Code List or Implementation Code List window appears. On this window, choose New from the File menu. Enter:

- Element – the number and description of the EDI element to which you are adding a code. When you enter an element, the program auto-populates the following text box with the description that corresponds to the element.
 - Code – the value of the EDI element code you are adding to the element. When you enter a code, the program auto-populates the following text box with the description that corresponds to the code.
- 3 Select Next to add another code or OK to return to the Create Transaction: *<transaction set>* window.

❖ **Modifying a code in the full code or implementation code list**

- 1 Highlight the element on the Elements tab of the Create Transaction: *<transaction set>* window.
- 2 Make the Full Code List or Implementation Code List tab active and choose Properties from the Edit menu.

In the Transaction Tree view, highlight the element on the *<transaction set>* window and choose Full Code List or Implementation Code List from the Code List menu. Either the Full Code List or Implementation Code List window appears. On this window, choose Properties from the Edit menu.

Either the Element Code Properties or the Implementation Code Properties window displays.

- 3 On the Element Code Properties or the Implementation Code Properties window, change the same information you entered on the New Element Code or New Implementation Code window.

❖ **Selecting the segments to be included in the EDI message**

- 1 Choose Create Transaction from the Build menu on the main EMap window.

If you are using the default or chose the tabular presentation, the Create Transaction: *<transaction set>* window displays. If you chose the tree presentation, the *<transaction set>* window displays.

- 2 In the tabular format, the Create Transaction: *<transaction set>* window has five tabs - Transactions, Segments, Elements, Full Code List, and Implementation Code List. The Segments tab is active when the window is opened since customizing a transaction set or message involves selecting the segments to be included in the transaction set/message.

- 3 On the Create Transaction: *<transaction set>* window, right-click the segments you want to include in your transaction. You can also use the two choices on the Options menu:

- To select all of the segments in a transaction, choose Check All.
- To deselect all of the currently selected segments (except those that are mandatory), choose Erase Checkmarks.

- 4 To create your transaction with the segments displayed in a tree presentation, choose Standard Tree from the View menu.

The Transaction window and *<transaction set>* window (in tree presentation) display side-by-side.

- 5 The transaction for the current map is selected on the Transaction window and the segments for that transaction display on the *<transaction set>* window. Select the check box next to each segment you want to include in your customized transaction.

- 6 In the tree view, use the Segment menu choices to select all of the segments in a transaction (Check All) or deselect all of the currently selected segments (Erase Checkmarks). You can switch back to the tabular format by choosing Standard Table from the View menu.

❖ **Making the transaction**

- 1 Choose Make Transaction from the File menu on the Create Transaction: *<transaction set>* window if you are in the Transaction Table view, or the *<transaction set>* window if you are in the Transaction Tree view.

If you previously created this transaction, you are asked to confirm that you want to overwrite the original transaction.

- 2 If you answer No, the flow tables are not overwritten, and you return to Create Transaction: window.

If you answer Yes, new flow tables are generated. A second message box appears, asking you whether you want to overwrite the earlier flow tables.

Note Although you usually choose to overwrite the flow tables, there may be instances where you would like to preserve the original flow.

The Mapping window displays.

- 3 If you are mapping an inbound EDI message, the segments and elements in the EDI template you created display on the right side of the Mapping window – under the section labeled Source: X12 (or Source: HL7 or Source: EDIFACT).
- 4 If you are mapping an outbound EDI message, the segments and elements in the EDI template you created display on the left side of the Mapping window – under the section labeled Destination: X12. (or Destination: HL7 or Destination: EDIFACT).

Working with the standards library

This section provides instructions for working in the standards library.

Note Working in the standards library makes changes to an instance of the standard itself. Once changes are made, the standard itself is changed. Deleting or modifying an element, transaction, or segment in the library function deletes this instance from the actual standards. To restore the original transaction, manually reenter it or reinstall the entire version of this standard from the EMap installation CD.

❖ Entering the standards library

Sometimes you need to alter the standards library to add a new transaction, segment, element, or code, or modify or delete a transaction, segment, element, or code. For example, the HL7 Standard allows users to create “Z” segments.

To change the standards, you must be in the standards library. There are two ways to enter the library:

- 1 Choose Transaction Table or Transaction Tree from the View menu on the main EMap window. If you choose:
 - Transaction Table – the tabular presentation of the standards library displays.
 - Transaction Tree – the library displays in a tree presentation.
- 2 Choose Create Transaction from the Build menu on the main EMap window and enter the standards library from the Create Transaction: *<transaction set>* window. If the default standards view is:
 - Transaction Table – the standards library displays in a tabular presentation.
 - Transaction Tree – the standards library displays in a tree presentation.

❖ **Accessing the standards library**

- 1 In the Transaction Table view, choose Library from the Options menu on the Create Transaction: *<transaction set>* window, then choose View from the Library submenu.
- 2 Enter the standards library. The Create Transaction: *<transaction set>* window looks the same as it does when you are creating a customized EDI message. However, when you are creating a transaction, only the segments, elements, and codes associated with the current map display on the tabs of the window. When you are editing the standards library, all of the transactions, segments, elements, and codes in the standard display.

Any changes in these screens, regardless of whether you are in the library view or not, affects the entire standard.

The menu options at the top of the window change based on which tab is active:

- When the Transactions tab is active, you can perform transaction-related actions. See the following section entitled Transaction – Transaction Table View for instructions on performing transaction-related actions from the Create Transaction: *<transaction set>* window.

- When the Segments tab is active, you can perform segment-related actions. See the following section entitled Segments – Transaction Table View for instructions on performing segment-related actions from the Create Transaction: <transaction set> window.
- When the Elements tab is active, you can perform element-related actions. See the following section entitled Element – Transaction Table View for instructions on performing element-related actions from the Create Transaction: <transaction set> window.
- When the Full Code List tab is active, you can perform full code-related actions. See the following section entitled Code lists – Transaction Table View for instructions on performing code-related actions from the Create Transaction: <transaction set> window.
- When the Full Code List tab is active, you can perform implementation code-related actions. See the following section entitled CODE LISTS – Transaction Table View for instructions on performing code-related actions from the Create Transaction: <transaction set> window.

Administering transactions in the transaction table view

The Transactions tab on the Create Transaction: <transaction set> window contains five menu options:

- File
- Edit
- View
- Reports
- Help

When the Transactions tab is active, you can add, modify, delete, or search for a transaction; change to the tree view of the standards library; or create two transaction-related reports.

❖ Adding a new transaction

- 1 Choose New from the File menu on the Create Transaction: <transaction set> window when the Transaction tab is active. The New Transaction window displays.
- 2 Enter the following information on the New Transaction window:

- Transaction Number – the code that identifies this transaction set or message.
 - Transaction Code – the 2-character group-level code for this transaction set or message.
 - Transaction Description – a brief textual description of the transaction set or message.
- 3 Once you have entered all the information, select OK to return to the Create Transaction: <transaction set> window.

❖ **Modifying a transaction**

- 1 Select a transaction and choose Properties from the Edit menu on the Create Transaction: <transaction set> window when the Transaction tab is active. The Transaction Properties window displays.
- 2 On the Transaction Properties window, you can change the same information you entered on the New Transaction window.
 - Transaction Number – the code that identifies this transaction set or message.
 - Transaction Code – the 2-character group-level code for this transaction set or message.
 - Transaction Description – a brief textual description of the transaction set or message.
- 3 Once you have made your changes, select OK to return to the Create Transaction: <transaction set> window.

❖ **Deleting a transaction**

- 1 Choose Delete from the Edit menu on the Create Transaction: <transaction set> window when the Transaction tab is active.
- 2 When you choose Delete, the program asks you to confirm that you want to delete the transaction before it is permanently removed. If you select:
 - Yes – the transaction is permanently removed from the standard, and you return to the Create Transaction: <transaction set> window.
 - No – the transaction is not deleted, and you return to the Create Transaction: <transaction set> window.

❖ **Finding a transaction**

- 1 Choose Find or Find Next from the Edit menu on the Create Transaction: <transaction set> window when the Transaction tab is active. The Find dialog box displays.
- 2 In the Find dialog box, enter the text you want to find and select Find to search for the first occurrence and Find Next to search for additional occurrences, or select Cancel to return to the Create Transaction: <transaction set> window.

❖ **Changing to the transaction tree view of the standards library**

- Choose Standard Tree from the View menu on the Create Transaction: <transaction set> window. When you change views, you automatically exit from the standards library and must reenter in the new view.

❖ **Producing transaction-related reports**

- When the Transaction tab is active, you can create two reports from the Reports menu on the Create Transaction: <transaction set> window. You can print the Selected Transaction Template report or the Transaction Listing report from the Reports menu. See the Reports chapter for an explanation of these reports.

Segments – transaction table view

The Segments tab on the Create Transaction: <transaction set> window contains six menu options—File, Edit, View, Options, Reports, and Help. When the Segments tab is active, you can add, modify, delete, or search for a segment; change to the tree view of the standards library; or create two segment-related reports.

❖ **Adding a new segment**

- 1 Choose New from the File menu on the Create Transaction: <transaction set> window when the Segments tab is active. The New Segment window displays.
- 2 Enter the following on the New Segment window:
 - Segment – the EDI segment identifier.
 - Segment Description – a brief textual description of the new segment that you are adding to the library.

- 3 Select Next to add another segment to the library, or OK if you are finished. You return to the Create Transaction: *<transaction set>* window, and the new segment appears on the window.

❖ **Modifying a segment**

- 1 Highlight a segment and choose Properties from the Edit menu on the Create Transaction: *<transaction set>* window when the Segments tab is active. The Segment Properties window displays.
- 2 All of the elements in the highlighted segment are listed. From the menus on the Segment Properties window, add, modify, delete, or search for a new element to the segment, as well as open the Element Library window.

❖ **Deleting a segment**

- 1 Select a segment and choose Delete from the Edit menu on the Create Transaction: *<transaction set>* window when the Segments tab is active.
- 2 When you choose Delete, the program asks you to confirm that you want to delete the segment before it is permanently removed. If you select:
 - Yes – the segment is permanently removed from the standard, and you return to the Create Transaction: *<transaction set>* window.
 - No – the segment is not deleted, and you return to the Create Transaction: *<transaction set>* window.

❖ **Finding a segment**

- 1 Choose Find or Find Next from the Edit menu on the Create Transaction: *<transaction set>* window when the Segments tab is active. The Find dialog box displays.
- 2 In the Find dialog box, enter the text you want to find and select Find to search for the first occurrence of the text and Find Next to search for additional occurrences.

Select Cancel to return to the Create Transaction: *<transaction set>* window.

❖ **Changing to the transaction tree view of the standards library**

- Choose Standard Tree from the View menu on the Create Transaction: *<transaction set>* window.

When you change views, you automatically exit from the standards library and must reenter in the new view.

❖ **Producing segment-related reports**

- When the Segments tab is active, you can create two reports from the Reports menu on the Create Transaction: <transaction set> window - the Selected Segment report and the Make Transaction report. See the Reports chapter for an explanation of these reports.

Elements – transaction table view

The Elements tab contains six menu options—File, Edit, View, Options, Reports, and Help. (When you are not in the standards library, there are only five options. The Reports option is no longer present.) When the Elements tab is active, you can add, modify, delete, or search for an element; or change to the tree view of the standards library.

❖ **Adding a new element**

- 1 Choose New from the File menu on the Create Transaction: <transaction set> window when the Elements tab is active. The New Element window displays.
- 2 Enter the following information on the New Element window:
 - Under Element:
 - The first text box contains the numeric identifier of the new element.
 - The second text box contains a brief textual description of the new element
 - Under Options:
 - Minimum specifies the least number of characters that can make up the element (minimum length).
 - Maximum specifies the most number of characters that can make up the element (maximum length).
 - Type is the EDI-specific data type of the element. Choose from the following drop-down list of choices:
 - AN – Alphanumeric
 - ID – Identifier code
 - DT – Date
 - TM – Time

- R – Real number
 - N0 to N9 – Implied decimal (0 – 9 decimal places)
- 3 Select Next to add another element to the library or OK if you are finished adding elements to the library. You return to the Transaction: *<transaction set>* window, and the new element appears on the window.

❖ **Modifying an element**

- 1 Select an element and choose Properties from the Edit menu on the Create Transaction: *<transaction set>* window when the Elements tab is active. The Element Properties window displays.
- 2 On the Element Properties window, you can change the same information you entered on the New Element window.
- 3 Select Next to modify another element or OK if you are finished modifying elements in the library. You return to the Create Transaction: *<transaction set>* window, and the modified element appears on the window.

❖ **Deleting an element**

- 1 Select an element and choose Delete from the Edit menu on the Create Transaction: *<transaction set>* window when the Elements tab is active.
- 2 When you choose Delete, the program asks you to confirm that you want to delete the element before it is permanently removed. If you select:
 - Yes – the element is permanently removed from the standard, and you return to the Create Transaction: *<transaction set>* window.
 - No – the element is not deleted, and you return to the Create Transaction: *<transaction set>* window.

❖ **Finding an element**

- 1 Choose Find or Find Next from the Edit menu on the Create Transaction: *<transaction set>* window when the Elements tab is active. The Find dialog box displays.
- 2 In the Find dialog box, enter the text you want to find, and select Find to search for the first occurrence of the text and Find Next to search for additional occurrences. Select Cancel to return to the Create Transaction: *<transaction set>* window.

❖ **Changing to the transaction tree view of the standards library**

- Choose Standard Tree from the View menu on the Create Transaction: <transaction set> window.

When you change views, you automatically exit from the standards library and must reenter in the new view.

Code lists – transaction table view

Both the Full Code List tab and the Implementation Code List tab contain five menu options on the Full Code List tab – File, Edit, View, Reports, and Help. When either tab is active, you can add a code, modify a code, delete a code, or search for a code for a specific element; change to the tree view of the standards library; or create two code-related reports.

❖ **Adding a new code to the full code or implementation code list**

- 1 Highlight the element on the Elements tab of the Create Transaction: <transaction set> window.
- 2 Make the Full Code List or Implementation Code List tab active and choose New from the File menu.

Either the New Element Code or the New Implementation Code window displays

- 3 Enter the following information on the New Element Code/New Implementation Code window:
 - Element – the number and description of the EDI element to which you are adding a code.
 - Code – the EDI element code you are adding to the element.
- 4 Select Next to add another code or OK to return to the Create Transaction: <transaction set> window.

❖ **Modifying code in the full code or implementation code list**

- 1 Highlight the element on the Elements tab of the Create Transaction: <transaction set> window.
- 2 Make the Full Code List or Implementation Code List tab active and choose Properties from the Edit menu. Either the Element Code Properties or the Implementation Code Properties window displays.

- 3 On the Element Code Properties or the Implementation Code Properties window, you can change the same information you entered on the New Element Code or New Implementation Code window.

❖ **Deleting code from the full code or implementation code list**

- 1 Highlight the element on the Elements tab of the Create Transaction: <transaction set> window.
- 2 Make the Full Code List or Implementation Code List tab active and choose Delete from the Edit menu.

When you choose Delete, the program asks you to confirm that you want to delete the code before it is permanently removed.

- If you select Yes, the code is permanently removed from the selected element in the library, and you return to the Create Transaction: <transaction set> window.
- If you select No, the code is not deleted, and you return to the Create Transaction: <transaction set> window.

❖ **Finding code in the full code or implementation code list**

- 1 Choose Find or Find Next from the Edit menu on the Create Transaction: <transaction set> window when either the Full Code List tab or the Implementation Code List tab is active. The Find dialog box displays.
- 2 In the Find dialog box, enter the text you want to find, and select:
 - Find to search for the first occurrence of the text
 - Find Next to search for additional occurrences.
- 3 Select Cancel to return to the Create Transaction: <transaction set> window.

❖ **Changing to the transaction tree view of the standards library**

- Choose Standard Tree from the View menu on the Create Transaction: <transaction set> window.

When you change views, you automatically exit from the standards library and must reenter in the new view.

❖ **Producing a code-related report**

- You can produce one report when either the Full Code List tab or the Implementation Code List tab is active. The Element Values report includes detailed information about the values in the elements in the full code list. See the Reports chapter for an explanation of this report.

The standards library – transaction tree view

In the Transaction Tree view, you enter the standards library from different menus for transactions, segments, elements, and codes.

When you first enter the standards library in the Transaction Tree view, the Transaction window and the *<transaction set>* window display side-by-side.

The *<transaction set>* window looks the same as it does when you are creating a customized EDI message in the Transaction Tree view. However, when you create a transaction, you modify the segments, elements, and codes only in the transaction you are creating. When you are in the standards library, you are making changes to the standards themselves.

To perform actions on:

- Transactions in the standards library – use the menu options at the top of the Transaction window. See the following section entitled Transaction – Transaction Tree View for instructions on performing transaction-related actions from the Transaction window.
- Segments in the standards library – use the menu options at the top of the Segment Library window. See the following section entitled Segments – Transaction Tree View for instructions on accessing and performing segment-related actions from the Segment Library window.
- Elements in the standards library – use the menu options at the top of the Element Library window. See the following section entitled Elements – Transaction Tree View for instructions on accessing and performing element-related actions from the Element Library window.
- Codes in the standards library – use the menu options at the top of the Full Code List or Implementation Code List window. See the following section entitled Codes – Transaction Tree View for instructions on accessing and performing code-related actions from the Full Code List or Implementation Code List window.

Transactions – transaction tree view

The Transaction window displays when you first enter the standards “Library” in the Transaction Tree view. There are four menu options on the Transactions window - File, Edit, Reports, and Help. On this window, you can add, modify, delete, or search for a transaction; or create two transaction-related reports. On the *<transaction set>* window, you can change to the table view of the standards library.

❖ Adding a new transaction

- 1 Choose New from the File menu on the Transaction window. The New Transaction window displays.
- 2 Enter the following on the New Transaction window:
 - Transaction Number – the code that identifies this transaction set or message.
 - Transaction Code – the 2-character group-level code for this transaction set or message.
 - Transaction Description – a brief textual description of the transaction set or message.
- 3 After you enter all the information, select OK to return to the Transaction window.

❖ Modifying a transaction

- 1 Select a transaction and choose Properties from the Edit menu on the Transaction window. The Transaction Properties window displays.
- 2 On the Transaction Properties window, you can change the same information you entered on the New Transaction window.
 - Transaction Number – the code that identifies this transaction set or message.
 - Transaction Code – the 2-character group-level code for this transaction set or message.
 - Transaction Description – a brief textual description of the transaction set or message.
- 3 After you make your changes, select OK to return to the Transaction window.

❖ Deleting a transaction

- 1 Select a transaction and choose Delete from the Edit menu on the Transaction window.
- 2 When you choose Delete, the program asks you to confirm that you want to delete the transaction before it is permanently removed. If you select:
 - Yes – the transaction is permanently removed from the standard, and you return to the Transaction window.
 - No – the transaction is not deleted, and you return to the Transaction window.

❖ **Finding a transaction**

- 1 Select a transaction and choose Find or Find Next from the Edit menu on the Transaction window. The Find dialog box displays.
- 2 In the Find dialog box, enter the text you want to find, and select:
 - Find to search for the first occurrence
 - Find Next to search for additional occurrences
- 3 Select Cancel to return to the Transaction window.

❖ **Changing to the transaction table view of the standards library**

- Choose Standard Table from the View menu on the <transaction set> window. When you change views, you automatically exit from the library and must reenter from the new view.

❖ **Producing transaction-related reports**

- You can create two reports from the Reports menu on the Transaction window. You can print the Selected Transaction Template report or the Transaction Listing report from the Reports menu. See the Reports chapter for an explanation of these reports.

Segments – transaction tree view

You perform segment-related actions in the standards library, from the Segment Library window.

❖ **Accessing the Segment Library window**

- 1 Choose Library from the Segment menu on the <transaction set> window. The Segment Library window displays.
- 2 Your four menu options are:
 - File
 - Edit
 - Reports
 - Help

From this window, you can add, modify, delete, or search for a segment; or create two segment-related reports. On the <transaction set> window, you can change to the table view of the standards library.

❖ Adding a new segment

- 1 Choose New from the File menu on the Segment Library window. The New Segment window displays.
- 2 Enter the following on the New Segment window:
 - Segment – the EDI segment code.
 - Segment Description – a brief textual description of the new segment that you are adding to the library.
- 3 Select Next to add another segment to the library or OK if you are finished adding segments. You return to the Segment Library window, and the new segment appears on the window.

❖ Modifying a segment

- 1 Select the segment and choose Properties from the Edit menu on the Segment Library window. The Segment Properties window displays.
- 2 All of the elements in the highlighted segment are listed. From the menus on the Segment Properties window, you can add, modify, delete, or search for an element in the segment, and open the Element Library window.

❖ Deleting a segment

- 1 Select the segment and choose Delete from the Edit menu on the Segment Library window.
- 2 When you choose Delete, the program asks you to confirm that you want to delete the segment before it is permanently removed from the standards library. If you select:
 - Yes – the segment is permanently removed from the library, and you return to the Segment Library window. The segment no longer displays on the window.
 - No – the segment is not deleted, and you return to the Segment Library window. The segment still displays on the window.

❖ Finding a segment

- 1 Choose Find or Find Next from the Edit menu on the Segment Library window. The Find dialog box displays.
- 2 In the Find dialog box:
 - a Enter the text you want to find
 - b Select Find to search for the first occurrence of the text

- c Find Next to search for additional occurrences.
 - 3 Select Cancel to return to the Segment Library window.
- ❖ **Changing to the transaction table view of the standards library**
 - Choose Standard Table from the View menu on the <transaction set> window. When you change views, you automatically exit from the library and must reenter in the new view.
 - ❖ **Producing segment-related reports**
 - You can create two reports from the Reports menu on the Segment Library window – Segment Library report and the Segment Cross-Reference report. See the Reports chapter for an explanation of these reports.

Elements – transaction tree view

You perform element-related actions in the standards library, from the Element Library window.

- ❖ **Accessing the Element Library window**
 - 1 Choose Library from the Element menu on the <transaction set> window. The Element Library window displays.
 - 2 There are five menu options on the Element Library window:
 - File
 - Edit
 - Code List
 - Reports
 - Help

On the Element Library window, you can add, modify, delete, or search for an element. On the <transaction set> window, you can change to the table view of the standards library.
- ❖ **Adding a new element**
 - 1 Choose New from the File menu on the Element Library window. The New Element window displays.
 - 2 Enter the following information on the New Element window:
 - Under Element:

- The first text box contains the numeric identifier of the new element.
 - The second text box contains a brief textual description of the new element
 - Under Options:
 - Minimum specifies the least number of characters allowed in an element (minimum length).
 - Maximum specifies the most number of characters allowed in an element (maximum length).
 - Type is the data type of the element. Choose from the drop-down list of choices:
 - AN Alphanumeric
 - ID Identifier code
 - DT Date
 - TM Time
 - R Real Number
 - N0 – N9 Decimal Explicit – “0” – “9” decimal places
- 3 Select Next to add another element to the library or OK if you are finished adding elements. You return to the Element Library window, and the new element appears on the window.

❖ **Modifying a element**

- 1 Select an element and choose Properties from the Edit menu on the Element Library window. The Element Properties window displays.
- 2 On the Element Properties window, you can change the same information you entered on the New Element window.
- 3 Select Next to modify another element in the library or OK if you are finished modifying elements in the library.
- 4 Return to the Element Library window, and the modified element appears on the window.

❖ **Deleting an element**

- 1 Select the element and choose Delete from the Edit menu on the Element Library window.

- 2 When you choose Delete, the program asks you to confirm that you want to delete the element before it is permanently removed. If you select:
 - Yes – , the element is permanently removed from the standard, and you return to the Element Library window. The element that you removed is no longer displayed.
 - No – the element is not deleted, and you return to the Element Library window.

❖ **Finding an element**

- 1 Choose Find or Find Next from the Edit menu on the Element Library window. The Find dialog box displays.
- 2 In the Find dialog box:
 - a Enter the text you want to find
 - b Select Find to search for the first occurrence of the text
 - c Select Find Next to search for additional occurrences.
- 3 Select Cancel to return to the Element Library window.

❖ **Changing to the transaction table view of the standards library**

- Choose Standard Table from the View menu on the *<transaction set>* window. When you change views, you automatically exit from the library and must reenter in the new view.

Code lists – transaction tree view

You perform code-related actions in the standards library from either the Full Code List or the Implementation Code List window. Even in the library, a code list is always related to a specific element.

❖ **Accessing the full code list or the implementation code list**

- 1 Highlight an element

Choose Full Code List or Implementation Code List from the Code List menu on the *<transaction set>* window.
- 2 The Full Code List or Implementation Code List window displays.
- 3 There are four menu options on both list windows:
 - File

- Edit
- Reports
- Help

From either window, you can add a code, modify a code, delete a code, or search for a code for a specific element in the library or create a code-related report.

On the *<transaction set>* window, you can change to the table view of the standards library. (The Full Code Lists windows are used as examples in the following explanation of code-related actions.)

❖ **Adding a new code to the full code or implementation code list**

- 1 Highlight the element on the *<transaction set>* window.
- 2 On either the Full Code List or the Implementation Code List window, choose New from the File menu.

Note Adding codes to the library is different from adding transactions, segments, and elements, because codes are always added for a specific element, whereas transactions, segments, and elements can be added to the standards library in general.

Either the New Element Code or the New Implementation Code window displays.

- 3 Enter the following information on the New Element Code/New Implementation Code window:
 - Element – the number and description of the EDI element to which you are adding a code.
 - Code – the EDI element code you are adding to the element.
- 4 Select Next to add another code or OK to return to the Full Code List or the Implementation Code List window.

❖ **Modifying code in the full code or implementation code list**

- 1 Highlight the element on the *<transaction set>* window.
- 2 Select a code and choose Properties from the Edit menu on the Full Code List or the Implementation Code List window. Either the Element Code Properties or the Implementation Code Properties window displays.

- 3 On the Element Code Properties or the Implementation Code Properties window, you can change the following information:
 - Element – the number and description of the EDI element to which you are adding a code.
 - Code – the EDI element code you are adding to the element.
- 4 Select Next to modify another code or OK to return to the Full Code List or the Implementation Code List window.

❖ **Deleting code from the full code or implementation code list**

- 1 Highlight the element on the Elements tab of <transaction set> window.
- 2 Select a code and choose Delete from the Edit menu on the Full Code List or the Implementation Code List window.
- 3 When you choose Delete, the program asks you to confirm that you want to delete the code before it is permanently removed. If you select
 - Yes – the code is permanently removed from the selected element in the library, and you return to the <transaction set> window.
 - No – the code is not deleted, and you return to the Full Code List or the Implementation Code List window.

❖ **Finding code in the full code or implementation code list**

- 1 Choose Find or Find Next from the Edit menu on Full Code List or the Implementation Code List window. The Find dialog box displays.
- 2 In the Find dialog box:
 - a Enter the text you want to find
 - b Select Find to search for the first occurrence of the text
 - c Select Find Next to search for additional occurrences.
- 3 Select Cancel to return to the Full Code List or the Implementation Code List window.

❖ **Changing to the transaction table view of the standards library**

- Choose Standard Table from the View menu on the <transaction set> window. When you change views, you automatically exit from the library and must reenter in the new view.

❖ **Producing a code-related report**

- You can produce one report from either the Full Code List tab or the Implementation Code List window. The Element Values report includes detailed information about the values in the elements in the full code list. See the Reports chapter for an explanation of this report.

Mapping

About this chapter

This chapter discusses mapping in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	174
Transaction mapping	180
Assigning levels	223
Any-to-any mapping	225

Introduction

For your map to convert data from one format into another, you must provide specific field-by-field instructions on how to convert the data. You do this on the transaction Mapping window or the Any-to-Any Map window.

In EMap, there are two types of maps:

- Any-to-any maps and
- Transaction maps

In any-to-any maps, data from one application is reformatted to be used directly by another application. In transaction maps, data reformatting takes place between an application and an EDI message.

Outbound transaction maps have application data as the source on the left side of the Mapping window and EDI standards data as the destination on the right-hand side. Inbound transaction maps have EDI standards data as the source on the left-hand side of the Mapping window and application data as the destination on the right-hand side. The Any-to-Any Map window has application data on both sides of the window, as both source and destination data.

The mapping mechanism is very different for transaction maps and any-to-any maps. In any-to-any maps, all mapping takes place in a rule - each field is mapped by an individual command in one mapping rule. The source data can be a record field, a memory variable, or a system variable. In transaction maps, mapping takes place in an entirely different way. EMap uses its built-in knowledge of standards, together with information that you supply about the application data, to link EDI elements to record fields. In addition to mapping the data, you can manipulate the data and test it. You can use memory variables, conditional variables, system variables, rules, and cross reference tables.

Actions from the Transaction Mapping window

Segment actions

From the transaction Mapping window, you can perform the following actions on segments:

- Insert a selected segment into the transaction that is being mapped; delete a segment from the transaction; create a simple copy of a segment in the transaction; create a copy of a segment in the transaction that includes all associated mapping, or find a segment.

- Ignore (not map) only the unmapped elements in the current segment or every unmapped element in the transaction.
- Perform conditional mapping on segments or make them trading partner-specific.
- View the segment detail.
- Assign levels to segments. (Levels are discussed in the Map Flow chapter.)
- Enter informational notes for a specific segment.

Element actions

From the transaction Mapping window, you can perform the following mapping-related actions on elements:

- Create a simple copy of an element in a segment; create a copy of an element in a segment, including all associated mapping, cross-reference tables, and rules; delete an element from a segment; or find an element.
- Map an EDI element to a field in a record or to a memory variable. You can also add a new memory variable or modify or delete a memory variable.
- Ignore (not map) or unmap an element in the current segment.
- Create or remove an association between a cross-reference table and an element. You can also add a new cross-reference table, modify a cross-reference table, or select a table.
- Attach a rule to an element or remove a rule from an element. You can specify that a rule be performed before or after the element is mapped.
- Perform conditional mapping on elements or make them trading partner-specific.
- Enter informational notes for a specific element.
- Update the mapping for all elements to include any changes that have been made to a record field or memory variable to which the element is mapped.
- Simultaneously map an EDI element to a field, while creating the field based on the element.
- Copy mapping instructions from one element to another.
- Add codes to an element or view the codes associated with an element.
- Perform any of the following options on an element:
 - Zero-fill the data in the element.
 - Make the element mandatory.

- Truncate the data in the element.
- Justify the data in the element.
- “Save and paste” data into the element.

Application data
actions

From the transaction Mapping window, you can perform the following mapping-related actions on application data:

- Add a new record, or modify or delete a record.
- Add a new field, or modify or delete a field.

Actions from the Any-to-Any Map window

On the Any-to-Any Map window, you can map one of the following to a field in an output application record:

- Field in an input application record.
- System variable.
- Memory variable.

When you map fields that are different data types, ECTMap automatically makes the conversion for you. When a DI (Decimal Implicit) field is mapped, ECTMap first converts the field to a DE (Decimal Explicit) before performing mapping activities.

Transaction mapping and any-to-any mapping

In this chapter, transaction mapping and any-to-any mapping are discussed separately. First, transaction mapping is broken down into inbound transaction maps and outbound transaction maps. (There are slight differences between the mapping-related actions available in each and the way in which some actions are performed.) Within the two types of transaction map, there is a detailed explanation of actions related to EDI segments, actions related to EDI elements, actions related to the application data, mapping actions, and the “Find” function. After both types of transaction maps are discussed, any-to-any mapping is explained.

You can get to the transaction Mapping window or the Any-to-Any Map window from the main EMap window either by selecting on the toolbar or on Work Space or by choosing Mapping from the Build menu. If the current map is a transaction map, the transaction Mapping window displays. If the current map is an any-to-any map, the Any-to-Any Map window displays. In addition, the Mapping window is automatically displayed after you create a transaction.

The Mapping window for transaction maps is divided into three sections – Source, Destination, and Mapping – with Source data on the left, Destination data on the right, and the Mapping section at the bottom.

For the Mapping section at the bottom of the window, a description for each column is listed below.

Table 13-1: Mapping window description

Column	Description
Level	Hierarchical number that is used in map flow
Segment	EDI Standards identifier for the segment
Element	The position of the element within the segment
Sub-Element	Position of the component within the element
Repeat No.	If an element can repeat, this number indicates which number is in the sequence of the repeating element
Record	The name of the record that contains the field that is mapped to the element
Field Name	Name of the field mapped to the element
Type	Data type of the element
Requirement	Specifies the status of the segment M – a mandatory segment must occur at this point in the transaction. O – an optional segment may occur at this point in the transaction. C – a conditional segment is required only if a specific condition exists. N – a segment is not used because the standards were created from an implementation guide R – a segment is required because the standards were created from an implementation guide S – a segment may occur if the standards were created from an implementation guide.
Minimum	Specifies the minimum length of the element as defined in the standard
Maximum	Specifies the maximum length of the element as defined in the standard
Number	Numeric identifier of the EDI element
Description	Name of the EDI element
Rule Type	The type of rule to be performed, for example Prior or Post
Rule	Number of the rule attached to an element
Table	Name of the cross-reference table associated with an element
Condition 1	When you set a condition for a segment in the Condition 1 field, the condition is shown in this field
Condition 2	When you set a condition for a segment in the Condition 2 field, the condition is shown in this field
TP Type	If a segment or element is linked to a specific trading partner, the “Trading Partner Type” (blank or NOT) is shown in this field
TP	If a segment or element is linked to a trading partner, the name of the trading partner is shown in this field

Column	Description
Notes	If you have entered informational notes about an element, they are shown in this field
Mandatory	Specifies whether the mapping is mandatory for the application field
Truncate	Specifies whether EMap truncates data (without issuing a warning) when the field length to which data is mapped is shorter than the field length from which it is mapped. (This is a good choice if you know truncation will occur and do not want to see warnings in your log.)
Zero Fill	If you choose Zero Fill, EMap adds leading zeroes to a field and output a zero if the application field is a numeric zero
Justification	<p>For Justification, you can make one of three choices:</p> <ul style="list-style-type: none"> • Left Justification – indicates that data begins in the leftmost position of the record field. • Left Justification without Trim – indicates that data begins without removing any leading spaces in the leftmost position of the record field. • Right Justification – indicates that data ends in the rightmost position of the record field. <p>If you choose None (the default justification), EMap uses default justification for the record field. The default justification for alpha-numeric fields is left-justification, and the default justification for numeric fields is right-justification.</p>

The Any-to-Any Map window is divided into four sections. There is a Source data section on the left, and a Destination data section on the right. Rather than extending all the way across the bottom of the window, the Mapping section is between the Source and Destination sections at the bottom. Above the Mapping section is a split window that allows you to view and perform intermediate mapping steps.

Transaction mapping

On the top part of the transaction mapping window, EDI standard data and application data are side by side - with source data on the left and destination data on the right. All of the data is presented in a tree structure that allows you to select a level to expose the level beneath - revealing data mapping, levels, rules, cross reference tables and conditions for EDI data and the record/table names for application data. In a tabular presentation at the bottom of the window, the Mapping section displays the data linkages and associated levels and rules as mapping takes place.

The Mapping section includes all of the segments in the transaction set or message you created, with each element displayed on a separate line. There are also fields for EDI data, such as segment, element, data type, and length; application data, such as record name, field name, and attributes; and trading partner data to be displayed as mapping takes place, as well as fields for levels and rules. As you map the source data to the destination data, each record and field of the application data appears on the appropriate line at the bottom of the window, next to the segment and element to which it is linked in the map. As you assign levels and add rules to levels or elements, these also appear on the appropriate lines in the Mapping section. (Refer to the Rules chapter for instructions concerning rules and the Map Flow chapter for instructions concerning levels.)

Working with inbound map segments

On inbound transaction maps, Source: <Standard> data is on the left side and Destination: Application Data is on the right. To access segment-related actions, highlight a segment and use the Segment menu at the top of the window; highlight a segment and use the File and Edit menus at the top of the window; or right-click a segment and use the drop-down list that displays.

When you right-click a segment in the Source: <standard> data section, you can select and insert a new segment into the current transaction; create and insert into the current transaction a simple copy of a segment; create and insert into the current transaction a copy of a segment that includes all associated mapping, rules, and cross-reference tables; delete a segment in the current transaction; ignore (do not map) either unmapped elements in the highlighted segment or all unmapped elements in the current transaction; make the segment trading partner-specific in the current transaction; or enter informational notes for the segment. (Condition is grayed out because it is not an option for segments in an inbound transaction map.)

When you highlight a segment on the Mapping window and use the Segment menu, you have the same set of menu choices as when you right-click the segment.

When you highlight a segment and use the File menu and Edit menus at the top of the window, you have some of the same choices and one additional one. When you use the File menu, you can create and insert a new segment. When you use the Edit menu, you can create and insert a simple copy of a segment, create and insert a copy of a segment with all associated mapping actions, or delete a segment. In addition, you can search for a segment from the Edit menu.

Adding, copying, and deleting segments

❖ Adding a new segment

1 Do one of the following:

- Choose New from the Segment menu at the top of the window.
- Right-click a segment and choose New from the drop-down list.

The Select a Segment window displays.

2 Drag a segment from the Select a Segment window and drop it onto the Source pane.

If you drop the new segment onto:

- A level, the new segment is inserted at the very end of the level.
- An existing segment, the new segment is inserted directly after the existing segment.

❖ Copying a segment

• Do one of the following:

- Highlight the segment and choose Edit>Copy>Segment.
- Highlight a segment and choose Segment>Copy.
- Right-click a segment and choose Copy from the drop-down list.

A copy of the segment is created and inserted directly following the segment that was highlighted.

❖ Copying a segment with all mapping-related data

• Do one of the following:

- Highlight a segment and choose Edit>Copy with Data>Segment.

- Highlight a segment and choose Segment>Copy with Data.
- Right-click a segment and choose Copy with Data from the drop-down list.

A copy of the segment is created and inserted after the segment that was highlighted.

Double-click the segment to display the mapping activity that was copied.

❖ **Deleting a segment**

- Do one of the following:
 - Highlight the segment and choose Edit>Delete>Segment.
 - Highlight the segment and choose Segment>Delete.
 - Right-click a segment and choose Delete from the drop-down list.

You are asked to confirm that you want to delete the segment before it is permanently removed.

Ignoring unmapped elements

Use the Ignore option after you finish all of your element-to-field mapping. Ignore specifies that unmapped elements are not to be mapped. The All option on the Ignore submenu causes all unmapped elements in all segments to not be mapped. The Current option on the Ignore submenu specifies that only unmapped elements in the currently high-lighted segment should not be mapped. When an element is ignored, IGNVAR appears in the record column on the Mapping section at the bottom of the Mapping window and IGNORE appears in the Field Name column.

❖ **Ignoring unmapped elements in the current segment or in all segments**

- Do one of the following:
 - Highlight the segment and choose Segment>Ignore>Current.
 - Right-click the segment and choose Ignore>Current from the drop-down list.

❖ **Ignoring all unmapped elements in all segments**

- Do one of the following:
 - Choose Ignore from the Segment menu and then choose All from the Ignore submenu.

- Right-click a segment, choose Ignore from the drop-down list that displays, and then choose All from the Ignore submenu.

Linking segments and trading partners

The Trading Partner option allows you to specify that a segment should be mapped only for a particular trading partner.

This allows you to use the same map for multiple trading partners. When the main body of mapping is the same, you can tailor the mapping of specific segments to each trading partner.

❖ Linking a segment to a specific trading partner

- 1 Highlight the segment and choose Segment>Trading Partner.

or

Right-click the segment and choose Trading Partner from the drop-down list.

The Select Segment Trading Partner window displays.

- 2 Select the Trade Partner Type from the drop-down list. The choices are:
 - Blank – The segment is used only for the trading partner listed in the Trading Partner Name.
 - NOT – The segment is used for all trading partners except for the trading partner listed in the Trade Partner Name.
- 3 For the Trade Partner Name, do one of the following:
 - Accept the All Trading Partner, which is the default.
 - Choose another trading partner by selecting the up arrow and choosing a trading partner from the Trading Partners window:
- 4 Select OK to return to the Mapping window.

To view the linkage that you have created, you must choose Segment Detail from the View menu.

Viewing and annotating segments

❖ Viewing segment detail

- Select a segment and choose View>Segment Detail.

The Segment window displays. You can view details about the segment and perform actions on it using the menu options at the top of the window.

❖ **Changing the loop ID view**

- To toggle between displaying the loop ID for each segment, choose View>Loop IDs.

Creating notes

The Notes option allows the user to enter brief notes related to the selected segment. The notes are used for documentation purposes and are informational only. They show up on the Map Implementation Guide report and when you choose Segment Detail on the View menu.

❖ **Entering notes about a segment**

- 1 Do one of the following:
 - Highlight the segment and choose Segment>Notes.
 - Right-click the segment and choose Notes from the drop-down list.

The Notes for Segment window displays.

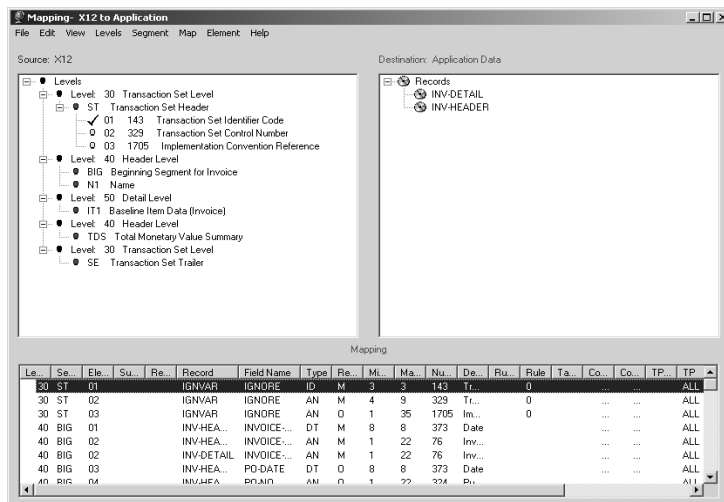
- 2 Enter the information about the segment in the text box at the bottom of the window.

A maximum of 80 characters can be entered on the window. For convenient display, a maximum of twenty characters is suggested.

- 3 Select OK to return to the Mapping window.

Working with inbound map elements

On inbound transaction maps, Source: <Standard> data is on the left side of the window and Destination: Application Data is on the right.



You can access element-related actions on the Mapping window in several ways. You can highlight an element and use the Element, File, or Edit menu at the top of the window, or you can right-click an element and choose from the drop-down list that displays. Some actions are available by only one of these methods, while others can be performed in more than one way.

Using the right-click menu

When you right-click an element in the Source: <standard> data section, a drop-down list gives you choices of:

- Mapping the element to a field in the output application record
- Mapping the element to a memory variable
- Mapping the element to a special memory variable called a conditional variable
- Copying mapping from one element to another
- Adding element codes to an element
- Ignoring the element
- Automatically creating a field in the output record based on the element: associating a rule with the element
- Linking the element to a cross-reference table

- Attaching a condition to the element; making the element trading-partner-specific
- Entering notes for the segment. (Constant is grayed out because it is not an option for elements in an inbound transaction map.)

Using the Element menu

When you highlight an element and use the Element menu, you have some of the same choices that the drop-down list gave you, as well as one additional choice.

However, from the Element menu, you cannot map the element or use the element to automatically create a field in the output. You can associate a rule with the element; link the element to a cross-reference table; attach a condition to the element; make the element trading-partner-specific; and enter informational notes for the element.

You can also:

- Create and insert into a segment a simple copy of an element
- Create and insert into a segment a copy of an element that includes all associated mapping, rules, cross-reference tables, and associated conditions
- Delete an element from a segment; update mapping for the element to include any changes that have been made to a record field or memory variable since the element was mapped
- Make the element mandatory
- Zero-fill the data in the element, justify it, or truncate it; save the contents of an element and paste it into another element.

Using the File or Edit menu

When you highlight an element and use the File and Edit menus at the top of the Mapping window, you have some of the same choices and one additional one.

When you use the:

- File menu – you can create and insert a new element.
- Edit menu – you can create and insert a simple copy of an element, create and insert a copy of an element with all associated mapping actions, or delete a segment. In addition, you can search for an element.

Adding, copying, and deleting elements

❖ Inserting a new element on the Mapping window

- Highlight an element and choose File>New>Element.

A copy of the highlighted element is inserted directly after the highlighted element.

❖ Copying an element

- Do one of the following:
 - Highlight the element and choose Edit>Copy.
 - Highlight the element and choose Element>Copy.

A copy of the element is created and inserted directly following the element that was copied.

❖ Copying an element with all map-related data

- Do one of the following:
 - Highlight the element and choose Edit>Copy with Data.
 - Highlight the element and choose Element>Copy with Data.

A copy of the element is created and inserted directly following the element that was copied.

❖ Deleting an element

- Do one of the following:
 - Highlight the element and choose Edit>Delete.
 - Highlight the element and choose Element>Delete.

You are asked to confirm that you want to delete the element before it is permanently removed.

Note EMap does not allow you to delete an element if it is the only one present in a segment. If you attempt to do this, an error message displays.

Adding element conditions

The Condition option on the Element menu allows you to attach a condition to an element that controls whether or not the element is mapped. On the Condition window, specify which record fields or memory variables are to be checked against what other record fields, memory variables, or constants before mapping the element.

If one or more of the conditions is dependent upon the value of data in another element, you must make that other element a conditional variable. For more information about conditional variables, see “Mapping to a conditional variable” on page 195.

You can set up one or two conditions. If you set up two conditions, both must be true for the mapping to occur. This is the equivalent of linking the two conditions with a Boolean AND.

❖ Attaching a condition to an element

- 1 Do one of the following:
 - Highlight the element and choose Element>Condition.
 - Right-click the element and choose Condition from the drop-down list.

The Element Conditions window displays.
- 2 Select the top the up arrow to choose the type of data to be evaluated in the condition. Choices are:
 - Record/Field to select field from the Records/Tables window. The program enters the record and field names into the top text boxes.
 - Memory Variable to select a memory variable from the Memory Variables window. The program enters MEMVAR into the top text box and the name of the memory variable into the next text box.
 - Clear Condition to remove the information entered in the condition.
- 3 Select the middle up arrow to choose the operator to be used in the condition. Choices are:
 - EQ – equal to
 - NE – not equal to
 - LT – less than
 - LE – less than or equal to

- GT – greater than
 - GE – greater than or equal to
- 4 Under Condition 1 (or 2), select the bottom up arrow to choose the type of data against which the data in the condition is evaluated. If you choose:
 - Record/Field to select field from the Records/Tables window. The program enters the record and field names into the bottom text boxes.
 - Memory Variable to select a memory variable from the Memory Variables window. The program enters MEMVAR into the bottom text box and the name of the memory variable into the next text box.
 - Constant – the Constant window displays. Enter the constant value in the Constant text box.
 - 5 Select OK, and the program enters STRVAR and the constant value into the bottom two text boxes.

Adding and removing element rules

The Rule option on the Element menu of the Mapping window allows you to attach a rule to an element. There are three choices on the Rule submenu:

- Prior rules – performed before the element is mapped.
- Post rules – performed after the element is mapped. Post rules are generally used for inbound maps.
- Remove – removes a rule that was previously attached to the element.

On inbound transaction maps, rules attached to an element are not performed if the element does not contain data.

❖ Attaching a rule to an element

- 1 Do one of the following:
 - Highlight the element, choose Element>Rule>Prior or Element>Rule>Post.
 - Right-click the element, choose Rule>Prior or Rule>Post from the drop-down list.

The Rule Definitions pane is superimposed over the application data.

- 2 Drag a rule and drop it onto the element.

- ❖ **To remove a rule from an element**
 - Right click the element and choose Rule>Remove drop-down list.

Linking elements and trading partners

The Trading Partner option allows you to link the mapping of an element to a particular trading partner.

- ❖ **Linking an element to a specific trading partner**
 - 1 Do one of the following:
 - Highlight the element and choose Element>Trading Partner from the Element menu or
 - Right-click the element and choose Trading Partner from the drop-down list.
 - The Select Element Trading Partner window displays.
 - 2 Select the Trading Partner Type. The choices are:
 - blank – the element is used for only the trading partner entered in the Trading Partner Name text box below.
 - NOT – the element is used for all trading partners except for the trading partner entered in the Trading Partner Name text box below.
 - 3 For Trading Partner Name, do one of the following:
 - Accept the All Trading Partner, which is the default.
 - Choose another trading partner by selecting the up arrow and choosing a trading partner from the Trading Partners window:
 - 4 Select OK to return to the Mapping window.

Updating an element

If an element is mapped to either a record field or a memory variable, the Update option can be used to automatically update the element database to reflect any changes that have been made to either the record field or the memory variable since the element was originally mapped.

This option is usually used after field characteristics, such as size or type, have been modified. When you generate your map and receive an error message saying that things are not in sync, use this option to quickly remap the application data to the EDI elements.

❖ Updating the mapping for an element

- Choose Element>Update.

The program automatically updates the element database.

Adding a cross-reference table to an element

The Cross Reference Table option on the Element menu allows you to associate an element with a cross-reference table that converts source values to destination values.

❖ Associating a cross-reference table with an element

- 1 Perform either of the following:

- Highlight the element and choose Element>Cross Reference Table>New.
- Right-click the element and choose Cross Reference Table>New from the drop-down list.

The New Cross Reference Table displays.

- 2 Enter the File Name (up to 8 characters), application field length, and a brief textual Description:

- 3 Select Create.

The Cross Reference Table Properties window displays.

The program automatically creates the entries for the cross reference table, populating the EDI Field Value with the EDI code values for this element and the Description with the description associated with each EDI code value. To complete the process:

- Select Options>Fill Fields From>Description or Options>Fill Fields From>Description>EDI value.

ECMap populates the Cross Reference Value with either the description or the EDI value based on your choice.

For more information on creating, modifying and deleting cross reference tables, see Chapter 22, “Working with Cross-Reference Tables.”

Annotating an element

The Notes option on the Element menu allows you to enter brief notes related to the selected element. For easy display, use a limit of 20 characters. The notes are used for documentation purposes and are informational only, and show up on the Map Implementation Guide report.

❖ Adding informational notes about an element

- 1 Do one of the following:
 - Highlight the element and choose Element>Notes
 - Right-click an element and choose Notes from the drop-down list.The Notes for Element window displays.
- 2 Enter the information about the element in the text box. You can enter up to 80 characters.
- 3 Select OK to return to the Mapping window.

Applying element options

❖ Applying options for elements

- 1 Highlight the element and choose Options from the Element menu. The option you have chosen is checked on the menu itself.
- 2 You have the following choices of options related to element data:
 - Save
 - Paste
 - Mandatory
 - Truncate
 - Zero Fill
 - Justification

Save and paste

Choose Options>Save and Options>Paste to give you the capability to copy mappings of elements from one group to another. It is often referred to as “Block Save and Paste” because you must block the elements to be saved, save them, and then paste them where they are required.

❖ **Using a block save and paste**

- 1 Press the Ctrl (Control) key while left-clicking the first element in the block. Then press the Shift key while left-clicking the last element in the block. (The block is not highlighted.)
- 2 Right-click the last element and choosing Options>Save from the drop-down list.
- 3 Right-click the element after which you want to paste the element mappings and choose Options>Paste from the drop-down list.

A message displays, asking you whether you want to paste the saved elements.

- 4 Select Yes to complete the action.

Mandatory	Choose Element>Options>Mandatory to identify the element as mandatory, or required, in this segment.
Truncate	Choose Element>Options>Truncate to truncate data (without issuing a warning) when the field length to which data is mapped is shorter than the field length from which it is mapped. (This is a good choice if you know truncation will occur and do not want to see warnings in your log.)
Zero fill	Choose Element>Options>Zero Fill to add leading zeroes to a field and output zero if the application field is a numeric zero.
Justification	Justification – choose one of the following: <ul style="list-style-type: none"> • Element>Options>Justification>Left to begin data in the leftmost position of the record field. • Element>Options>Justification>Left Without Trim to begin data without removing any leading spaces in the leftmost position of the record field. • Element>Options>Justification>Right to end data in the rightmost position of the record field. • Element>Options>Justification>None to use the default justification for the record field. The default justification for alpha-numeric fields is left-justification, and the default justification for numeric fields is right-justification.

Working with inbound map application data

When you right-click a record or field in the Application Data section of the inbound transaction Mapping window, a drop-down list displays that allows you to add, edit, or delete a record or a field in a record.

To perform actions on records, if you highlight a record and select:

- **New** – the New Record window displays. From this window, you can add a new record. (Refer to the Records/Tables chapter for detailed instructions on adding a new record.)
- **Properties** – the Record Properties window displays. From this window, you can modify the highlighted record. (Refer to the Records/Tables chapter for detailed instructions on editing a record.)
- **Delete** – a message displays asking you to confirm that you want to delete the record before it is permanently removed. (Refer to the Records/Tables chapter for detailed instructions on deleting a record.)

To perform actions on field, if you highlight a field and select:

- **New** – the New Field window displays. From this window, you can add a new field. (Refer to the Records/Tables chapter for detailed instructions on adding a new field.)
- **Properties** – the Field Properties window displays. From this window, you can modify the highlighted record. (Refer to the Records/Tables chapter for detailed instructions on editing a field.)
- **Delete** – a message displays asking you to confirm that you want to delete the field before it is permanently removed. (Refer to the Records/Tables chapter for detailed instructions on deleting a field.)

Mapping EDI elements to application data

On the inbound transaction Mapping window, you can perform EDI element to field mapping actions by either:

- Highlighting an element and using the Map menu,
- Right-clicking an element and using the top part of the drop-down list.

You have the same mapping choices on both menus. You can:

- Map an element to a field in a record, to a memory variable, or to a special memory variable called a conditional variable.

- Repeat the mapping of an element or choose to not map (or ignore) an element.
- Have EMap automatically create an output field based on an element – at the same time that the element is being mapped to the newly created field.

Mapping to a field

❖ Mapping an element to a field

- 1 Open both the record and the segment containing the element by double-clicking on them.
- 2 Drag the element and drop it onto the field.

This is commonly known as record/field drag-and-drop mapping.

Mapping to a memory variable

Often, the data to be mapped to an element is held in a user-defined memory variable and must be mapped from there.

❖ Mapping an element to a memory variable

- 1 Expand the segment containing the element.
- 2 Perform one of the following:
 - Right-click the element and select Memory Variable from the drop-down list.
 - Highlight the element and choose Map>Memory Variable.

The Memory Variable pane displays.

- 3 Drag a memory variable from the Memory Variable pane and drop it onto an element.

Mapping to a conditional variable

There are times when the mapping of one EDI element depends upon the value in another (generally related) EDI element. The EDI element whose value determines the mapping of the other EDI element is called a conditional variable. The EDI element whose mapping is determined by the conditional variable has a “condition” or conditions” placed on it.

A conditional variable is a special type of memory variable that is used in conditional mapping. When you specify that an element is a conditional variable, EMap creates a special memory variable field name of XXXNNSS, where XXX is the current Segment ID, NN is a sequence number that gives the position of the element in the segment, and SS is the sub-element. EMap first checks for the existence of the variable name and then creates it if it does not already exist. All field names are case sensitive - a lower case beg01 is not the same variable as an upper case BEG01.

❖ **Mapping an element to a conditional variable**

- Right-click the element containing the data that will be evaluated, and select Conditional Variable from the drop-down menu.

The results of your selection displays in all three sections of the Mapping window:

- In the Mapping section at the bottom of the window on the line containing the element you selected, CNDVAR displays under Record, and the name of the conditional variable under Field Name.
- If you double-click the element in the standards section of the window, you see a mapping icon and the name of the conditional variable that was created. The memory variable created appears in the Destination: Memory Variable pane of the Mapping window.

Repeating an element mapping

The repeat map function allows you to copy the mapping from one element to another. It is often used when a segment contains multiple copies of the same element or collection of elements, and each occurrence of a given element has multiple fields to which it might map. You would conditionally map the first of the multiple elements to its various locations, and then mark the other instances of the same element as a repeat of the first.

❖ **Automatically repeating the mapping of an element**

- 1 Right-click the element to which mapping will be copied, and choose Repeat from the drop-down list.

The Repeat Map window displays.

- 2 Enter the number of the element (and sub-element number and repeat element number if applicable) whose mapping will be copied.
- 3 Select OK.

The mapping will be copied, and you return to the Mapping window.

Note You cannot currently use the repeat map function on the first element in a segment because there is no “Prior Element” whose mapping will be copied.

Displaying code lists

For inbound transaction maps, you can display the code lists for an element from the Mapping window, but there are no actions available.

❖ **Displaying the full code list or the implementation code list**

- Right-click an element, and choose Full Code List or Implementation Code List from the drop-down list.

The Full Code List displays or Implementation Code List displays.

Note The Implementation Code List is available only when you are using an implementation guide standard. Implementation guide standards can be obtained by using the SEF import facility or by purchasing an electronic implementation guide.

Removing element mapping

The Ignore option specifies that an element is not mapped. The Ignore option for element mapping is different from the Ignore option for segments in that it actually unmaps previously mapped elements, allowing it to serve a dual role at this point. When the Ignore option is chosen, IGNVAR appears in the record column on the Mapping section of the Mapping window for the element and IGNORE appears in the Field Name column.

❖ **Ignoring (not mapping) an element**

- Right-click the element, and choose Ignore from the drop-down list.

Creating an element-based field

The Make Field option is a capability in EMap that allows you to simultaneously map and create new fields inside records by dragging elements and dropping them onto a record. In many cases, the data layout – whether it is a flat file or a database table – is not known or is non-existent prior to mapping. In this situation, EMap allows you to create record or table layouts based on the critical elements inside the transaction set or message. This functionality creates fields of the same name, type, and size as the EDI element.

You might need a more powerful Make Field capability when the field definition or name can be derived from the code list description of a different element. For example, the first two elements of the N1 segment are a qualifier and a generic element called Name. The first element in the N1 segment determines the kind of name that appears in the second element.

The Make Field option has a mechanism where the field created in the target record could have the name Ship To if, for example, the qualifier ST informs us that the name is a Ship To name.

This more powerful functionality is enabled by making the qualifier a conditional variable and then selecting conditional mapping (without actually mapping the element) and specifying that this particular element – the name – is conditional on the previous element. In other words, you would set up the condition that N102 is conditionally mapped based on N101 equaling ST. At this point dragging and dropping the element would create the field name Ship To in the target record.

❖ Automatically creating a field based on an EDI element

- 1 You can turn the Make Field switch on and off in two ways:
 - Right-click the element, and choose Make Field from the drop-down list
 - Highlight the element and choose Map>Make Field.

When the Make Field switch is turned on, the only record/field mapping that is allowed is drag-and-drop mapping from an element onto a record.

- 2 Drag and drop the element to a record.

A new record is created using the element name and description.

Searching for a string

You can use the Find command to search for a text string in:

- Record/Field
- Memory Variable
- Mapping
- Code
- Rule
- Cross-Reference Table

The first three items are always available, while the last three are grayed out unless they have been enabled and the associated pane is visible.

To open the Code pane, highlight an element and choose Code from the Map menu.

To open the Rule or Cross-Reference panes, highlight an element and choose Rule or Cross Reference Table from the Element menu.

If the program:

- Does not find the character string, it gives you a message telling you that the character string was not found.
- Finds the character string, it highlights and displays the appropriate record/field, memory variable, code, rule, or cross reference table on the appropriate pane.

❖ **Searching for a character string during inbound mapping**

- Perform these steps when finding searching for a text string:
 - Choose Edit>Find>Record/field – when you choose Record/Field from the Find submenu at the top of the Mapping window, the Find dialog box appears. Enter the criteria for which you are searching, and EMap searches the Application Data to find the search string. If it finds the search string, the program highlights and displays the appropriate record/field in the Application Data section of the Mapping window.

If the Application Data pane is not open, the program automatically opens it when it finds the first instance of the character string. When you select All, the program searches record fields.

- Choose Edit>Find>Memory variable – when you choose Memory Variable from the Find submenu at the top of the Mapping window, the Find dialog box appears. Enter the criteria for which you are searching, and EMap searches the memory variables used in the current map to find the search string. If it finds the search string, the program highlights and displays the appropriate memory variable on the Memory Variable window.

If the Memory Variables data pane is not open, the program automatically opens it when it finds the first instance of the character string. When you select All, the program searches memory variables.

- Choose Edit>Find>Mapping – when you choose Mapping from the Find submenu at the top of the Mapping window, the Find dialog box appears. Enter the criteria for which you are searching, and EMap searches the Mapping section at the bottom of the window to find the search string. If it finds the search string, the program opens the appropriate window and displays the highlighted line(s) containing the search string. When you select All, the program searches mapping.

Select the table view

In order to search for a mapping string, the Mapping window view must be set to Table view (View>Element>Table).

- Choose Edit>Find>Code – when either Full Code or Implementation Code is chosen from the Code submenu of the Map menu, the Full Code List or Implementation Code List pane for the highlighted element is superimposed over the Application Data. If you then choose Code from the Find submenu of the Edit menu, the Find dialog box appears. You enter the criteria for which you are searching, and EMap searches the code list.

If it finds the search string, the program displays on the Full Code List or Implementation Code List pane the code in which the character string was found.

- Choose Edit>Find>Rule – when either Prior or Post is chosen from the Rule submenu of the Element menu, the Rule Definitions pane is superimposed over the Application Data). If you then choose Rule from the Find submenu of the Edit menu, the Find dialog box appears. You enter the criteria for which you are searching, and EMap searches the rules definitions used in the current map.

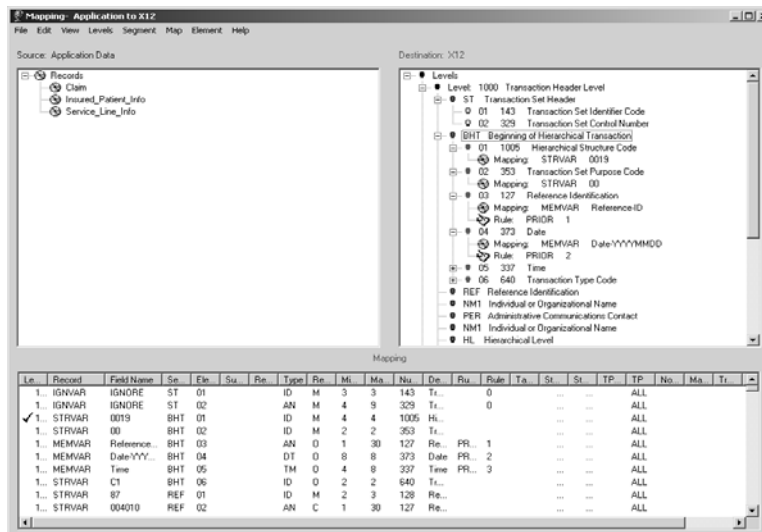
If it finds the search string, the program highlights and displays on the Rules Definitions pane the rule in which the character string was found.

- Choose Edit>Find>Cross reference table – when Select is chosen from the Cross Reference Table submenu of the Element menu, the Cross Reference Table pane is superimposed over the Application Data. If you then choose Cross Reference Table from the Find submenu of the Edit menu, the Find dialog box appears. You enter the criteria for which you are searching, and ECMa searches the cross-reference tables used in the current map.

If it finds the search string, the program highlights on the Cross Reference Tables pane the cross reference table in which the character string was found.

Working with outbound map segments

On outbound transaction maps, Source: Application Data is on the left side of the window and Destination: <standard> data is on the right.



Segment-related mapping actions

You can perform segment-related actions the following ways:

- Highlight a segment and use the Segment, File, and Edit menus on the Mapping window.

Using the right-click menu

- Right-click a segment and choose from the drop-down list that displays.
- When you right-click a segment in the Destination: <standard> data section, a drop-down list allows you to:
- Select and insert a new segment into the current transaction
 - Create and insert into the current transaction a simple copy of a segment
 - Create and insert into the current transaction a copy of a segment that includes all associated mapping, rules, and cross-reference tables
 - Delete a segment in the current transaction
 - Ignore (not map) either the unmapped elements in the segment or all of the unmapped elements in the current transaction
 - Attach a condition to the segment
 - Make the segment trading-partner-specific in the current transaction
 - Enter informational notes for the segment.

Using the Segment menu

When you highlight a segment and use the Segment menu at the top of the Mapping window, you have the same set of choices as when you right-click the segment.

Using the File or edit menu

When you highlight a segment and use the File menu and Edit menus at the top of the window, you have some of the same choices and one additional one. When you use the:

- File menu – you can create and insert a new segment
- Edit menu – you can create and insert a simple copy of a segment, create and insert a copy of a segment with all associated mapping actions, or delete a segment. In addition, you can search for a segment.

Adding, copying, and deleting segments

❖ Adding a new segment

- 1 Do one of the following:
 - Choose Segment>New
 - Right-click a segment and choose New from the drop-down list.

In each case, the Select a Segment window displays.
- 2 Drag a segment from the Select a Segment window and drop it onto the Destination: pane.

If you drop the new segment onto:

- A level, the new segment is inserted at the very end of the level
- An existing segment, the new segment is inserted directly after the existing segment.

❖ **Copying a segment**

- Do one of the following:
 - Highlight the segment and choose Edit>Copy>Segment
 - Highlight a segment and choose Segment>Copy
 - Right-click a segment and choose Copy from the drop-down list

A copy of the segment is created and inserted directly following the segment that was copied.

❖ **Copy a segment with all mapping-related data**

- Do one of the following:
 - Highlight the segment and choose Edit>Copy with Data>Segment
 - Highlight a segment and choose Segment>Copy with Data
 - Right-click a segment and choose Copy with Data from the drop-down list

A copy of the segment is created and inserted directly following the segment that was copied.

Double-click the segment to display the mapping activity that was copied.

❖ **Deleting a segment**

- Do one of the following:
 - Highlight a segment and choose Edit>Delete>Segment
 - Choose Delete from the Segment menu at the top of the window.
 - Right-click a segment and choose Delete from the drop-down list that displays.

You are asked to confirm that you want to delete the segment before it is permanently removed.

Ignoring unmapped elements

The Ignore option is generally used after you have finished all of your element-to-field mapping. Ignore specifies that unmapped elements are not to be mapped. The All option on the Ignore submenu causes all unmapped elements in all segments to not be mapped. The Current option on the Ignore submenu specifies that only unmapped elements in the currently high-lighted segment should not be mapped. When an element is ignored, IGNVAR appears in the record column on the Mapping section at the bottom of the Mapping window and IGNORE appears in the Field Name column.

- ❖ **Ignoring unmapped elements in the current segment or in all segments**
 - Do one of the following:
 - Highlight the segment and choose Segment>Ignore>Current
 - Right-click a segment and choose Ignore>Current from the drop-down list
- ❖ **Ignoring all unmapped elements in all segments**
 - Do one of the following:
 - Highlight the segment and choose Segment>Ignore>All
 - Right-click a segment and choose Ignore>All from the drop-down list

Creating a conditional segment

ECMap allows you to designate that a segment is to be generated only if a specified condition is true.

A condition consists of a left operand, an operator, and a right operand. The value in the left operand is tested against the value in the right operand, using one of the following operators – equal to, not equal to, less than, less than or equal to, “greater than, or greater than or equal to. If the condition is true, the segment is mapped.

- ❖ **Attaching a condition to a segment**
 - 1 Do one of the following:
 - Highlight the segment and choose Segment>Condition
 - Right-click the segment and choose Condition from the drop-down list

The Segment Condition window displays.

- 2 Enter the following on the Segment Condition window:
 - Segment text boxes – the program automatically populates the identifier and description of the segment to which the condition will be attached.
 - Left Operand – contains the data to be evaluated. Select the up arrow and select from the following choices:
 - If you choose Record/Field, the Records/Tables window displays. Double-click a record field to select it. You return to the Segment Condition window and the names of the record and field you selected display in the Left Operand text boxes.
 - If you choose Memory Variable, the Memory Variables window displays. Double-click a memory variable to select it. You return to the Segment Condition window and MEMVAR and the name of the memory variable you selected display in the Left Operand text boxes.
 - If you choose Clear Condition, all the values currently entered in the text boxes on the window are removed. This option removes any conditions previously defined for this segment.
 - Operator – defines how the data is evaluated. Select the up arrow and select from the following choices:
 - EQ – equal to
 - NE – not equal to
 - LT – less than
 - LE – less than or equal to
 - GT – greater than
 - GE – greater than or equal to
 - Right Operand – contains the data against which the data in the Left Operand will be evaluated. Select the up arrow and select from the following choices:
 - If you choose Record/Field, the Records/Tables window displays. Double-click a record field to select it. You return to the Segment Condition window and the names of the record and field you selected display in the Right Operand text boxes.

- If you choose Memory Variable, the Memory Variables window displays. Double-click a memory variable to select it. You return to the Segment Condition window and MEMVAR and the name of the memory variable you selected display in the Right Operand text boxes.
 - If you choose Constant, the Constant window displays. Enter the character string and select OK. You return to the Segment Condition window and STRVAR and the character string you entered display in the Right Operand text boxes.
- 3 Select Next to enter another condition; Select OK to return to the Mapping window.

Linking segments with a trading partner

The Trading Partner option allows you to specify that a segment (and its associated mapping, rules, etc.) should be mapped only for a particular trading partner.

This feature allows you to use the same map for multiple trading partners. When the main body of mapping is the same, you can tailor the mapping of specific segments to each trading partner

❖ Making a segment trading-partner-specific

- 1 Do one of the following:
 - Highlight the segment and choose Segment>Trading Partner
 - Right-click the segment and choose Trading Partner from the drop-down list

The Select Segment Trading Partner window displays.

- 2 Select the Trading Partner Type from the drop-down list:
 - Blank – the segment is used for only the trading partner entered in the Trading Partner Name text box below.
 - NOT – the segment is used for all trading partners except for the trading partner entered in the Trading Partner Name text box below.

For Trading Partner Name, the default is ALL (the All Trade Partner). If you select the up arrow, the Trading Partners window displays.

- 3 For the Trade Partner Name, do one of the following:
 - Accept the All Trading Partner, which is the default.

- Choose another trading partner by selecting the up arrow and choosing a trading partner from the Trading Partners window:
- 4 Select OK to return to the Mapping window.
- To view the linkage that you have created, you must choose Segment Detail from the View menu.

Viewing and annotating segments

❖ Viewing segment detail

- Select a segment and choose View>Segment Detail.

The Segment window displays. You can view details about the segment and perform actions on it using the menu options at the top of the window.

❖ Changing the loop ID view

- To toggle between displaying the loop ID for each segment, choose View>Loop IDs.

Creating notes

The Notes option allows the user to enter brief notes related to the selected segment. The notes are used for documentation purposes and are informational only. They show up on the Map Implementation Guide report and when you choose Segment Detail on the View menu.

❖ Entering notes about a segment

- 1 Do one of the following:
 - Highlight the segment and choose Segment>Notes.
 - Right-click the segment and choose Notes from the drop-down list.

The Notes for Segment window displays.
- 2 Enter the information about the segment in the text box at the bottom of the window.

A maximum of 80 characters can be entered on the window. For convenient display, a maximum of twenty characters is suggested.
- 3 Select OK to return to the Mapping window.

Working with outbound map elements

On outbound transaction maps, Source: Application Data is on the left side of the window and Destination: <standard> data is on the right.

Element-related actions

You can perform element-related actions the following ways:

- Highlight an element and use the Element, File, or Edit menu on the Mapping window
- Right-click an element and choose from the drop-down list that displays.

Some actions are available by only one of these methods, while others can be performed in more than one way.

Using the right-click menu

Right-clicking an element in the Destination: <standard> data section provides a drop-down list in which you can:

- Map the element to a field in the output application record
- Map the element to a memory variable
- Map the element to a special memory variable called a conditional variable
- Copy mapping from one element to another
- Add element codes to an element; ignoring (not mapping) the element
- Automatically create a field in the output record based on the element: associating a rule with the element
- Link the element to a cross-reference table; attaching a condition to the element
- Make the element trading-partner-specific
- Enter informational notes for the segment.

Using the Element menu

Highlighting an element and using the Element menu provides some additional choices. Although you cannot map the element or use the element to automatically create a field in the output, you can

- Associate a rule with the element; link the element to a cross-reference table; attach a condition to the element
- Make the element trading-partner-specific
- Enter informational notes for the segment
- Create and insert into a segment a simple copy of an element

- Create and insert into a segment a copy of an element that includes all associated mapping, rules, and cross-reference tables
- Delete an element from a segment
- Update mapping for the element to include any changes that have been made to a record field or memory variable since the element was mapped
- Make the element mandatory
- Zero-fill the data in the element, justify it, or truncate it
- Save data and paste it into the element.

Using the File or Edit menu

Highlighting an element and using the File and Edit menus at the top of the Mapping window also allows you perform the following.

When you use the:

- File menu – create and insert a new element.
- Edit menu – create and insert a simple copy of an element, create and insert a copy of an element with all associated mapping actions, or delete a segment.

You can also search for an element.

Adding, copying, and deleting elements

❖ Inserting a new element

- Highlight the element and choose File>New>Element

A copy of the highlighted element is inserted directly after the highlighted element.

❖ Copying an element

- Do one of the following:
 - Highlight the element and choose Edit>Copy
 - Highlight the element and choose Element>Copy

A copy of the element is created and inserted directly following the element that was copied.

❖ Copying an element with all map-related data

- Do one of the following:
 - Highlight the element and choose Edit>Copy with Data

- Highlight the element and choose Element>Copy with Data.

A copy of the element is created and inserted directly following the element that was copied.

❖ **Deleting an element**

- Do one of the following:
 - Highlight the element and choose Edit>Delete
 - Highlight the element and choose Element>Delete

You are asked to confirm that you want to delete the element before it is permanently removed.

Note You cannot delete an element if it is the only one present in a segment.

Adding element conditions

When a Condition is attached to an element in an outbound map, the program checks to see whether data was mapped to that EDI element and if so, maps a specified constant value to one or two other related EDI elements. The Condition option is generally used in conjunction with a Conditional Variable and is sometimes used with the Repeat Map function.

The EDI element into which the constant value is stored is specified as the Conditional Variable, a special memory variable to which EMap gives the name XXXNNSS, where:

- XXX – the current segment ID
- NN – a sequence number that gives the position of the field name in the segment
- SS – the sub-element

When you make an element a conditional variable, EMap checks to see if the name already exists for this element, and creates one if it does not exist. The Condition is attached to the EDI element that will be checked for the presence of data, or which is mapped to a field that will be checked for the presence of data. The two EDI elements are generally related and follow one another in a segment – such as a qualifier and its associated data - and there can be the possibility of multiple pairs occurring.

When multiple pairs are possible, you use the Repeat Map function to copy the mapping of the first pair to as many additional pairs as might exist in the application data.

During mapping, the program keeps track of whether a prior pair has been mapped, and if so, adjusts the mapping position for the next pair accordingly. The value of this option is that qualifiers that have no data are not mapped.

You can set up one or two conditional mappings that specify the conditional variable that is mapped if the element contains data and the constant value that is stored in the conditional variable.

❖ **Attaching a condition to an element**

1 Do one of the following:

- Highlight the element and choose Condition from the Element menu
- Right-click the element and choose Condition from the drop-down list that displays.

The Element Conditions window displays.

2 For each condition you want to set:

- Select the top up arrow and the Memory Variables window displays. Double-click a conditional variable. (Remember that a conditional variable is a special type of memory variable.). You return to the Element Conditions window, and the program enters MEMVAR and the name of the conditional variable into the top two text boxes.
- Select the middle up arrow and the program enters EQ in the middle text box on the Element Conditions window. This is an “equals” operator.
- Select the bottom up arrow and the Constant window displays. Enter a character string in the text box and, and select OK. The program enters STRVAR and the constant value into the bottom two text boxes on the Element Conditions window.

Adding and removing element rules

The Rule option allows you to attach a rule to an element. There are three options on the Rules submenu—Prior, Post, and Remove. Prior rules are performed before the element is mapped, and Post rules are performed after the element is mapped. Use Remove to remove a rule that was previously attached to the element. Post rules are generally used for inbound maps.

Note On outbound transaction maps, rules attached to an element are not performed if the field that is mapped to the element does not contain data.

❖ Attaching a rule to an element

- 1 Do one of the following:
 - Highlight the element and choose Element>Rule>Prior or Element>Rule>Post
 - Right-click the element and choose Rule>Prior or Rule>Post from the drop-down list

The Rule Definitions pane displays.

- 2 Drag the rule and drop it onto the element.

❖ Removing a rule from an element

- Right-click the element, choose Rule>Remove from the drop-down list

Linking elements and trading partners

❖ Linking an element to a specific trading partner

The Trading Partner option allows you to specify that an element (and its associated mapping, rules, etc.) will be mapped for a particular trading partner.

- 1 Do one of the following:
 - Highlight the element and choose Element>Trading Partner
 - Right-click the element and choose Trading Partner from the drop-down list.

The Select Element Trading Partner window displays.

- 2 Select a Trade Partner Type. The choices are:
 - blank – the element is used for only the trading partner entered in the Trading Partner Name text box below.

- NOT – the element is used for all trading partners except for the trading partner entered in the Trading Partner Name text box below.)
- 3 For Trading Partner Name, do one of the following:
 - Accept the All Trading Partner, which is the default
 - Choose another trading partner by selecting the up arrow and choosing a trading partner from the Trading Partners window.
 - 4 Select OK to return to the Mapping window.

Updating an element

If an element is mapped to either a record field or a memory variable, the Update option can be used to automatically update the element database to reflect any changes that have been made to either the record field or the memory variable since the element was originally mapped.

Use this option after field characteristics, such as size or type, have been modified. When you generate your map and receive an error message saying that things are not in sync, you exercise this option to quickly remap the application data to the EDI elements.

❖ Updating the mapping for an element

- Choose Element>Update.

The program automatically updates the element database.

Adding a cross-reference table to an element

The Cross Reference Table option on the Element menu lets you associate an element with a cross-reference table to convert source values to destination values.

❖ Associating a cross-reference table with an element

- 1 Do one of the following
 - Highlight the element and choose Element>Cross Reference Table>New
 - Right-click the element and choose Cross Reference Table>New from the drop-down list.
- 2 Enter the File Name (up to 8 characters) and a brief textual Description:
- 3 Select Create.

The program automatically creates the entries for the cross reference table, populating the EDI Field Value with the EDI code values for this element and the Description with the description associated with each EDI code value.

- 4 To complete the automatic creation of the cross-reference table, use the Options menu of the Cross Reference Tables window.

ECMap populates the Cross Reference Value with either the description or the EDI value.

For more information on creating, modifying and deleting cross reference tables, see Chapter 22, “Working with Cross-Reference Tables.”

Annotating an element

The Notes option on the Element menu allows the user to enter brief notes related to the selected element. The notes are used for documentation purposes and are informational only. They show up on the Map Implementation Guide report.

❖ Adding informational notes about an element

- 1 Do one of the following:
 - Highlight the element and choose Element>Notes
 - Right-click an element and choose Notes from the drop-down list.

The Notes for Element window displays.

- 2 Enter the information about the element in the text box.

A maximum of 80 characters can be entered. For convenient display, a maximum of twenty (20) characters is recommended.

- 3 Select OK to return to the Mapping window.

Applying element options

❖ Applying options for elements

- 1 Highlight the element and choose Element>Options from the Element menu. The option you have chosen is checked.
- 2 You have the following choices of options related to element data:
 - Save

	<ul style="list-style-type: none"> • Paste • Mandatory • Truncate • Zero Fill • Justification
Save and paste	<p>Choose Options>Save and Options>Paste to give you the capability to copy mappings of elements from one group to another. It is often referred to as “Block Save and Paste” because you must block the elements to be saved, save them, and then paste them where they are required.</p> <p>❖ Using a block save and paste</p> <ol style="list-style-type: none"> 1 Press the Ctrl (Control) key while left-clicking the first element in the block. Then press the Shift key while left-clicking the last element in the block. (The block is not highlighted.) 2 Right-click the last element and choosing Options>Save from the drop-down list. 3 Right-click the element after which you want to paste the element mappings and choose Options>Paste from the drop-down list. A message displays, asking you whether you want to paste the saved elements. 4 Select Yes to complete the action.
Mandatory	Choose Element>Options>Mandatory to identify the element as mandatory, or required, in this segment.
Truncate	Choose Element>Options>Truncate to truncate data (without issuing a warning) when the field length to which data is mapped is shorter than the field length from which it is mapped. (This is a good choice if you know truncation will occur and do not want to see warnings in your log.)
Zero fill	Choose Element>Options>Zero Fill to add leading zeroes to a field and output zero if the application field is a numeric zero.
Justification	<p>Justification – choose one of the following:</p> <ul style="list-style-type: none"> • Element>Options>Justification>Left to begin data in the leftmost position of the record field. • Element>Options>Justification>Left Without Trim to begin data without removing any leading spaces in the leftmost position of the record field.

- **Element>Options>Justification>Right** to end data in the rightmost position of the record field.
- **Element>Options>Justification>None** to use the default justification for the record field. The default justification for alpha-numeric fields is left-justification, and the default justification for numeric fields is right-justification.

Working with outbound map application data

When you right-click a record or field in the Source: Application Data section of the outbound transaction Mapping window, a drop-down list displays that allows you to add, edit, or delete a record or a field in a record.

Performing actions on records

If you highlight a record and select:

- **New** – the New Record window displays. From this window, you can add a new record.
- **Properties** – the Record Properties window displays. From this window, you can modify the highlighted record.
- **Delete** – a message displays asking you to confirm that you want to delete the record before it is permanently removed.

For more information on creating, editing, and deleting records, see Chapter 9, “Working with Records and Tables.”

Perform actions in fields within records

If you highlight a field and select:

- **New** – the New Field window displays. From this window, you can add a new field.
- **Properties** – the Field Properties window displays. From this window, you can modify the highlighted record.
- **Delete** – a message displays asking you to confirm that you want to delete the field before it is permanently removed.

For more information on creating, editing, and deleting fields, see Chapter 9, “Working with Records and Tables.”

Mapping EDI data to application data

On the outbound transaction Mapping window, perform EDI element to field mapping actions by either:

- Highlighting an element and using the Map menu
- Right-clicking an element and using the top part of the drop-down list

You have the same mapping choices on both menus. You can map an element to a field in a record, to a memory variable, or to a special memory variable called a conditional variable. You can repeat the mapping of an element or choose to not map (or ignore) an element. You can even have EMap automatically create an output field based on an element – at the same time that the element is being mapped to the newly created field.

Mapping an field

❖ Mapping an element to a field in a record

- 1 Double-click to expand the record and segment to display the field and element you wish to map.
- 2 Drag the element and drop it onto the field.

Mapping a memory variable

Often, the data to be mapped to an element is held in a user-defined memory variable and must be mapped from there.

❖ Mapping an element to a memory variable

- 1 Expand the segment containing the element.
- 2 Perform one of the following:
 - Right-click the element and select Memory Variable from the drop-down list
 - Highlight the element and choose Map>Memory Variable

The Memory Variable pane displays.

- 3 Drag a memory variable from the Memory Variable pane and drop it onto an element.

Mapping a conditional variable

There are times when the mapping of one EDI element depends upon the value in another (generally related) EDI element. The EDI element whose value determines the mapping of the other EDI element is called a Conditional Variable. The EDI element whose mapping is determined by the conditional variable has a Condition placed on it.

A conditional variable is a special memory variable to which ECTMap gives the name XXXNNSS, where XXX is the correct segment ID, NN is a sequence number that gives the position of the field name in the segment, and SS is the sub-element (if it exists). When you make an element a conditional variable, ECTMap checks to see if the name already exists for this element and creates one if it does not exist. (All field names are case sensitive - a lower case beg01 is not the same variable as an upper case BEG01.)

The program checks to see whether data was mapped to an EDI element to which a Condition is attached and if so, maps a specified constant value to one or two related EDI elements that have been specified as Conditional Variables.

❖ Mapping an element to a conditional variable

- Right-click the element containing the data to will be evaluated and select Conditional Variable from the drop-down list.

You see the results of your selection in all three sections of the Mapping window:

- In the Mapping section at the bottom of the window on the line containing the element you selected, CNDVAR is displayed under Record and the name of the conditional variable under Field Name.
- If you double-click the element in the standards section of the window, you see a mapping icon and the name of the conditional variable that was created. The memory variable created appears in the Destination: Memory Variable pane of the Mapping window.

Repeating an element mapping

The repeat map function allows you to copy the mapping from one element to another. It is often used when a segment contains multiple copies of the same element or collection of elements, and each occurrence of a given element has multiple fields to which it might map. You would conditionally map the first of the multiple elements to its various locations, and then mark the other instances of the same element as a repeat of the first. The Repeat Map function is often used with a Conditional Variable and the Condition option.

❖ Automatically repeating the mapping of an element

- 1 Right-click the element to which mapping will be copied, and choose Repeat from the drop-down list.

The Repeat Map window displays.

- 2 Enter the number of the element (and sub-element number and repeat element number if applicable) whose mapping will be copied.
- 3 Select OK.

The mapping will be copied, and you return to the Mapping window.

Note You cannot currently use the repeat map function on the first element in a segment because there is no “Prior Element” whose mapping will be copied.

Displaying code lists

For outbound transaction maps, you can display the code lists for an element from the Mapping window, but there are no actions available.

❖ Displaying the code list for an element

- Right-click an element and choose Full Code List or Implementation Code List from the drop-down list.

The Full Code List or the Implementation Code List displays.

Note The Implementation Code List is available only when you are using an implementation guide standard. Implementation guide standards can be obtained by using the SEF import facility or by purchasing an electronic implementation guide.

Mapping a constant

There are times when a constant value must be mapped to an element in an outbound map.

❖ Mapping a constant to an element

- 1 Perform one of the following:
 - Right-click the element and choose Constant from the drop-down list

- Highlight the element and choose Map>Constant.

A Constant pop-up window displays.

- 2 Enter the value or string and select OK.

This option is available only on outbound maps. On inbound maps, you must use a rule to put a constant into an application field

Removing element mapping

The Ignore option specifies that an element is not mapped. The Ignore option for element mapping is different from the Ignore option for segments in that it actually unmaps previously mapped elements, allowing it to serve a dual role at this point. When the Ignore option is chosen, IGNVAR appears in the record column on the Mapping section of the Mapping window for the element and IGNORE appears in the Field Name column.

❖ Ignoring (not mapping) an element

- Right-click the element, and choose Ignore from the drop-down list.

Creating an element-based field

The Make Field option is a capability in ECMap that allows you to simultaneously map and create new fields inside records by dragging elements and dropping them onto a record. In many cases, the data layout – whether it is a flat file or a database table – is not known or is non-existent prior to mapping. In this situation, ECMap allows you to create record or table layouts based on the critical elements inside the transaction set or message. This functionality creates fields of the same name, type, and size as the EDI element.

There are times when a more powerful Make Field capability is needed. This occurs when the field definition or name can be derived from the code list description of a different element.

For example, the first two elements of the N1 segment are a qualifier and a generic element called Name. The first element in the N1 segment determines the kind of name that appears in the second element. The Make Field option has a mechanism whereby the field created in the target record could have the name Ship To if, for example, the qualifier ST informs us that the name is a Ship To name.

This more powerful functionality is enabled by making the qualifier a conditional variable and then selecting conditional mapping (without actually mapping the element) and specifying that this particular element – the name – is conditional on the previous element. In other words, you would set up the condition that N102 is conditionally mapped based on N101 equaling ST. At this point dragging and dropping the element would create the field name Ship To in the target record.

❖ **Automatically creating a field based on an EDI element**

1 Do one of the following to Toggle the Make Field switch:

- Right-click the element and choose Make Field from the drop-down list
- Highlight the element and choose Map>Make Field.

When the Make Field switch is turned on, the only record/field mapping that is allowed is drag-and-drop mapping from an element onto a record.

2 Drag and drop the element on a record.

A new record is created using the element name and description.

Searching for a string

You can use the Find command to search for a text string in a:

- Record/Field
- Memory Variable
- Mapping
- Code
- Rule
- Cross-Reference Table

The first three items are always available, while the last three are grayed out unless they have been enabled and the associated pane is visible.

To open the Code pane, highlight an element and choose Code from the Map menu.

To open the Rule or Cross-Reference panes, highlight an element and choose Rule or Cross Reference Table from the Element menu.

If the program:

- Does not find the character string, it gives you a message telling you that the character string was not found.
- Finds the character string, it highlights and displays the appropriate record/field, memory variable, code, rule, or cross reference table on the appropriate pane.

❖ **Finding a record/field**

- 1 Choose Edit>Find>Record/Field.

The Find dialog box appears.

- 2 Enter the criteria and choose Find.

ECMap searches the Application Data to find the search string. If it finds the search string, the program highlights and displays the appropriate record/field in the Application Data section of the Mapping window.

❖ **Finding a memory variable**

- 1 Choose Edit>Find>Memory Variable

The Find dialog box appears.

- 2 Enter the criteria and choose Find.

ECMap searches the memory variables used in the current map to find the search string. If it finds the search string, the program highlights and displays the appropriate memory variable on the Memory Variable window. (If the Memory Variables data pane is not open, the program automatically opens it when it finds the first instance of the character string.) When you select All, the program searches memory variables.

❖ **Finding a mapping**

- 1 Choose Edit>Find>Mapping

The Find dialog box appears.

- 2 Enter the criteria and choose Find.

ECMap searches the Mapping section at the bottom of the window to find the search string. If it finds the search string, the program opens the appropriate window and displays the highlighted line(s) containing the search string. When you select All, the program searches mapping.

❖ **Finding a code**

- 1 Choose Edit>Find>Full Code or Edit>Find>Implementation Code

The Find dialog box appears.

- 2 Enter the criteria and choose Find.

ECMap searches the code list. If it finds the search string, the program displays on the Full Code List or Implementation Code List pane the code in which the character string was found.

❖ **Finding a rule**

- 1 Choose Edit>Find>Rule

The Find dialog box appears.

- 2 Enter the criteria and choose Find.

ECMap searches the rules definitions used in the current map. If it finds the search string, the program highlights and displays the rule in which the character string was found.

❖ **Finding a cross-reference table**

- 1 Choose Edit>Find>Cross Reference Table

The Find dialog box appears.

- 2 Enter the criteria and choose Find.

ECMap searches the cross-reference tables used in the current map. If it finds the search string, the program highlights on the Cross Reference Tables pane the cross reference table in which the character string was found.

Assigning levels

The Assign option on the Levels menu is used to assign levels to the EDI data. EDI Standards group segments into different levels—normally header, detail, and summary. For example, in X12 transaction sets, the data in Table 1 is header level data, the data in Table 2 is detail level data, and the data in Table 3 is summary level data. The levels you define in ECMap should be based on the EDI standard for the specific message that you are mapping.

In ECMap, levels reflect the relationship between EDI segments and the application data being mapped. As a result, the levels you assign to segments in ECMap may correspond to the levels used in the Standards or they may be different. Levels also define the context for interpreting data at a given point in time.

The level-related actions and menu options are the same on the outbound Mapping window and the inbound Mapping window.

❖ **To assign a level to a segment**

- 1 Select Level>Assign

The Levels window and Segment window display side by side. If you have already created levels, they display on the Levels window.

- 2 Do one of the following:

- Drag a level from the Levels window and drop it onto a segment on the Segments window
- Drag a segment and drop it onto a level.

You can also highlight a block of segments, drag them to the Levels window, and drop them onto a level.

Performing Levels window actions

On the Levels window, you can add a new level, or modify or delete a level.

❖ **Adding a new level**

- 1 Choose File>New on the Levels window.

The New Level window displays.

- 2 Enter a number of the new level.
- 3 Enter the description of the new level
- 4 To add another level, select Next. Select OK to return to the Levels window.

❖ **Modifying a level**

- 1 Highlight a level and choose Edit>Properties.

The Level Properties window displays.

- 2 Modify the level number and description.
- 3 Select Next to modify the next level on the list. Select OK to return to the Levels window.

❖ **Deleting a level**

- Highlight a level and choose Edit>Delete.

You are asked to confirm that you want to delete the level before it is permanently removed.

Levels play a major part in the creation of map flow. Refer to Chapter 15, “Creating a Map Flow,” for more information about level-related actions.

Any-to-any mapping

To open the Any-to-Any Map window, select Map on the Toolbar or on Workspace, or choose Mapping from the Build menu on the main EMap window.

When the mapping window is opened for a new map, the Rule Definition-New window displays.

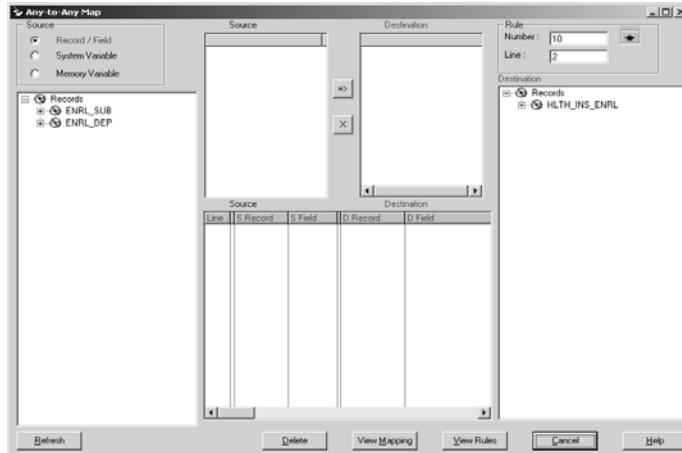
On the Rule Definition-New window, enter the Rule Number and Rule Description of the rule that contains the mapping instructions and select OK. The Any-to-Any Map window displays.

Under Rule in the top right corner of the Any-to-Any Map window, the rule number that you entered on the Rule Definition – New window displays in the Number text box, and the program enters a 1 in the Line text box. Each time that source data is mapped to destination data, a new command is created in the rule and the line number is incremented by 1. (When the mapping window is reopened, the next available number displays in the Line text box. This is the next consecutive number after the number of the last rule command created.) This rule is the Before rule in the map flow and contains all of the mapping instructions.

The Any-to-Any Map window is similar to the transaction Mapping window, but the mapping mechanism is very different. Each instance of mapping in an any-to-any map takes place on a separate command line in a rule - the rule you created the first time the mapping window was opened for the map. On the window, there is still source data on the left, destination data on the right, and a mapping section at the bottom. However, there is no EDI data, and data is mapped directly from one application to another in individual rule commands.

When you originally define the application records for an any-to-any map, you specify a Record Type on the New Records window to indicate whether the record is used as a Source record, a Destination record, or Both. (Refer to the Records/Tables chapter for detailed instructions on defining new records.)

On the Any-to-Any Map window, records designated as either Source or Both appear in the Source pane. Records designated as either Destination or Both appear in the Destination pane. If the record type is left blank, the record appears as both Source and Destination.



The Any-to-Any Map window is divided into four sections. Records that have the record property of Source or Both appear in the Source section on the left side of the Any to Any Map window. Records that have the property Destination or Both appear on the right side of the Any to Any Map window in the Destination section. (Records that are not marked as either Source or Destination are given the property Both and appear in both sections.) Between the two sections is a temporary work area with a Source side and a Destination side. Beneath the Work Space is another section with both Source and Destination sides, where the results of the mapping display. Each instance of mapping has a separate line number in the Mapping section that shows both the Source and Destination record and field (or Source memory variable or system variable).

There are several buttons at the bottom of the window, each of which serve a single special purpose.

❖ **Mapping a source field to a destination field**

- 1 Open a Source record by double-clicking the record name to display the fields in the record.
- 2 Open a record in the Destination section by double-clicking that record name.
- 3 Drag a field from the Source section and drop it onto a Destination field.

- 4 As you drop a source field onto a destination field, a new line appears in the Mapping section at the bottom of the window, showing the results of your actions. This is a new command line in the current any-to-any mapping rule.
- 5 Another way to map Source fields onto Destination fields is to double-click successive Source fields, and then double-click successive Destination fields. As you do this, you see these fields displayed on the two sides of the temporary work area above the Mapping section in the middle of the window.
- 6 When you have finished selecting the fields, select the button between the two sides of the work area. Each Source field is mapped to the Destination field directly across from it in the temporary work area, and the results display in the Mapping section underneath the temporary Work Space. If you select X between the Source and Destination, all of the entries in the work area are removed.

❖ **Mapping a memory variable to a destination field**

- 1 Map a memory variable to a destination record field by selecting Memory Variable in the box at the top left side of the Any to Any Map window above the Source section. (Record/Field is the default selection.)
- 2 A list of all available memory variables is superimposed over the Source section of the window. Map a memory variable to a record field by dragging a memory variable and dropping it onto a Destination record field in the same way that you mapped a field onto a field.

❖ **Mapping a system variable to a destination field**

- 1 Map a system variable to a destination record field by selecting System Variable in the box at the top left side of the Any to Any Map window above the Source section. (Record/Field is the default selection.)
- 2 A list of all available system variables is superimposed over the Source section of the window. Map a system variable to a record field by dragging a system variable and dropping it onto a Destination record field in the same way that you mapped a field onto a field.

❖ **Refreshing the mapping commands**

- The Refresh button at the bottom of the Any-to-Any Map window updates the window to reflect any changes that have been made to the current mapping commands.

❖ **Removing a command line in the mapping rule**

Use Delete button at the bottom of the Any-to-Any Map window remove one instance of mapping, or one command in the mapping rule.

- Highlight the command and select Delete. The highlighted command line is removed.

❖ **Refreshing the mapping window**

- Use the View Mapping button at the bottom of the Any-to-Any Map window to refresh the mapping section of the window with the contents of a mapping rule.

❖ **Switching between mapping and rule functionality with View Rules**

The View Rules button at the bottom of the Any-to-Any Map window is used to switch between mapping and rule functionality. This allows you to add additional logic to your map and perform operations on fields that allow them to be converted correctly.

- 1 Select View Rules. The Rule Definitions window displays.
- 2 To return to the Any-to-Any Map window from the Rule Definitions window, select Mapping (which appears when you highlight a rule command), or you can highlight a rule command and choose Any-to-Any from the Command menu.

Conditional Mapping

About this chapter

This chapter discusses inbound and outbound conditional mapping, including trading partner conditions.

Topics

This chapter includes the following topics:

Topic	Page
About conditional mapping	230
Inbound conditional mapping	231
Outbound conditional mapping	236
Trading partner conditionals: inbound and outbound	242
Mapping segments	242
Mapping elements	243
Trading partner-specific examples	244

About conditional mapping

When conditional mapping is optional	<p>Conditional mapping lets you perform mapping activities based on a condition or conditions. The condition might be whether data exists in a field or element or identify the trading partner with whom the mapped data is being exchanged.</p> <p>Conditional mapping lets you process segments and elements differently based on the presence or absence of data in a field, on the identity of the trading partner, or on the specific data in an element, field or memory variable. For example, if the only N1 segment you expect to receive and process is for Ship To name information, you need to map only one N1 segment and can skip conditional mapping.</p>
When conditional mapping is required	<p>On the other hand, to process two N1 segments (both the Ship To and Bill To name information) and map the data to different fields, you must use conditional mapping to recognize and separately process the two segments. Another example is the varying requirements of different trading partners for mapping a segment. One trading partner may never want to see an N1 segment containing Bill To information. Another trading partner may want to see a name in N102, and a third may want to see a location in N104 and no data in N102. Conditional mapping takes care of all these requirements in one map. It lets you create one map that produces more than one output from the same input, based on conditions that you specify.</p>
Conditional mapping with segments and elements	<p>Segment and element conditional mapping allows you to account for many data conditions in one map. Conditional mapping that is trading-partner-specific allows you to tailor one map to accommodate varying individual trading partner requirements. While you often map one EDI data element to one application field or memory variable, you sometimes map an incoming EDI data element to multiple fields simultaneously. And often you do not want to map a certain element unless another element is also present.</p>
Conditional mapping with inbound and outbound maps	<p>Conditional mapping is different for inbound and outbound maps.</p> <ul style="list-style-type: none"><li data-bbox="373 1199 1208 1260">• Inbound maps use conditional mapping at the segment level to map trading partner-specific segments to application fields.<li data-bbox="373 1277 1208 1366">• Outbound maps generate trading partner-specific segments, and they also use conditional mapping at the segment level to generate a segment only if a record/field or memory variable condition is satisfied.

At the element level, inbound conditional mapping performs element to field mapping only if specified conditions are true. Outbound maps use conditional mapping at the element level to check whether a record/field or memory variable contains data, and if it does, to map the current element and either one or two other elements designated as conditional variables. Conditional variables are memory variables that the system creates for you.

Inbound conditional mapping

On inbound maps, you can use conditional mapping at the segment and element level. You can link mapping of inbound segments and inbound elements to specific trading partners or based on data conditions you set up.

Segment mapping based on trading partner

You can make inbound segment mapping conditional based on a specific trading partner. For the specified trading partner, you can either map or not map the segment.

When you create a trading partner-specific segment, first create a copy of the segment you want to link to a specific trading partner:

- 1 In the main EMap window, select **Build > Mapping** or click the globe icon.
 - 2 In the Mapping window, highlight the segment you want to link to a trading partner, right click and select either **Copy** or **Copy With Data** from the submenu.
 - *If you select Copy*, the program creates a copy of the segment and inserts it after the highlighted segment.
 - *If you select Copy with Data*, the program creates a copy of the segment and all the mapping-related activity associated with it, such as rules or data mapped to a record/field or memory variable, and inserts it after the highlighted segment.
- ❖ **Linking the mapping (or *not* mapping) of a segment to a trading partner**
- 1 In the Mapping window, highlight the segment, right click it and select **Trading Partner** from the submenu.

The Select Segment Trading Partner window displays. The program automatically enters the name and description of the highlighted segment.

Notes

In the Options group, the Trade Partner Type provides the option to *not* map instances of this segment at this level for this trading partner, except for segments specifically tied to this trading partner.

Click the up arrow and NOT displays in the text box. Click the up arrow again and the text box reverts to blank. A blank text box is the default and means that the segment will be mapped for this trading partner.

- 2 Click the up arrow next to Trading Partner Name.
ALL TradePartner is the default value.
- 3 In the Trading Partner window, double-click the trading partner for whom you want to map (or *not* map) this segment.
The Trading Partner – Properties window opens and the program enters the trading partner name in the Internal ID text box.
- 4 Click OK to close all open windows.

Element mapping based on trading partner

You can make inbound element mapping conditional based on a specific trading partner. For the specified trading partner, you can either map or not map the element.

When you create a trading partner-specific element, first create a copy of the element you want to link to a specific trading partner:

- 1 In the main EMap window, select Build > Mapping or click the globe icon.
- 2 In the Mapping window, highlight the element you want to link to a trading partner, right click and select either Copy or Copy With Data from the submenu.
 - *If you select Copy*, the program creates a copy of the element and inserts it after the highlighted segment.

- *If you select Copy with Data*, the program creates a copy of the element and all the mapping-related activity associated with it, such as rules or data mapped to a record/field or memory variable, and inserts it after the highlighted segment.

❖ **Linking the mapping (or *not* mapping) of an element to a trading partner**

- 1 In the Mapping window, highlight the element, right click it and select Trading Partner from the submenu.

The Select Segment Trading Partner window displays. The program automatically enters the name and description of the highlighted element.

Notes

In the Options group, the Trade Partner Type provides the option to *not* map instances of this element at this level for this trading partner, except for elements specifically tied to this trading partner.

Click the up arrow and NOT displays in the text box. Click the up arrow again and the text box reverts to blank. A blank text box is the default and means that the element will be mapped for this trading partner.

- 2 Click the up arrow next to Trading Partner Name.
ALL TradePartner is the default value.
- 3 In the Trading Partner window, double-click the trading partner for whom you want to map (or *not* map) this element.
The Trading Partner – Properties window opens and the program enters the trading partner name in the Internal ID text box.
- 4 Click OK to close all open windows.

Element mapping based on data criteria

Inbound conditional mapping at the element level can also be based on the evaluation of data criteria you create and attach to an element as a condition or conditions. The program evaluates the condition or conditions and maps the data if they are true.

If/then and AND conditions

You may create up to two conditions. If you define one condition, the condition is applied as an “if ... then”. If you create two conditions, they will be applied as an AND condition (“if ... AND if... then”). Both conditions must be true for the mapping to occur.

OR conditions

You can create an OR condition by duplicating the element and modifying the condition. You can duplicate the element by highlighting it and selecting Copy or Copy With Data from the Element menu on the Mapping window. Duplicate the element and modify its conditions as many times as you have possible destinations and/or conditions for this element.

Note Refer to Chapter 3, “Working with Maps” for more detailed instructions on copying elements.

❖ **Setting up conditions**

Use the following procedure to set up the condition or conditions that will determine whether the element will be mapped.

- 1 Map the element to the destination application field. Then, right-click the element and select Condition from the submenu.

The Element Conditions window displays, automatically populated with the identifier and description of the element whose mapping depends on the condition or conditions you specified.

Under Options, you may set up one or two conditions – Condition 1 and Condition 2.

- 2 Click the top up arrow to choose the data to be evaluated in the condition.

If you choose Record/Field, the Records/Tables window displays.

- 3 Double-click a record/field to select it, and the program automatically enters the record and field names into the top two text boxes.

If you choose Memory Variable, the Memory Variables window displays.

- 4 Double-click a memory variable to select it, and the program automatically enters MEMVAR into the top text box and the name of the memory variable into the next text box.

If one or more of the conditions depend on the value of data in another element, you must make that other element a conditional variable. A conditional variable is a special type of memory variable that is used in conditional mapping. When you specify an element as a conditional variable, EMap creates a special memory variable field name of XXXNNSS, where XXX is the current Segment ID, NN is a sequence number that gives the position of the element in the segment, and SS is the sub-element. EMap first checks for the existence of the variable name and then creates it if it does not already exist. All field names are case sensitive; a lower case beg01 is not the same variable as an upper case BEG01.

- 5 To conditionalize an element variable, on the Mapping window, right-click the element containing the data that will be evaluated and select Conditional Variable from the submenu.

You see the results of your selection in all three sections of the Mapping window. In the Mapping section at the bottom of the window on the line containing the element you selected, CNDVAR displays under Record and the name of the conditional variable under Field Name. If you double-click the element in the standards section of the window, you see a mapping icon and the name of the conditional variable that was created. The memory variable created appears in the Destination: Memory Variable pane of the Mapping window.

Clear Condition 1 (or 2) lets you remove information entered in a condition.

- 6 Click the middle up arrow and choose an operator from the drop-down list. The program will insert the operator you select into the middle text box. The choices are:
 - EQ – Equal to
 - NE – Not equal to
 - LT – Less than
 - LE – Less than or equal to
 - GT – Greater than
 - GE – Greater than or equal to
- 7 Click the bottom up arrow to choose the value against which the data in the condition will be evaluated.

If you choose Record/Field, the Record/Tables window displays.

- 8 Double-click a record/field to select it, and the program automatically enters the names of the record and field into the top bottom text boxes.
If you choose Memory Variable, the Memory Variables window displays.
- 9 Double-click a memory variable to select it, and the program automatically enters MEMVAR into the first bottom text box and the name of the memory variable into the next text box.
If you choose Constant, the Constant window displays.
- 10 Enter a constant value and select OK. The program enters STRVAR and the constant value into the bottom two text boxes.
- 11 Select OK again to return to the Mapping window

Outbound conditional mapping

On outbound maps, you can use conditional mapping at the segment and element level. You can link mapping of outbound segments to specific trading partners or based on data criteria. You can also link mapping of outbound elements to specific trading partners or based on the presence of data.

When you create conditional segments or elements, you generally first create copies of the segments or elements that you want to link to specific trading partners or map based on data. You can duplicate a segment or element by highlighting it and selecting Copy or Copy With Data from the Segment or Element menu on the Mapping window. You can duplicate the segment or element and modify its conditions as many times as you have possible destinations and/or conditions for this segment or element.

For conditional mapping based on trading partners, the mapping of that segment or element depends upon the value in the field flagged as the trading partner ID in the application data. You set this as a field attribute on the New Field or Field Properties window when you define the application data.

Note Because it is universally recognized, the EDI group ID is stored in the Segment/Element database and used for trading partner-specific conditional mapping.

Data is mapped to that segment or element only when the selected trading partner is the receiver of the EDI data. If you have multiple instances of the same segment marked as ALL, you need a NOT to turn them off. You must be sure to mark segments or elements other than the trading-partner-specific segment or element to NOT be mapped for this trading partner.

Segment mapping based on trading partner

You can make outbound segment mapping conditional based on a specific trading partner. For the specified trading partner, you can either map or not map the segment.

When you create a trading partner-specific segment, first create a copy of the segment you want to link to a specific trading partner:

- 1 In the main EMap window, select Build > Mapping or click the globe.
- 2 In the Mapping window, highlight the segment you want to link to a trading partner, right click and select either Copy or Copy With Data from the submenu.
 - *If you select Copy*, the program creates a copy of the segment and inserts it after the highlighted segment.
 - *If you select Copy with Data*, the program creates a copy of the segment and all the mapping-related activity associated with it – such as rules or data mapped to a record/field or memory variable – and inserts it after the highlighted segment.

❖ Linking the mapping of a segment to a trading partner

- 1 In the Mapping window, highlight the segment, right click it and select Trading Partner from the submenu.

The Select Segment Trading Partner window displays. The program automatically enters the name and description of the highlighted segment.

In the Options group, the Trade Partner Type provides the option to *not* map instances of this segment at this level for this trading partner, except for segments specifically tied to this trading partner.

Click the up arrow and NOT displays in the text box. Click the up arrow again and the text box reverts to blank. A blank text box is the default and means that the segment will be mapped for this trading partner.

- 2 Click the up arrow next to Trading Partner Name.

ALL TradePartner is the default value.

- 3 In the Trading Partner window, double-click the trading partner for whom you want to map (or *not* map) this segment.

The Trading Partner – Properties window opens and the program enters the trading partner name in the Internal ID text box.

- 4 Click OK to close all open windows.

Segment mapping based on data criteria

Outbound conditional segment mapping lets you to designate that a segment is to be generated only if a condition specified is true. You use the Segment Condition window to specify a record/field or memory variable that is to be checked against another record/field, memory variable or constant value before generating the segment.

❖ Attaching a condition to a segment on an outbound map

- 1 From the Segment menu, right-click the segment and choose Condition from the submenu.

The Segment Condition window displays. The program automatically enters the identifier and description of the segment to which the condition will be attached. The condition consists of a left operand (record/field or memory variable), an operator (equal, not equal, less than, less than or equal to, greater than, or greater than or equal to) and a right operand (record/field, memory variable, or constant).

- 2 To choose the Left Operand, click the up arrow and select from the following choices:

- If you choose Record/Field, the Records/Tables window displays.

Double-click a record field to select it, and the program automatically enters the record and field names into the Left Operand text boxes.

- If you choose Memory Variable, the Memory Variables window displays.

Double-click a memory variable to select it, and the program automatically enters MEMVAR and the name of a memory variable into the Left Operand text boxes.

- If you choose Clear Condition, you remove a condition previously defined for this segment.

- 3 To choose the Operator that will be used in the condition, select the up arrow.
A menu of Operator choices displays.
- 4 Select the operator you want to use. You will return to the Segment Condition window and the operator you chose displays in the Operator text box.
- 5 To choose the Right Operand, click the up arrow and select from the following choices:
 - If you choose Record/Field, the Record/Tables window displays.
Double-click a record/field to select it, and the program automatically enters the names of the record and field into the Right Operand text boxes.
 - If you choose Memory Variable, the Memory Variables window displays.
Double-click a memory variable to select it, and the program automatically enters MEMVAR and the name of the memory variable in the Right Operand text boxes.
 - If you choose Constant, the Constant window displays.
Enter a constant value and select OK. The program enters STRVAR and the constant value into the Right Operand text boxes.
- 6 If you want to enter another condition, select Next. If you are finished entering conditions, select OK to return to the Mapping window.

Element mapping based on trading partner

You can make outbound element mapping conditional based on a specific trading partner. For the specified trading partner, you can either map or not map the element.

When you create a trading partner-specific element, first create a copy of the element you want to link to a specific trading partner:

- 1 In the main ECMap window, select Build > Mapping or click the globe icon.
- 2 In the Mapping window, highlight the element you want to link to a trading partner, right click and select either Copy or Copy With Data from the submenu.

- If you select *Copy*, the program creates a copy of the segment and inserts it after the highlighted segment.
- If you select *Copy with Data*, the program creates a copy of the segment and all the mapping-related activity associated with it, such as rules or data mapped to a record/field or memory variable, and inserts it after the highlighted segment.

❖ **Linking the mapping (or *not* mapping) of an element to a trading partner**

- 1 In the Mapping window, highlight the element, right click it and select Trading Partner from the submenu.

The Select Segment Trading Partner window displays. The program automatically enters the name and description of the highlighted element.

Notes

In the Options group, the Trade Partner Type provides the option to *not* map instances of this element at this level for this trading partner, except for elements specifically tied to this trading partner.

Click the up arrow and NOT displays in the text box. Click the up arrow again and the text box reverts to blank. A blank text box is the default and means that the element will be mapped for this trading partner.

- 2 Click the up arrow next to Trading Partner Name.
ALL TradePartner is the default value.
- 3 In the Trading Partner window, double-click the trading partner for whom you want to map (or *not* map) this element.
The Trading Partner – Properties window opens and the program enters the trading partner name in the Internal ID text box.
- 4 Click OK to close all open windows.

Element mapping based on presence of data

Outbound conditional element mapping allows you to conditionally map constant values to EDI elements based on the presence or absence of data in a specific application field or memory variable. If data is present in the record/field or memory variable, then a specified constant value will be mapped into a specified element.

You can fill two other elements on each mapped element. The element into which the constant value is to be stored must first be defined as a special type of memory variable known as a conditional variable. ECTMap creates a special memory variable field name of XXXNNS, where XXX is the current Segment ID, NN is a sequence number that gives the position of the element in the segment, and SS is the sub-element. When you choose Conditional Variable, ECTMap will check for the existence of the memory variable name and create it if it does not exist.

❖ **Making an element a conditional variable**

- 1 Right-click the element and select Conditional Variable from the submenu.

You will see the results of your selection in all three sections of the Mapping window.

In the Mapping section at the bottom of the window on the line containing the element you selected, CNDVAR displays under Record and the name of the conditional variable under Field Name.

If you double-click the element in the standards section of the window, you will see a mapping icon and the name of the conditional variable that was created. The memory variable also displays in the Source: Memory Variable pane of the Mapping window.

After you have created the conditional variable, you specify what will happen if the program finds data in an application field or memory variable.

- 2 Map the element, right-click it, and select Condition from the submenu.

Note You can also highlight the element after it has been mapped and choose Condition from the Element menu.

The Element Conditions window displays, already populated with the identifier and description of the element whose mapping depends on the condition.

Under Options, you may assign values to two conditional variables – under Store Into 1 and Store Into 2.

- 3 Click the top up arrow to display the Memory Variables window.

Remember, a conditional variable is a special type of memory variable.

- 4 Double-click a memory variable to select it, and the program automatically enters MEMVAR into the top text box and the name of the memory variable into the next text box.
- 5 Click the middle up arrow and EQ is entered into the text box.
Equal is the only operator that can be used, since you can only assign a value here.
- 6 Click the bottom up arrow to display the Constant window.
- 7 Enter the value that you want to assign to the memory variable.
- 8 Click OK and the constant value is entered in the bottom text box.
- 9 Select OK again to return to the Mapping window.

Trading partner conditionals: inbound and outbound

Both segment and element mapping may be trading partner-specific.

Trading Partner
Owning

Trading partner-specific mapping lets you designate which trading partner uses this segment exclusively (called Trading Partner Owning).

Trading Partner NOT
Owning

Choosing NOT as the Trade Partner Type lets you designate which trading partner exclusively does not use this segment (called Trading Partner Not Owning).

Mapping segments

On inbound maps, only one occurrence of a segment is allowed on a single level. On outbound maps multiple copies of a segment are allowed. In other words, segment mapping is single for inbound maps and multiple for outbound maps.

The DTM segment provides an excellent example. On outbound maps, you can specify that two DTM segments for two dates be generated at a particular level, whereas on inbound maps, the mapping of the dates would be dependent on the qualifier in the DTM segment itself.

Mapping elements

On inbound maps, you can have multiple unconditional maps of a particular element. On outbound maps, only one mapping is allowed for a particular element. For example, a field such as the Purchase Order Number could populate multiple fields in multiple records on an inbound map. However, on an outbound map, only one BEG 03 can be generated.

Trading-partner-specific mapping varies for inbound and outbound maps, and for segments and elements. Table 14-1 and Table 14-2 illustrate what happens for each type of trading partner-specific mapping:

Table 14-1: Methods to use for mapping types

If mapping	Method used
Inbound Segment	Single; only one occurrence allowed
Outbound Element	Single; only one occurrence allowed
Inbound Map Flow	Multiple; many occurrences allowed
Inbound Element	Multiple; many occurrences allowed
Outbound Segment	Multiple; many occurrences allowed

Table 14-2: Trading partner-specific mapping

Method used	Mapping assignments	Trading-partner-specific	Other trading partners
Method Used	Mapping Assignments	Trading-partner-specific	Other Trading Partners
Single	ALL TradePartner	Ignore	Execute
	Trading Partner Owing	Execute	Ignore
Single	ALL TradePartner	Ignore	Execute
	Trading Partner Owing	Ignore	Ignore
Multiple	ALL TradePartner	Execute	Execute
	Trading Partner Owing	Execute	Ignore
Multiple	ALL TradePartner	Ignore	Execute
	Trading Partner Owing	Ignore	Ignore

Inbound maps can have only one occurrence of a segment on a single level, but they can have multiple elements mapped to several fields. Outbound maps can have multiple copies of a segment on one level, but only one element may be mapped per segment.

Trading partner-specific examples

This section describes four trading partner-specific examples.

Inbound conditional segment mapping

This section provides examples for inbound conditional segment mapping.

Example 1 In this example, one 40 REF segment is mapped. In this case, everyone maps this way.

Segment	Description
40 REF	Indicates ALL under the Trade Partner column.

Example 2 In this example, two 40 REF segments are mapped. In this case, everyone EXCEPT XYZ gets the first 40 REF and only XYZ receives the second 40 REF.

Segment	Description
40 REF	Indicates ALL under the Trade Partner column
40 REF	Indicates XYZ under the Trade Partner column

Example 3 In this example, three 40 REF segments are mapped. In this case, the third segment is not needed as only XYZ would get the mapping unique to them. They would NOT get the segment designated as ALL. No one will get the segment set as Not XYZ.

Segment	Description
40 REF	Indicates ALL under the Trade Partner column
40 REF	Indicates XYZ under the Trade Partner column
40 REF	Indicates NOT XYZ under the Trade Partner column

Example 4 In this example, four 40 REF segments are mapped. In this case, ABC gets nothing. The other trading partners map the same as 2 and 3 above.

Segment	Description
40 REF	Indicates ALL under the Trade Partner column
40 REF	Indicates XYZ under the Trade Partner column
40 REF	Indicates NOT XYZ under the Trade Partner column
40 REF	Indicates NOT ABC under the Trade Partner column

Inbound conditional element mapping

This section provides examples for inbound conditional element mapping.

Example 1

In this example, two 02 elements in the 40 REF segment are mapped. In this case, everyone gets this information.

Segment	Element	Description
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates ALL under the Trade Partner column.

Example 2

In this example, three 02 elements in the 40 REF segment are mapped. In this case, the ALL TradePartner would get the first and second elements. XYZ would get all three.

Segment	Element	Description
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates XYZ under the Trade Partner column.

Example 3

In this example, four 02 elements in the 40 REF segment are mapped. In this case, XYZ would only receive the third element.

Segment	Element	Description
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates XYZ under the Trade Partner column.
40 REF	Element 02	Indicates NOT XYZ under the Trade Partner column.

Example 4

In this example, five 02 elements in the 40 REF segment are mapped. In this case, XYZ would only receive the third element, and ABC would receive nothing.

Segment	Element	Description
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates XYZ under the Trade Partner column.
40 REF	Element 02	Indicates NOT XYZ under the Trade Partner column.
40 REF	Element 02	Indicates NOT ABC under the Trade Partner column.

Outbound conditional segment mapping

This section provides examples for outbound conditional segment mapping.

Example 1

In this example, two 40 REF segments are mapped. In this case, everyone receives the information.

Segment	Description
40 REF	Indicates ALL under the Trade Partner column.
40 REF	Indicates ALL under the Trade Partner column.

Example 2

In this example, three 40 REF segments are mapped. In this case, the ALL TradePartner would get the first and second segments. XYZ would get all three.

Segment	Description
40 REF	Indicates ALL under the Trade Partner column.
40 REF	Indicates ALL under the Trade Partner column.
40 REF	Indicates XYZ under the Trade Partner column.

Example 3

In this example, four 40 REF segments are mapped. In this case, XYZ would only receive the third segment. Everyone EXCEPT XYZ would get the first and second segments.

Segment	Description
40 REF	Indicates ALL under the Trade Partner column.
40 REF	Indicates XYZ under the Trade Partner column.
40 REF	Indicates XYZ under the Trade Partner column.
40 REF	Indicates NOT XYZ under the Trade Partner column.

Outbound conditional element mapping

This section provides examples for outbound conditional element mapping.

Example 1

In this example, one 02 element for the 40 REF segment is mapped. In this case, everyone gets this information.

Segment	Element	Description
40 REF	Element 02	Indicates ALL under the Trade Partner column.

Example 2

In this example, two 02 elements for the 40 REF segment are mapped. In this case, everyone EXCEPT XYZ receives the first element and only XYZ receives the second element.

Segment	Element	Description
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates XYZ under the Trade Partner column.

Example 3

In this example, three 02 elements for the 40 REF segment are mapped. In this case, the third segment is not needed since only XYZ would get the mapping unique to them. They would NOT get the segment designated as ALL.

Segment	Element	Description
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates XYZ under the Trade Partner column.
40 REF	Element 02	Indicates NOT XYZ under the Trade Partner column.

Example 4

In this example, four 02 elements for the 40 REF segment are mapped. In this case, ABC gets nothing.

Segment	Element	Description
40 REF	Element 02	Indicates ALL under the Trade Partner column.
40 REF	Element 02	Indicates XYZ under the Trade Partner column.
40 REF	Element 02	Indicates NOT XYZ under the Trade Partner column.
40 REF	Element 02	Indicates NOT ABC under the Trade Partner column.

Creating a Map Flow

About this chapter

This chapter discusses map flow in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	250
Assigning levels to EDI segments	253
Creating outbound map flow	254
Creating inbound map flow	263
Creating an any-to-any map flow	270

Introduction

Flow levels

Map flow is how data is interpreted during mapping. Flow points are those instances in the map flow at which the data being mapped causes the context for interpreting that data to change. In ECMap, this change in context is called a change in the level. If you think of map flow as an outline, the major entries in the outline represent levels; the places where the levels change represent flow points. While you remain at one level, data is interpreted in the manner dictated by the syntax, semantics, and business logic attached to the data in that level, whether it is EDI data or application data.

When the level changes, the manner in which the data is interpreted changes as well. The syntax, semantics, and business logic attached to the data in the new level assume control of how the data is interpreted.

Automatic map flow creation

ECMap can create basic map flow for transaction maps using information that it derives from the standards data being used in the map and from information that you supply. You can have ECMap create the basic map flow and then customize it, or you can build the flow yourself.

Manual map flow creation

When you manually create flow, you must assign levels to segments in your EDI message and to flow points. In addition, you must also create rules that tell the program how to read, manipulate, and write the data, and attach these rules to the flow and to elements within the EDI segments.

The levels you assign to EDI segments are associated with the application data through the source and destination data linkages created in the mapping process. For any-to-any maps, you must manually create the flow using levels and rules.

Levels group related data objects—data objects that are interpreted in the same way by the program. For outbound transaction maps, this grouping of related data objects is determined by the context of the input—the sequence in which the input must be read in order to create the outgoing EDI message.

For inbound transaction maps, the grouping of related data objects is determined by the specific structure of the incoming EDI message, the format of the output, and the sequence in which the output must be written. For any-to-any maps, the levels are determined by the context of the input and the sequence in which these records must be read in order to create output records with the correct content and in the correct sequence.

In addition to defining the context for interpreting the data at specific points in a message, levels trigger mapping actions in transaction maps; mapping takes place each time a level changes. For example, levels allow you to map a segment (such as a DTM) to multiple record types, depending upon where the segment occurs. Levels can be obvious, such as Header, Detail, or Trailer—or subtle, such as distinguishing between a Ship To and a Bill To address.

Levels and rules

Levels and rules work together to create the flow. While levels provide the grouping and sequencing of data, rules tell the program what actions it should perform on the data. A rule is a user-defined series of commands that can be attached either to a flow point in the map flow or to a specific data object. Rules that are attached to a flow point are executed when there is a change in the level, often called a level break or transition. Rules that are attached to data objects cause actions to take place when the program encounters these data objects during map execution. The commands within rules can perform many actions, such as reading and writing data, performing arithmetic operations and logical functions, calling other rules, and even performing no action.

Three kinds of rules

Three kinds of rules can be associated with a flow point:

- I/O rule
- Before rule
- After rule.

When a flow point is encountered in an outbound transaction map, the I/O rule is executed first and usually reads the input data. If the I/O rule is executed successfully, the Before rule is processed. The mapping then takes place. Finally, the After rule is processed. In inbound transaction maps, there are no I/O rules because the system implicitly knows how to read the Standards data. In any-to-any maps, the I/O rule is executed first and generally reads the input data; the Before rule usually performs the mapping; and the After rule normally writes the output data. For inbound maps, the Before rule and the After rule are executed on every flow point. For outbound maps and any-to-any maps, the Before rule and the After rule are executed only at flow points where the I/O rule has executed successfully. (Refer to the Rules chapter for a detailed explanation of how to perform rule-related actions.)

Levels and rules, and the flow associated with them, differ for inbound and outbound maps. The flow for any-to-any maps is the same as the flow for outbound maps.

Inbound map flow For inbound maps, the flow is driven primarily by the EDI data used as input—you assign levels to the EDI data and define rules based on the output records you want to create. ECTMap knows how to read the EDI data, but it must use the flow to determine level transitions and execute your rules to write the output.

Outbound map flow For outbound maps and any-to-any maps, the process is reversed; the flow is based on how the data must be read. For outbound maps, ECTMap knows how to write the EDI data, but you must define rules and assign levels to the EDI data so that the segments can be written at the points where the input data is read. For any-to-any maps, you must tell the program how to read and write the data, but the flow is driven by how the data must be read in order to create the correct output.

ECTMap can automatically create basic map flow for you, or you can manually create the flow yourself. If ECTMap creates the basic map flow, you can:

- Modify the flow
- Add or modify a new level, delete a level
- Assign a level to a segment or remove a level that was previously assigned to a segment
- Attach a rule to a flow point or to an EDI element
- Remove a rule that was previously attached to either of them.

ECTMap provides a variety of ways to use conditions to imbed business logic in your map. You can use conditional logic in your rules; you can make segments, elements, and flow points trading-partner-specific; and you can attach conditions to segments, elements, and flow points.

The flow for inbound, outbound, and any-to-any maps is different. In this chapter, we first discuss assigning levels to EDI segments, which is the same for both inbound and outbound transaction maps. Then, we discuss creating flow for each of the three types of map. Within flow creation, we first discuss creating map and record flow for outbound maps, then for inbound maps, and finally for any-to-any maps.

Assigning levels to EDI segments

On outbound maps, you control the levels. On inbound maps, the standard controls the levels. EDI standards group segments into different tables, or levels. The levels you define and assign in ECMap should be based on the EDI standard for the transaction being mapped.

Where to begin

For both inbound and outbound transaction maps, levels are assigned to EDI segments from the transaction Mapping window.

❖ **Opening the Mapping window**

- Choose Build>Mapping from the main menu.

The Mapping window displays.

❖ **Assigning levels to EDI segments**

- 1 Choose Levels>Assign from Mapping window.

The Levels and Segment windows display side by side. Levels that have already been created display on the Levels window.

- 2 To assign a level to a segment:

- Drag a level from the Levels window and drop it onto a segment on the Segments window
- Drag a segment and drop it onto a level
- Highlight a block of segments and drag and drop them onto a level.

Adding, modifying, and deleting levels

On the Levels window, you can add a new level, delete a level, or modify an existing level.

❖ **Adding a new level**

- 1 Choose File>New.

The New Level window displays.

- 2 Enter a number of the new level.

- 3 Enter the description of the new level

- 4 To add another level, select Next. Select OK to return to the Levels window.

❖ **Modifying an existing level**

- 1 Highlight a level and choose Edit>Properties.
The Level Properties window displays.
- 2 Modify the level number and description.
- 3 Select Next to modify the next level on the list. Select OK to return to the Levels window.

❖ **Deleting an existing level**

- Highlight a level and choose Edit>Delete.
You are asked to confirm that you want to delete the level before it is permanently removed.

Creating outbound map flow

This section provides instructions on how to create an outbound map flow.

Automatically creating outbound map flow

The Create Flow function works very well when the input is a sequential file and the record types match the EDI sequence. ECTMap automatically creates and assigns rules and defines level numbers based on the map segment information and the element field mapping. Based on the type of outbound flow entry being added (Child/Sibling), ECTMap automatically populates the information used for switching levels and most of the information about the Parent record. However, you must populate the match field if you checked the Multiple Files option.

Outbound flow basically gives the looping structure of the data. When you generate this structure automatically, ECTMap uses the looping structure in the Standard for the basic definition of the flow. However, this basic definition does not necessarily reflect the looping structure in the application data for this particular map.

Since ECTMap does not inherently know information such as which records are the parents of which other record or which records repeat, you may need to make some adjustments to the flow after it is created. In particular, you may need to change the depth of the first segments in loops. The first segment of any loop has a non-zero depth. Unless the application data dictates that a level change should take place at this segment (such as when you change from a header segment to a detail segment that starts a loop), you need to change the depth to zero. Level transitions can be points at which data must be written, but they can also be points to which the program must return but where data is not written.

❖ **Automatically creating outbound map flow**

- 1 Choose Build>Map Flow from the main menu.

The Outbound Map Flow window displays.

- 2 Choose File>Create Flow.

ECTMap creates the basic map flow. It enters information in the five text boxes at the bottom of the window and displays the map flow (in outline form) on the window.

Note The automatic outbound option creates flow, records, and read rules according to the levels in the Segment Table. Occasionally, there may be problems with loops such as “LS” and “LE” segments where manual adjustments need to be made to the outbound generated flow. The Outbound Flow should be considered as a tool to give you a head start when working with reading sequential files. Note that the automatic outbound flow is not to be used when reading XML or ODBC files.

Manually creating outbound map flow

For outbound maps, the flow is driven by how the input records containing the input data must be read in order to create the output. You can either read the records in the input order or use various techniques to read them in a different order to create the desired output. You first look at the EDI message you want to create. You then look at the application data that will be used to create the message and ask yourself, “How must I read these records to create this output?” Based on this analysis, you assign levels and create rules.

Flow hierarchy

When you create an outbound transaction map, you assign levels that group the application data together in a way that is logically related to the grouping of the segments and elements in the transaction set or message you are creating. ECTMap uses family terminology to describe the levels assigned to application data—Parent, Child, and Sibling.

In addition, a special term is used for the first record—the Master record. The Master record is the very first Parent record. Other than the Master record, a Parent record is never explicitly defined in ECTMap; instead, a record implicitly becomes a Parent when it is defined as having a Child. A Child record is always one level below its Parent record. A Child record that has the same Parent record as another different Child record is called a Sibling record.

There can be multiple Child records for the same Parent record, and they are all Sibling records. A Child record can be directly below the Master record or it can be below another Child (who implicitly becomes that Child's Parent) record. The Parent/Child/Sibling terminology used to describe the relationship between the records in the application data is analogous to the Level/Sublevel/Loop terminology applied to the relationship between the segments in EDI data.

You control the order in which the input application data is read to produce the output EDI data by the assignment of the Parent/Child/Sibling levels and by the rules you create—the commands they contain and where the rules occur in the flow. Three rules can take place at a flow point in an outbound map—an I/O rule, a Before rule, and an After rule. If the I/O rule for the flow point is executed successfully, then the Before rule is executed, mapping is done, and the After rule is executed. If the I/O rule is not successful, no mapping takes place and no rules are executed.

Once you have decided how to group your data, you create and assign levels to the flow.

Adding, deleting, and modifying map flow levels

❖ Adding a new outbound map flow entry

- 1 Perform one of the following:
 - If this is the first map flow entry, Choose File>New>Add Master Level
 - For subsequent entries, choose File>New>Add Sibling Level or File>New>Add Child Level.

The Outbound Record Flow window displays, with two tabs - Required and Advanced. The Required tab is active initially.

- 2 Under Levels on the Required tab, enter:
 - Current level – the level assigned to the current record. Select the up arrow and the Levels window displays. Double-click an existing level entry to enter it in the Current level text box, or if you have not defined your levels, define a new level on the Levels window and then select it. The program automatically enters the level number in the first text box and the description of the level in the second text box.
 - Parent level – the level attached to the parent record. Select the up arrow and the Levels window displays. Double-click an existing level entry or define a new level and then select it. The program automatically enters the level number in the first Parent level text box and the description of the level in the second text box.
- 3 Under Record Type on the Required tab, the current record and level can be designated as having one of the following attributes:
 - Mandatory – record is required at this level. If it is not found, the transaction is rejected.
 - Optional (Error Message) – record may or may not be found in the application data file. If it does not exist, a warning message appears indicating that it is missing.
 - Optional (No Message) – record may or may not be found in the application data file. No message appears if it is missing.
 - Link Only – indicates that this level is linked to the Master level or a repeating parent level. The link level can only be reached after a break occurs (the key value changes or the end-of-file is reached). When a break occurs, the flow returns to the Master (or repeating parent) level, and proceeds to the link level. If the break was an end of file the processing ends, otherwise the flow returns to the link level parent.
 - Repeating – there are multiple records of this record type (children/siblings) for each instance of its parent record (i.e., multiple details to one header). Processing at this level ends when the break field (such as a Purchase Order Number) changes. If a repeating record has children, the system reads the repeating record and then the child records before reading another repeating record. The repeating record may be either mandatory or optional.
- 4 Under Records on the Required tab, enter the following information:

- Current record – the record that is read in the I/O Rule. Select the up arrow and the Records/Tables window displays. Double-click a record, and the program automatically enters it in the Current record text box.
- Parent record – the parent record for the current flow entry. It has no parent. Select the up arrow and the Records/ Tables window displays. Double-click a record, and the program automatically enters it in the Parent record text box.
- Key Field – used to match on a field in the parent record. Select the up arrow and the Records/Tables window displays. Double-click a field to enter it in the Key Field text box. This field is required if the Multiple Files option is checked.
- Match Field – a field in the parent record that matches the key field in the current record. Select the up arrow and the Records/Tables window displays. Double-click a field to enter it in the Key Field text box. This field is required if the Multiple Files option is checked.

When the key field and match field no longer match, you are reading a new record.

5 Under Rules on the Required tab, enter the following information:

- I/O rule – generally contains commands that include reading (or fetching) the current application record. It is always performed. If the commands in the I/O rule are successfully executed, the associated segments are generated (mapped). Select the up arrow and the Rules window displays. Double-click a rule, and the program automatically enters the rule number in the first I/O rule text box and the rule description in the second text box.
- Before rule – contains commands that are performed before segments are generated (mapped). These commands might include actions such as initializing a counter. The Before rule is performed only if the commands in the I/O rule were successful. Select the up arrow and the Rules window displays. Double-click a rule, and the program automatically enters the rule number in the first Before rule text box and the rule description in the second text box.

- After rule – contains commands that are performed after completion of the I/O rule and the Before rule and after the segments (if any) are generated (mapped). The After rule is performed only if the commands in the I/O rule were successful. Select the up arrow and the Rules window displays. Double-click a rule, and the program automatically enters the rule number in the first After rule text box and the rule description in the second text box.

Reviewing advanced tab settings

As you create flow (by adding entries in the flow table), EMap automatically sets the levels entered on the Advanced tab of the Outbound Record Flow window – Next Level and Break Level – based on the parent-child relationships you create between your flow entries. EMap adjusts these values automatically as the flow is built. When you have finished creating the flow, you can review the levels, but you rarely have to change them.

The Advanced tab of the Outbound Record Flow window contains information about the levels to which you switch based on certain conditions. If the I/O is successful (you have read the record), the Before rule is executed, the mapping occurs, the After rule is executed, and the flow goes to the Next level. If the I/O is not successful (you did not read the record), you go to the Break level.

- Next Level – the level to which the program goes if another record of the same record type is read. The flow goes to the Next Level when the I/O is successful. Select the up arrow and the Levels window displays.

Double-click an existing level entry to enter it in the Next Level text box, or if you have not defined your levels, define a new level on the Levels window and then select it. The program automatically enters the level number in the first text box and the level description in the second text box.

- Break Level – the level to which the program goes if no more records of this record type are found. The flow goes to the Break Level when the I/O rule is not successful. Select the up arrow and the Levels window displays. Double-click an existing level entry to enter it in the Break Level text box, or if you have not defined your levels, define a new level on the Levels window and then select it. The program automatically enters the level number in the first text box and the level description in the second text box.

❖ **Deleting an outbound map flow entry**

- Highlight an entry and choose Edit>Delete.

You are asked to confirm whether you want to delete the current map flow level.

❖ **Modifying an outbound map flow entry**

- 1 Select an entry and choose Edit>Properties.

The Outbound Record Flow of Level: <level> window displays. It has the same two tabs as the Outbound Record Flow window – Required and Advanced.

- 2 Change the information as needed following the steps discussed in “Adding a new outbound map flow entry” on page 256.

Viewing map flow information

❖ **Viewing outbound map flow**

- Choose View>Outflow Details.

The Outbound Map Flow Detail window displays. It contains all of the information about the outbound map flow on one window. Use the scroll bar to view all of the information.

❖ **Viewing the rules used in the current outbound map**

- Do one of the following:
 - Choose View>Rules.
 - Right-click one of the rule number text boxes at the bottom of the window – I/O Rule, Before Rule, or After Rule.

The Rule Definitions window displays.

❖ **Viewing the levels used in the current outbound map**

- Do one of the following:
 - Choose View>Levels.
 - Right-click one of the level number text boxes at the bottom of the window – Next Level or Break Level.

The Levels window displays.

❖ **Viewing the records and fields associated with the current outbound map**

- Choose View>Record/Field.

The Records/Tables window displays.

Using the multiple files option

Use the Multiple Files option in the following instances:

- When a single application file is used as input to create a single EDI message, and the input file contains multiple record types that need to be processed out of sequence.
- When multiple application files are used as input to create a single EDI message, and each input file contains one record type.
- When a single application file is used as input to create a single/multiple EDI message(s), and the input file contains one record type with specific fields that cause level changes.

Match and Key fields

ECMap uses two special fields to keep related information grouped together and to alert the program that a level change must take place - Match Field and Key Field.

Parent and child records are linked using the Match Field in the parent record and the Key Field in the child records. When these two fields match, the program knows that the input data being processed should be used together to create an EDI message. When the Key Field and Match Field no longer match, the program knows that all of the information for one EDI message has been processed. The program ends the processing of that message and begins the processing of a new EDI message.

When to use the Multiple Files option

The Multiple Files option should be checked in the following circumstances:

For sequential input

When the input is a flat file (sequential file), and:

- There are multiple files with one record type per file. The record types are related via key and match fields. When a child record is read and its key field no longer matches the match field of the parent record, the program knows it has the next parent record. The program automatically ends the processing of the current EDI message and switches to the first level in a new EDI message.
- There is one file that has one record type, but the order of the output to be created is different from the order of the input records in the file. A change in a specific field (called a break field) in the single record type causes a change in levels. When the program encounters a break field, it automatically ends the processing of the current EDI message and switches to the first level in a new EDI message.

- There is one file that has multiple record types, but the order of the records in the input file is different from the order of the data in the EDI message to be created. The program splits the input file into multiple files, with one file per record type, to allow you to define a flow that is not in the same sequence as the input.

For database input

When the input is from a database and:

- Multiple tables are joined into one view, creating one virtual table that contains all of the record types in the database. When the program encounters a change in a key field, it automatically ends the processing of the current EDI message and switches to the first level in a new EDI message.
- Multiple Select Once commands are performed at the beginning of the map execution (at the I/O Rule on the Master Level.). The sets created by the Select Once statements are ordered by the same match/key linkage fields and are traversed according to these linkage fields.

When not to use the Multiple Files option

The Multiple Files option should be unchecked in the following circumstances:

- When there is one sequential input file with multiple record types, and the order of the records in the input file corresponds to the sequence of the data in the EDI message to be created.

Each time the record type changes, the program automatically ends the processing of the current EDI message and switches to the first level in a new EDI message.

- The input is from a database and the child records for each parent are retrieved by doing a new Select command based on each row of the parent. Levels change based on the presence of rows in the set created by the Select commands.

❖ Selecting the multiple files option

- Choose Option>Multiple Files to toggle the multiple files option on and off.

When the multiple files option is selected, a check mark appears next to Multiple Files on the menu.

Creating inbound map flow

Automatic map flow creating

The Create Flow function can automatically generate inbound map flow in most cases, using ECTMap's inherent knowledge of the EDI standards and additional information that you supply. Along with the map flow, Create Flow creates and assigns rules, and defines level numbers based on the map segment information and the element-to-field mapping. You may need to make some adjustments to the flow after it is created.

❖ **Automatically creating inbound map flow**

- 1 Select Build>Map Flow.

The Inbound Map Flow window displays.

- 2 Select File>Create Flow.

The map flow is generated and displayed.

Manually creating inbound map flow

For inbound maps, the flow is driven by what you must do to create the output records containing the application data. You first look at the content and sequence of the application data you want to create. You then look at the EDI transaction set or message that is providing the input data and ask yourself, "How must I group this EDI data so that, when it is read by the program, I will be able to map the necessary data to the application records and write them in the proper sequence?" You generally need to take the looping structure into account when assigning levels. Based on this analysis, you assign levels (group the EDI data) and create rules.

You assign levels to the segments in the incoming transaction set or message so that they are grouped in a way that allows the application records to be created and written in the correct order. You also use the commands contained in rules and the placement of the rules in the flow to control the order in which the application data records are created and written to the output file. Rules can be invoked by the presence of a segment at a particular level, by the occurrence of a specific value in an element or sub-element, or by encountering a flow point. Flow points are those instances in map flow at which the level changes. Two rules always take place at flow points in inbound maps - the Before Rule and the After Rule. The Before Rule is invoked before the segment referenced by the flow point is mapped; the After Rule is invoked after mapping.

Inbound flow defines the relationship between EDI messages you receive and your application data. You create flow points in inbound flow, and each flow point is an entry in the inbound flow table. The entry contains the current level code, the segment (optionally qualified by an element content) that causes processing to advance to the new level code, and the Before and After Rules that are performed at the flow point.

Each of the inbound segments is linked to a data flow level. An occurrence of a particular segment or qualification within a segment can cause the level to change. For example, on an inbound 850 transaction, the occurrence of a PO1 segment causes the level to change from header to detail. The occurrence of a default segment can also cause the level to change. The special flow point associated with this kind of level change is called a Default inbound map flow entry, and it is invoked by the occurrence of a segment that is not defined as being in the current level. In this case, the presence of the default segment causes the level to change and the rules specified for that flow point to be executed. For instance, since you cannot be certain whether the N1, N3, or N4 will be the final segment in an N1 loop, a default segment may be used to exit the loop if any segment but N1, N3, or N4 are received.

Once you have decided how to group your data, you create and assign levels and rules to the map flow.

Adding, deleting, and modifying map flow entries

❖ Adding a new inbound map flow entry

- 1 Do one of the following:
 - If this is the first map flow entry, Choose File>New>Add Master Level

- For subsequent entries, choose File>New>Add Sibling Level or File>New>Add Child Level.

The Inbound Record Flow - New window displays. There are two tabs on the window – Levels and Advanced. The Levels tab is active initially.

2 Enter the following information on the Levels tab

Under Levels:

- The Current level is the which any attached conditions will be checked. Select the up arrow and the Levels window displays. Drag a level and drop it into the Current level text box, or double-click an existing level to enter it in the Current level text box.

If you have not defined your levels, you can define a new level on the Levels window and then select it. The program automatically enters the level number in the first text box and the level description in the second text box.

- The Segment is the segment whose presence in the current level invokes the transition to the new level specified. This field is required if Segment is checked as the Flow Type. If Default is checked as the Flow Type, Segment is grayed out.
- The Next level is the level that receives control when a specified condition is met (receipt of a segment or contents of the segment/sequence). The Next level may be the same as the Current level. If you wish to perform a rule for every iteration of a loop or anytime you may wish to perform I/O within a level on a particular segment without changing level, then the flow point would have the same Current and Next levels or values.

Select the up arrow and the Levels window displays. Drag a level and drop it into the Next level text box, or double-click an existing level to enter it in the Next level text box. You can define a new level on the Levels window and then select it. The program automatically enters the level number in the first text box and the description in the second text box.

Under Flow Type:

- If you choose Segment, the presence of the specified segment in the current level (optionally qualified by an element and sub-element) invokes this entry (i.e., change the level). Certain segments imply a level change – such as switching from a header segment to a detail segment. If you choose Segment, you must make an entry in the Segment text box under Levels in the top section of the window.
- If you choose Default, the receipt of any segment not defined as being at the current level in mapping invokes this entry (i.e., change the level). It may be used to change to a new level and perform rules specified for this flow point. If you choose Default, the Segment text box under Levels in the top section of the window is grayed out.

Under Rules:

- The Before rule is performed at this level before mapping.
- The After rule is performed at this level after mapping.

Select the up arrow and the Rule Definitions window displays. Double-click a rule to select it. If you do not want to perform an action (either before or after mapping), enter 0 - the Do Nothing rule. You return to the Levels tab of the Inbound Record Flow window, and the program populates the first rule text box with the number of the rule you selected and the second text box with the description of the rule.

If you want to create a new rule, you can also do this from the Rule Definitions window. After you have created the rule, you can select it as described above.

3 Enter optional information on the Advanced tab:

Under Element Condition:

- The Element is the element in the segment whose presence in the current level invokes this entry. Select the up arrow, and the Select A Segment window displays. Double-click an element, or drag an element and drop it on the Element text box.

When you select an element, the program automatically populates the following text boxes:

- The Subelement is the sub-element whose presence in the current level invokes this entry.
- The Field Number is the field whose presence in the current level invokes this entry.
- The Type is the code associated with the field.

- The Element/Sub-Element Value Is: is the value in the element or sub-element specified above that invokes this entry.

Choose from the drop-down list of logical operators that relates the value of the data whose presence creates the condition to the value entered in the second text box. Choices include:

- EQ – Equal to
- NE – Not equal to
- LT – Less than
- LE – Less than or equal to
- GT – Greater than
- GE – Greater than or equal to

The second text box contains the value that is compared to the data whose presence creates the condition.

Under Trading Partner

In the first text box, enter the trading partner who “owns” this segment or level. Select the up arrow, and the Trading Partners window displays, with a list of trading partners from which to choose. Either double-click the trading partner or highlight it and choose Select from the File menu, and the program automatically populates the two Trading Partner text boxes. In this way, you place a condition on the flow point, making it trading-partner-specific.

Examples

For inbound maps, when you build conditions in the map flow, all possible conditions must be defined for a specific segment or loop and that one of the key elements must contain a unique value. The key fields are: Current Level, Segment, Before Rule, After Rule, and New Level.

For example, if you have a segment that repeats, you need to specify the value of the element in the Element/Subelement Value Is: field on the Advanced Tab of the Inbound Record Flow - Properties tab. For each element condition, you need to associate a unique rule with this element on the same window. The rule must be unique or the map won't run correctly. To make the rule unique, assign a unique rule number for each element condition. For example, enter "5" for the rule number and "Do Nothing" for the function of the rule. By creating a unique rule number for each condition of the element, your map will run successfully. If you have two entries with the same segment with the same Before Rule and After Rule with the same current and next levels, the map will not run correctly because non-unique rows were created in the inbound map flow table.

Here are three examples of unique rows. For example, in the Inbound Map Flow window you want to execute logic for specific occurrences of a repeating segment, for example REF (01-9F) or REF (01-G1). In order for this to work properly, all other occurrences of the REF segment must be defined.

As you can see for the REF (01 = D9):

- Current level = 1400
- Segment = REF
- Before Rule = 10
- After Rule = 0
- New Level = 1400

For REF (01 = G1):

- Current level = 1400,
- Segment = REF
- Before Rule = 8
- After Rule = 21401
- New Level = 1400

When you need to add REF (01 = DG) to a REF segment needs, you need to define a new "Do Nothing" rule and place it in the Before or After Rule field on the Inbound Record Flow - Properties window.

❖ **Deleting an inbound map flow entry**

- Highlight the entry and choose Edit>Delete.

You are prompted to confirm the deletion before it is permanently removed.

❖ **Modifying an inbound map flow entry**

- 1 Highlight the entry and choose Edit>Properties.
The Inbound Record Flow – Properties window displays.
- 2 Modify the information as described in “Adding a new inbound map flow entry” on page 264.
- 3 Select OK to return to the Inbound Map Flow window.

Copying an inbound map flow entry

The Template option on the Map Flow window allows you to create a copy of the current flow point/entry in the flow point table, including all associated levels and rules.

❖ **Creating a template copy of an inbound map flow entry**

- 1 Choose Edit>Template.
The Inbound Record Flow – Template window displays.
- 2 Modify any information that you want to change.
- 3 Select OK to return to the Inbound Map Flow window.

Viewing inbound map flow information

❖ **Viewing inbound map flow**

- Choose View>Inflow Detail.

The Inbound Map Flow Detail window displays. It contains all of the information about the inbound map flow on one window. Use the scroll bar to view all of the information

❖ **Viewing rules associated with the current inbound map**

- Choose View>Rules.

The Rules Definition window displays.

❖ **Viewing levels associated with the current inbound map**

- Choose View>Levels.

The Levels window displays.

❖ **Viewing the mapping for the current inbound map**

- Choose Segment>Mapping.

The Mapping window displays. You can view all of the mapping details for the current map

❖ **Viewing the segment detail for the current inbound map**

- Choose Segment>Segment Detail.

The Segment window displays. On this window you can see a summary view of all of the segments, levels, and mapping information.

Creating an any-to-any map flow

The flow for any-to-any maps is basically the same as the flow for outbound transaction maps – it is driven by how the input data must be read in order to create and write the output data. You first look at the output data you want to create. You then look at the input data and ask yourself, “How must I read these input records in order to write these output records?” Based on this analysis, you assign levels and create rules. You assign levels so that the input data is grouped in a way that allows you to read it and create the required output data (i.e., the input data is logically related to the output data). Your rules tell the program when and how to read the data, how to map the data, and when and how to write the data.

Manually creating and any-to-any map flow

As discussed earlier, EMap can automatically create the basic map flow for transaction maps because it has built-in knowledge of the structure of the EDI standard being used. However, since application data is not required to conform to any special standard, EMap cannot automatically generate flow for any-to-any maps. You must manually create the map flow for any-to-any maps, and you must do this from the Any-to-Any Map Flow window. Unlike transaction maps, there is no ability to assign levels on the Any-to-Any Map window - because of the difference in how data is mapped in the two types of maps. Levels are assigned to data - not rule commands - and in any-to-any maps, rule commands do the actual mapping.

In any-to-any maps, application data is mapped directly to application data, and rule commands perform the mapping. The first time that you open the Any-to-Any Map window, you must define the rule that holds the mapping instructions. Each time that you link source and destination data, a separate rule command is created within the mapping rule.

I/O rule sequence

In addition to the rule that performs the mapping, you must add rule commands to read the input data, write the output data, and perform any necessary manipulations on the data during mapping. Since an any-to-any map involves no Standards data, you must tell the system how and when to read and write the data. You tell the program how to read and write the data when you specify the type of read and write rule commands that should be used.

In any-to-any maps, the I/O rule generally reads the input, the Before Rule usually performs the mapping, and the After Rule normally writes the output. The program automatically invokes the I/O Rule when the any-to-any map is run. If the I/O Rule is successful, the Before Rule is executed and then the After Rule is executed.

Assign levels

When you create an any-to-any map, you assign levels that group the input application data together in a way that is logically related to the grouping of the output application data you are creating. ECTMap uses family terminology to describe the levels assigned to application data – Parent, Child, and Sibling. In addition, a special term is used for the first record – it is called the Master record. The Master record is the very first Parent record.

Other than the Master record, a Parent record is never explicitly defined in ECTMap; instead, a record implicitly becomes a Parent when it is defined as having a Child. A Child record is always one level below its Parent record. A Child record that has the same Parent record as another different Child record is called a Sibling record. There can be multiple Child records for the same Parent record, and they are all Sibling records. A Child record can be directly below the Master record or it can be below another Child (who implicitly becomes that Child's Parent) record.

You control the order in which the input application data is read to produce the output application data by the Parent/Child/Sibling levels you assign and the rules you create – the commands they contain and where the rules occur in the flow. Three rules can take place at a flow point in an any-to-any map – an I/O rule, a Before rule, and an After rule. The I/O rule is always executed. The Before and After rules are executed only if the I/O rule is successful.

Adding a new any-to-any map flow entry

❖ To add a new any-to-any map flow entry

- Do one of the following:
 - If this is the first map flow entry, Choose File>New>Add Master Level
 - For subsequent entries, choose File>New>Add Sibling Level or File>New>Add Child Level.

The Any-to-Any Record Flow window displays, with two tabs - Required and Advanced. The Required tab is active initially.

Entering Required tab information

Under Levels, enter the following information:

- Current level – the level assigned to the current record. Select the up arrow and the Levels window displays.

This window is normally blank the first time it is invoked since levels are not usually assigned until map flow is created.

Define a new level on the Levels window and then select it. The program automatically enters the level number in the first text box and the description of the level in the second text box. (Once you have entered levels, you can double-click an existing level on the Levels window to enter it in the Current level text box.)

- Parent level – the level attached to the parent record. (All levels except the master level have a parent level.) This level is automatically entered by the program based on earlier choices that you made. If you want to override the level entered by the program, select the up arrow, and the Levels window displays. Double-click an existing level entry or define a new level and then select it. The program automatically enters the level number in the first Parent level text box and the description of the level in the second text box.

Under Record Type specify the attributes for the current flow:

- Mandatory record – must occur at this level. If it is not found, the transaction is rejected.
- Optional (Error Message) record – may or may not be found in the application data file at this level. If the record does not exist, a warning message appears indicating that it is missing.

- Optional (No Message) record – may or may not be found in the application data file at this level. No message appears if the record is missing.
- Link Only – indicates that this level is linked to the Master level or a repeating parent level. The link level can only be reached after a break occurs (the key value changes or the end-of-file is reached). When a break occurs, the flow returns to the Master (or repeating parent) level, and proceeds to the link level. If the break was an end of file the processing ends, otherwise the flow returns to the link level parent.
- A record is Repeating if there are multiple records of the current type (children/siblings) for each instance of its parent record (for example, multiple details records exist for one header record). Processing at this level ends when a break occurs. A break can occur in two different ways:
 - If the “Multiple Files” option is checked, processing at this level ends when the break field (such as a Purchase Order Number) changes. If a repeating record has children, the system reads the repeating record and then the child records before reading another repeating record. The repeating record may be either mandatory or optional.
 - If the input data contains multiple records in a single file and the records are being read in sequence, then the value of the record type determines when a break occurs. For example, if you are processing detail records and a header record is read, this causes a break to occur. (The key field entries are blank.)

Under Records, enter the following information:

- Current record – the record that is read in the I/O rule attached to this flow point. You can enter the Current record or you can have the program enter it for you when you choose the Key Field. To select the Current record yourself, select the up arrow and the Records/Tables window displays.

On the Records/Tables window, double-click a record, and the program automatically enters it in the Current record text box.

- Parent record – automatically entered by the program based on earlier choices that you made. There is no Parent record if this is the first flow entry (i.e., you chose Add Master Level on the Any-to-Any Map Flow window), because the current record is the master record and it has no parent. If this is a subsequent flow entry (i.e., you chose Add Sibling Level or Add Child Level on the Any-to-Any Map Flow window), there will be a Parent record. If you want to override the record entered by the program, select the up arrow and the Records/Tables window displays. Double-click a record, and EMap automatically enters it in the Parent record text box. (You can also have the program enter the Parent record for you when you choose the key field.)
- Key Field – the field in a child record that is used to link that child record with its parent record. Select the up arrow and the Records/Tables window displays. Open the record and double-click a field to enter it in the Key Field text box. This field is required if the Multiple Files option is checked. (If the Current record text box is blank, the record in which the Key Field is located is also entered in that text box at the same time. If there is already an entry in the Current record text box, it is not replaced. At this point, the program also populates the Parent record text box if it is not already populated.)
- Match Field – the field in the parent record that is used to link that parent record with its child records (i.e., if the Match Field in a parent record and the Key Field in a child record are the same, the two records are linked). Select the up arrow and the Records/Tables window displays. Double-click a field to enter it in the Key Field text box. This field is required if the Multiple Files option is checked.

Under Rules on the Required tab, you attach rules to the flow point. These rules contain commands that read data, write data, map data, manipulate data, or do nothing. Enter the following information about the three rules attached to the flow point:

- I/O rule – generally contains the commands that read the input application data. In order for EMap to read the input data, you must use the appropriate read command – an SQL FETCH command for data stored in an ODBC database, a Sequential Read command for data stored in flat files, or a command to read HTML or XML data. The I/O rule is always performed and is performed before any other rule. When the I/O rule executes successfully, the Before and After rules are also executed. Select the up arrow and the Rules window displays. Double-click a rule, and the program automatically enters the rule number in the first I/O rule text box and the rule description in the second text box. If you will not read data at this flow point, use the Do Nothing (0) rule as the I/O rule. (This usually occurs on a Link Only level.)
- Before rule – usually contains the mapping commands that perform the actual data conversion. This rule can also include commands to set conditions or manipulate the data being mapped, as well as commands to write the data. The Before rule is performed after the I/O rule and is performed only if the I/O rule was executed successfully. Select the up arrow and the Rules window displays. Double-click a rule, and the program automatically enters the rule number in the first Before rule text box and the rule description in the second text box. If you will not map data at this flow point, use the Do Nothing (0) rule as the Before rule.
- After rule – generally contains the commands that write (or insert) the output application data. In order for EMap to write the output data, you must use the appropriate write command – an SQL Insert command for data to be stored in an ODBC database, a Sequential Write command for data to be written to a flat file, or a command to write HTML or XML data. The After rule is performed after the I/O rule and the Before rule and only if the I/O rule was executed successfully. Select the up arrow and the Rules window displays. Double-click a rule, and the program automatically enters the rule number in the first After rule text box and the rule description in the second text box. When the After rule has successfully executed, you have generated the output data of the map. If you will not write data at this flow point, use the Do Nothing (0) rule as the After rule.

Entering Advanced tab information

The Advanced tab of the Any-to-Any Record Flow window contains the levels to which the program proceeds based on certain conditions. EMap automatically places values in these fields as the map flow is built, and these values change as entries are added to the map flow. Normally, you would manually enter information on this tab only when you want to override the entries that EMap automatically places in these fields. (Changes to the defaults entered by the program are generally made as an exception for processing.)

- The Next Level is the level to which the program proceeds if another record of the same record type is read. Select the up arrow and the Levels window displays.

On the Levels window, double-click an existing level entry to enter it in the Next Level text box, or if you have not defined your levels, define a new level on the Levels window and then select it. When you select or double-click a level, the program automatically enters the level number in the first Next Level text box and the level description in the second text box.

- The Break Level is the level to which the program proceeds at this flow point if the record type of the current record is different from the record type of the previous record. Select the up arrow and the Levels window displays. Double-click an existing level entry to enter it in the Break Level text box, or if you have not defined your levels, define a new level on the Levels window and then select it. When you select or double-click a level, the program automatically enters the level number in the first Break Level text box and the level description in the second text box.

Deleting and modifying any-to-any

❖ Deleting an any-to-any map flow entry

- Highlight an entry and choose Edit>Delete.

You are asked to confirm whether you want to delete the current record (map flow entry) before it is permanently removed.

❖ Modifying an any-to-any map flow entry

- 1 Highlight an entry and choose Edit>Properties.

The Any-to-Any Record Flow of Level: window displays.

- 2 Modify the information as discussed in “Adding a new any-to-any map flow entry” on page 272.

Viewing any-to-any map flow information

❖ Viewing any-to-any map flow

- 1 Choose View>Outflow Details.

The Any-to-Any Map Flow Detail window displays all of the information about the any-to-any map flow on one window. Use the scroll bar to view all of the information.

- 2 In addition to viewing the information, you can also perform actions from this window. From the menu options at the top of the Any-to-Any Map Flow Detail window, you have all of the same options that you have on the Any-to-Any Map Flow window. From the File menu, you can add a new flow entry. From the Edit menu, you can delete or modify an existing flow entry. From the View menu, you can access rules, levels, and records. From the Options menu, you can choose the Multiple Files option.

❖ Viewing the rules used in the current any-to-any map

- Choose View>Rules.

The Rule Definitions window displays.

❖ Viewing the levels used in the current any-to-any map

- Choose View>Levels.

The Levels window displays.

❖ Viewing the records and fields associated with the current any-to-any map

- Choose View>Record/Field.

The Records/Fields window displays.

Using the multiple files option

Use the Multiple Files option when you use:

- A single application file as input to create a single output application file, and the input file contains multiple record types that need to be processed out of sequence.

- Multiple application files as input to create a single output application file, and each input file contains one record type.
- A single application file as input to create single/multiple output application files, and the input file contains one record type with specific fields that cause level changes.

Match and Key fields

ECMap uses two special fields to keep related information grouped together and to alert the program that a level change must take place - Match Field and Key Field.

Parent and child records are linked using the Match Field in the parent record and the Key Field in the child records. When these two fields match, the program knows that the input data being processed should be used together to create an output application file.

When the key field and the match field no longer match, the program knows that all of the information for one output application file has been processed. The program senses a level change that ends the processing of one output application file and begins the processing of a new output application.

When to use the Multiple Files option

For sequential input

When the input is a flat file (sequential file) and:

- There are multiple files with one record type per file. The record types are related via key and match fields. When a child record is read and its key field no longer matches the match field of the parent record, the program knows it has the next parent record. The program automatically ends the processing of the current output application file and switches to the first level in a new output application file.
- There is one file that has one record type. A change in a specific field (called a break field) in the single record type causes a change in levels. When the program encounters a break field, it automatically ends the processing of the current output application file and switches to the first level in a new output application file.
- There is one file that has multiple record types, but the order of the output to be created is different from the order of input records in the file. The program splits the input file into multiple files, with one file per record type, to allow you to define a flow that is not in the same sequence as the input.

For database input

When the input is from a database and:

- Multiple tables are joined into one view, creating one virtual table that contains all of the record types in the database. When the program encounters a change in a key field, it automatically ends the processing of the current output application file and switches to the first level in a new output application file.
- Multiple Select Once commands are performed at the beginning of the map execution (at the I/O Rule on the Master Level.). The sets created by the Select statements are ordered by the same match/key linkage fields and are traversed according to these linkage fields.

When not to use the Multiple Files option:

Multiple Files option should be unchecked when:

- The input is a flat file (sequential file) and there is one file with multiple record types, and the order of the records in the input file corresponds to the sequence of the records in the output file to be created. Each time the record type changes in the input file, the program automatically ends the processing of the current record in the output file and switches to the first level in a new output record.
- The input is from a database and the child records for each parent are retrieved by doing a new Select command based on each row of the parent. Levels change based on the presence of rows in the set created by the Select command.

❖ Selecting the multiple files option

- Choose Option>Multiple Files to toggle the multiple files option on and off.

When the multiple files option is selected, a check mark appears next to Multiple Files on the menu.

Generating Maps

About this chapter

This chapter provides instructions for generating maps in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	282
Generating maps	282

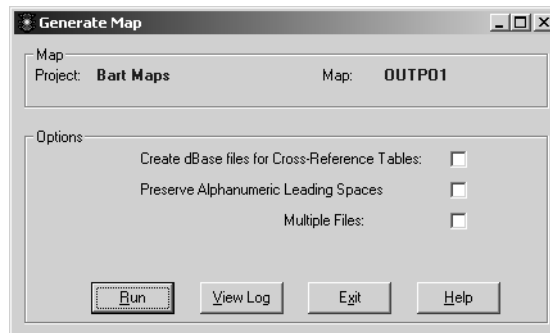
Introduction

Before you run a map, you must generate, or compile, the map. When you generate a map, ECTMap creates a .map text file that includes all the rules, mapping, records, and processing instructions associated with that map. Map generation compiles all of the mapping instructions and related information into the internal representation required by ECTMap for efficient execution or “running” of the map. At runtime, ECRTP (the ECTMap RunTime Program) loads the .map file into memory and uses it to process the data.

Generating maps

❖ Generating a map

- 1 Choose Build>Generate Map from the main menu.



The Generate Map window displays. The names of the current Project and Map display in the Map section at the top of the window.

- 2 Select any of the following options, if applicable to your map:
 - Create dBase files for Cross-Reference Tables – Select this option to have the program create a copy of each cross-reference table in a .dbf file. The default is not to create the dBase tables, but to save them in the map file.
 - Preserve Alphanumeric Leading Spaces – Select this option to preserve all leading spaces for your alphanumeric fields.

- Multiple Files – Select this option if you checked this option on the Outbound Map Flow window. This option is only available for outbound and any-to-any maps. (See Chapter 15, “Creating a Map Flow” for information about the Multiple Files option.)
- 3 Select Run on the Generate Map window.

If the map was generated successfully, a `Map Generation Completed Successfully` message displays.

If the map did not generate successfully, a `No Map Generated. All Errors Need To Be Corrected.` message displays.
 - 4 Select View Log on the Generate Map window to see the actions that took place during map generation and any error conditions that were found.

Carefully review all the Warnings, “Not Referenced” messages, and Errors that were encountered during map generation. As mentioned earlier, warnings and messages do not prevent your map from being generated, but they should be looked at. Errors must be corrected. Once you have corrected all Errors and serious Warnings and “Not Referenced” messages, generate your map again.
 - 5 Once your map has generated successfully, Select Exit.

Reviewing log file entries

The program creates a log file while it attempts to generate the map, it is creating a log file. This log file includes a list of all the Warnings, “Not Referenced” messages, or Errors that were found during map generation. At the end of the log, there is a summary of the problems encountered and a message indicating whether or not the map generated successfully.

Even if a map generated successfully, the log may contain Warnings or “Not Referenced” messages, which you should review. If they indicate any real problems, you should make the necessary changes to your mapping instructions and generate your map again. Select View Log to see the actions that took place during map generation and any error conditions that were found.

- Errors cause map generation to fail. All Errors must be checked and fixed before the map is generated again.
- Warnings and “Not Referenced” messages indicate potential problems. They do not stop map generation nor necessarily have to be corrected, but they should be checked.

For example, you receive a “Not Referenced” message if you have records that are not used in the map. If you intentionally excluded these records from your map, ignore the message. However, if you intended to use them, you must change your mapping instructions and regenerate the map.

Running Maps

About this chapter

This chapter describes the Run Map function of EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	286
Running an inbound map	287
Building a functional acknowledgement	314
Working with TA1 interchange acknowledgements	324
Running an outbound map	330
Running an any-to-any map	358

Introduction

After a map has been successfully generated, it is ready to run.

Run Map windows

There are three Run Map windows

- Inbound
- Outbound
- Any-to-Any

Acknowledgement maps

In addition, there is a Build Acknowledgement window for running outbound 997 (Functional Acknowledgement) and TA1 (Interchange Acknowledgement) maps.

When the current map is an inbound transaction map, the Run Inbound Map window displays. When the current map is an outbound transaction map, the Run Outbound Map window displays. When the current map is an any-to-any map, the Run Any-to-Any Map window displays. The Build Acknowledgement window displays for 997 Functional Acknowledgement maps. The windows are different for each map, they share most of the same options.

❖ Running a map:

- Select Build>Run Map from the main menu.

The Run Map window displays.

The screenshot shows the 'Run Outbound Map' dialog box with the following details:

- Map Name:** OUTP01
- Map Name Extension:** (empty)
- Transaction Name:** 850
- Output EDI File:** C:\aaa\Copied\Data\Output_OUTP01a.txt
- Map Directory:** C:\maps\bart\OUTP01
- Log Type:** ODBC Log
- Non ODBC Trading Partner:**
- Run EDI Adapter:**
- Trading Partner Directory:** (empty)
- Buttons:** Store RunTime Trading Partner Data, Retrieve RunTime Trading Partner Data, Run Map, View Trace, View Trans Log, View EDI File, Cancel, Help

Running an inbound map

When you choose Run Map from the Build menu and the current map is an inbound transaction map, the Run Inbound Map window displays.

The Run Inbound Map window has eight tabs: Required, Option 1, Option 2, File Alias, ODBC Alias, Parameters, I/O Redirect, and Web Script. There are also six buttons at the bottom of the window: Run Map, View Trace, View Trans Log, View EDI File, Cancel, and Help.

Required tab

Enter the following on the Required tab of the Run Inbound Map window:

The screenshot shows the 'Run Inbound Map' dialog box with the 'Required' tab selected. The fields are populated as follows:

- Inbound EDI File:** C:\aaa\Copied\Data\invin.x12
- Map Directory:** C:\maps\bar\NINVIN
- Map Name Extension:** (empty)
- Log Type:** ODBC Log
- Compliance Check:**
- NCPDP Batch:**
- Print EDI:**
- Run EDI Adapter:**
- NCPDP Telecommunications:**
- Non ODBC Trading Partner:**
- Trading Partner Directory:** (empty)
- Buttons:** Store RunTime Trading Partner Data, Retrieve RunTime Trading Partner Data
- Bottom Buttons:** Run Map, View Trace, View Trans Log, View EDI File, Cancel, Help

- **Inbound EDI File** – the full path name of the file containing the EDI data to be translated by the map. You can enter the file name or search for it using the Browse button. (You must enter the full directory path, including the file name with file extension.)

- Map Directory – the full path name of the directory containing the generated map (.map file). This is the map directory that you defined on the Map Directories tab of the Maps window, and it is automatically populated by the program. If you choose, you can enter a different directory or search for one using the Browse button.
- Map Name Extension – the unique 8-character extension that identifies this map from any other maps using the same transaction. For more information see, “Creating a new trade agreement” on page 67.
- Log Type – while the program is processing the EDI data, it places information into a log file. The Log Type specifies how that information is recorded. Choices are:
 - ODBC Log – select ODBC log if you set up an ODBC-compliant log database when you set up your map
 - Text Log – to write a short log file
 - No Log – no log file is written when the map is run
 - Expanded text log – to write a text file that includes all the entries

If you plan to build and send a Functional Acknowledgement (997) for the transaction being processed, you must choose either ODBC Log or Expanded Text Log.

See *ECRTP Reference Guide* for formats of these log files.

- Print EDI – if you are going to run a print map. You also must have set the purpose of the map as PRT on the General tab of the Trade Agreement window.
- Run EDI Adapter – to run ECRTP as an adapter with e-Biz 2000, e-Biz Integrator, or MQSI. When this box is selected, enter in the full directory path that contains the Acquire, Deliver, or Process mode configuration file. For more information about modes and configuration files, see *ECRTP Reference Guide*.
- Compliance Check – select it if you want the RTP to perform a compliance check, comparing the EDI data being sent or received with the applicable EDI standard. To use this option, you must generate the map as a compliance map, using CMP as the purpose on the General tab of the Trade Agreement window.
- NCPDP Batch – select this if your map is an inbound NCPDP map that uses batch processing

- NCPDP Telecommunications – check this if your map is an inbound NCPDP map that uses interactive processing
- Non-ODBC Trading Partner – select it if you want the RTP to run against a dBaseIII trading partner database file.

You must select this button if you did not set up an ODBC-compliant trading partner database on the Map DSN tab of the Maps window.

You may also select this button when you set up an ODBC database, but prefer to run the RTP against a dBase III file. Users sometimes choose this option because the RTP runs faster with the dBase III file and is portable to other platforms that do not support ODBC drivers.

- Trading Partner Directory – is required if you selected Non-ODBC Trading Partner. This directory contains your dBase III trading partner (customer.dbf) and trade agreement (tradstat.dbf) files. You can enter the directory path or search for it using the Browse button.
- Store Run Time Trading Partner Data button – active only when Non-ODBC Trading Partner is selected. When you select this button, the system creates or overwrites information in the dBase III trading partner file (customer.dbf) and trade agreement file (tradstat.dbf) with information from the Access trading partner table and trade agreement table. You use this button in two instances:

Before you run the map for the first time, you must initially populate the dBase III database file. (When you select this button, the program creates customer.dbf and tradstat.dbf, but the wixset.dat file is not created until you select the Run Map button. The program needs to know which of the multiple company profiles to use for the wixset.dat file, and you enter that information on the Option 1 tab.)

Each time you perform actions within EMap that affect trading partner information, you must transfer these changes from the Access database to the dBase III file.

It is very important to first select the Retrieve Run Time Trading Partner Data button before selecting the Store Run Time Trading Partner Data button to prevent the loss of control counts. (See the explanation for the Retrieve Run Time Trading Partner Data button below.)

- Retrieve Run Time Trading Partner Data button – is active only when Non-ODBC Trading Partner is selected. When you select this button, the system retrieves the control count information (ISA_IN_NO, ISA_OUT_NO, and GSNO) from the dBase III trading partner file (customer.dbf) and uses it to update the control count information in the Access trading partner database (tp.mdb).

When you run the RTP using a dBase III trading partner file, the control counts are incremented and saved in the dBase III file, but not in the Access database. As a result, even though the control counts have been saved properly by the system, the information displayed on the window and in reports is not always correct - because the information used in reports and on windows is retrieved from the Access database. You can remedy this potential problem by selecting the Retrieve Run Time Trading Partner Data button each time you have finished running your maps. When you do this, you ensure that the control counts in the Access database are always the same as those in the dBase III file, and that all windows and reports always reflect up-to-date control count information. (However, if you have ever previously incorrectly overwritten the dBase III file without first retrieving the control counts, the control count information is incorrect in both places.)

Note If you are running the RTP against a dBase III file (Non-ODBC Trading Partner is selected), always select Retrieve Run Time Trading Partner Data before you select Store Run Time Trading Partner Data, in order to preserve the correct control counts used in the EDI envelopes. As an extra precaution, you can routinely select Retrieve Run Time Trading Partner Data after each execution of Run Map. We recommend using an ODBC Trading Partner database to avoid potential problems.

Option 1 tab

Enter the following on the Option 1 tab of the Run Inbound Map window:

The screenshot shows the 'Run Inbound Map' dialog box with the 'Option 1' tab selected. The settings are as follows:

- All Trading Partner Default:
- Overwrite Output User Files:
- Ignore Trading Partner MailBox:
- Ignore Trade Agreement MailBox:
- ST03 (X.12):
- Create Bad Transaction Log:
- Enhanced Return Codes:
- Trace Type: Short Trace
- Route EDI Type: No Routing
- Run Inbound Map: INVIN
- Company Identification: 1
- Max Memory Cross Reference: 10000
- Number of Maps in Memory: []
- Number of ODBC Cache: []
- Start Processing at Byte Count: []
- End Processing at Byte Count: []

- All Trading Partner Default – when selected, the program defaults to the ALL Trade Partner if it does not find a trade agreement for the trading partner when the map is run. This is useful when all data will first be processed through a specific map (such as a compliance map) and then processed through an actual map or when data may be received from trading partners that have not been set up or linked with this map. It allows the data to be processed when no valid trade agreement is in the trading partner database for this data set.
- Ignore Trade Agreement Mailbox – when selected, the program does not use the override mailbox set up on the Overrides tab of the Trade Agreement window. EDI data is routed to the trading partner mailbox unless it is not ignored. (The trading partner mailbox is set up on the General tab of the Trading Partner window.) If the trading partner mailbox is ignored, no routing takes place.

- ST03(X12) – when selected, the RTP uses ST03 as part of the trading partner lookup for X12 maps. The default is not to use ST03 since it was not added until version 4030 and is an optional element. If ST03(X12) is selected and the X12 data does not contain an ST03 element, the program still includes ST03 in the lookup key, but the lookup is done on a blank value. The ST03(X12) check box is active only when the type of map is X12.
- Ignore Trading Partner Mailbox – when selected, the program does not use the mailbox set up for the trading partner on the General tab of the Trading Partner window. Instead, the program routes EDI data to the trade agreement IN mailbox — if it exists and is not ignored. (The trade agreement mailbox is set up on the Overrides tab of the Trade Agreement window.) If the trade agreement mailbox does not exist or is ignored, no routing takes place.
- Overwrite Output User Files – when selected, the output created when the map is run overwrites previous output, rather than being appended to the file.
- Create Bad Transaction Log – select to create a log of transactions that fail to meet compliance standards
- Enhanced Return Codes – select to add new return codes (for compliance maps only) 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:
 - 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
- Trace Type – can be Short Trace, Long Trace, or No Trace. If you choose:
 - Short Trace – the program produces a listing of only the errors that were encountered during mapping or the message “Maprun complete. No errors detected.” You would generally choose this option after debugging the map.
 - Long Trace – the program produces a very detailed listing of the program activities that took place during mapping. You would generally choose this option while you are debugging the map.

- No Trace – the program produces no listing. You would generally choose this option after the map goes into production.
- Route EDI Type – you may leave the text box blank or you may choose No Routing, Route In, Route Out, Route Good, Route Bad, or Route Other. Choosing a routing option allows you to pass the EDI data directly into a specific trading partner mailbox (IN, OUT, GOOD, BAD, or OTHER) and into a specific file in that mailbox (if you specified a file name on the Overrides tab of the Trade Agreement window). If Ignore Trading Partner Mailbox is selected, this option is not available.
- Run Inbound Map – allows you to enter the name of a map (without the.map extension) to run. Although you are still required to enter a Trading Partner Directory on the Required tab and Company Identification on this tab, neither trading partner nor company information is referenced when the map is run. The RTP automatically runs this map and does not perform a trading partner lookup to find the map to run.
- Company Identification – allows you to associate a specific company profile with this map. ECMap allows you to create multiple company profiles and associate each of them with a unique Profile Number, which is entered on the Company ID window during company setup. This allows a company to have multiple “Sender IDs” - and thus to respond to the varying requirements of multiple trading partners. If you choose Browse, the Select Company Id window displays a list of all company profiles available. Double-click a profile, and the program enters the profile number associated with the profile in this text box.
- Max Memory Cross Reference – is the maximum allowable number of entries in a cross-reference table for memory lookups. If the entries in any single table exceed the number entered in this text box, then that table is not stored in memory and the lookups for that table go to disk. The default size is 10,000 entries. (This option is ignored when the RTP is run on a Unix machine, where all tables are stored in memory.)
- Number of Maps in Memory – is the maximum number of maps that can be cached in memory at one time. An entry in this text box informs the RTP to keep multiple maps open simultaneously, up to the number entered in this text box. If no entry is made, the default of 0 maps is used and the program does not cache any maps in memory. Once the maximum number of open maps is reached, each new map must replace an open map. The program uses an algorithm to determine which of the open maps is the least frequently used and replaces that map with the new map.

- Number of ODBC Cache – Specifies the maximum number of ODBC connections cached. The default is 0.
- Start Processing at Byte Count – is used to designate the specific character (byte) at which processing of the incoming EDI file begins. When only specific portions of large data files need to be processed, this option saves the time of reading through the preceding data.
- End Processing at Byte Count is used to designate the specific character (byte) at which processing of the incoming EDI file ends. When only specific portions of large data files need to be processed, this option saves the time of reading through the following data.

Option 2 tab

Enter the following on the Option 2 tab of the Run Inbound Map window:

The screenshot shows the 'Run Inbound Map' window with the 'Option 2' tab selected. The window has a title bar with a question mark and a close button. Below the title bar are several tabs: 'Required', 'Option 1', 'Option 2' (selected), 'File Alias', 'ODBC Alias', 'Parameters', 'I/O Redirect', and 'Web Script'. The main area contains several sections of controls:

- Zero Handling Options:** Three radio buttons: 'Zero Fill EDI Non-Null Numbers', 'No Zero Fill on Null or Non-Null Numbers', and 'None'. The first option is selected.
- Transaction Control Number Check:** Three radio buttons: 'Increasing Control Numbers', 'Unique Control Numbers', and 'None'. The first option is selected.
- Validate Control Number Sequence:** A checkbox, currently unchecked.
- Run Acknowledgement Map:** A checkbox, currently unchecked.
- Generate Error for Invalid Leading Zeros:** A checkbox, currently unchecked.
- Output the Elapsed Time:** A checkbox, currently unchecked.
- Trading Partner Search Option:** A dropdown menu showing 'Group Sender'.
- Substitute Output Filename:** A text box with a 'Browse...' button.
- Substitute User File Directory:** A text box with a 'Browse...' button.
- Substitute Map and TP Directory:** A text box with a 'Browse...' button.
- Temporary Files Directory:** A text box with a 'Browse...' button.
- Substitute Good Filename:** A text box with a 'Browse...' button.
- Substitute Bad Filename:** A text box with a 'Browse...' button.

At the bottom of the main area are several buttons: 'Create Batch Command File', 'Delete Transaction Log', 'Archive Transaction Log', 'Acknowledgement Options', and 'Create Adapter Configuration File'. At the very bottom of the window are buttons for 'Run Map', 'View Trace', 'View Trans Log', 'View EDI File', 'Cancel', and 'Help'.

- Zero Handling Options – choose one of the following:
 - Zero Fill EDI Non-Null Numbers to fill incoming elements that are not blank with zeroes

- No Zero Fill on Null or Non-Null Numbers to map numeric data exactly as it appears in the map
- None to do nothing to incoming elements
- Transaction Control Number Check – choose one of the following:
 - Increasing Control Numbers to check for ST control numbers within a group segment
 - Unique Control Numbers to check for unique ST control numbers
 - None for no additional checking on the ST control number
- Validate Control Number Sequence – checks the control numbers on the inbound EDI interchange and group-level envelopes to confirm that the current control number is one greater than the last control number received.
- Run Acknowledgement Map – when selected, the program generates and returns a 997 Functional Acknowledgement to the sender. A Functional Acknowledgement tells the sender the status of the transaction that was sent.
- Generate Error for Invalid Leading Zeros – Flags leading zeros in numeric X12 fields as an error on HIPAA compliance maps.
- Output the Elapsed Time – when selected, the program saves the elapsed time of the run into the trace file when Short Trace is selected on the Option 1 tab. (The elapsed time is automatically saved when Long Trace is selected, but you must select this option to have the time saved when Short Trace is selected.)
- The Trading Partner Search Option – specifies which sender and receiver fields in the incoming EDI envelopes are to be used in the trading partner lookup. Most EDI messages have both an inner and outer envelope, and each envelope includes both sender and receiver codes. For some standards, the sender and receiver codes on the outer envelope have qualifiers that define the type of code used. (The outer envelope is considered an enterprise address and the inner envelope is a division address. ECTMap calls the outer envelope “interchange” and the inner envelope “group”.)

Using criteria specified in the Trading Partner Search Option text box, the program matches the data in the incoming EDI envelope to data in the trading partner database to determine which map to run and how to route the data. There are two trading partner lookup options for non-ODBC users. There are 15 options, and each uses a different combination of sender and receiver information. The seven reverse lookups are generally used in conjunction with compliance checking on outbound EDI. In reverse lookups, the sender information in the envelope is compared to the corresponding receiver information in the trading partner database and vice versa. The trading partner lookup options are listed on the next page:

The following are normal lookups used in compliance checking:

Normal lookups by	Lookup made on	Comments
Group Sender	<ul style="list-style-type: none"> group-level sender code (GS 02, MSH 00004, UNB S002 0042) 	Default lookup
Interchange Sender	<ul style="list-style-type: none"> interchange-level sender qualifier (ISA 05, <no HL7>, UNB S002 0007) interchange-level sender code (ISA 06, MSH 00003, UNB S002 0004) 	The -e5 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Interchange/Group Sender	<ul style="list-style-type: none"> interchange-level sender qualifier (ISA 05, <no HL7>, UNB S002 0007) interchange-level sender code (ISA 06, MSH 00003, UNB S002 0004) group-level sender code (GS 02, MSH 00004, UNB S002 0042) 	The -e2 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Interchange Receiver	<ul style="list-style-type: none"> interchange-level receiver qualifier (ISA 07, <no HL7>, UNB S003 0007) interchange-level receiver code (ISA 08, MSH 00005, UNB S003 0004) 	The -e6 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Interchange/Group Receiver	<ul style="list-style-type: none"> interchange-level receiver qualifier (ISA 07, <no HL7>, UNB S003 0007) interchange-level receiver code (ISA 08, MSH 00005, UNB S003 0004) group-level receiver code (GS 03, MSH 00006, UNB S003 0046) 	The -e4 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Group Sender and Receiver	<ul style="list-style-type: none"> group-level sender code (GS 02, MSH 00004, UNB S002 0042) group-level receiver code (GS 03, MSH 00006, UNB S003 0046) 	The -e1 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.

Normal lookups by	Lookup made on	Comments
Interchange Sender and Receiver	<ul style="list-style-type: none"> • interchange-level sender qualifier (ISA 05, <no HL7>, UNB S002 0007) • interchange-level sender code (ISA 06, MSH 00003, UNB S002 0004) • interchange-level receiver qualifier (ISA 07, <no HL7>, UNB S003 0007) • interchange-level receiver code (ISA 08, MSH 00005, UNB S003 0004) 	The -e7 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Interchange/Group Sender and Receiver	<ul style="list-style-type: none"> • interchange-level sender qualifier (ISA 05, <no HL7>, UNB S002 0007) • interchange-level sender code (ISA 06, MSH 00003, UNB S002 0004) • group-level sender code (GS 02, MSH 00004, UNB S002 0042) • interchange-level receiver qualifier (ISA 07, <no HL7>, UNB S003 0007) • interchange-level receiver code (ISA 08, MSH 00005, UNB S003 0004) • group-level receiver code (GS 03, MSH 00006, UNB S003 0046) 	The -e3 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Batch Sender	<ul style="list-style-type: none"> • fourth field in the record. It includes the starting delimiter and has a maximum length of 24 characters (NCPDP) 	The -e14 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Batch Receiver	<ul style="list-style-type: none"> • tenth field in the record. It includes the starting delimiter and has a maximum length of 24 characters (NCPDP) 	The -e15 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Batch Sender and Receiver	<ul style="list-style-type: none"> • fourth field in the record. It includes the starting delimiter, and has a maximum length of 24 characters (NCPDP) • tenth field in the record. It includes the starting delimiter and has a maximum length of 24 characters (NCPDP) 	The -e16 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Bin Number	<ul style="list-style-type: none"> • first element in the Transaction header segment. The element number is 101-A1 (NCPDP) 	The -e17 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.

Normal lookups by	Lookup made on	Comments
Processor Control Number	<ul style="list-style-type: none"> fourth element in the transaction header segment. The element number is 104-A4 (NCPDP) 	The -e18 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Batch Sender and Bin Number	<ul style="list-style-type: none"> fourth field in the record. It includes the starting delimiter, and has a maximum length of 24 characters (NCPDP) first element in the Transaction header segment. The element number is 101-A1 (NCPDP) 	The -e19 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Batch Sender and Processor Control Number	<ul style="list-style-type: none"> fourth field in the record. It includes the starting delimiter, and has a maximum length of 24 characters (NCPDP) fourth element in the transaction header segment. The element number is 104-A4 (NCPDP) 	The -e20 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Batch Receiver and Bin Number	<ul style="list-style-type: none"> tenth field in the record. It includes the starting delimiter and has a maximum length of 24 characters (NCPDP) first element in the Transaction header segment. The element number is 101-A1 (NCPDP) 	The -e21 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Batch Receiver and Processor Control Number	<ul style="list-style-type: none"> tenth field in the record. It includes the starting delimiter and has a maximum length of 24 characters (NCPDP) fourth element in the transaction header segment. The element number is 104-A4 (NCPDP) 	The -e22 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.

The following are reverse lookups used in compliance checking:

Reverse lookups by	Lookup made on	Comments
Reverse: Group Receiver	<ul style="list-style-type: none"> group-level receiver code (GS 02, MSH 00004, UNB S002 0042) 	The -er switch on the command line or in a batch command file.
Reverse: Interchange Receiver	<ul style="list-style-type: none"> interchange-level receiver qualifier (ISA 05, <no HL7>, UNB S002 0007) interchange-level receiver code (ISA 06, MSH 00003, UNB S002 0004) 	The -e10 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Reverse: Interchange/Group Receiver	<ul style="list-style-type: none"> interchange-level receiver qualifier (ISA 05, <no HL7>, UNB S002 0007) interchange-level receiver code (ISA 06, MSH 00003, UNB S002 0004) oup-level receiver code (GS 02, MSH 00004, UNB S002 0042) 	The -e13 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.

Reverse lookups by	Lookup made on	Comments
Reverse: Interchange Sender	<ul style="list-style-type: none"> interchange-level sender qualifier (ISA 07, <no HL7>, UNB S003 0007) interchange-level sender code (ISA 08, MSH 00005, UNB S003 0004) 	The -e11 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Reverse: Interchange/Group Sender	<ul style="list-style-type: none"> interchange-level sender qualifier (ISA 07, <no HL7>, UNB S003 0007) interchange-level sender code (ISA 08, MSH 00005, UNB S003 0004) group-level sender code (GS 03, MSH 00006, UNB S003 0046) 	The -e12 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Reverse: Interchange Sender and Receiver	<ul style="list-style-type: none"> interchange-level sender qualifier (ISA 07, <no HL7>, UNB S003 0007) interchange-level sender code (ISA 08, MSH 00005, UNB S003 0004) interchange-level receiver qualifier (ISA 05, <no HL7>, UNB S002 0007) interchange-level receiver code (ISA 06, MSH 00003, UNB S002 0004) 	The -e8 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.
Reverse: Interchange/Group Sender and Receiver	<ul style="list-style-type: none"> interchange-level sender qualifier (ISA 07, <no HL7>, UNB S003 0007) interchange-level sender code (ISA 08, MSH 00005, UNB S003 0004) group-level sender code (GS 03, MSH 00006, UNB S003 0046) interchange-level receiver qualifier (ISA 05, <no HL7>, UNB S002 0007) interchange-level receiver code (ISA 06, MSH 00003, UNB S002 0004) group-level receiver code (GS 02, MSH 00004, UNB S002 0042) 	The -e9 switch on the command line or in a batch command file. Available only for ODBC trading partner databases.

- Substitute Output Filename – allows you to override the name of the output application file. This should not be used if you are creating multiple output files. You can Browse for this.
- Substitute User File Directory – allows you to override the location of the output files. This option may be used when you are creating multiple output files. You can Browse for this.

- Substitute Map and TP Directory – allows you to specify a single location for the trading partner files and the map files and overwrite the map and trading partner directories specified on the Required tab of this window. You can Browse for this.
- Temporary Files Directory – allows you to specify the location of the temporary split files that are created when the Multiple Files option is selected. You can Browse for this.
- Substitute Good Filename – allows you to override the trade agreement and trade partner mailbox options by specifying the path and filename to substitute for the default good file of a compliance map. You can browse for this.
- Substitute Bad Filename – Allows you to override the trade agreement and trade partner mailbox options by specifying the path and filename to substitute for the default bad file of a compliance map. You can browse for this.

Note The maximum length of the Good and Bad filename path is 255 characters.

- Create Batch Command File button – creates a DOS batch command file to automatically run the map from a command line. After you have selected all of the options for running the map, select Create Batch Command File button, and the Batch File Name window displays.

In the text box, enter the full path file name where the runmap script is stored and select OK. The program automatically creates a file that includes remarks and the runmap executable followed by the switches that correspond to the options selected on the tabs of the Run Inbound Map window.

A sample batch command file would appear as follows:

```
REM Windows/UNIX Inbound Run Command

REM Inbound Trace file will be in c:\maps\megaproject\bigcompany

REM Log Messages will be in ODBC With Connection String "DSN=ODBCLog"
wrmi32 c:\maps\data\hlthing.x12 -sl "DSN=ODBCLog"
-dg c:\maps\megainsurance\biglab
-dt "c:\maps\megainsurance\biglab" -it -o -b -wx 0
```

If you are using an EMap batch command file in a UNIX environment, replace wrmi32 (the runmap executable) with mapinrun. Refer to *ECRTP Reference Guide* for an explanation of individual map switches.

The batch command file can be:

- Executed separately from EMap
- Incorporated in another batch file
- Called from another application.

You can edit the switches in the file, and you can cut and paste information from the file – such as the runmap switches - for use elsewhere. To run the map from a DOS command line, you must add .bat to the file name.

If you are using an EMap batch command file in a Windows NT environment and the map must execute before other processing continues, insert && before the wrmi32 command line.

- Archive Transaction Log button – allows you to archive the sequential transaction log data. Enter the Destination File Name (full path) where you will store the archived transaction logs in the runmap dialog box.
- Delete Transaction Log button – allows you to delete the previous sequential transaction log data. This is mostly used when you are testing.
- Acknowledgement Options button – when selected, displays the Build Acknowledgement window. It has seven of the eight tabs on the Run Map window – Required, Option 1, Option 2, File Alias, ODBC Alias, Parameters, and I/O Redirect. It also has buttons at the bottom of the window – Run Map, OK, View Trace, View Trans Log, View EDI File, Cancel, and Help.

Refer to the section Building a Functional Acknowledgment, which follows this section on Running an Inbound Map. A functional acknowledgement is a transaction set (997) transmitted by the receiver of an EDI transmission to the sender, indicating receipt and syntactical acceptability of data transmitted according to the ASC X12 standards. The Functional Acknowledgement allows the receiving party to report back to the sending party problems encountered by the syntax analyzer as the data is interpreted. It is not intended to serve as an acknowledgement of data content.

Create adapter configuration file

When you select the Create Adapter Configuration File button, ECTMap prompts you through a series of dialogs to create a customized configuration file. The first dialog to appear is the Configuration File Options

Adapter use two methods of transferring data: buffer and NDO.

- The buffer method moves raw data to and from the transport with no additional processing.
- The NDO method moves the data in a hierarchical (tree) structure. The term NDO (New Data Object) is the Sybase-specific name given to the tree method of exchanging data.

If you select Buffer or NDO, the following modes are available:

- Acquire mode to get data from the application and put that data to a transport.
- Deliver mode to get information from the transport and deliver it to the application.
- Process Mode to get data from a transport, enrich the data or submit a request and get a response, and put the data to another transport.

The Input Buffer Name and Output Buffer Name's availability depends on whether you select Buffer or NDO. For example, if you select Buffer and Acquire because you want to get a data buffer from an application, the available Output Buffer Name would be EDI File.

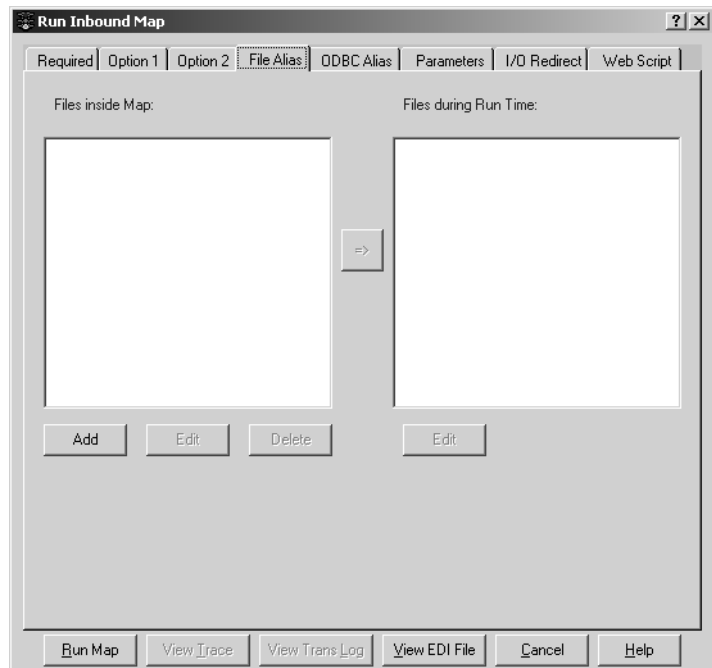
If you select Buffer and Deliver because you want to get data in the form of a buffer from a transport and delivers it to an application, the available Input Buffer Name would be EDI File.

For Buffer and Process, the Adapter Shell constructs two buffers, one used during input and one used during output, and passes them to the processBuffer() function. The data is retrieved from the transport into a buffer, the adapter plug-in is called for processing, and the processing output is put to the transport. The result of the processing should provide an indication of whether looping should continue or not. If you select Buffer and Process, both the Input Buffer Name and the Output Buffer Name would be EDI File:

See the *Adapter Runtime Environment for EDI User Guide* and *ECTRP Reference Guide* for more information about the contents of the configuration files.

File Alias tab

On the File Alias tab, you can create aliases for application files that are hard-coded in the map (the file names are written to the *.map* file when the map is generated), allowing you to dynamically change files at runtime. When the program encounters a file for which an alias has been set up, it automatically substitutes the alias file for the hard-coded file. (This is the `-af` switch explained in *ECRTP Reference Guide*. There is also a `-du` switch that allows you to provide a substitute for the directory in which the application files are located. These switches allow you to adapt your maps between machines and to different platforms.)



❖ Adding a file alias

- 1 Select Add to search for a file.

The Open window displays.

- 2 When you find the map file that you want to replace at runtime, double-click it and the file is automatically entered under Files inside Map.

- 3 Select the right arrow

The Open window displays.

- 4 When you find the substitute file that you want to use at runtime, double-click it and the file is automatically entered under Files during Run Time.

Note File alias names are case sensitive.

You have now created an alias for the original application file. When the program encounters the original application file, it uses the new alias application file in place of the original file.

❖ **Deleting a file alias**

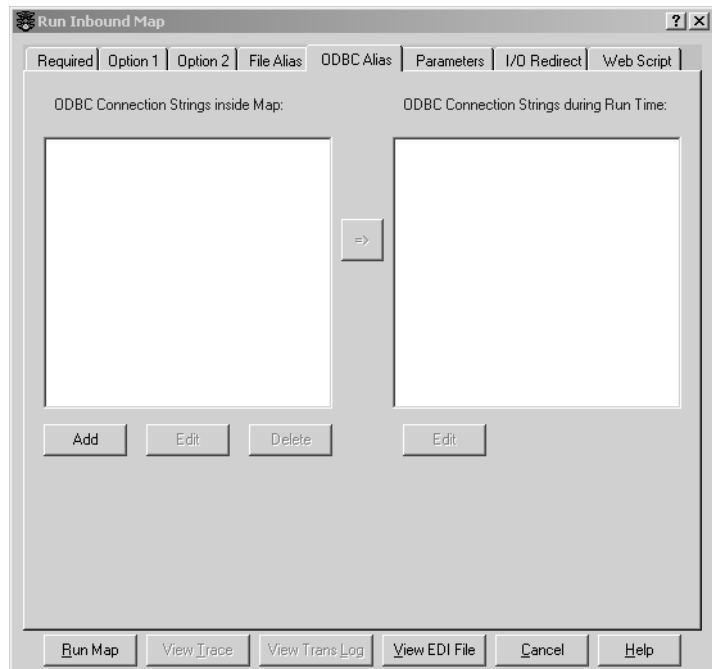
- Highlight the alias pair and select Delete.

❖ **Editing a file alias**

- 1 Highlight a file and select Edit button.
The runmap window displays.
- 2 Edit the full path and select OK.

ODBC Alias tab

On this window, you can create aliases for ODBC connection strings that are hard-coded in the map (the DSNs are imbedded in the .map file when the map is generated), allowing you to change the files dynamically at runtime. (This is the `-ad` switch explained in ECRTF Reference Guide. This switch allows you to adapt your maps between machines and to different platforms.)



❖ Creating an ODBC alias

- 1 Select Add

The runmap window displays.

- 2 Enter the DSN that you want to replace at runtime and select OK.

The DSN is automatically entered under ODBC Connection Strings inside Map.

- 3 Select the right arrow.

The runmap window displays.

- 4 Enter the DSN that you want to use at runtime and select OK.

The DSN is automatically entered under ODBC Connection Strings during Run Time.

You have now created an alias for the original Driver Connect String. When the program encounters the original DSN, it uses the new alias DSN in its place.

❖ **Deleting an ODBC alias**

- To remove a pair of DSNs on one line, highlight the line and select Delete.

❖ **Editing an ODBC Alias**

- 1 Highlight the DSN you want to edit and select Edit.

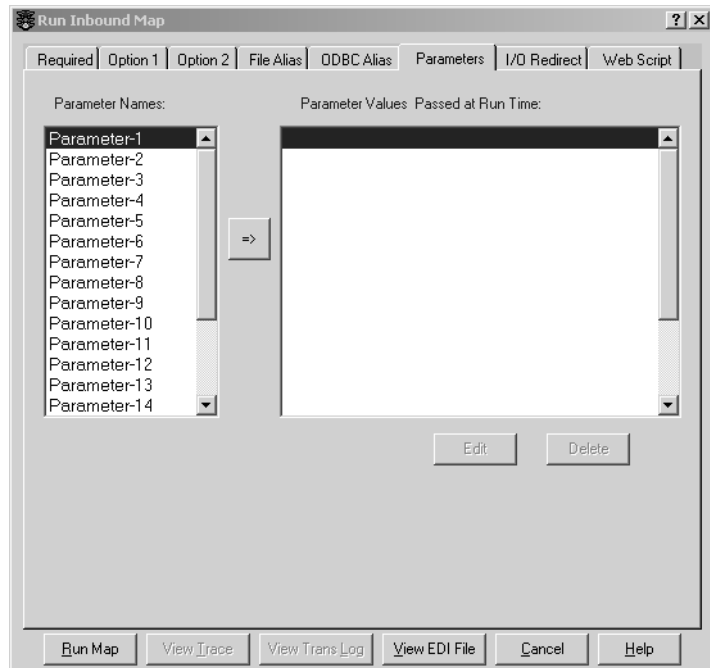
The ODBC Alias window displays.

- 2 On this window, you can change the name of the DSN.

Parameters tab

Use the Parameters tab to pass information to the RTP. The information is passed in the form of values to be substituted for parameters that are used by the map at run time. The parameters are originally created in an Assignment rule command. A parameter can be assigned to a field in a record, a memory variable, or a system variable. EMap allows you to use up to 20 such parameters. On the Parameters tab, you specify the values to be substituted for parameters in this instance of running this particular map.

By imbedding parameters in a map, rather than actual values, you can use the same map for different trading partners or different purposes. For example, you might use a parameter for the directory location of a file that you read or write in the map. Or, the map may serve dual purposes, and you could pass in a flag that specifies the purpose of the map.



❖ Entering a parameter value

- 1 Highlight a parameter and select the right arrow

The Runmap window displays.

- 2 Enter the value for the parameter and select OK.

You return to the Run Inbound Map window, and the value you entered displays across from the parameter for which it will be substituted at run time.

❖ Modifying a parameter value

- 1 Highlight the value and select Edit.

The Runmap window displays with the current value of the parameter displayed in the text box.

- 2 Edit the value and select OK.

You return to the Run Inbound Map window, and the modified value displays.

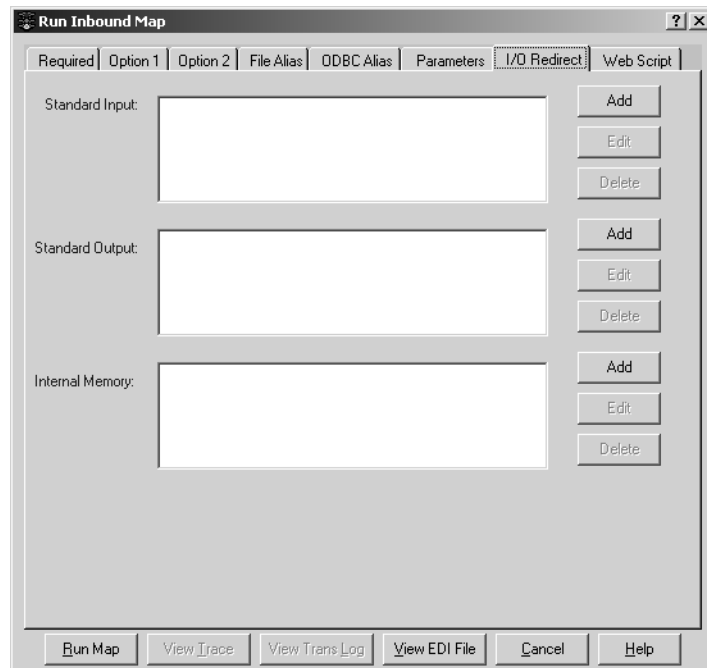
❖ **Deleting a parameter value**

- Highlight the value and select Delete.

The value is removed from the Run Inbound Map window.

I/O Redirect tab

The I/O Redirect tab allows you to use memory locations, rather than files, for reading, writing, and storing data during runtime. This option has many uses. For example, you use this option when the input to the map comes from an entry on a terminal or over the Internet. With this option, the same map can be used with files and with memory—by simply redirecting the input, output, and storage files to memory locations at runtime.



On this window, you can specify that:

- Standard Input is used in place of a specified input file.
- Standard Output is used in place of a specified output file.
- Internal Memory is used in place of a specified storage file.

You can add, modify, or delete any of the three I/O options.

❖ **Adding a new entry for any of the three I/O options**

- 1 Select Add next to the option you want to change.

The Directories/Mailboxes window displays.

- 2 Either double-click the name of the file for which a memory location is used

or

Enter it in the File name text box.

- 3 Select OK.

You return to the Run Inbound Map window and the file name displays in the appropriate section of the window.

❖ **Modifying the name of an input file, output file, or storage file for which a memory location is used**

- 1 Highlight the file on the window. Select Edit.

The Redirect window displays. The full path location of the current file displays in the text box.

- 2 Make changes to the entry and select OK.

You return to the Run Inbound Map window, and the modified file location displays in the appropriate section of the window.

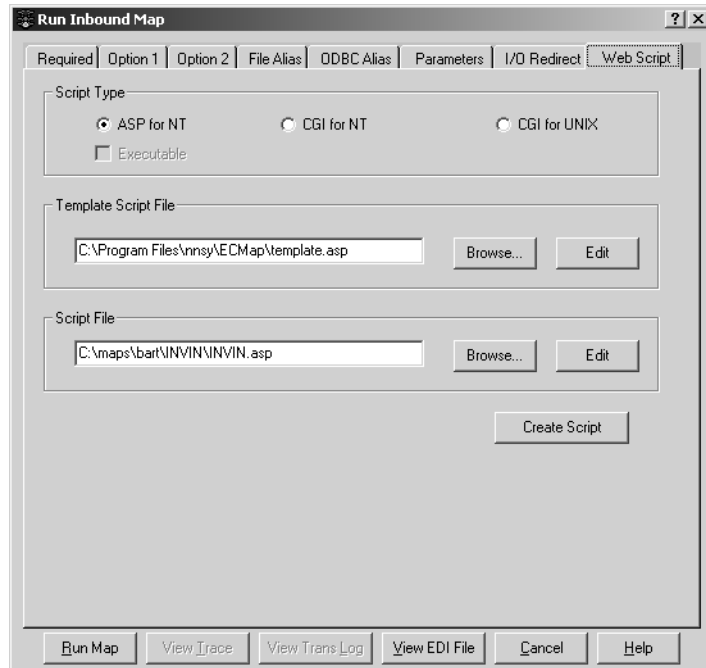
❖ **Deleting the name of an input file, output file, or storage file for which a memory location is used**

- Highlight the file and select Delete.

The file is removed from the Run Inbound Map window.

Web Script tab

Enter the following on the Web Script tab of the Run Inbound Map window:



- Script Type – identifies where the script resides:
 - ASP for NT – An Active Server Page contains the script.
 - CGI for NT – The script is contained in a Common Gateway Interface to the Windows NT operating system.
 - CGI for UNIX – The script is contained in a Common Gateway Interface to a UNIX operating system
- Executable – Check to toggle between compiled and uncompiled code:
 - Uncompiled C Code – The script is uncompiled C code. You can make changes to the code, but you must compile it before you use it. The map switches are in the code itself.
 - Compiled Executable Code – The script is an executable file that has already been compiled. The map switches are in a separate data file outside the code. You cannot make changes to the code.

- **Template Script File** – the full path location of the template script file for the Script Type selected above.

You can Browse for this on the Open window. On the Open window, the program automatically selects the Files of Type that corresponds to the Script Type you selected. Double-click the template file and it is automatically entered in the Template Script File text box.

You can Edit the script file if you have chosen to use ASP or uncompiled C code (CGI for NT or CGI for UNIX) as the template file. The script file displays on a Notepad window. You can perform any actions that are normally available with Notepad. Make changes to the script file and then save it.

- **Script File** – the full path location of the actual script file that is used in the program. It is created from the Template Script File.

You can Browse for this on the Open window. On the Open window, the program automatically selects the Files of Type that corresponds to the Script Type you selected. Double-click the script file and it is automatically entered in the Script File text box.

You can Edit the script file if you have chosen to use ASP or uncompiled C code (CGI for NT or CGI for UNIX). The script file displays on a Notepad window. You can perform any actions that are normally available with Notepad. Make changes to the script file and then save it.

When you entered the required information, select the Create Script. The program automatically creates the script.

Running an inbound map

After you have entered all of the required information on the seven tabs of the Run Inbound Map window, you can Run your map.

❖ **Running and inbound map:**

- Select Run Map.

Run processing

First, the program creates the wixset.dat (company data) file, based on the company profile that you entered in the Company Identification text box on the Option 1 tab of the Run Inbound Map window.

Then, the RTP begins processing the EDI data. You do not receive a message on the window telling that your map has run successfully, but you can select View Trace at the bottom of the window to verify the outcome.

Viewing the trace file for an inbound map

The View Trace button allows you to review the events that took place during the translation process, as they were written to the trace file. The trace file tells you whether the map has run successfully or failed. There are three options for the trace file – short, long, and none. You set this option under Trace Type on the Option 1 tab of the Run Inbound Map window. When you select View Trace, the incoming.err – Notepad window displays for non-ODBC logs and the TrNNN.dat – Notepad window displays for ODBC logs.

If you choose Short Trace and the map:

- Does not encounter errors, the trace file displays a message telling you that there are 0 errors and that the map ran is complete.
- Encounters any errors other than a fatal error, the trace file lists and gives a summary of all the errors that occurred and displays a message telling you that the map run is complete.
- Encounters a fatal error, the trace file lists and gives a summary of all the errors that occurred up to the point that the fatal error occurred. All processing of data stops at that point so that there may be other errors beyond that point.

If you choose Long Trace and the map:

- Does not encounter errors, the trace file lists each step that took place in the mapping process and displays a message at the end telling you that there are 0 errors and that the map run is complete.
- Encounters any errors other than a fatal error, the trace file lists each step that took place in the mapping process and displays error messages at the appropriate points in the listing. At the end, it gives a summary of all the errors that occurred and displays a message telling you that the map run is complete.
- Encounters a fatal error, the trace file lists each step that took place in the mapping process and displays error messages at the appropriate points in the listing. At the point that the fatal error occurred, all processing of data stops so that there may be other errors beyond that point.

If you choose No Trace, the trace file is not created.

Viewing the transaction log for an inbound map

The View Trans Log button allows you to review the information placed in the log file. The amount of information in the log is dependent upon the Log Type option (Text Log, No Log, or Expanded Text Log) that you selected on the Required tab of the Run Inbound Map window. The transaction log tells you whether the map has run successfully or failed, but it also performs other functions. The results of any Write Log rule commands that were used in the map display. If you are using a 997 functional acknowledgement, `translog.in` is used as the input to the 997. When you select View Trans Log, the `translog.in` – Notepad window displays for non-ODBC logs and the View Transaction Log with Run ID: <run ID #> window displays for ODBC logs.

The following displays on the window:

- A list of the errors and warnings that occurred during map execution.
- The results of any user-generated entries created with Write Log rule commands.
- System-generated messages for each ST and SE segment.

Viewing the EDI file for an inbound map

The View EDI File button displays the EDI Viewer window.

On this window, you can print the EDI file, find character strings in the file, replace character strings, view the file in a “wrapped” or “unwrapped” format, and save the file. You can edit the EDI file if required – for example, if errors occurred and corrections had to be made. You can also open another EDI file and perform the same operations on it.

From the File menu on the EDI Viewer window, you can Open another EDI file (or any file), Save (or Save As) the open file, or Print the open file.

- When you choose Open from the File menu on the EDI Viewer window, the Browse – EDI File window displays.

From the Edit menu, either Find a character string or you can find a string and Replace it with another character string.

- If you choose Find, the Find dialog box displays. Enter the character string you want to find in the Find What text box and press the Enter key. The character string for which you are searching displays and highlighted. To find another occurrence of the character string, select Find Next or select Cancel to exit the dialog box.

- If you choose Replace, the Find dialog box displays, with a new Replace With text box. Enter the character string you want to replace in the Find What text box and the character string which is to be replaced, displayed, and highlighted. To replace another occurrence of the character string, select Find Next and then Replace again. To exit the dialog box, select Cancel.
From the Options menu, choose to View EDI in either a Blocked or an Unblocked format.
- If you choose to view the EDI file in a blocked format, the EDI Block Size dialog box displays. In the text box, enter a block size between 40 and 300 and select OK. The EDI file in the EDI Viewer window now displays in blocks the length that you just specified.

Building a functional acknowledgement

An X12 functional acknowledgement is a transaction set (997) transmitted by the receiver of EDI data back to the sender, indicating receipt and syntactical acceptability of the X12 data that was originally transmitted according to the ASC X12 standards. The Functional Acknowledgement allows the receiver to report back to the sender any problems encountered by the syntax analyzer as the data is interpreted. It is not intended to serve as an acknowledgement of data content.

In EMap, the input to a 997 map is the log file created by the inbound map that requested the Functional Acknowledgement. To create a correct 997, you must use either an ODBC log or an expanded text log in the inbound map requesting the 997. If you used a short text log, the 997 will be created, but it will be incorrect.

Required tab

Enter the following on the Required tab of the Build Acknowledgement window:

- Map Name – the name of the 997 map being run. EMap includes four different 997 maps on the installation CD. If you used the default installation directory, these maps are located in c:\Program Files\Sybase\EMap. The names of the four maps are listed below, with a brief description of when you use each map:

- 997ODBCCT – Run this map when you are using an ODBC-compliant log database and want to lookup on a specific trading partner
- 997ODBCA – Run this map when you are using an ODBC-compliant log database and want to lookup on the ALL Trade Partner
- 997XLGTP – Run this map when you are using an expanded text log file and want to lookup on a specific trading partner
- 997XLGA – Run this map when you are using an expanded text log file and want to lookup on the ALL Trade Partner
- Transaction Name – 997, and the Code is FA (Functional Acknowledgement). The program automatically enters this information.
- Output EDI File – the full path file name of the Functional Acknowledgement that is created by the 997 map. Rather than using routing, you would generally specifically choose the destination folder for your Functional Acknowledgement and enter it in this text box. You can enter the full path or Browse for it.
- Map Directory – the full path name of the directory containing the 997 map you are running. If you used the default directory when you installed the maps, the entry in this text box is c:\Program Files\Sybase\ECMap. Otherwise, it is the directory structure that you chose and set up. You can Browse for this.
- Functional Acknowledgement – while the program is creating this, it places information into a log file. You have three choices for Log Type – Text Log, No Log or Expanded Text Log. For Functional Acknowledgements, Text Log, is generally used during development. (See EC RTP Reference Guide for formats of these log files.)
- Non-ODBC Trading Partner – select to run the RTP against a dBase III trading partner file. You must click this button if you didn't set up an ODBC-compliant trading partner database, but you may also click it if you set up an ODBC database, but prefer to run the RTP against a dBase III file. Users sometimes choose the latter option because the RTP runs faster with the dBase III file and is portable to other platforms that do not support ODBC drivers.
- Trading Partner Directory – enter if you selected Non-ODBC Trading Partner. This is the same directory that you used in the inbound map that created the input to this 997 map. This directory contains your dBase III trading partner (customer.dbf) and trade agreement (tradstat.dbf) files. You can enter the directory path or Browse for it.

- RTP – if you run this against an ODBC-compliant trading partner database, enter the DSN for that database in the Trading Partner Connection text box. This is the same DSN that you used in the inbound map that created the input to this 997 map. (This option is not available if you selected Non ODBC Trading Partner.) You can enter the DSN or Browse for it.
- Log Connection text box – not active for Functional Acknowledgements.

Option 1 tab

Enter the following on the Option 1 tab of the Build Acknowledgement window:

- No Trading Partner – not applicable for Functional Acknowledgements.
- All Trading Partner Default – when selected, the program defaults to the ALL Trade Partner if it does not find a trading partner/trade agreement match when the map is run.
- Ignore Trading Partner Mailbox – when selected, the mailbox set up for the trading partner on the General tab of the Trading Partner window is not used. This is generally the choice for Functional Acknowledgements. You can effectively route data when you enter the full path of the EDI output file on the Required tab.
- No EDI File option – not applicable for Functional Acknowledgements.
- Update All Trading Partner Record – select to use the control counts of the All Trade Partner - regardless of whether a specific trading partner match was found. If a trading partner match was found, all of the information for that specific trading partner is used to populate the EDI envelopes - except for the ISA_OUT_NO. Instead, the ISA_OUT_NO for the All Trade Partner is used. This option is chosen (and used every time a map is run) when a company must have all of its transactions in sequential order.
- Ignore Trade Agreement Mailbox – when selected, the override mailbox set up on the Overrides tab of the Trade Agreement window is not used. This is generally the choice for Functional Acknowledgements. You effectively choose a mailbox when you enter the full path of the EDI output file on the Required tab.
- Trace Type – can be Short Trace, Long Trace, or No Trace. You generally choose Short Trace for Functional Acknowledgements during development.

- **Route EDI Type** – choices are No Routing, Route Out, or Route In. Routing allows you to pass the EDI data directly into the Trading Partner mailboxes. If Ignore Trading Partner Mailbox is selected, this option is grayed out. This is generally the choice for Functional Acknowledgements since you can effectively route data when you enter the full path of the EDI output file on the Required tab.
- **Company Identification** – allows you to associate a specific company profile with this map. You should use the same company profile that you used on the inbound map that created the input to the 997 map.

Option 2 tab

Enter the following on the Option 2 tab of the Build Acknowledgement window:

- **No UNG, UNE Segments** – applicable only for EDIFACT messages.
- **Map Numeric Zero** – not applicable for Functional Acknowledgements.
- **Output the Elapsed Time** – when selected, the program saves the elapsed time of the run into the trace file when you use Short Trace.

Note This check box is active only if you have selected Short Trace as the Trace Type on the Option 2 tab. The time is automatically saved for Long Trace.

- **Delete Inbound Trans Log Before Run** – choose for Functional Acknowledgements. This causes the program to delete the previous translog.in file before it runs the inbound map that invokes the Functional Acknowledgement. If you don't choose this option, you create 997 transactions for all of the other transactions in the file.
- **Max Memory Cross Reference** – the maximum allowable cross-reference table entries for memory lookups. If tables exceed the size of this parameter, then the table lookups go to disk. The default size is 10,000 entries. You can use the default for Functional Acknowledgements.
- **Substitute Company Directory** – allows you to override the location of the company directory. You can Browse for this. You should use the same company directory that you used on the inbound map that created the input to the 997 map.

- Substitute Input Filename – allows you to override the name of the input file that is hard-coded in the 997 map. Enter `translog.in` - the name of the log file that is produced by the inbound map and used as the input to the 997 map.
- Substitute User File Directory – allows you to override the directory location of the input file. This should be the map directory. You can Browse for this.
- Substitute Map and TP Directory – allows you to specify a single location for the trading partner files and the map files. This is not usually used for Functional Acknowledgements.

File Alias tab

On this window, aliases for application files that are hard-coded in the 997 map (the file names are written to the `.map` file when the map is generated), allowing you to change the files dynamically at runtime. (This is the `-af` switch explained in EC RTP Reference Guide. There is also a `-du` switch that lets you specify a substitute for the directory in which the application files are located. These switches allow you to adapt your maps between machines and to different platforms.)

❖ Adding a file alias

- 1 Select Add to search for a file.

The Open window displays.

- 2 When you find the map file that you want to replace at runtime, double-click it and the file is automatically entered under Files inside Map.

- 3 Select the right arrow

The Open window displays.

- 4 When you find the substitute file that you want to use at runtime, double-click it and the file is automatically entered under Files during Run Time.

You have now created an alias for the original application file. When the program encounters the original application file, it uses the new alias application file in place of the original file.

❖ Deleting a file alias

- Highlight the alias pair and select Delete.

❖ Editing a file alias

- 1 Highlight a file and select Edit button.
The runmap window displays.
- 2 Edit the full path and select OK.

ODBC Alias tab

On this window, you can create aliases for ODBC connection strings that are hard-coded in the 997 map (997 maps use 997 as the DSN, which is written to the .map file when the map is generated), allowing you to change the files dynamically at runtime. (This is the `-ad` switch explained in ECRTP Reference Guide. This switch allows you to adapt maps between machines and platforms.)

❖ Creating an ODBC alias

- 1 Select Add
The runmap window displays.
- 2 Enter the DSN that you want to replace at runtime and select OK.
The DSN is automatically entered under ODBC Connection Strings inside Map.
- 3 Select the right arrow.
The runmap window displays.
- 4 Enter the DSN that you want to use at runtime and select OK.
The DSN is automatically entered under ODBC Connection Strings during Run Time.

You have now created an alias for the original Driver Connect String. When the program encounters the original DSN, it uses the new alias DSN in its place.

❖ Deleting an ODBC alias

- To remove a pair of DSNs on one line, highlight the line and select Delete.

❖ Editing an ODBC Alias

- 1 Highlight the DSN you want to edit and select Edit.
The ODBC Alias window displays.
- 2 On this window, you can change the name of the DSN.

Parameters tab

The Parameters tab is used to pass information to the RTP. The information is passed in the form of values to be substituted for parameters that are used by the map at run time. The parameters are originally created in an Assignment rule command. A parameter can be a field in a record, a memory variable, or a system variable. EMap allows you to use up to 20 such parameters. On the Parameters tab, you specify the values to be substituted for parameters in this instance of running this particular map.

By imbedding parameters in a map, rather than actual values, you can use the same map for different trading partners or different purposes. For example, you might use a parameter for the directory location of a file that you read or write in the map. Or, the map may serve dual purposes, and you could pass in a flag that specifies the purpose of the map.

❖ Entering a parameter value

- 1 Highlight a parameter and select the right arrow

The Runmap window displays.

- 2 Enter the value for the parameter and select OK.

You return to the Run Inbound Map window, and the value you entered displays across from the parameter for which it will be substituted at run time.

❖ Modifying a parameter value

- 1 Highlight the value and select Edit.

The Runmap window displays with the current value of the parameter displayed in the text box.

- 2 Edit the value and select OK.

You return to the Run Inbound Map window, and the modified value displays.

❖ Deleting a parameter value

- Highlight the value and select Delete.

The value is removed from the Run Inbound Map window.

I/O Redirect tab

The I/O Redirect tab allows you to use memory locations - rather than files - for reading, writing, and storing data during run time. This option has many uses. For example, you use this option when the input to the map comes from an entry on a terminal or over the Internet. With this option, the same map can be used with files and with memory – by simply redirecting the input, output, and storage files to memory locations at run time.

On this window, you can specify that:

- Standard Input is used in place of a specified input file.
- Standard Output is used in place of a specified output file.
- Internal Memory is used in place of a specified storage file.

You can add, modify, or delete any of the three I/O options.

❖ **Adding a new entry for any of the three I/O options**

- 1 Select Add next to the option you want to change.

The Directories/Mailboxes window displays.

- 2 Either double-click the name of the file for which a memory location is used

or

Enter it in the File name text box.

- 3 Select OK.

You return to the Run Inbound Map window and the file name displays in the appropriate section of the window.

❖ **Modifying the name of an input file, output file, or storage file for which a memory location is used**

- 1 Highlight the file on the window. Select Edit.

The Redirect window displays. The full path location of the current file displays in the text box.

- 2 Make changes to the entry and select OK.

You return to the Run Inbound Map window, and the modified file location displays in the appropriate section of the window.

❖ **Deleting the name of an input file, output file, or storage file for which a memory location is used**

- Highlight the file and select Delete.

The file is removed from the Run Inbound Map window.

Running a functional acknowledgement

After you have entered all of the required information on the seven tabs of the Build Acknowledgement window, you can run your map.

❖ **Running a functional acknowledgement**

- Select Run Map at the bottom of the window.

The RTP begins processing the functional acknowledgement. You do not receive a message on the window telling that your map has run successfully, but you can select View Trace at the bottom of the window to verify the outcome.

Viewing the trace file for a functional acknowledgement

The View Trace button allows you to review the events that took place during the translation process, as they were written to the trace file. The trace file tells you whether the map has run successfully or failed. There are three options for the trace file – short and long. You set this option under Trace Type on the Option 1 tab of the Build Acknowledgement window. When you select View Trace, the outgoing.err – Notepad window displays for non-ODBC logs and the TrNNN.dat – Notepad window displays for ODBC logs.

If you choose Short Trace and the map:

- Does not encounter errors, the trace file displays a message telling you that there are 0 errors and that the map ran is complete.
- Encounters any errors other than a fatal error, the trace file lists and gives a summary of all the errors that occurred and displays a message telling you that the map run is complete.
- Encounters a fatal error, the trace file lists and gives a summary of all the errors that occurred up to the point that the fatal error occurred. All processing of data stops at that point so that there may be other errors beyond that point.

If you Choose Long Trace:

- and the map does not encounter errors, the trace file lists each step that took place in the mapping process and displays a message at the end telling you that there are 0 errors and that the map run is complete.
- and the map encounters any errors other than a fatal error, the trace file lists each step that took place in the mapping process and displays error messages at the appropriate points in the listing. At the end, it gives a summary of all the errors that occurred and displays a message telling you that the map run is complete.
- and the map encounters a fatal error, the trace file lists each step that took place in the mapping process and displays error messages at the appropriate points in the listing. At the point that the fatal error occurred, all processing of data stops so that there may be other errors beyond that point.

If you choose No Trace, the trace file is not created.

Viewing the transaction log when building a functional acknowledgement

The View Trans Log button allows you to review the information placed in the log file. The amount of information in the log is dependent upon the Log Type option (Text Log, No Log, or Expanded Text Log) that you selected on the Required tab of the Build Acknowledgement window. The transaction log also tells you whether the map has run successfully or failed, but it also performs several other functions. The results of any Write Log rule commands that were used in the map display. When you select View Trans Log, the translog.out – Notepad window displays for non-ODBC logs and the View Transaction Log with Run ID: <run ID #> window displays for ODBC logs.

The following displays:

- A list of the errors and warnings that occurred during map execution.
- The results of any user-generated entries created with Write Log rule commands.
- System-generated messages for each ST and SE segment.

Working with TA1 interchange acknowledgements

A TA1 interchange acknowledgement is a message transmitted by the receiver of EDI data back to the sender that contains information about the acceptability of the interchange segments of the original message. The interchange acknowledgement reports whether the interchange segment was valid (accepted) or invalid (rejected) and, if appropriate, identifies errors.

The TA1 segment appears following the interchange segment in X12 messages. It can appear between the ISA and GS segments of the file. It is a fixed length, and contains these five elements:

Element	Name	Required	Max/min length	Type	Description
01	Interchange control number	Yes	9	N0	Control number from the ISA of the original message
02	Interchange date	Yes	6	DT	Interchange date from the ISA of the original message
03	Interchange time	Yes	6	TM	Interchange time from the ISA of the original message
04	Interchange acknowledgement code	Yes	1	ID	Code indicating whether the original message/ISA was accepted or rejected
05	Interchange note code	Yes	3	ID	Code specifying errors reported in the original ISA

To set up TA1 interchange acknowledgements, you must:

- 1 At the partner site, create the TA1 interchange acknowledgement map, using the Build Acknowledgement screen. This procedure is similar to that used to build the 997 functional acknowledgement map.
- 2 At the originating site, include a request for a TA1 interchange acknowledgement in the trade agreement for all outbound EDI maps that require an interchange acknowledgement.
- 3 At the originating site, create a TA1 reconciliation map to read the interchange acknowledgement and report on the ISA segment of the original message.
- 4 At the originating site, the ISA14 element value must be set to 1 in the outbound EDI message. This indicates that this message requests an Interchange Acknowledgement.

Each of these processes is described in subsequent sections.

Setting up an interchange acknowledgement

When you set up an interchange acknowledgement, you must define the interchange acknowledgement on the Build Acknowledgement screen.

Defining the interchange acknowledgement

You create a TA1 interchange acknowledgement as you would a 997 functional acknowledgement—on the Build Acknowledgement screen. The instructions that follow describe entries on the Build Acknowledgement tabs that are specific to the TA1 interchange acknowledgement.

Required tab

Enter the following information on the Required tab:

- The Map Name is the name of the TA1 map being run. ECTMap includes two different outbound TA1 maps on the installation CD. If you used the default installation directory, these maps are located in *c:\ProgramFiles\Sybase\ECTMap*. They are:
 - TA1ODBCA – run this map to look up on the ALL trading partner. This is the recommended map.
 - TA1ODBCT – run this map to look up on a specific trading partner.

- The Transaction Name is TA1.
- Enter FA or XX.

Note Leave blank all other fields that do not apply.

- The Output EDI File is the full path name of the interchange acknowledgement that is to be created by the TA1 map.
- The Map Directory is the full path name of the directory containing the TA1 map you are running.
- While the program is creating the interchange acknowledgement, it places information into a log file. You have these choices for Log Type: Text Log, No Log, Expanded Text Log, or ODBC Log. For interchange acknowledgements, Expanded Text Log is used during development.

Note Interchange acknowledgements are written to the ODBC log, TRLOG.

- Check the Non-ODBC Trading Partner check box if your trading partner is not an ODBC trading partner.
- Leave the Trading Partner Directory box blank.
- Enter the DSN for the ODBC-compliant database in the Trading Partner Connection textbox.
- Log Connection field is active if you are using an ODBC log.

Option 1, Option 2, and File Alias tabs

Fill in information on the Option 1, Option 2, and File Alias tabs as described in “Building a functional acknowledgement” on page 314.

ODBC Alias tab

On this window, you can create aliases for ODBC connection strings that are hard-coded in the TA1 map. Creating an alias allows you to change the files dynamically at runtime. The default name for the ODBC connection string is 997Log.

❖ Creating an alias for the ODBC connection string

- 1 Click Add.

The RunMap ODBC Connection String Inside Map window displays.

- 2 Enter 997Log, the original ODBC connection string, and click OK.
997Log is automatically entered in the ODBC Connection Strings inside Map box of the ODBC Alias tab. The Delete and Edit buttons become active.
- 3 Click the right-arrow key.
The RunMap ODBC Connection String During Run Time box displays.
- 4 Enter the name of the substitute DSN that you want to use at runtime, and click OK.
The new DSN name is entered automatically in the ODBC Connection Strings during Run Time box.

When the program encounters the original DSN, it uses the new alias DSN in its place.

Parameters tab

You can use the Parameters tab to specify a RunID to generate acknowledgements for a specific execution of ECRTP. The information is passed in the form of values to be substituted at run time. See the section “Parameters tab” on page 320 for information on how to enter values on the Parameters tab.

You can use the Parameters tab to select Run IDs to acknowledge specific transactions.

❖ Specifying a Run ID for an interchange acknowledgement

- 1 Select Parameter-1 in the Parameter Names box.
- 2 Click the right-arrow key.
- 3 Enter E in the Parameter Values Passed at Run Time box. Click OK.
An E tells ECTMap that the second parameter contains a Run ID.
- 4 Select Parameter-2 in the Parameter Names box.
- 5 Click the right-arrow key.
- 6 Enter the appropriate Run ID value in the Parameter Values Passed at Run Time box. Click OK.

Note You can also specify a Run ID on the Build Acknowledgement screen in ECTMap.

I/O Redirect tab

The I/O Redirect tab allows you to use memory locations—rather than files—for reading, writing, and storing data during run time. Fill in information on the I/O Redirect tab as described in ““I/O Redirect tab” on page 321.

Setting the interchange control number in the trade agreement

You set up the trade agreement for the interchange acknowledgement as described for X12 outbound maps in Chapter 4, “Working with the Address Book.”

Do not select the Request TA1 interchange acknowledge check box. TA1 acknowledgement maps should not request a TA1.

Requesting a TA1 acknowledgement for outbound EDI maps

The X12 tab of the Trade Agreements screen for outbound maps lets you specify whether a TA1 interchange acknowledgement should be requested for communications with the trade partner.

To require a TA1 interchange acknowledge, select the Request TA1 Interchange Acknowledgement check box on the X12 tab of the Trade Agreement screen. A check in this box indicates that there is an agreement between the sender and receiver that the receiver will send an interchange acknowledgement confirming the receipt of the ISA envelope.

Building a reconciliation map

An update map handles the reconciliation process for interchange acknowledgements. This map searches the TRLOG table for TA1 information present in the inbound EDI file processed by an inbound map. If TA1 information is present, it updates the original log entry to indicate receipt of a TA1 acknowledgement.

You create the reconciliation map as an any-to-any map. The instructions that follow describe entries on the New Map Definition screen that are specific to the TA1 reconciliation map.

Map Properties tab

The following information is required on the Map Properties tab:

- The Project name is entered by the program.
- The Map name is TA1_UPDATE.
If the inbound file contains any TAI acknowledgements, TA1_UPDATE marks the original interchange as acknowledged.
- The Map Type is Any-to-Any.
- In the Options section, enter values for the Century Minimum box and, if desired, the Description box.

Map Directories tab

Complete the information on the Map Directories tab as described in “Defining map directories” on page 31.

The Map DSN tab

The Map DSN tab is used when log information is stored in a database accessed through ODBC.

- In the Trade Partner section, enter the Data Source Name for the Trade Partner Driver Connection.
- In the Log section, enter the Data Source Name. If the log has already been created, you can choose 997Log from the drop-down menu. If the DSN has not been created, you can do so directly from this screen by clicking the Configure Data Source button.

Note You can override the DSN name on the Required tab of the Run Any-to-Any Map screen. Enter the DSN name entered on this window for 997Log.

- When you have completed this tab, click OK.

The Project/Map screen displays.

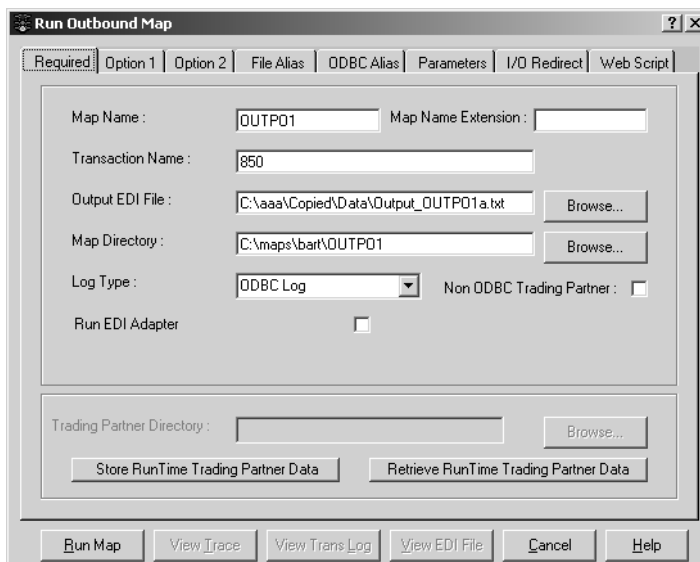
You have defined the TA1 reconciliation map.

Running an outbound map

When you choose Run Map from the Build menu and the current map is an outbound map, the Run Outbound Map window displays. The Run Outbound Map window has eight tabs: Required, Option 1, Option 2, File Alias, ODBC Alias, Parameters, I/O Redirect, and Web Script. There are also six buttons at the bottom of the window: Run Map, View Trace, View Trans Log, View EDI File, Cancel, and Help.

Required tab

Enter the following on the Required tab of the Run Outbound Map window:



The screenshot shows the 'Run Outbound Map' dialog box with the 'Required' tab selected. The window has a title bar with a question mark and close button. Below the title bar are eight tabs: 'Required', 'Option 1', 'Option 2', 'File Alias', 'ODBC Alias', 'Parameters', 'I/O Redirect', and 'Web Script'. The 'Required' tab contains the following fields and controls:

- Map Name: Map Name Extension:
- Transaction Name:
- Output EDI File: - Map Directory: - Log Type:
- Run EDI Adapter:
- Trading Partner Directory: -

At the bottom of the window are six buttons: , , , , , and .

- The Map Name – the name of the outbound map to be run.
- Map Name Extension – the unique 8-character extension that identifies this map from any other maps using the same transaction. For more information see, “Creating a new trade agreement” on page 67.
- The Transaction Name – the identifier of the EDI transaction set or message used in the map.
- Output EDI File – the full path file name of the output EDI file. You can Browse for it.

- Map Directory – the full path of the directory containing the generated map (.map file). You can Browse for this.
- Log Type – while the program is processing the EDI data, it places information into a log file. The Log Type specifies how that information is recorded. Choices are:
 - ODBC Log – select ODBC log if you set up an ODBC-compliant log database when you set up your map
 - Text Log – to write a short log file
 - No Log – no log file is written when the map is run
 - Expanded text log – to write a text file that includes all the entries
- Non-ODBC Trading Partner – select you want the RTP to run against a dBase III trading partner database file.

You must click this button if you did not set up an ODBC-compliant trading partner database on the Map DSN tab of the Maps window.

You may also click this button when you set up an ODBC database, but prefer to run the RTP against a dBase III file. Users sometimes choose this option because the RTP runs faster with the dBase III file and is portable to other platforms that do not support ODBC drivers.

- Run EDI Adapter – select to run ECRTP as an adapter with e-Biz 2000, e-Biz Integrator, or MQSI. When this box is selected, enter in the full directory path that contains the Acquire, Deliver, or Process mode configuration file. For more information about modes and configuration files, see *ECRTP Reference Guide*.
- Trading Partner Directory – required if you selected Non-ODBC Trading Partner. This directory contains your dBase III trading partner (*customer.dbf*) and trade agreement (*tradstat.dbf*) files. You can enter the directory path or search for it using the Browse button.
- Store Run Time Trading Partner Data button – active only when Non-ODBC Trading Partner is selected. When you select this button, the system creates or overwrites information in the dBase III trading partner file (*customer.dbf*) and trade agreement file (*tradstat.dbf*) with information from the Access trading partner table (*tp.mdb*) and trade agreement table (*tradstat.mdb*). You use this button in two instances:
 - Before you run the map for the first time, you must initially populate the dBase III database file.

When you select this button, the program creates customer.dbf and tradstat.dbf, but the wixset.dat file is not created until you select the Run Map button. The program needs to know which of the multiple company profiles to use for the wixset.dat file, and you enter that information on the Option 1 tab.

- Each time you perform actions within EMap that affect trading partner information, you must transfer these changes from the Access database to the dBase III file.

Note It is very important to first select the Retrieve Run Time Trading Partner Data button before selecting the Store Run Time Trading Partner Data button to prevent the loss of control counts. (See the explanation for the Retrieve Run Time Trading Partner Data button below.)

- Retrieve Run Time Trading Partner Data button – active only when Non-ODBC Trading Partner is selected. When you select this button, the system retrieves the control count information (ISA_IN_NO, ISA_OUT_NO, and GSNO) from the dBase III trading partner file (customer.dbf) and uses it to update the control count information in the Access trading partner database (tp.mdb).

When you run the RTP using a dBase III trading partner file, the control counts are incremented and saved in the dBase III file, but not in the Access database. As a result, even though the control counts have been saved properly by the system, the information displayed on the window and in reports is not always correct - because the information used in reports and on windows is retrieved from the Access database. You can remedy this potential problem by selecting the Retrieve Run Time Trading Partner Data button each time you have finished running your maps. When you do this, you ensure that the control counts in the Access database are always the same as those in the dBase III file, and that all windows and reports always reflect up-to-date control count information. (However, if you have ever previously incorrectly overwritten the dBase III file without first retrieving the control counts, the control count information is incorrect in both places.)

Note If you are running the RTP against a dBase III file (Non-ODBC Trading Partner is selected), always select Retrieve Run Time Trading Partner Data before you select Store Run Time Trading Partner Data, in order to preserve the correct control counts used in the EDI envelopes. As an extra precaution, you can routinely select Retrieve Run Time Trading Partner Data after each execution of RunMap. We recommend using an ODBC Trading Partner database to avoid potential problems.

Option 1 tab

Enter the following on the Option 1 tab of the Run Outbound Map window:

The screenshot shows the 'Run Outbound Map' dialog box with the 'Option 1' tab selected. The 'Option 1' tab contains the following settings:

- No Trading Partner
- All Trading Partner Default
- Ignore Trading Partner MailBox
- Lookup Trading Partner from ECRTTP Log
- No EDI File
- Update All Trading Partner Record
- Ignore Trade Agreement MailBox
- Pad EDI Data to Minimum Length

Additional settings include:

- Trace Type: Short Trace (dropdown)
- Route EDI Type: No Routing (dropdown)
- Company Identification: 1 (text field)
- Max Memory Cross Reference: 10000 (text field)
- Number of Maps in Memory: (empty text field)
- Number of ODBC Cache: (empty text field)

Buttons at the bottom: Run Map, View Trace, View Trans Log, View EDI File, Cancel, Help.

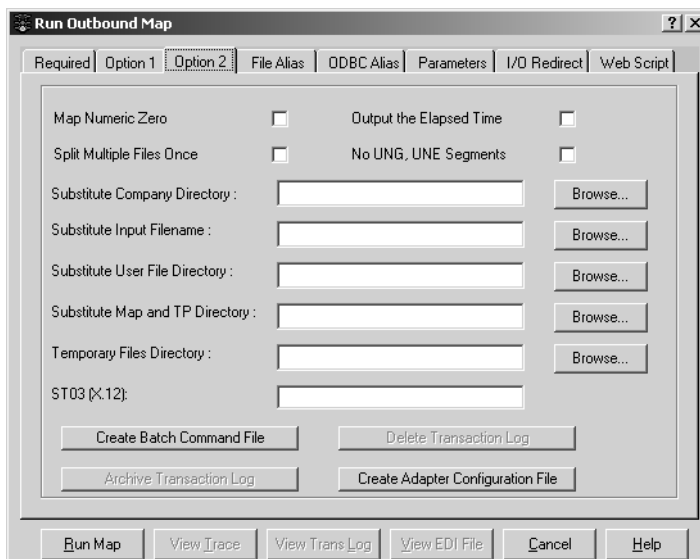
- No Trading Partner – select when you run an outbound map without using a trading partner. The map does not require trading partner records in order to run, although they may be present. When this box is selected, none of the four check boxes below it are active.
- All Trading Partner Default – when selected, the program defaults to ALL Trade Partner if it does not find a trading partner/trade agreement match when the map is run.
- Ignore Trading Partner Mailbox – when selected, the mailbox set up for the trading partner on the General tab of the Trading Partner window is not used. The outbound EDI files go to the Output EDI File location specified on the Required tab of the Run Outbound Map window. (This applies to writing EDI files, not reading them.)
- Lookup Trading Partner from ECRTTP Log – when selected, triggers a look up against the trading partner database. If a trading partner match is found, outbound processing uses the entry to populate the EDI envelope.
- No EDI File check box – disregard.

- Update All Trading Partner Record – when selected, only the control counts of the ALL Trade Partner are updated. (The All Trading Partner Default must be selected.) This function is used when getting control numbers during concurrent runs.
- Ignore Trade Agreement Mailbox – when selected, the override mailbox set up on the Overrides tab of the Trade Agreement window is not used.
- Trace Type – can be Short Trace, Long Trace, or No Trace.
 - If you choose Short Trace, the program produces a listing of only the errors that were encountered during mapping or the message “Maprun complete. No errors detected.” You would generally choose this option after debugging the map.
 - If you choose Long Trace, the program produces a very detailed listing of the program activities that took place during mapping. You would generally choose this option while you are debugging the map.
 - If you choose No Trace, the program produces no listing. You would generally choose this option after the map goes into production.
- Route EDI Type – you may leave the text box blank or you may choose No Routing, Route In, Route Out, Route Good, Route Bad, or Route Other. Choosing a routing option allows you to pass the EDI data directly into a specific trading partner mailbox (IN, OUT, GOOD, BAD, or OTHER) and into a specific file in that mailbox (if you specified a file name on the Overrides tab of the Trade Agreement window). If Ignore Trading Partner Mailbox is selected, this option is not available.
- Company Identification – lets you associate a company profile with this map. You create company profiles on the Company ID window - each associated with a Profile Number and each a unique “Sender ID”. Choose Browse, and the Select Company Id window displays, with a listing of all company profiles available. Double-click a profile to select it, and the profile number associated with the profile is entered in this text box.
- Max Memory Cross Reference – the maximum allowable number of entries in a cross-reference table for memory lookups. If the entries in any single table exceed the number entered in this text box, then that table is not stored in memory and the lookups for that table go to disk. The default size is 10,000 entries. (This option is ignored when the RTP is run on a Unix machine, where all tables are stored in memory.)

- Number of Maps in Memory – the maximum number of maps that can be cached in memory at one time. An entry in this text box informs the RTP to keep multiple maps open simultaneously, up to the number entered in this text box. If no entry is made, the default of 0 maps is used and the program does not cache any maps in memory. Once the maximum number of open maps is reached, each new map must replace an open map. The program uses an algorithm to determine which of the open maps is the least frequently used and replaces that map with the new map.
- Number of ODBC Cache – Specifies the maximum number of ODBC connections cached. The default is 0.

Option 2 tab

Enter the following on the Option 2 tab of the Run Outbound Map window:



- Map Numeric Zero – when selected, the program zero-fills outgoing numeric fields that are not blank. (Otherwise, a “0” in a numeric field is treated as if no data was sent.)
- Split Multiple Files Once – when selected, temporary split files are created only once when the Multiple Files option is selected.

- Output the Elapsed Time – when selected, the program saves the elapsed time of the run into the trace file when Short Trace is selected on the Option 1 tab. (The elapsed time is automatically saved when Long Trace is selected, but you must select this option to have the time saved when Short Trace is selected.)
- No UNG, UNE Segments – when selected, the program does not create UNG or UNE segments for an EDIFACT message.
- Substitute Company Directory – allows you to override the location of the company directory. You can Browse for this.
- Substitute Input Filename – allows you to override the name of the input file that contains the field designated as the trading partner ID. (You set this as a field attribute when you defined your application records.) You can Browse for this.
- Substitute User File Directory – allows you to override the directory location of the input file. You can Browse for this.
- Substitute Map and TP Directory – allows you to specify a single location for the trading partner files and the map files. You can Browse for this.
- Temporary Files Directory – allows you to specify the location of the temporary split files that are created when the Multiple Files option is selected. You can Browse for this.
- ST03(X12) – use only for X12 maps. ST03 (Implementation Convention Reference) was introduced as an optional element in X12 version 4020. Enter the value that you want placed in the ST03 element in the outgoing X12 transactions produced by this map.
- Create Batch Command File button – creates a DOS batch command file to automatically run the map from a command line. After you have selected all of the options for running the map, select Create Batch Command File button, and the Batch File Name window displays.

Enter the full path file name containing the runmap script and select OK. The program automatically creates a file that includes remarks, the runmap executable, and switches that correspond to the options selected on the tabs of the Run Outbound Map window.

A sample batch command file for an outbound map would appear as follows:

```
REM Windows/UNIX Outbound Run Command
```

```
REM Outbound Trace and Log files will be in
```

```
c:\maps\megaproject\bigcompany  
  
wrmo32 c:\temp.x12 880OUT GP -t ANY -dg  
c:\maps\megaproject\bigcompany -dt  
c:\maps\megaproject\bigcompany -nt -ne -wx 0
```

The batch command file can be executed separately from EMap, be incorporated in another batch command file, or be called from another application. You can edit the switches in the file, and you can cut and paste information from the file – such as the runmap switches – for use elsewhere. To run the map from a DOS command line, you must add .bat to the file name. (If you edit the file from Windows Explorer, you must remove the .bat extension. If you do not remove it, selecting the file name causes the program to execute.)

If you are using an EMap batch command file in a UNIX environment, replace wrmo32 (the runmap executable) with rmapout. If you are using an EMap batch command file in a Windows NT environment and the map must execute before other processing continues, insert && before the wrmo32 command line.

- Archive Transaction Log – saves a copy of the current transaction log to a special file. When you choose this option, a runmap window displays.

Enter in the text box on this window the full path file name in which you want to save the current transaction log and select OK.

- Delete Transaction Log – permanently removes the contents of the transaction log file. Since new entries to the transaction log are appended to entries, rather than overwriting them, it is important to control the size of this log by periodically purging the contents.

Note If you want to save the contents of the log, you should archive it prior to deleting it. Once you have deleted the log, you cannot recover it.

You receive no message saying that the log has been deleted.

- Create Adapter Configuration File button – when selected, EMap prompts you through a series of dialogs to create a customized configuration file. For a detailed description of these dialogs and which options to select, see “Create adapter configuration file” on page 302.

File Alias tab

On this window, you create aliases for application files that are hard-coded in the map (the file names are written to the .map file when the map is generated), allowing you to dynamically change the files at runtime. (This is the `-af` switch explained in ECRTP Reference Guide. There is also a `-du` switch that allows you to provide a substitute for the directory in which the application files are located. These switches allow you to adapt your maps between machines and to different platforms.)

❖ Adding a file alias

- 1 Select Add to search for a file.

The Open window displays.

- 2 When you find the map file that you want to replace at runtime, double-click it and the file is automatically entered under Files inside Map.

- 3 Select the right arrow

The Open window displays.

- 4 When you find the substitute file that you want to use at runtime, double-click it and the file is automatically entered under Files during Run Time.

You have now created an alias for the original application file. When the program encounters the original application file, it uses the new alias application file in place of the original file.

❖ Deleting a file alias

- Highlight the alias pair and select Delete.

❖ Editing a file alias

- 1 Highlight a file and select Edit button.

The runmap window displays.

- 2 Edit the full path and select OK.

ODBC Alias tab

On this window, you can create aliases for ODBC connection strings that are hard-coded in the map (the DSNs are imbedded in the .map file when the map is generated), allowing you to change the files dynamically at runtime. (This is the `-ad` switch explained in ECRTF Reference Guide. This switch allows you to adapt your maps between machines and to different platforms.)

❖ **Creating an ODBC alias**

- 1 Select Add

The runmap window displays.

- 2 Enter the DSN that you want to replace at runtime and select OK.

The DSN is automatically entered under ODBC Connection Strings inside Map.

- 3 Select the right arrow.

The runmap window displays.

- 4 Enter the DSN that you want to use at runtime and select OK.

The DSN is automatically entered under ODBC Connection Strings during Run Time.

You have now created an alias for the original Driver Connect String. When the program encounters the original DSN, it uses the new alias DSN in its place.

❖ **Deleting an ODBC alias**

- To remove a pair of DSNs on one line, highlight the line and select Delete.

❖ **Editing an ODBC Alias**

- 1 Highlight the DSN you want to edit and select Edit.

The ODBC Alias window displays.

- 2 On this window, you can change the name of the DSN.

Parameters tab

The Parameters tab is used to pass information to the RTP. The information is passed in the form of values to be substituted for parameters that are used by the map at run time. The parameters are originally created in an Assignment rule command. A parameter can be a field in a record, a memory variable, or a system variable. ECTMap allows you to use up to 20 such parameters. On the Parameters tab, you specify the values to be substituted for parameters in this instance of running this particular map.

By imbedding parameters in a map, rather than actual values, you can use the same map for different trading partners or different purposes. For example, you might use a parameter for the directory location of a file that you read or write in the map. Or, the map may serve dual purposes, and you could pass in a flag that specifies the purpose of the map.

❖ Entering a value for a parameter

- 1 Highlight a parameter and select the right arrow

The Runmap window displays.

- 2 Enter the value for the parameter and select OK.

You return to the Run Inbound Map window, and the value you entered displays across from the parameter for which it will be substituted at run time.

❖ Modifying the value entered for a parameter

- 1 Highlight the value and select Edit.

The Runmap window displays with the current value of the parameter displayed in the text box.

- 2 Edit the value and select OK.

You return to the Run Inbound Map window, and the modified value displays.

❖ Deleting the value entered for a parameter

- Highlight the value and select Delete.

The value is removed from the Run Inbound Map window.

I/O Redirect tab

The I/O Redirect tab allows you to use memory locations - rather than files - for reading, writing, and storing data during run time. This option has many uses. For example, you use this option when the input to the map comes from an entry on a terminal or over the Internet. With this option, the same map can be used with files and with memory – by simply redirecting the input, output, and storage files to memory locations at run time.

On this window, you can specify that:

- Standard Input is used in place of a specified input file.
- Standard Output is used in place of a specified output file.
- Internal Memory is used in place of a specified storage file.

You can add, notify, or delete any of the three I/O options.

❖ Adding a new entry for any of the three I/O options

- 1 Select Add next to the option you want to change.

The Directories/Mailboxes window displays.

- 2 Either double-click the name of the file for which a memory location is used

or

Enter it in the File name text box.

- 3 Select OK.

You return to the Run Inbound Map window and the file name displays in the appropriate section of the window.

❖ Modifying the name of an input file, output file, or storage file for which a memory location is used

- 1 Highlight the file on the window. Select Edit.

The Redirect window displays. The full path location of the current file displays in the text box.

- 2 Make changes to the entry and select OK.

You return to the Run Inbound Map window, and the modified file location displays in the appropriate section of the window.

❖ **Deleting the name of an input file, output file, or storage file for which a memory location is used**

- Highlight the file and select Delete.

The file is removed from the Run Inbound Map window.

Web Script tab

Enter the following on the Web Script tab of the Run Outbound Map window:

- **Script Type** – identifies where the script resides:
 - **ASP** – an Active Server Page contains the script.
 - **CGI for NT** – the script is contained in a Common Gateway Interface to the Windows NT operating system.
 - **CGI for UNIX** – the script is contained in a Common Gateway Interface to a UNIX operating system
- **Executable** – select to toggle between compiled and uncompiled code:
 - **Uncompiled C Code** – The script is uncompiled C code. You can make changes to the code, but you must compile it before you use it. The map switches are in the code itself.
 - **Compiled Executable Code** – The script is an executable file that has already been compiled. The map switches are in a separate data file outside the code. You cannot make changes to the code.
- **Template Script File** – the full path location of the template script file for the Script Type selected above.

You can Browse for this on the Open window. On the Open window, the program automatically selects the Files of Type that corresponds to the Script Type you selected. Double-click the template file and it is automatically entered in the Template Script File text box.

You can Edit the script file if you have chosen to use ASP or uncompiled C code (CGI for NT or CGI for UNIX) as the template file. The script file displays on a Notepad window. You can perform any actions that are normally available with Notepad. Make changes to the script file and then save it.

- **Script File** – the full path location of the actual script file that is used in the program. It is created from the Template Script File.

You can Browse for this on the Open window. On the Open window, the program automatically selects the Files of Type that corresponds to the Script Type you selected. Double-click the script file and it is automatically entered in the Script File text box.

You can Edit the script file if you have chosen to use ASP or uncompiled C code (CGI for NT or CGI for UNIX). The script file displays on a Notepad window. You can perform any actions that are normally available with Notepad. Make changes to the script file and then save it.

When you entered the required information, select the Create Script button. The program automatically creates the script.

Running an outbound map

After you have entered all of the required information on the seven tabs of the Run Outbound Map window,

❖ Running an outbound map

- Select Run Map at the bottom of the window.

The RTP begins processing the application data. You do not receive a message on the window telling that your map has run successfully, but you can select View Trace at the bottom of the window to verify the outcome.

Viewing the trace file for an outbound map

The View Trace button allows you to review the events that took place during the translation process, as they were written to the trace file. The trace file tells you whether the map has run successfully or failed. There are three options for the trace file – short, long, and none. You set this option under Trace Type on the Option 1 tab of the Run Outbound Map window. When you select View Trace, the outgoing.err – Notepad window displays for non-ODBC logs and the TrNNN.dat – Notepad window displays for ODBC logs.

If you choose Short Trace and the map:

- Does not encounter errors, the trace file displays a message telling you that there are 0 errors and that the map ran is complete.
- Encounters any errors other than a fatal error, the trace file lists and gives a summary of all the errors that occurred and displays a message telling you that the map run is complete.

- Encounters a fatal error, the trace file lists and gives a summary of all the errors that occurred up to the point that the fatal error occurred. All processing of data stops at that point so that there may be other errors beyond that point.

If you choose Long Trace and the map:

- Does not encounter errors, the trace file lists each step that took place in the mapping process and displays a message at the end telling you that there are 0 errors and that the map run is complete.
- Encounters any errors other than a fatal error, the trace file lists each step that took place in the mapping process and displays error messages at the appropriate points in the listing. At the end, it gives a summary of all the errors that occurred and displays a message telling you that the map run is complete.
- Encounters a fatal error, the trace file lists each step that took place in the mapping process and displays error messages at the appropriate points in the listing. At the point that the fatal error occurred, all processing of data stops so that there may be other errors beyond that point.

If you choose No Trace, the trace file is not created.

Viewing the transaction log for an outbound map

The View Trans Log button allows you to review the information in the log file. The amount of information in the log depends on the Log Type option (Text Log, No Log, or Expanded Text Log) that you selected on the Required tab of the Run Outbound Map window. The transaction log tells you whether the map ran successfully or failed, but it also performs several other functions. The results of any Write Log rule commands that were used in the map display. When you select View Trans Log, the translog.out – Notepad window displays for non-ODBC logs and the View Transaction Log with Run ID: <run ID #> window displays for ODBC logs.

The following displays:

- A list of the errors and warnings that occurred during map execution.
- The results of any user-generated entries created with Write Log rule commands.
- System-generated messages for each ST and SE segment.

Viewing the EDI file for an outbound map

From the Run Outbound Map window, you can select View EDI File to display the EDI Viewer window. On this window, you can print the EDI file, find character strings in the file, replace character strings, view the file in a “wrapped” or “unwrapped” format, and save the file. You can edit the EDI file if required – for example, if errors occurred and corrections had to be made. You can also open another EDI file and perform the same operations on it.

From the File menu on the EDI Viewer window, you can Open another EDI file (or any file), Save (or Save As) the open file, or Print the open file.

- When you choose Open from the File menu on the EDI Viewer window, the Browse – EDI File window displays.

From the Edit menu, you can either Find a character string or you can find a string and Replace it with another character string.

- If you choose Find, the Find dialog box displays. Enter the character string to find in the Find What text box and press Enter. The character string for which you are searching is highlighted. To find another occurrence of the character string, select Find Next or select Cancel to exit.
- If you choose Replace, the Find dialog box displays, with a new Replace With text box. Enter the character string to replace in the Find What text box and the character string to be substituted in the Replace With text box. Press the Replace key. The character string is replaced, displayed, and highlighted. To replace another occurrence of the character string, select Find Next and then Replace again. To exit the dialog box, select Cancel.

From the Options menu, you can choose to View EDI in a Blocked or an Unblocked format.

- If you choose to view the EDI file in a blocked format, the EDI Block Size dialog box displays. In the text box, enter a block size between 40 and 300 and select OK. The EDI file in the EDI Viewer window now displays in blocks the length that you just specified.

The EDI data displays in an “unwrapped” format. Each segment begins on a separate line and has an incremental line number. In addition, position markers that appear above every 10th character help you to easily identify the character positions within the segments.

You can “wrap” the “unwrapped” EDI data by selecting Wrap EDI. The EDI data appears as one long wrapped record on the EDI Viewer window.

- If you select Options on the EDI Viewer window, the Options window displays. On the Options window, you can set the starting character (Start Byte), the ending character (End Byte), and length (Wrap Size) of the EDI records you are viewing.

Running an any-to-any map

When you choose Run Map from the Build menu and the current map is an any-to-any map, the Run Any-to-Any Map window displays. The Run Any-to-Any Map window has eight tabs: Required, Option 1, Option 2, File Alias, ODBC Alias, Parameters, I/O Redirect, and Web Script. There are also six buttons at the bottom of the window: Run Map, View Trace, View Trans Log, View EDI File, Cancel, and Help.

Required tab

Enter the following on the Required tab of the Run Any-to-Any Map window:

- Map Name – the name of the any-to-any map to be run.
- Map Name Extension – the unique 8-character extension that identifies this map from any other maps using the same transaction. For more information see, “Creating a new trade agreement” on page 67.
- Map Directory – the full path of the directory containing the generated map (.map file). You can Browse for this.

While the RTP is processing the EDI data, it places information into a log file. The Log Type specifies how that information is recorded. If you set up an ODBC-compliant log database on the Maps DSN tab of the New Map window when you created the map, the File Type must be ODBC Log. If you set up a non-ODBC log database, you have three choices - Text Log, No Log, or Expanded Text Log. If you choose Text Log, the program writes entries to a short log (containing fewer fields than the expanded text log) when the map is run. If you choose No Log, the program does not write entries to the log when the map is run. If you choose Expanded Text Log, the program writes entries to a sequential log file that contains all the same fields as the ODBC log.

- Non-ODBC Trading Partner – select if you want the RTP to run against a dBase III trading partner database file. You must click this button if you did not set up an ODBC-compliant trading partner database on the Map DSN tab of the Maps window. You may also click this button when you set up an ODBC database, but prefer to run the RTP against a dBase III file. Users sometimes choose the latter option because the RTP runs faster with the dBase III file and is portable to other platforms that do not support ODBC drivers.
- Run EDI Adapter – select to run ECRTP as an adapter with e-Biz 2000, e-Biz Integrator, or MQSI. When this box is selected, enter in the full directory path that contains the Acquire, Deliver, or Process mode configuration file. For more information about modes and configuration files, see ECRTP Reference Guide.
- Trading Partner Directory – required if you selected Non-ODBC Trading Partner. This directory contains your dBase III trading partner (customer.dbf) and trade agreement (tradstat.dbf) files. You can enter the directory path or search for it using the Browse button.
- Store Run Time Trading Partner Data button – active only when Non-ODBC Trading Partner is selected. When you select this button, the system creates or overwrites information in the dBase III trading partner file (customer.dbf) and trade agreement file (tradstat.dbf) with information from the Access trading partner table and trade agreement table. You use this button in two instances:
 - Before you run the map for the first time, you must initially populate the dBase III database file. (When you select this button, the program creates customer.dbf and tradstat.dbf, but the wixset.dat file is not created until you select the Run Map button. The program needs to know which of the multiple company profiles to use for the wixset.dat file, and you enter that information on the Option 1 tab.)
 - Each time you perform actions within EMap that affect trading partner information, you must transfer these changes from the Access database to the dBase III file.

It is very important to first select the Retrieve Run Time Trading Partner Data button before selecting the Store Run Time Trading Partner Data button to prevent the loss of control counts. (See the explanation for the Retrieve Run Time Trading Partner Data button below.)
- The Retrieve Run Time Trading Partner Data button is active only when Non-ODBC Trading Partner is selected.

When you select this button, the system retrieves the control count information (ISA_IN_NO, ISA_OUT_NO, and GSNO) from the dBase III trading partner file (customer.dbf) and uses it to update the control count information in the Access trading partner database (tp.mdb).

When you run the RTP using a dBase III trading partner file, the control counts are incremented and saved in the dBase III file, but not in the Access database. As a result, even though the control counts have been saved properly by the system, the information displayed on the window and in reports is not always correct - because the information used in reports and on windows is retrieved from the Access database. You can remedy this potential problem by selecting the Retrieve Run Time Trading Partner Data button each time you have finished running your maps. When you do this, you ensure that the control counts in the Access database are always the same as those in the dBase III file, and that all windows and reports always reflect up-to-date control count information. (However, if you have ever previously incorrectly overwritten the dBase III file without first retrieving the control counts, the control count information is incorrect in both places.)

Note If you are running the RTP against a dBase III file (Non-ODBC Trading Partner is selected), always select Retrieve Run Time Trading Partner Data before you select Store Run Time Trading Partner Data, in order to preserve the correct control counts used in the EDI envelopes. As an extra precaution, you can routinely select Retrieve Run Time Trading Partner Data after each execution of RunMap. We recommend using an ODBC Trading Partner database to avoid potential problems.

Option 1 tab

Enter the following on the Option 1 tab of the Run Any-to-Any Map window:

- No Trading Partner – when selected when you run an any-to-any map without using a trading partner. The map does not require trading partner records in order to run, although they may be present. When this box is selected, none of the four check boxes below it are active.
- All Trading Partner Default – when selected, the program defaults to ALL Trade Partner if it does not find a trading partner/trade agreement match when the map is run.

- Ignore Trading Partner Mailbox – when selected, the mailbox set up for the trading partner on the General tab of the Trading Partner window is not used. The any-to-any EDI files go to the Output EDI File location specified on the Required tab of the Run Any-to-Any Map window. (This applies to writing EDI files, not reading them.)
- Lookup Trading Partner from ECRTP Log – when selected, triggers a lookup against the trading partner database. If a trading partner match is found, outbound processing uses the entry to populate the EDI envelope.
- Update All Trading Partner Record – when selected, only the control counts of the ALL Trade Partner are updated. (The All Trading Partner Default must be selected.) This function is used when getting control numbers during concurrent runs.
- Ignore Trade Agreement Mailbox – when selected, the override mailbox set up on the Overrides tab of the Trade Agreement window is not used.
- Trace Type – can be Short Trace, Long Trace, or No Trace.
 - If you choose Short Trace – the program produces a listing of only the errors that were encountered during mapping or the message “Maprun complete. No errors detected.” You would generally choose this option after debugging the map.
 - If you choose Long Trace – the program produces a very detailed listing of the program activities that took place during mapping. You would generally choose this option while you are debugging the map.
 - If you choose No Trace – the program produces no listing. You would generally choose this option after the map goes into production.
- Company Identification – lets you associate a company profile with this map. You create company profiles on the Company ID window - each associated with a Profile Number and each a unique “Sender ID”. Choose Browse, and the Select Company Id window displays, with a listing of all company profiles available. Double-click a profile to select it, and the profile number associated with the profile is entered in this text box.
- Max Memory Cross Reference – the maximum allowable number of entries in a cross-reference table for memory lookups. If the entries in any single table exceed the number entered in this text box, then that table is not stored in memory and the lookups for that table go to disk. The default size is 10,000 entries. (This option is ignored when the RTP is run on a Unix machine, where all tables are stored in memory.)

- **Number of Maps in Memory** – the maximum number of maps that can be cached in memory at one time. An entry in this text box informs the RTP to keep multiple maps open simultaneously, up to the number entered in this text box. If no entry is made, the default of 0 maps is used and the program does not cache any maps in memory. Once the maximum number of open maps is reached, each new map must replace an open map. The program uses an algorithm to determine which of the open maps is the least frequently used and replaces that map with the new map.
- **Number of ODBC Cache** – Specifies the maximum number of ODBC connections cached. The default is 0.

Option 2 tab

Enter the following on the Option 2 tab of the Run Any-to-Any Map window:

- **Map Numeric Zero** – when selected, the program zero-fills outgoing numeric fields that are not blank. (Otherwise, a “0” in a numeric field is treated as if no data was sent.)
- **Split Multiple Files Once** – when selected, temporary split files are created only once when the Multiple Files option is selected.
- **Output the Elapsed Time** – when selected, the program saves the elapsed time of the run into the trace file when Short Trace is selected on the Option 1 tab. (The elapsed time is automatically saved when Long Trace is selected, but you must select this option to have the time saved when Short Trace is selected.)
- **Substitute Company Directory** – allows you to override the location of the company directory. You can Browse for this.
- **Substitute Input Filename** – allows you to override the name of the input file that contains the field designated as the trading partner ID. (You set this as a field attribute when you defined your application records.) You can Browse for this.
- **Substitute User File Directory** – allows you to override the directory location of the input file. You can Browse for this.
- **Substitute Map and TP Directory** – allows you to specify a single location for the trading partner files and the map files. You can Browse for this.
- **Temporary Files Directory** – allows you to specify the location of the temporary split files that are created when the Multiple Files option is selected. You can Browse for this.

- Create Batch Command File button – creates a DOS batch command file to automatically run the map from a command line. After you have selected all of the options for running the map, select Create Batch Command File button, and the Batch File Name window displays.

Enter the full path file name containing the runmap script and select OK. The program automatically creates a file that includes remarks, the runmap executable, and switches that correspond to the options selected on the tabs of the Run Any-to-Any Map window.

A sample batch command file for an any-to-any map would appear as follows:

```
REM Windows/UNIX Any-to-Any Run Command

REM Any-to-Any Trace and Log files will be in
c:\maps\megaproject\bigcompany

wrmo32 c:\temp.x12 880OUT GP -t ANY -dg
c:\maps\megaproject\bigcompany -dt
c:\maps\megaproject\bigcompany -nt -ne -wx 0
```

The batch command file can be executed separately from EMap, be incorporated in another batch command file, or be called from another application. You can edit the switches in the file, and you can cut and paste information from the file – such as the runmap switches – for use elsewhere. To run the map from a DOS command line, you must add .bat to the file name. (If you edit the file from Windows Explorer, you must remove the .bat extension. If you do not remove it, selecting the file name causes the program to execute.)

If you are using an EMap batch command file in a UNIX environment, replace wrmo32 (the runmap executable) with rmapout. If you are using an EMap batch command file in a Windows NT environment and the map must execute before other processing continues, insert && before the wrmo32 command line.

- Archive Transaction Log – saves a copy of the current transaction log to a special file. When you choose this option, a runmap window displays.

Enter in the text box on this window the full path file name in which you want to save the current transaction log and select OK.

- Delete Transaction Log – permanently removes the contents of the transaction log file. Since new entries to the transaction log are appended to entries, rather than overwriting them, it is important to control the size of this log by periodically purging the contents.

Note If you want to save the contents of the log, you should archive it prior to deleting it. Once you have deleted the log, you cannot recover it.

You receive no message saying that the log has been deleted.

- Create Adapter Configuration File button – when is selected, EMap prompts you through a series of dialogs to create a customized configuration file. For a detailed description of these dialogs and which options to select, see “Create adapter configuration file” on page 302.

File Alias tab

On this window, you create aliases for application files that are hard-coded in the map (the file names are written to the .map file when the map is generated), allowing you to dynamically change the files at runtime.

❖ Adding a file alias

- 1 Select Add to search for a file.

The Open window displays.

- 2 When you find the map file that you want to replace at runtime, double-click it and the file is automatically entered under Files inside Map.

- 3 Select the right arrow

The Open window displays.

- 4 When you find the substitute file that you want to use at runtime, double-click it and the file is automatically entered under Files during Run Time.

You have now created an alias for the original application file. When the program encounters the original application file, it uses the new alias application file in place of the original file.

❖ Deleting a file alias

- Highlight the alias pair and select Delete.

❖ **Editing a file alias**

- 1 Highlight a file and select Edit button.
The runmap window displays.
- 2 Edit the full path and select OK.

ODBC Alias tab

On this window, you can create aliases for ODBC connection strings that are hard-coded in the map (the DSNs are imbedded in the .map file when the map is generated), allowing you to change the files dynamically at runtime. (This is the `-ad` switch explained in EC RTP Reference Guide. This switch allows you to adapt your maps between machines and to different platforms.)

❖ **Creating an ODBC alias**

- 1 Select Add
The runmap window displays.
- 2 Enter the DSN that you want to replace at runtime and select OK.
The DSN is automatically entered under ODBC Connection Strings inside Map.
- 3 Select the right arrow.
The runmap window displays.
- 4 Enter the DSN that you want to use at runtime and select OK.
The DSN is automatically entered under ODBC Connection Strings during Run Time.

You have now created an alias for the original Driver Connect String. When the program encounters the original DSN, it uses the new alias DSN in its place.

❖ **Deleting an ODBC alias**

- To remove a pair of DSNs on one line, highlight the line and select Delete.

❖ **Editing an ODBC Alias**

- 1 Highlight the DSN you want to edit and select Edit.
The ODBC Alias window displays.
- 2 On this window, you can change the name of the DSN.

Parameters tab

The Parameters tab is used to pass information to the RTP. The information is passed in the form of values to be substituted for parameters that are used by the map at run time. The parameters are originally created in an Assignment rule command. A parameter can be a field in a record, a memory variable, or a system variable. ECTMap allows you to use up to 20 such parameters. On the Parameters tab, you specify the values to be substituted for parameters in this instance of running this particular map.

By imbedding parameters in a map, rather than actual values, you can use the same map for different trading partners or different purposes. For example, you might use a parameter for the directory location of a file that you read or write in the map. Or, the map may serve dual purposes, and you could pass in a flag that specifies the purpose of the map.

❖ **Entering a parameter value**

- 1 Highlight a parameter and select the right arrow

The Runmap window displays.

- 2 Enter the value for the parameter and select OK.

You return to the Run Inbound Map window, and the value you entered displays across from the parameter for which it will be substituted at run time.

❖ **Modifying a parameter value**

- 1 Highlight the value and select Edit.

The Runmap window displays with the current value of the parameter displayed in the text box.

- 2 Edit the value and select OK.

You return to the Run Inbound Map window, and the modified value displays.

❖ **Deleting a parameter value**

- Highlight the value and select Delete.

The value is removed from the Run Inbound Map window.

I/O Redirect tab

The I/O Redirect tab allows you to use memory locations - rather than files - for reading, writing, and storing data during run time. This option has many uses. For example, you use this option when the input to the map comes from an entry on a terminal or over the Internet. With this option, the same map can be used with files and with memory – by simply redirecting the input, output, and storage files to memory locations at run time.

On this window, you can specify that:

- Standard Input is used in place of a specified input file.
- Standard Output is used in place of a specified output file.
- Internal Memory is used in place of a specified storage file.

You can add, notify, or delete any of the three I/O options.

❖ Adding a new entry for any of the three I/O options

- 1 Select Add next to the option you want to change.

The Directories/Mailboxes window displays.

- 2 Either double-click the name of the file for which a memory location is used

or

Enter it in the File name text box.

- 3 Select OK.

You return to the Run Inbound Map window and the file name displays in the appropriate section of the window.

❖ Modifying the name of an input file, output file, or storage file for which a memory location is used

- 1 Highlight the file on the window. Select Edit.

The Redirect window displays. The full path location of the current file displays in the text box.

- 2 Make changes to the entry and select OK.

You return to the Run Inbound Map window, and the modified file location displays in the appropriate section of the window.

❖ **Deleting the name of an input file, output file, or storage file for which a memory location is used**

- Highlight the file and select Delete.

The file is removed from the Run Inbound Map window.

Web Script tab

Enter the following on the Web Script tab of the Run Any-to-Any Map window:

- **Script Type** – identifies where the script resides:
 - **ASP** – an Active Server Page contains the script.
 - **CGI for NT** – the script is contained in a Common Gateway Interface to the Windows NT operating system.
 - **CGI for UNIX** – the script is contained in a Common Gateway Interface to a UNIX operating system
- **Executable** – Check to toggle between compiled and uncompiled code:
 - **Uncompiled C Code** – The script is uncompiled C code. You can make changes to the code, but you must compile it before you use it. The map switches are in the code itself.
 - **Compiled Executable Code** – The script is an executable file that has already been compiled. The map switches are in a separate data file outside the code. You cannot make changes to the code.
- **Template Script File** – the full path location of the template script file for the Script Type selected above.

You can Browse for this on the Open window. On the Open window, the program automatically selects the Files of Type that corresponds to the Script Type you selected. Double-click the template file and it is automatically entered in the Template Script File text box.

You can Edit the script file if you have chosen to use ASP or uncompiled C code (CGI for NT or CGI for UNIX) as the template file. The script file displays on a Notepad window. You can perform any actions that are normally available with Notepad. Make changes to the script file and then save it.

- **Script File** – the full path location of the actual script file that is used in the program. It is created from the Template Script File.

You can Browse for this on the Open window. On the Open window, the program automatically selects the Files of Type that corresponds to the Script Type you selected. Double-click the script file and it is automatically entered in the Script File text box.

You can Edit the script file if you have chosen to use ASP or uncompiled C code (CGI for NT or CGI for UNIX). The script file displays on a Notepad window. You can perform any actions that are normally available with Notepad. Make changes to the script file and then save it.

Running an any-to-any map

After you have entered all of the required information on the seven tabs of the Run Any-to-Any Map window, you can run your map.

❖ Running an any-to-any map

- Select Run Map.

The program attempts to run the map. You do not receive a message on the window telling that your map has run successfully, but you can select View Trace at the bottom of the window to verify the outcome.

Viewing the trace file for any-to-any maps

The View Trace button allows you to review the events that took place during the translation process, as they were written to the trace file. The trace file tells you whether the map has run successfully or failed. There are three options for the trace file – short, long, and none. You set this option under Trace Type on the Map tab of the Run Any-to-Any Map window. When you select View Trace, the outgoing.err – Notepad window displays for non-ODBC logs and the TrNNN.dat – Notepad window displays for ODBC logs.

If you chose Short Trace and the map:

- Does not encounter errors, the trace file displays a message telling you that there are 0 errors and that the map ran is complete.
- Encounters any errors other than a fatal error, the trace file lists and gives a summary of all the errors that occurred and displays a message telling you that the map run is complete.

- Encounters a fatal error, the trace file lists and gives a summary of all the errors that occurred up to the point that the fatal error occurred. All processing of data stops at that point so that there may be other errors beyond that point.

If you chose Long Trace and the map:

- Does not encounter errors, the trace file lists each step that took place in the mapping process and displays a message at the end telling you that there are 0 errors and that the map run is complete.
- Encounters any errors other than a fatal error, the trace file lists each step that took place in the mapping process and displays error messages at the appropriate points in the listing. At the end, it gives a summary of all the errors that occurred and displays a message telling you that the map run is complete.
- Encounters a fatal error, the trace file lists each step that took place in the mapping process and displays error messages at the appropriate points in the listing. At the point that the fatal error occurred, all processing of data stops so that there may be other errors beyond that point.

If you chose No Trace, the trace file is not created.

Viewing the transaction log for any-to-any maps

The View Trans Log button allows you to review the information placed in the transaction log file. The amount of information in the log is dependent upon the Log Type option (Text Log, No Log, or Expanded Text Log) that you selected on the Required tab of the Run Any-to-Any Map window. The transaction log tells you whether the map has run successfully or failed, but it also performs other functions. The results of any Write Log rule commands that were used in the map display. When you select View Trans Log, the translog.out – Notepad window displays for non-ODBC logs and the View Transaction Log with Run ID: <run ID #> window displays for ODBC logs.

The following displays:

- A list of the errors and warnings that occurred during map execution.
- The results of any user-generated entries created with Write Log rule commands.
- In addition to viewing the information you can also perform actions from this window, using the File, Edit, and Search menu options.

Working with Memory Variables

About this chapter

This chapter discusses memory variables for EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	362
Using the Memory Variables window	363
Using the Memory Variable pane	368

Introduction

Memory variables are temporary variables that are used during the mapping process. They serve many purposes. They can be used to create data that is required for mapping, but is not present in the data being mapped. For example, line item numbers may be required on outbound maps, but may not be present in the user files. Memory variables can also hold the intermediate results of data manipulation during the mapping process. For example, input data may have to be multiplied and then summed before it is mapped; you would use memory variables to hold the intermediate results of these arithmetic operations. Or memory variables may hold data that must be read, stored, and used with data that is read later. For example, you may have to retrieve a percentage from a header record and apply it later to each detail record to calculate a value that will be mapped at the detail level.

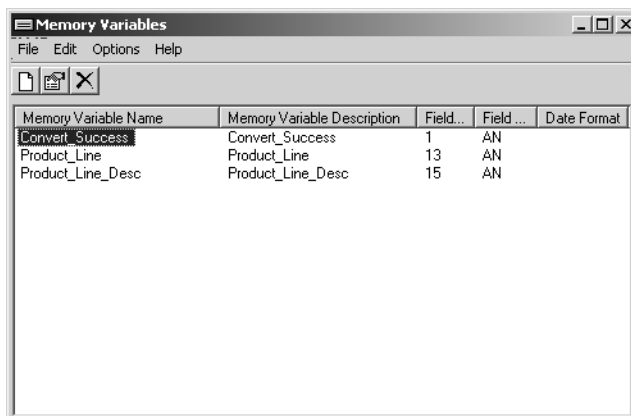
Where to begin

In EMap, memory variable-related actions are initiated from either the Memory Variable window or the Memory Variable pane on the Mapping window.

❖ Displaying the Memory Variable window:

- Choose Tools>Memory Variable from the main menu.

The Memory Variables window displays.



❖ Displaying the Memory Variable pane:

- Do one of the following:
 - Right-click an element and choose Memory Variable from the drop-down list.
 - Choose Map>Memory Variable from the Mapping window menu

Using the Memory Variables window

From the Memory Variables window you can you can:

- Add a new memory variable
- Delete a memory variable
- Import a memory variable
- Modify a memory variable
- Search for a memory variable

Adding memory variables

❖ Adding a new memory variable

- 1 Choose File>New.

The New Memory Variables window displays. There are two tab pages: Fields and Options.

- 2 Enter the following information on the Fields Tab:

- **Memory Variable Name** – the name of the new memory variable you are creating. Although the only explicit naming rule is that there can be no imbedded blanks in the name, memory variables are generally easier to use in mapping if they have a descriptive name. For example, the name could reflect the kind of data being stored, such as `ext_price` or `temp_name`. Or the name could be the name of a related field with a prefix, such as `mv_partno`.
- **Memory Variable Description** – a brief textual description of the memory variable.
- **Field Type** – the type of data in the memory variable. Choose from the drop-down list of the data types. Depending on the data type selected, additional text boxes display.

Field type	Description
AN	Alphanumeric – Field whose characters are comprised of the letters A – Z, numbers 0-9, and special characters (-,/). (ex. ABC12-3D)
CD	Computational (reserved for future use)

Field type	Description
DE	<p>Decimal Explicit – Numeric field containing an actual decimal point (ex. 123.45)</p> <p>In the No. Decimals text box, enter the number of places to the right of the actual decimal point.</p> <hr/> <p>Note To get rid of the decimal point (explicit or implicit) in a rule, multiply the memory variable by 10 raised to the power of the number of decimal places. For example, if there are 3 decimal places, multiply the memory variable by 1000.</p>
DI	<p>Decimal Implicit – Numeric field with the position of an implied decimal point defined by the number of characters to the right of the decimal point. (ex. 12345 with 3 decimal places = 12.345)</p> <p>In the No. Decimals text box, enter the number of places to the right of the implied (not actual) decimal point. Tip: Memory variables defined as DI are stored in the memory variable with leading zeroes and no decimal places. Mappers use this feature to map directly from a memory variable to EDI when they need a zero-filled number.</p> <hr/> <p>Note When you assign a Decimal Explicit (DI) memory variable to an alphanumeric (AN) memory variable, the RTP does a type conversion that puts the decimal point back - so the data retains its original meaning. If you want to get rid of the decimal point and cannot map directly from the memory variable, multiply the memory variable by 10 raised to the power of the number of decimal places. (For example, if there are 2 decimal places, multiply the memory variable by 100.)</p>
DT	<p>Date – Numeric field containing the date in a specified format.</p>

Field type	Description
DF	<p data-bbox="624 262 1241 404">Date Format – choose from the following entries in the drop-down list (where YY are the last 2 digits of the year, YYYY are all 4 digits of the year, MM is the month expressed as a 2-digit number, MON is the 3-alpha abbreviation for the month name, and DD is the 2-digit day of the month):</p> <ul data-bbox="624 418 1241 847" style="list-style-type: none"> <li data-bbox="624 418 772 442">• YYMMDD <li data-bbox="624 456 806 480">• YYYYMMDD <li data-bbox="624 494 772 519">• MMDDYY <li data-bbox="624 532 772 557">• DDMMYY <li data-bbox="624 571 1241 626">• JULIAN – EMap only supports 7-digit Julian dates (4-digit year + 3-digit days) <li data-bbox="624 640 806 664">• MMDDYYYY <li data-bbox="624 678 786 703">• MM/DD/YY <li data-bbox="624 716 786 741">• DD/MM/YY <li data-bbox="624 755 813 779">• MONDDYYYY <li data-bbox="624 793 813 817">• DDMONYYYY <li data-bbox="624 831 813 855">• YYYYMONDD <p data-bbox="624 862 1241 942">Century Minimum Type – defines the manner in which 2-digit years in specified date fields are handled. Select one of the following three choices from the drop-down list:</p> <p data-bbox="624 956 1241 1098">Use global century minimum – use value specified when map was defined to change all 2-digit years to 4-digit years. (All years prior to global century year are treated as next century. All years including and after global century year are treated as current century.)</p> <p data-bbox="624 1112 1241 1432">Use field century minimum – specify a year to be used as global century year (overriding the value specified when map was defined) to change all 2-digit years to 4-digit years. An additional text box appears. Enter year you want to use as global century year for this date field in the Field Century Minimum text box that appears. The Field Century Minimum is a numeric value between 0 and 99 used to determine the century when translating a 2-digit year to a 4-digit year. Every 2-digit year less than or equal to this value is assumed to have a century of 20 and every 2-digit year greater than this value is assumed to have a century of 19. Enter the numeric value in the text box.</p> <p data-bbox="624 1446 1241 1494">No century minimum – specify that no century type logic is to be used. (2-digit years are not converted to 4-digit years.)</p>

Field type	Description
TM	Time – Numeric field containing time in 24-hour format (ex., 9 pm = 2100)
PD	Packed Decimal (reserved for future use)
R	Real Number – Numeric field that allows for a floating decimal
SD	Signed Decimal – Decimal field with the last character corresponding to an overpunch field (for COBOL- based systems) In the No. Decimals text box, enter the number of places to the right of the implied decimal point.
SI	Signed Integer – Numeric field including a plus or minus sign (ex. -123)
UI	Unsigned Integer – Numeric field without a plus or minus sign (ex. 123)

- The Field Length – the length of the memory variable.

3 Enter the following on the Options tab:

- Select the Justification of the memory variable from the drop-down list:

Type	Definition	Description
L	Left-justified	Data begins in the left-most column
R	Right-justified	Data ends in the right-most column
N	No justification	The program uses default justification

The default for characters is left-justification, and the default for numbers is right-justification.

- Today's Date +/- n Days – select this attribute if the memory variable is a calculated date based on today's date. This date is derived from the system time. If you select this attribute, an additional text box displays following Today's Date +/- n Days. Enter the value for the number of days to be added to or subtracted from today's date. The limit for the number of days is between -999 to +999 (-999 to 9999 if you do not include the plus sign).
- Fixed String attribute – check if the memory variable is a fixed string, or constant. If you select this attribute, an additional text box displays following Fixed String. Enter a character string in the text box.

Modifying memory variables

❖ Modifying a memory variable

- 1 Highlight a memory variable on the Memory Variables window and choose Edit>Properties.

The Memory Variable Properties window displays. The two tabs of the Memory Variable Properties window are populated with the current information.

- 2 Modify the Field or Options tabs as needed.

See “Adding memory variables” on page 363 for more information about the Fields and Options tabs.

- 3 Select OK to return to the Memory Variables window.

Deleting memory variables

❖ Deleting a memory variable

- Highlight a memory variable on the Memory Variables window and choose Edit>Delete from the Edit menu.

You are prompted to confirm the memory variable deletion before continuing.

Importing memory variables from another map

❖ Importing memory variables

- 1 Choose Options>Import.

The Select Project and Map window displays.

- 2 Double-click the map that has the memory variable you want to import, or highlight the memory variable and choose Select from the File menu.

The Import Memory Variables window displays.

- 3 Drag and drop a memory variable into the Memory Variables window.

The memory variable you selected is now in the Memory Variables window.

Searching for memory variables

❖ Finding a memory variable

- 1 Choose Find or Find Next from the Edit menu.

The Find dialog box displays.

- 2 Enter the character string you want to search for and select Find.

After the program has found the first occurrence of the character string, you can select Find Next to continue searching for additional occurrences or Cancel to return to the Memory Variables window.

Using the Memory Variable pane

From the Memory Variables window you can you can:

- Add a new memory variable
- Delete a memory variable
- Modify a memory variable

❖ Adding, modifying or deleting the memory variables:

- Right-click a memory variable to display a drop-down list. Choose one of the following:
 - New – the New Memory Variable window displays. Refer to “Adding memory variables” on page 363.)
 - Properties – the Memory Variable Properties window displays. You can modify the memory variable you right-clicked. See the procedure, “Modifying memory variables” on page 367.
 - Delete – you are asked to confirm that you want to permanently remove the memory variable you right-clicked.

Working with System Variables

About this chapter

This chapter discusses system variables in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	370
Working with system variables	370
Using a system variable on the Any-to-Any Map window	375
Alphabetical listing of EMap system variables	376

Introduction

ECMap maintains a set of variables called system variables. These system variables hold information such as company identification data, trading partner data, trade agreement data, EDI envelope data, current date and time, a hard-coded form feed, and record level control for outbound flow. Each system variable has a predefined size and field type and is given a name that reflects the information it contains.

You cannot add a new system variable, nor can you delete a system variable. You cannot change the characteristics of a system variable, but you can modify the value. You can use system variables in certain rule commands, view a list of system variables, and map from a system variable to an output application field in an any-to-any map. You cannot explicitly map from a system variable in a transaction map, but you can assign a system variable to a memory variable and map from the memory variable.

System variables are used to populate EDI envelopes. If you assign a value to a system variable in the Before Rule of the master level in the map flow, you can override the current value of the system variable that would have been inserted into the EDI envelope. While you must use this feature with caution, it can be extremely useful. For example, if you are running a map with the All Trade partner and your application data contains a trading partner ID (such as the DUNS number), you can load that information into the appropriate system variable in the Before Rule and it is used on the EDI envelope.

Working with system variables

This section describes how to

- View a list of system variables
- Use system variables in specific rule commands

Viewing system variables

- ❖ **Viewing the list of system variables**
 - Choose Tools>System Variable from the main menu.

The System Variables window displays.

The name, description, field length, and field data type are displayed for each system variable. You can scroll through the list of system variables, or you can search the list.

❖ **Searching for system variables**

- 1 Choose File>Find from the System Variables window menu.

The Find window displays.

- 2 Type character string you want to search for in the Search String text and select Find.

The program highlights the memory variable containing the first occurrence of the search string. Select Find Next and the program displays and highlights the memory variable containing the next occurrence of the search string. Select Cancel to exit from the Find dialog box.

Using a system variable in a rule command

You can use system variables in the following EMap rule commands:

- Arithmetic
- Assignment
- Concatenate Strings
- If Condition
- Keyed Record I/O
- Perform Rule
- Substring
- Table Conversion

Arithmetic rule command

In an Arithmetic rule command, a system variable can be used as the Left parameter, the Right parameter, and the Destination parameter in the arithmetic expression. It can be used as one, two, or all of these values in one command.

Parameter name	Variable
Left	<ul style="list-style-type: none">• Memory Variable• Record/Field• System Variable
Right	<ul style="list-style-type: none">• Memory Variable• Record/Field• System Variable• Constant
Destination	<ul style="list-style-type: none">• Memory Variable• Record/Field• System Variable

Assignment rule command

When a system variable is used as the Source parameter in a rule command, you assign the value in the system variable to a record/field, a memory variable, or another system variable. When a system variable is the Destination parameter in a rule command, you assign the value of a constant, a record/field, a memory variable, or even another system variable to the system variable.

Parameter name	Variable
Source	<ul style="list-style-type: none">• Memory Variable• Record/Field• System Variable• Constant• Parameter
Destination	<ul style="list-style-type: none">• Memory Variable• Record/Field• System Variable

Concatenate Strings rule command

In a Concatenate Strings rule command, a system variable can be used as the Left Source parameter, the Right Source parameter, and the Destination parameter.

Parameter name	Variable
Left Source	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable • Constant
Right Source	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable • Constant
Destination	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable

If Condition rule command

In an If Condition rule command, a system variable can be used as the Left parameter and the Right parameter in the condition.

Parameter name	Variable
Left	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable • Log Error Occurs
Right	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable • Constant • Error Condition – Use the right arrow on this variable to pick an error condition

Keyed Record I/O rule command

In a Keyed Record I/O rule command, a system variable can be used as the Keyed Field Value.

Parameter name	Variable
Keyed Field Value	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable • Constant

Perform Rule rule command

In a Perform Rule command, a system variable can be used as the Left parameter and the Right parameter in the condition linked to the rule to be performed.

Parameter name	Variable
Left	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable
Right	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable • Constant • Error Condition – Use the right arrow on this variable to pick an error condition

Substring rule command

In a Substring rule command, a system variable can be used as the Source String parameter and the Destination String parameter.

Parameter name	Variable
Source String	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable
Destination String	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable

Table Conversion rule command

In a Table Conversion rule command, a system variable can be used as the Source Variable parameter and the Destination Variable parameter.

Parameter name	Variable
Source Variable	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable
Destination Variable	<ul style="list-style-type: none"> • Memory Variable • Record/Field • System Variable

Using a system variable on the Any-to-Any Map window

In an any-to-any map, a system variable can be mapped directly to an application field. Under Source in the upper left corner of the window, select System Variable. The System Variable pane is superimposed over the Source application data on the left side of the window. This pane contains most of the same information as the System Variables window: name, field length, and data type. You can drag a system variable and drop it onto a Destination application field, or you can use the temporary mapping window in the upper middle part of the window. To use the temporary mapping area, double-click a system variable and then double-click the field to which it will be mapped. The two values appear across from one another under Source and Destination in the upper middle part of the window. You can continue to select pairs of data to be mapped (record fields, system variables, or memory variables). When you have entered all of the mapped pairs, select the arrow between the two sides and the program automatically performs all of the mapping. The data no longer appears in the upper middle section of the window, but it appears in the lower middle section.

Refer to the Any-to-Any Mapping section in the Mapping chapter for more detailed instructions.

Alphabetical listing of EMap system variables

Table 19-1: EMap 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_INTERCHANGE	1	AN	Flag that specifies that an entire interchange should be aborted, on an outbound 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 • = 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 Loaded to on outbound: 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 • = 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 • = 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 Loaded to on outbound: 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>

Name	Size	Type	Description
APP_RECV_CODE	35	AN	Receiver ID code on the group-level envelope Loaded from on outbound: tp/GSID (if blank, use tradstat/RCV_GSID) Loaded to on outbound/loaded from on inbound: X12: GS 03 EDIFACT: UNB S003 0046. UNG S007 0044 HL7: MSH 0006. FHS 0072. BHS 0086
APP_RECV_QUAL	4	AN	Receiver ID code qualifier on the group-level envelope Loaded to on outbound/loaded from on inbound: EDIFACT: UNG S007 0007
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_QUAL	4	AN	Sender ID code qualifier on the group-level envelope Loaded to on outbound/loaded from on inbound: EDIFACT: UNG S006 0007
ASSOC_CODE	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_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
B_RECV_ID	24	AN	Batch receiver ID
B_SEND_ID	24	AN	Batch sender ID
B_VERS	2	AN	Batch version
BIN_NUMB	6	AN	Bin number

Alphabetical listing of EMap system variables

Name	Size	Type	Description
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.]
CLIST_VER	6	AN	Code List Directory Version Number EDIFACT: UNH S009 0110
CNT_AG1	3	AN	Controlling Agency, Coded EDIFACT: UNH S009 0051
CNT_AG2	3	AN	Controlling Agency, Coded EDIFACT: UNH S016 0051
CNT_AG3	3	AN	Controlling Agency, Coded EDIFACT: UNH S017 0051
CNT_AG4	3	AN	Controlling Agency, Coded EDIFACT: UNH S018 0051
COMBINED_BREAK	1	UI	End-of-file (EOF) indicator 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. 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. 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.
COMM_AC_REF	35	AN	Common access reference User-assigned value for outbound use Loaded to on outbound/loaded from on inbound: EDIFACT: UNB S005 0026

Name	Size	Type	Description
COMM_AGM_ID	35	AN	Communication agreement ID User-assigned value for outbound use Loaded to on outbound/loaded from on inbound: EDIFACT: UNB S005 0032
CONTACT1	35	AN	Name of trading partner contact #1 Loaded from on inbound: tp/CONTACT1
CONTACT2	35	AN	Name of trading partner contact #2 Loaded from on inbound: tp/CONTACT2
DATA_ELEM_SEP	1	AN	X12 element delimiter Loaded from on outbound: tp/ELE_DELIMIT (if blank, the program uses the default) Loaded from on inbound: ISA Column 4
EDFT_A_COMP_SEP	1	AN	EDIFACT composite element delimiter – syntax A (The syntax identifier is in UNB S001 0002.) 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 (Changing this value does affect the envelope because the same information is loaded into an internal program variable and that variable is used to populate the envelopes.)
EDFT_A_DEC_IND	1	AN	EDIFACT decimal indicator – syntax A (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.)

Alphabetical listing of ECTMap system variables

Name	Size	Type	Description
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_REL_IND	1	AN	<p>EDIFACT release character – syntax A (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_A_SEG_TERM	1	AN	<p>EDIFACT segment delimiter – syntax A (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>
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)</p> <p>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>

Name	Size	Type	Description
EDFT_B_DEC_IND	1	AN	<p>EDIFACT decimal indicator – syntax B (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>
EDFT_B_ELEM_SEP	1	AN	<p>EDIFACT element delimiter – syntax B (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_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_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>

Alphabetical listing of ECTMap system variables

Name	Size	Type	Description
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 will be 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 will be 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 will be 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 outbound: The file to which the mapped EDI data will be 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>
FIRST_LAST_TRAN	1	AN	<p>First and last transfers</p> <p>Loaded to on outbound/loaded from on inbound: EDIFACT: S010 0073</p>
FORM_FEED	1	AN	<p>Form feed character - <CTRL> I</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 and it will cause a 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 on outbound/loaded from on inbound: X12: GS 04</p> <p>EDIFACT: UNG S004 0017</p>

Name	Size	Type	Description
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_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_ID	6	AN	Functional identifier code Loaded to on outbound/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)
FUNC_ST_REF	35	AN	Value of ST 03 Supplied by the user on outbound as the <value> in the -s3 switch Loaded to on outbound/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	
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

Alphabetical listing of ECTMap 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_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_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)
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
LOG_BYTES	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_ERRS	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_LEVEL	5	UI	Flow level number Loaded from log file on inbound and outbound Available for interrogation by users

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_RECNAME	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_SEGMENT	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
LOG_SUBELEMENT	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
MSG_IMPID	14	AN	Message Implementation Guide Identification EDIFACT: UNH S017 0121
MSG_IMPREL	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_SUBID	14	AN	Message Subset Identification EDIFACT: UNH S016 0115

Alphabetical listing of ECTMap system variables

Name	Size	Type	Description
MSG_SUBREL	3	AN	Message Subset Release Number EDIFACT: UNH S016 0118
MSG_SUBVER	3	AN	Message Subset Version Number EDIFACT: UNH S016 0116
MSG_TYPE	6	AN	Message Type Sub-Function EDIFACT: UNH S009 0113
NCPDP_BATCH_MSG	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. (i.e., 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
PROC_NUMB	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
PROPRIETARY_LENGTH	3	AN	Length of proprietary header
PROPRIETARY_TEXT	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_QUAL	4	AN	Interchange/Outer Envelope Receiver ID code qualifier Loaded from on outbound: tp/IDQUAL (if blank, use tradstat/RCV_IDQUAL) (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

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
RECVR_SUBID	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 will load "X") Loaded to on outbound/loaded from on inbound: X12: GS 07
SCEN_ID	14	AN	Scenario Identification EDIFACT: UNH S018 0127
SCEN_REL	3	AN	Scenario Release Number EDIFACT: UNH S018 0130
SCEN_VER	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
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
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
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: X12: GS 06 EDIFACT:

Alphabetical listing of ECTMap system variables

Name	Size	Type	Description
SEND_CODE	35	AN	<p>Sender ID code on interchange-level envelopes</p> <p>Loaded from on outbound: tp/SND_IDCODE (if blank, use wixset/WIX_IDCODE)</p> <p>Loaded to on outbound/loaded from on inbound: X12: ISA 06</p> <p>EDIFACT: UNB S002 0004</p>
SEND_QUAL	4	AN	<p>Sender ID code qualifier on interchange-level envelopes</p> <p>Loaded from on outbound: tp/SND_IDQUAL (if blank, use wixset/WIX_IDQUAL)</p> <p>Loaded to on outbound/loaded from on inbound: X12: ISA 05</p> <p>EDIFACT: UNB S002 0007</p>
SEND_REV_ROUTE	14	AN	<p>Sender reverse routing code</p> <p>Loaded to on outbound/loaded from on inbound:</p> <p>EDIFACT: UNB S002 0008</p>
SEQ_TRAN	1	AN	<p>Sequence of transfers</p> <p>Loaded to on outbound/loaded from on inbound:</p> <p>EDIFACT: UNH S010 0070</p>
SERV_CODE	6	AN	<p>Service Code List Directory Version Number</p> <p>EDIFACT: UNB S001 0080</p>
SERV_ID	15	AN	Service provider ID
SERV_QUAL	2	AN	Service provider ID qualifier
SKIP_SEGMENT_WRITE	1	AN	<p>Flag that specifies to not write this EDI segment</p> <p>Generated by the program on outbound and inbound: In outbound maps, sets the SKIP_SEGMENT_WRITE SYSVAR in a Before or After segment element EDIT rule to a “Y”, then when the level which contains this segment is generated, the rule will turn it to a “Y” and this segment won't be written. Note SKIP_SEGMENT_WRITE is reset to “ “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 would just affect the first segment at that level.</p> <p>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 will be placed in the EDI OUT File.</p>

Name	Size	Type	Description
SNDR_SUBID	35	AN	Interchange Sender Internal Sub-Identification EDIFACT: UNB S002 0042
SOFT_ID	10	AN	Software/vendor ID
SQL_RETURN	2	UI	SQL return code Returns the status of SQL commands 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.
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)
SYS_ADDR1	35	AN	First line of trading partner street address Loaded to on inbound: tp/ADDR1
SYS_ADDR2	35	AN	Second line of trading partner street Loaded to on inbound: tp/ADDR2
SYS_BILLIDEN	15	AN	Trading partner Bill To ID code Loaded to on inbound: tp/BILLIDEN
SYS_BILLQUAL	2	AN	Trading partner Bill To ID code qualifier Loaded to on inbound: tp/BILLQUAL
SYS_CHG_TRANS	6	AN	Indicator that transaction has changed Generated by the program On Outbound, this triggers a trading partner lookup. (i.e., 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_COUNTRY	25	AN	Country where the trading partner is located Loaded to on inbound: tp/COUNTRY

Alphabetical listing of ECTMap system variables

Name	Size	Type	Description
SYS_DATE	8	Date	Runtime system date Loaded with on inbound and outbound: Computer date
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_IDCODE	35	AN	Trading partner ID code Not currently used
SYS_IDQUAL	2	AN	Trading partner ID code qualifier Not currently used
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_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_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
SYS_SHIPQUAL	2	AN	Trading partner Ship To identification code qualifier Loaded to on inbound: tp/SHIPQUAL
SYS_STATE	2	AN	State where the trading partner is located Loaded to on inbound: tp/STATE
SYS_TIME	4	Time	System Time in format HHMM Loaded with: Computer time

Name	Size	Type	Description
SYS_TRADNO	35	AN	<p>Internal ID number for trading partner generated by the trading partner lookup</p> <p>Loaded from on inbound and outbound: 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 envelope(s) with the appropriate fields in the trading partner database, to find the CUSTNO.</p> <p>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.</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 <message/ transaction set></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>(If this value is changed, it will 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: <2-character transaction code></p>
SYS_USER_FIELD	35	AN	USER DEFINED FIELD
SYS_ZIP	9	AN	<p>Zip code of trading partner location</p> <p>Loaded to on inbound: tp/ZIP</p>
TELEPHONE1	22	AN	<p>Trading partner telephone number #1</p> <p>Loaded to on inbound: tp/TELEPHONE1</p>
TELEPHONE2	22	AN	<p>Trading partner telephone number #2</p> <p>Loaded to on inbound: tp/TELEPHONE2</p>
TEST_IND	1	AN	<p>Transaction mode (T, P, or I)</p> <p>Loaded from on outbound: tradstat/STAT</p> <p>Loaded to on outbound/loaded from on inbound: X12: ISA 15</p> <p>EDIFACT: UN B S005 0035</p>

Name	Size	Type	Description
TOT_ERRS	9	UI	<p>Total number of errors during a map run</p> <p>Loaded from log file on inbound and outbound</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</p> <p>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 will be used to match the Functional Acknowledgment 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 will be used as input to the map that creates the Functional Acknowledgment.</p> <p>On outbound Functional Acknowledgments: 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 in order to prevent sending duplicate acknowledgments.</p> <p>On inbound Functional Acknowledgements: When an Interchange Acknowledgment 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 acknowledgment 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
TRANSMISSION_TYPE	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
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

Alphabetical listing of ECTMap system variables

Name	Size	Type	Description
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 will be written and which is generated by the program using the Run ID number. On outbound: The file to which the mapped EDI data will be 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
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

Name	Size	Type	Description
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_ID	8	AN	Company ADP user identification
WIX_ADP_PASS	8	AN	Company ADP password
WIX_APP_SND_QL	4	AN	
WIX_AUTH_CODE	10	AN	Company authorization code
WIX_AUTH_QUAL	2	AN	Company authorization code qualifier
WIX_B_SEND_ID	24	AN	Company batch sender ID
WIX_BIN_NUMB	6	AN	Company bin number
WIX_COMPANY_NAME	35	AN	Company name
WIX_GSID	35	AN	Company group/inner envelope sender ID code
WIX_IDCODE	35	AN	Company interchange/outer envelope sender ID code
WIX_IDQUAL	2	AN	Company interchange/outer envelope sender ID code qualifier
WIX_MODEM_BAUD	4	AN	Company modem baud rate
WIX_MODEM_PORT	2	AN	Company modem port
WIX_PHONE_TONE	1	AN	Company modem tone
WIX_PROC_NUMB	10	AN	Company process control number
WIX_PRODUCER_ID	12	AN	Company producer ID code
WIX_SECU_CODE	10	AN	Company security code
WIX_SECU_QUAL	2	AN	Company security code qualifier
WIX_SERV_ID	15	AN	Company service ID
WIX_SERV_QUAL	2	AN	Company service ID qualifier
WIX_SNDR_ROUTE	15	AN	Sender internal ID

Alphabetical listing of EMap system variables

Name	Size	Type	Description
WIX_SNDR_SUBID	15	AN	Sender internal subID
WIX_SOFT_ID	10	AN	Company software ID
WIX_TEL_NUMBR	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

About this chapter

This chapter discusses rules in EMap.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	398
Overview of rule commands	399
Creating and using rules	403
Working with rule commands	409
Rule commands reference	411

Introduction

Rules are a series of commands that perform actions. EMap includes a wide variety of commands that can read and write several different types of records, perform logical and arithmetic operations, perform no action at all, invoke other rules, or hold comments. Rules can perform other rules unconditionally, or they can perform other rules conditionally, using `if..then..else` logic.

In EMap, you can add a new rule, modify or delete a rule, copy a rule, or import a rule from another project. Within a rule, you can add a new command, or modify or delete a command. Rules can be attached to the flow and to specific elements.

This chapter is organized into three sections. First, to give you an overview of the broad range of actions available to you through the rule function, the rule commands are listed alphabetically, with a brief summary of the actions that each command performs. Next, step-by-step instructions show you how to access the rules function and guide you through the actions of adding, deleting, and modifying rules and commands. Finally, there is a detailed explanation of the information that must be entered for each individual rule command.

Overview of rule commands

Table 20-1: Rule commands and descriptions

Command	Description
Abort Transaction	Stops the processing of the current EDI transaction and proceeds to the next one
Arithmetic	Performs arithmetic operations, including addition, subtraction, division, and multiplication
Assignment	Assigns a constant value or the value of a field to a variable or another field. Enables data to be moved from a source to a destination
Binary Data Placement	Stores the contents of a BIN segment in either: <ul style="list-style-type: none"> • A file predefined in ECMap • A file and directory combination that is stored in a memory variable.
Check Point	The Check Point (and associated Back out Point) are used to mark the location of processing at various points in the program, and pull out portions of sequential files that are being written
Clear Record	Clears all the fields in a record
Comment	Holds informational or explanatory text. (This field is for documentation purposes only and does not perform any action.)
Concatenate Strings	Allows two data strings to be combined into a single string
Do Nothing	Performs no action. Used in specific instances when a rule is required but no operation needs to take place. (For example, flow requires a Before Rule and an After Rule.) May also be used as a placeholder when testing commands
File Management	Allows file maintenance activities to take place within the RTP. <ul style="list-style-type: none"> • Copy Append Copies data from the source file and appends it to the destination file • Copy Overwrite Copies data from the source file and overwrites the destination file • Rename Renames the source file to the destination file • Close Closes the source file • Delete Deletes the source file
HTML I/O	Allows you to read and write HTML files. <ul style="list-style-type: none"> • Read HTML Data Reads HTML-coded data • Write HTML Form Creates an HTML form.

Command	Description
If Condition	Allows you to enter “if.. then” logic (“If” a specified condition is true, “then” a specified rule is performed.) and “if.. then.. else” logic into rules. (“If” a specified condition is true, “then” a specified rule is performed. Otherwise, the rule in “else” is performed.)
Keyed Record I/O	Used to read and write dBase-compatible files <ul style="list-style-type: none"> • Keyed Read Reads the next record in the file • Keyed Write Writes or appends the record to the file. (If the file does not exist, it is created) • Keyed Rewrite Rewrites the record in the file
Map Level	Executes all of the rules and mapping associated with the segments in a specified level
NDO File Commands	Used to read and write NDO files to and from memory, or pass an NDO files from memory to the Adapter Runtime Environment for EDI. <ul style="list-style-type: none"> • Put NDO File Converts an NDO sequential memory file to an NDO data tree. • Read NDO File On first use, converts an NDO data tree acquired from the Adapter to a sequential memory file. For subsequent uses, the program reads a record in the NDO sequential memory file. • Write NDO File Writes a record to an NDO sequential memory file
Perform Rule	Executes the specified rule number either one time or multiple times, based on a condition
Put File to Queue	Flags a file to be used in place of the queue file specified in the Adapter for EDI configuration file.
Sequential I/O	Allows you to read and write to ASCII text files, also called flat files <ul style="list-style-type: none"> • Read Record Reads the next record in the file • Read Record Once The read command is executed one time per execution of the map. (The next time this rule is encountered for the same record, no action is taken.) • Write Record Write or append the record to the file. (If the file does not exist, it is created.) • Rewrite Record Rewrite the record to the file. (Used for one-record files, such as counter type records.)

Command	Description
SQL	<p>Allows you to perform operations on ODBC databases</p> <ul style="list-style-type: none"> • Close Connection for File Closes the connection to the ODBC database • Close Cursor for File Record Closes the cursors (buffers). Each cursor represents a row or a group of rows from a Select or a Select and Update statement • Commit All Files Commits the inserted or updated rows for all databases. This physically writes the rows into the database so that they cannot be lost, making all changes to the database permanent. • Commit One File Commits the inserted or updated rows for the specified database. This physically writes the row to the database so that it cannot be lost, making changes to the database permanent • Execute SQL Command Executes an SQL command stored in a memory variable. • Fetch after Procedure Call Fetches the row after a Select statement is executed by a stored procedure. • Fetch after Prior Select Moves a single row from the cursor (buffer) to the record definition buffer. This command executes once per map execution. • Fetch Once after Prior Select Moves a single row from the buffer to the record definition buffer after a Select statement. Fetches a new row each time the rule is invoked. • Insert Record into Table Writes the row to the database buffer. • Rollback All Files Backs out all data to the last COMMIT command for all databases. • Rollback One File Backs out current data to the last COMMIT command for one database. • Select and Update Updates row(s) in the database table based on criteria for identifying a row. • Select for Retrieval Queries the ODBC database and creates a cursor (buffer) containing the rows that match the selection (“where”) criteria. Selects different rows each time the rule is invoked • Select Once for Retrieval Queries the database and creates a cursor (buffer) containing the rows that match the selection (“where”) criteria. This command executes once per map execution. • Stored Procedure Call Invokes specialized routines that are inside the database.

Command	Description
Stop Run	Stops the running of the run-time engine
String Operations	<p>Used to check string type, length, and position; to find, move, or replace a string; or to convert the case</p> <ul style="list-style-type: none"> • String Type Used to identify whether a field or variable is alpha or numeric • String Move Used to move a field, variable, or record to a destination field, variable, or record, based on start position and length • String Find Used to find a specific string • String Length Used to return the length of the string in a source field or variable. • String Replace Performs search and replace functions within a field or variable. Used to find the positions of a group of characters in a string • String Justification Used to left- or right-justify a field or variable • String Case Conversion Used to convert a field or variable string from upper- to lower-case or lower- to upper-case
Substring	Allows strings to be extracted from strings
Table Conversion	Allows data to be passed through a cross-reference table for conversion in a rule
User Exit	Allows data to be passed to and from routines that are performed outside of EMap
Write Log	Writes user-defined messages to the Transaction Log (Translog) files.
XML I/O	<p>Allows you to read and write XML files:</p> <ul style="list-style-type: none"> • Write XML Writes XML-coded data (includes start-of-record tag, data, and end-of-record tag) and attributes (within the start-of-record tag) • Write XML w/No End Record Tag Writes XML-coded data without an end-of-record tag (includes start-of-record tag and data) and attributes (within the start-of-record tag) • Write XML End Record Tag Only Writes only the XML end-of-record tag (includes only end-of-record tag) • Write XML Text Writes XML processing instructions, doctypes, comments, etc. • Read XML Reads XML-coded data

Creating and using rules

This section provides instructions for:

- Accessing the rules function
- Adding a rule
- Modifying a rule
- Deleting a rule
- Copying a rule
- Importing a rule from another project
- Modifying the commands within a rule
 - Adding a command
 - Inserting the new command before an existing command
 - Inserting the new command after an existing command
 - Deleting a command
 - Modifying a command.

Getting started

Rules, and the commands that make them up, are created and modified from the Rule Definitions window (or pane).

Access the Rule Definitions window (or pane) by using one of the following:

- From:
 - The main EMap window, choose Rules Definition from the Tools menu
 - The Mapping window (an element must be highlighted), choose Rule from the Element menu
 - Right-click an element and choose Rule from the drop-down list.
 - From:
 - The Map Flow window, choose Rules from the View menu.
 - The Record Flow window, select the up arrow next to any text box on the Rules tab.
- ❖ **Accessing the Rule Definitions window from the main menu**
- Choose Tools>Rules from the main menu.

The Rule Definitions window displays.

❖ **Accessing the Rule Definitions pane from the Mapping window**

- Highlight an element and choose one of the following:
 - Element>Rule>Prior – executed before mapping takes place
 - Element>Rule>Post – executed after mapping takes place
 - Element>Rule>Remove – allows you to remove a current rule association.

or

Right-click an element and choose Rule>Prior, Rule>Post, or Rule>Remove from the drop-down list.

The Rule Definitions pane displays on the Mapping window.

❖ **Accessing the Rule Definitions window from the Map Flow window**

- Choose View>Rules.

The Rule Definitions window displays.

Working with the Rule Definitions window

On the Rule Definitions window, you can add, delete, modify, copy, or import rules using menu options.

Working from the Rules definition pane

You can also add, delete, modify and copy rules from the Rule Definitions pane by right-clicking on a rule and selecting the task you want to perform from the drop-down list.

The following procedures describe how to perform the rules tasks from the Rule Definition window.

Adding modifying, deleting and copying rules

❖ **Adding a new rule**

- 1 Choose File>New Rule.

The Rule Definition – New window displays

- 2 Enter a unique Rule Number and a brief description for the rule.
- 3 Select OK to return to the Rule Definitions window.

❖ Modifying a rule

- 1 Highlight the rule and choose Rule>Properties from the Rule menu.
The Rule Definition – Properties: window displays.
- 2 Modify the Rule Number and/or the Rule Description.
- 3 Select OK to return to the Rules Definition window.

❖ Deleting a rule

- Highlight the rule and choose Rule>Delete.
You are prompted to confirm the rule deletion.

Note Since the Confirm message box does not identify the rule to be deleted, make sure that you are deleting the correct rule.

❖ Copying a rule

- 1 Highlight the rule and choose Rule>Copy.
The Rule Definition – Copy: window displays.
The Rule Number text box is blank. The description of the rule that you are copying displays in the Rule Description text box.
- 2 In the Rule Number text box, enter the number of the new rule being created. Use the Rule Description that was copied or modify it.
- 3 Select OK to complete the copy and return to the Rules Definition window.

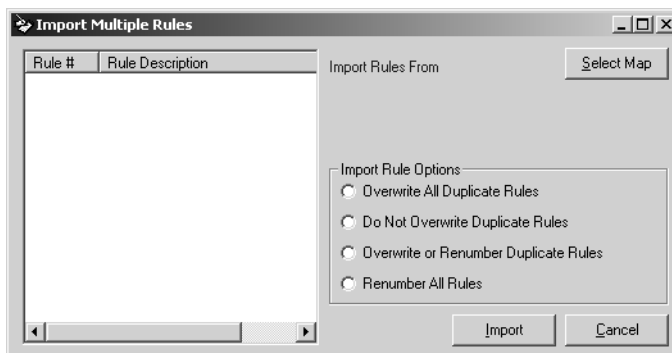
Importing rules

You can import one or more rules from another project. If you encounter rules with the same number as a rule in the current map, you can choose to skip the rule, overwrite the current rule, or renumber the rule you are importing.

❖ Importing rules from another project

- 1 Choose Rule>Import.

The Import Multiple Rules window displays.



- 2 Click the Select Map button.

The Choose Project and Map window displays.

- 3 Highlight a project and map from which to import rules and choose File>Select.

A list of all the rules from the map you selected displays on the Multiple Rules window.

- 4 Select one or more of the rules to import.
- 5 Select one of the following import rule options
 - Overwrite All Duplicate Rules
 - Do Not Overwrite Duplicate Rules
 - Overwrite or Renumber Duplicate Rules
 - Renumber All Rules

Overwriting all rules

If you select the Overwrite All Duplicate Rules option, You will automatically overwrite any rule in the current map with the same rule number as a rule you are importing from another map.

❖ To replace all duplicate rules in the current map

- 1 Select Overwrite All Duplicate Rules.
- 2 Click Import.

A window displays the message: The import rules process has completed. Would you like to view the log?

- 3 Click Yes to display the Import Rule Log window.

Not overwriting any rules

If you select the Do Not Overwrite Duplicate Rules option, you will not import any rule that has the same rule number as a rule in the current map.

❖ To retain the rules in the current map

- 1 Select Do Not Overwrite Duplicate Rules.
- 2 Click Import.

A window displays the message: The import rules process has completed. Would you like to view the log?

- 3 Click Yes to display the Import Rule Log window.

Interactively deciding how to handle duplicate rules

If you select the Overwrite or Renumber Duplicate Rules option, you will interactively specify what action to take for each duplicate rule:

- Overwrite the current map rule with the imported rule
- Skip the imported rule and keep the current map rule
- Keep the rule in the current map and renumber the imported rule to avoid conflicts

If you choose to renumber the imported rules, ECTMap will suggest a candidate rule number to use. The candidate rule number is always equal to the highest rule number in the current map plus one. For example, if the highest rule number in your current map is 750, the program will suggest 751 as the candidate rule number.

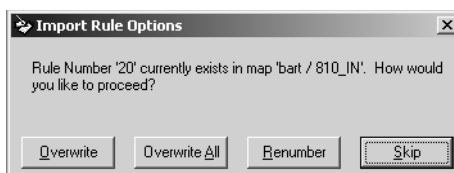
Note

The highest rule number that can be used in a map is 32767. If that number is used in your map, ECTMap will not be able to suggest a candidate number.

❖ To replace all duplicate rules in the current map

- 1 Select Overwrite or Renumber Duplicate Rules
- 2 Click Import.

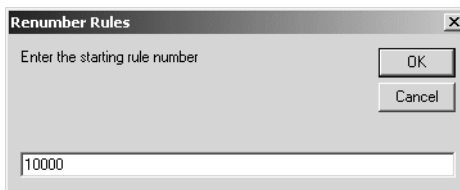
When a duplicate rule is encountered, the Import Rule Options window displays.



The Import Rules window displays the duplicate rule number and prompts you to specify how to proceed.

- 3 Select one of the following:
 - Overwrite – to overwrite the current rule with the one being imported
 - Overwrite All – to overwrite the current rule and all subsequent duplicate rules with the imported rules
 - Renumber – to specify a new number for the imported rule
 - Skip – to leave the current rule in the map and not import the rule

If you select Renumber, the Renumber Duplicate Rule window displays a candidate rule number for the current map.



Enter a number for the duplicate to use.

- 4 Repeat for each duplicate rule number in the current map.

After all duplicate rule numbers have been resolved, a window displays the message: The import rules process has completed. Would you like to view the log?

- 5 Click Yes to display the Import Rule Log window.

Renumbering imported duplicate rules

If you select the Renumber All Rules option, you specify new numbers for all imported rules to avoid any import conflicts.

❖ **To replace all duplicate rules in the current map**

- 1 Select Renumber All Rules
- 2 Click Import.

The Renumber Rules window displays the next available rule number.

- 3 Enter a number from which all subsequent renumbering will start.

After all duplicate rule numbers have been resolved, a window displays the message: `The import rules process has completed. Would you like to view the log?`

- 4 Click Yes to display the Import Rule Log window.

Using the log file

The Import Rule Log window contains the following information:

- Date and time of the import
- Source and destination maps used
- The action taken on each rule

Select Print to print the log file.

Sample log file

```
Rule Import Log
7/20/2004
```

```
Source Map: Metco / INVIN
Destination Map: Premier / 810_IN
```

```
Rule '10' was not imported
Rule '20' was overwritten
Rule '21' was successfully imported
Rule '22' was renumbered to rule '922' and successfully
imported
```

Working with rule commands

On the Rule Definitions window, you can add, delete, or modify rule commands.

If you're working in an Any-to-Any map, you can also open the mapping window from the rules definition window.

❖ **Adding a new rule command**

1 Do one of the following:

- Highlight a rule or command and choose File>New Command.

The new command is entered at the end of the commands in the rule.

- Highlight a command and choose Command>Insert Before to insert a new command before the highlighted rule or Command>Insert After to insert a new command after the highlighted rule.
- Right-click a rule and choose New Command from the drop-down list.
- Right-click a command and choose Insert Before or Insert After from the drop-down list.

The Rule Command – New window displays the number of the current rule and the command line number being assigned to the new command within the rule.

2 Select the command you want to add to the rule from the drop-down list.

Once you choose a command, the Rule Command – New dialog box changes. Depending on the particular command you choose, additional text boxes may be displayed. For more information on rule commands see “Rule commands reference” on page 411.

3 After you have entered all the information required for the particular command you selected, select

- Apply to save it and enter another command.
- OK to save the current command and return to the Rule Definitions window.

❖ **Modifying a rule command**

1 Do one of the following:

- Highlight the command and choose Command>Properties.
- Right-click a command and choose Properties from the drop-down list.

The Rule Command – Properties window displays.

2 Modify information in any of the text boxes.

3 Select OK to return to the Rules Definition window.

❖ **Deleting a rule command**

1 Do one of the following:

- On the Rule Definitions window – choose Delete from the Command menu at the top of the window.
- From the Rule Definitions pane – right-click a command and choose Delete from the drop-down list.

2 You are asked to confirm that you want to delete the command before it is removed permanently.

❖ **Accessing the Any-to-Any Map window**

- Choose Command>Any-to-Any.

The Any-to-Any Map window opens.

This menu option, together with the View Rules button on the bottom of the Any-to-Any Map window, allows you to toggle between the mapping window and the Rule Definitions window.

Rule commands reference

Each type of rule command has its own set of associated information that must be entered. When you choose a command type, the Rule Command window changes. There are different text boxes that reflect the specific information needed to execute the command that you have selected. The final section of this chapter contains a detailed explanation of the Rule Command windows for each of the commands available in EMap.

Abort transaction

The Abort Transaction command stops the processing of the current EDI transaction/message and proceeds to the next one. This command is generally used with conditional logic (error-checking or business logic) to end the processing of an EDI transaction/message based on the presence or absence of specific conditions.

Every time EMap reads or writes an EDI transaction/message (or an interchange or group envelope), it saves the location of current processing in all open output files. If an ODBC database is being used, it does a commit. (commit signals a successful end-of-transaction, letting the system know that all of the updates in the logical unit of work that has just been successfully completed can be made permanent. Any time before a commit occurs, the system can perform a rollback operation. rollback signals an unsuccessful end-of-transaction, telling the system to undo all of the updates made by the logical unit of work.)

When EMap encounters an Abort Transaction command, the program uses the results of the saved location to know where to begin processing or where to place the next output. If an ODBC database is being used, the program performs a rollback to the last commit.

X12, EDIFACT, and HL7 all have interchange level envelopes, group level envelopes, and transactions/ messages. In X12, an ST/SE segment is a transaction/message. In EDIFACT, a UNH or UIH segment is a transaction/message. In HL7, an MSH segment is a transaction/ message.

Inbound maps

On inbound maps, when neither the user nor the system halts the processing of an EDI transaction/message, the program reads the current input EDI transaction/message and maps data to the output file as it is being processed. EMap then reads the next EDI transaction/message. When the new transaction/message is read, the location of processing within the open output file is saved. If the program encounters an Abort Transaction rule, it halts processing of the current transaction/message and sends it to the BAD file. The program backs up to the previously saved location of processing in the output file, and a new transaction/message is read. If data was mapped to the output file before the Abort Transaction command was encountered, the program overwrites the data associated with the aborted transaction/ message. The program continues to read and process new transactions/messages, either mapping them to the output file or moving them to the BAD file. (The directory structure and creation of the BAD file depends on the runtime switches that were set.)

Outbound maps

On outbound maps, when neither the user nor the system halts the processing of the application data, the program maps the current application record and writes it to the EDI output file as it is being processed. EMap then reads the next application record. Whenever the program writes a new EDI transaction/message, the location of processing within the open EDI output file is saved. If the program encounters an Abort Transaction rule, it halts processing of the current application record and backs up to the previously saved location of processing in the output file. A new application record is read and the program begins overwriting any EDI data associated with the aborted transaction (if data was mapped and written before the Abort Transaction command was encountered). The program continues to read and process new application records, mapping them to the output EDI file if no user- or system-generated halts occur.

When the program has finished processing files, it makes sure that all bad data in text files has been overwritten, by checking to see whether the file size is larger than the current location. There is no need to check ODBC files since data is not actually saved to an ODBC database until a COMMIT is encountered. In addition, the program checks all interchange and group level envelopes to make sure that they contain at least one transaction/message. Empty envelopes are not processed.

The Abort Transaction command is usually used in response to conditional logic that has detected an undesirable situation.

Arithmetic

The Arithmetic command allows you to perform four basic arithmetic operations: addition (+), subtraction (-), division (/), and multiplication (*). The general form of the arithmetic command is:

<Left variable> <arithmetic Operator> <Right variable> = <Destination variable>

This command can also be used for date arithmetic, such as:

```
Date1 - Date2 = Number
Date1 +/- Number = Date2
```

Or incremental counters, such as

```
MemVar + 1 = MemVar
```

Variables

- *Left* – choices are:

- **Memory Variable** – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the Left text box.
- **Record/Field** – displays next to the Rule Command window. Double-click a field and the program automatically enters it in the Left text box.
- **System Variable** – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the Left text box.

- **Operator** – choices are:

+	Addition
-	Subtraction
/	Division
*	Multiplication

- **Right** – choices are:
 - **Memory Variable** – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the Right text box.
 - **Record/Field** – displays next to the Rule Command window. Double-click a field and the program automatically enters it in the Right text box.
 - **System Variable** – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the Right text box.
 - **Constant** – Enter a value in the text box on the ECMap window that displays and select OK. The value is automatically entered in the Right text box.
- **Destination** – holds the result of the arithmetic operation. Choose from:
 - **Memory Variable** – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the Destination text box.
 - **Record/Field** – displays next to the Rule Command window. Double-click a field and the program automatically enters it in the Destination text box.

- System Variable – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the Destination text box.

Assignment

The Assignment command assigns a constant value or the value of a field or variable to another field or variable.

Variables

- *Source* – contains the value that is assigned to the Destination variable. Choose from:
 - Memory Variable – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the Source text box.
 - Record/Field – displays next to the Rule Command window. Double-click a field and the program automatically enters it in the Source text box.
 - System Variable – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the Source text box.
 - Constant – Enter a value in the text box on the Constant Value window that displays and select OK. The value is automatically entered in the Source text box.
 - Parameter – Choose a parameter (up to 20 parameters can be used) from the drop-down list and it is entered in the Source text box.
- *Destination* – the object to which the value in the Source variable is assigned. Choose from:
 - Memory Variable – displays next to the Rule Command window. Drag a memory variable and drop it on the Destination text box.
 - Record/Field – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the Destination text box.
 - System Variable – displays next to the Rule Command window. Drag a system variable and drop it on the Destination text box.

Binary Data Placement

The Binary Data Placement command is used to specify a file and directory in which to store the binary data of a BIN segment.

Usage Notes

This command is available for inbound maps.

The Binary Data Placement command applies to the next BIN segment processed. If the command was executed in the Prior rule for a BIN segment, the command would apply to the current segment. However, if the command was executed as an After rule, the command would not apply to the current segment, it would affect the next BIN segment encountered.

The command is only active for one BIN segment. Any additional BIN segments in the transaction will be stored in the default location unless you set up Binary Data Placement command rules for them.

Parameters

Primary Data Location – where you will store the Binary data.

- **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the Binary Data Location text box.

The definition of the memory variable consists of a directory and file location.

- **Select File and Location** – the Directories/Mailboxes window displays. Drag a file and drop it onto the Source text box.

Check Point

The Check Point command is used to set a marker during the processing of data or to back out to the point where the marker was set. The two command types—Check Point and Backout Point—are used together to pull out portions of application files or EDI files that are being written. The Check Point command is very useful for testing the mapping process, because it allows you to back out to a specified level based on the errors that you have received.

You may assign up to five different check points at a time at various levels in the mapping process. Since these check points are dynamic, you may continue to reuse them. Each time you either read or write an EDI transaction/message, the program looks at the check points.

When you choose the Check Point command, a Parameters section appears at the bottom of the Rule Command – New/Properties window. In the Parameters section, there are two text boxes: Point Number (1-5) and Command Type.

When you select the up arrow next to the Point Number text box, you can choose either Memory Variable or Constant.

- If you choose Memory Variable, the Memory Variable window displays with a list of all currently defined memory variables. You can enter a memory variable in the Point Number text box by double-clicking it or by dragging it from the Memory Variable window and dropping it on the Point Number text box. You can select a memory variable, or you can first add a new one and then select it. The memory variable must contain the value 1, 2, 3, 4, or 5.
- If you choose Constant, the Constant Value window displays. Enter 1, 2, 3, 4, or 5 in the Constant text box and select OK. The value is entered in the Point Number text box.

When you select the up arrow next to the Command Type text box, you can choose either Check Point or Backout Point.

- Check Point tells the program to save this file location and associate it with the check point number entered in the text box above. When the program encounters a Backout Point associated with this same check point number, the program will revert back to this location in the processing.
- Backout Point tells the program to return to the location in the processing associated with the check point number in the text box above. The program rewinds all sequential files to the last Check Point.

The Check Point command works somewhat differently for inbound and outbound processing and for sequential and ODBC files.

For outbound maps:

- There are no limitations on how far you can back out a text file.
- You can back out only to the previous transaction/message in an ODBC database. Since a commit is performed each time you read a new transaction, you can perform a rollback only to the prior commit.

For all inbound maps:

- You can back out only to the previous transaction/message. (Each time ECMap reads an EDI transaction/message, the program looks at the check points. If it finds a check point earlier than the prior transaction/message, it changes the check point so that it points to the prior transaction/message.)

ECMap assumes that the Backout command only takes you back to a check point; never forward. As a result, when you back out to a check point that has other check points that occur after it, the program automatically assigns to those later check points the saved location to which you backed out.

Clear

The Clear command clears data from all or multiple fields in a record simultaneously.

The Record To Clear is the record where the fields are cleared. When you select the up arrow, all the currently defined records and fields are displayed on the Record/Fields window. You can drag a record from the Record/Fields window and drop it on the Record To Clear text box on the Rule Command window.

Variables

- *Type of Record To Clear* – choose from:
 - Clear All Fields – clears all the fields in the selected record.
 - Clear Unprotected Fields – clears all the fields in the record except those marked as protected. Protected is an attribute that is set on the Options tab of the New Field or Field Properties window.

Comment

The Comment command allows you to insert informational or explanatory text within your rule. This field is for documentation purposes only and does not perform any action.

When you select the up arrow, the Comment Rule dialog box displays. Enter your comment in the text box and select OK.

Concatenate Strings

The Concatenate Strings command allows you to combine two data strings into a third. The resulting string may be Trimmed (with trailing spaces eliminated) or Untrimmed. Strings may be combined at a fixed position specified by the Placement of Right Variable. The Destination may be either the left or the right source.

Variables

- *Left Source* – contains the first data string to be combined. Choose from:

- **Memory Variable** – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the Left Source text box.
- **Record/Field** – displays next to the Rule Command window. Double-click a field and the program automatically enters the record/field in the Left Source text box.
- **System Variable** – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the Left Source text box.
- **Constant** – enter a value in the text box on the Constant Value window that displays and select OK. The value is automatically entered in the Left Source text box.
- **Right Source** – contains the second data string to be combined. Choose from:
 - **Memory Variable** – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the Right Source text box.
 - **Record/Field** – displays next to the Rule Command window. Double-click a field and the program automatically enters it in the Right Source text box.
 - **System Variable** – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the Right Source text box.
 - **Constant** – enter a value in the text box on the Constant Value window that displays and select OK. The value is automatically entered in the Right Source text box.
- **Placement of Right Variable** – allows you to select the desired placement of the right variable data string:
 - **Delim(iter) Based Position in Dest(ination)** – places the right variable data string after the nth occurrence of a specified delimiter in the left variable data string (may overlay some of the left variable data).

When you make this choice, the Select a Delimiter window displays next to the Rule Command window with a numbered list of ASCII characters. See the appendices for a listing of ASCII characters that can be used as delimiters.

Double-click a delimiter, and the program automatically enters the number of that delimiter in the Placement of Right Variable (now called Delim Based Position in Dest) text box. The Delimiter Occurrence window displays. On that window, enter the number of times (n) the delimiter should appear in the left variable data string before the right variable data string is appended.

- **Memory Variable Position** – places the right variable data string after the position in the left variable data string specified by the value in a memory variable.

When you make this choice, the Memory Variable window displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the Placement of Right Variable (now called Memory Variable Position) text box.

- **Constant Destination Position** – places the right variable data string after the nth character in the left variable data string (may overlay some of the left variable data).

When you make this choice, the Constant Value window displays next to the Rule Command window. Enter a value in the text box, and the program automatically enters it in the Placement of Right Variable (now called Constant Destination Position) text box.

- **Untrimmed Left Variable** – places the right variable data string immediately after the left variable data string in the destination field.
- **Trimmed Left Variable** – trims leading and trailing blanks from the left variable data string and places the right variable data string immediately after it in the destination field.
- **Destination** – holds the combination of the left and right data strings. Choose from:

- **Memory Variable** – displays next to the Rule Command window. Double-click a memory variable, and the program automatically enters it in the Destination text box.
- **Record/Field** – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Double-click a field, and the program automatically enters the record/field in the Destination text box.
- **System Variable** – displays next to the Rule Command window. Double-click a system variable, and the program automatically enters it in the Destination text box.

Do Nothing

This command performs no action.

The Do Nothing command most commonly occurs by itself, in a rule containing only the Do Nothing command. This command is used when a rule is required but no operation needs to take place. For example, the Do Nothing command is often used in flow because flow requires a Before Rule and an After Rule. It may also be used as a placeholder when testing commands.

File Management

The File Management command allows file maintenance activities to take place within the RTP.

Variables

- *File Command* – choose from:
 - Copy Append – makes copies from the source file and appends to the target file.
 - Copy Overwrite – makes copies from the source file and overwrites the target file.
 - Rename – renames the target file using the name of the source file.
 - Close – closes the source file.
 - Delete – deletes the source file.
- *Source* data file location – can be a full directory path or a memory variable that contains a full path:
 - Memory Variable – the Memory Variable window displays. Drag a memory variable and drop it onto the Source text box.
 - File and Location – the Directories/Mailboxes window displays. Drag a file and drop it onto the Source text box. You can also drag and drop a directory, but it must contain only the file you want to use. It cannot be empty or contain multiple files.
- *Target* data file location – can be a full directory path or a memory variable that contains a full path.
 - Memory Variable – the Memory Variable window displays. Drag a memory variable and drop it onto the Target text box.
 - File and Location – the Directories/Mailboxes window displays. Drag a file and drop it onto the Target text box.

HTML I/O

The HTML I/O command allows you to:

- Read HTML Data – reads HTML-coded data.
- Write HTML Form – creates an HTML form.

The Rule Command window changes, depending on whether you are reading or writing HTML data.

Read HTML Data

The Read HTML Data option allows you to read HTML-coded data.

Variables

- *I/O Command* – tells the program that it is reading HTML data.
- *HTML File* – the full directory path to the file containing the record that is read. There are two choices in the drop-down list. If you choose:
 - *Memory Variable* – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable or drag and drop it onto the text box. The memory variable you select must contain a directory/file path, pointing to the record.
 - *Select File and Location* – the Directories (Mailboxes) window displays next to the Rule Command window. Double-click a directory or drag and drop it onto the text box.
- *Record* – contains the HTML data that is read. If you select the Record up arrow, the Records/ Tables window displays next to the Rule Command window. Double-click a record or drag and drop it onto the text box. The record you select must be located in the directory/file path specified in the HTML File text box above.

Write HTML form

The Write HTML Form option allows you to create an HTML form.

Variables

- *I/O Command* – tells the program that it is writing HTML data.
- *Input Template File* – the full directory path to the file containing the template that is used to create the HTML form. There are two choices in the drop-down list. If you choose:

- **Memory Variable** – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable or drag and drop it onto the text box. The memory variable you select must contain a directory/file path, pointing to the template.
- **Select File and Location** – the Directories (Mailboxes) window displays next to the Rule Command window. Double-click a directory or drag and drop it onto the text box.
- **Record** – contains the data used to populate the HTML template that creates the HTML form. If you select the Record up arrow, the Records/Tables window displays next to the Rule Command window. Double-click a record or drag and drop it onto the text box. The record you select must be located in the directory/file path specified in the HTML File text box above.
- **Output HTML File** – the full directory path to the file containing the HTML form that is being created. There are two choices in the drop-down list. If you choose:
 - **Memory Variable** – the Memory Variables window displays next to the Rule Command window. Double-click a memory variable or drag and drop it onto the text box. The memory variable you select must contain a directory/file path, pointing to the file to which the HTML form is written.
 - **Select File and Location** – the Directories (Mailboxes) window displays next to the Rule Command window. Double-click a directory or drag and drop it onto the text box.

If Condition

The If Condition command allows you to enter simple “if...then” (“if” the condition specified is true, the “then” rule is preformed) and “If..then..else” (“if” the condition specified is true, the “then” rule is preformed. Otherwise, the “else” rule is performed.) 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.

Variables

- **Left** – choose from:
 - **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the bottom text box below the Left text box.

- **Record/Field** – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the bottom text box below the Left text box.
- **System Variable** – displays next to the Rule Command window. Drag a system variable and drop it on the bottom text box below the Left text box.

A system variable that returns a status can be used to test for special conditions. For example, the system variable `SQL_RETURN` can be evaluated after each `SQL FETCH` is performed to see if data is present.

- **Log Error Occurs** – when you select **Log Error Occurs**, all the text boxes except **If True** are grayed out. An entry in the error log causes the rule specified in the **If True** text box to be performed. In the rule, you can specify additional actions to be taken when an error is encountered at runtime. For example, a report can be written to a user-defined file if an error occurs.

After an element is mapped or a rule is performed, the program checks to see if an error occurred. If a rule command caused multiple errors, **Log Error Occurs** reports only the last error condition that occurred. (The trace file and the log file have separate messages for each error.) Examples of multiple errors would be if the trading partner record is not found, if the control numbers on inbound don't match, if an incoming map segment does not match, or if the program terminates abnormally with an error message.

- *Logical* – choose from:
 - **EQ** – equal to.
 - **NE** – not equal to.
 - **GT** – greater than.
 - **GE** – greater than or equal to.
 - **LT** – less than.
 - **LE** – less than or equal to.
- *Right* – choose from:
 - **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the bottom text box below the Right text box.

- **Record/Field** – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the bottom text box below the Right text box.
- **System Variable** – displays next to the Rule Command window. Drag a system variable and drop it on the bottom text box below 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 automatically entered in the Right text box.
- **Error Condition** – choose from the drop-down list of error conditions. (Error conditions are represented internally in the system by 2-digit codes. You can use an If Condition to compare one of these internal error codes to the value in a record field, memory variable, or system variable.) The 2-digit codes and explanations for each error condition are shown below:

Number	Code	Explanation
	No Error	Normal completion, with no errors detected.
02	End of File	The program reached the end of the file being processed.
03	Duplicate Key	You are attempting to add a record that already exists.
04	Record Missing	The requested record was not found.
05	File Full	The file space allocated has been exhausted.
06	Device Error	A physical device error has occurred.
07	Open Error	You are attempting to open a file that is already open, or an I/O command was issued before a file was opened.
08	Out of Memory	There is not enough memory available.
09	File Conflict	There is a conflict between file attributes.
10	File Missing	The operating system cannot find a file.
	Other	An unspecified error condition has occurred.

- **If True rule** – performed when the If Condition is true or a Log Error Occurs. You can type the rule number in the If True text box, or you can select the up arrow to search for it. If you select the up arrow, the Rule Definitions window displays. On this window, you can select a rule or add a new one.

- Else rule – performed when the If Condition is not true. You can type the rule number in the Else text box, or you can select the up arrow to search for it. If you select the up arrow, the Rule Definitions window displays. On this window, you can select a rule or add a new one.

Keyed Record I/O

Keyed record commands are generally used only to maintain functionality in maps created in older versions of the EMap software. Keyed record I/O is used with dBase III files, and most current EMap users take advantage of the power and flexibility of ODBC-compliant databases.

When you select the Keyed Record I/O up arrow in the Parameters section of the Rule Command - New window, the Keyed I/O window displays.

Variables

- *Database* – the DBASE FILE check box is permanently checked.
- *Keyed I/O Command* – to use this option, the files must already exist. Choose from:
 - Keyed Read – reads the next record in the file.
 - Keyed Write – writes or appends the record to the file.
 - Keyed Rewrite – rewrites the record in the file.
- *File Location, File Name, and File Record* – see the file containing the field designated as the key field. The value of the key must have been previously set by an assignment rule. If you choose the Select button, the Directories/Mailboxes window displays. You can drag a directory, a file, or a record from the Directories/Mailboxes window and drop it onto the File Record text box, and the other two File text boxes are populated. If you drag and drop a file, information is entered in the File Location, File Name, and Index File text boxes. You can also double-click a record on the Directories/Mailboxes window and it is entered in the text box, but you must always drag and drop directories and files.
- *Index File* – the name of the dBase index file being used.
- *Status Variable* – a memory variable that holds the error condition returned by the Keyed I/O. This memory variable can be used in an If Condition. (See the If Condition for instructions on how to use this variable to determine whether or not the record was found.) If you choose Select, the Memory Variables window displays. You can drag a memory variable from the Memory Variables window and drop it onto the Status Variable text box.

- *Keyed Field Value* – choose from:
 - *Memory Variable* – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the bottom text box below the Keyed Field Value text box.
 - *Record/Field* – displays next to the Rule Command window. Double-click a field and the program automatically enters it in the bottom text box below the Keyed Field Value text box.
 - *System Variable* – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the bottom text box below the Keyed Field Value text box.
 - *Constant* – enter a value in the text box on the Constant Value window that displays and select OK. The value is automatically entered in the bottom text box below the Keyed Field Value text box.

Map Level

The Map command performs at one time all of the rules and mapping associated with the segments in a specified level. The user assigns a level to one or more segments that he wants to have processed as a unit. This level does not need to be in the flow, but can be used exclusively for aggregation purposes. When the user executes the command to perform the actions associated with that level, the RTP maps all of the elements in the segments at that level and executes all of the rules linked to those elements. The mapping of a specific level can be triggered by the existence of data or by a specific trading partner. This rule can be used only on outbound maps.

Under *Parameters*, select the up arrow next to the Map Level text box. The Levels window displays next to the Rule Command window, with a list of all current levels. Double-click the level that you want to map with this rule command and the program automatically enters 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 the Map Flow chapter for instructions on adding a new level.)

NDO File Command

The NDO File Command rule commands are used to interface with the Rules and Formatter. They allow a user to read an NDO file type from memory, write an NDO file type to memory, or pass an NDO file type from memory to the Sybase Adapter. Multiple NDO data trees can be processed in one inbound map run, but only one NDO data tree can be processed in each outbound map run.

Variables

- *NDO File Command* – choose from:
 - Put NDO File – converts a sequential memory file (of NDO file type) to an NDO data tree for delivery to the Sybase Adapter. After the Put NDO File command is executed, the sequential memory file is empty and ready to be populated again with Write NDO File commands.

Note For a file to be delivered to the Rules and Formatter, the map must encounter and execute a Put NDO File command. If the system encounters and executes only Write NDO File commands, the file will never be converted and delivered to the Sybase Adapter.

- Read NDO File – the first time this command is executed, the program converts an NDO data tree acquired from the Sybase Adapter to a sequential memory file (of NDO file type) that has the same name as the schema node of the NDO data tree. After this, and every other time the command is executed, the program reads a record in the sequential memory file (of NDO file type).
- Write NDO File – writes a record to a sequential memory file (of NDO file type).

Note The Write NDO File command only creates an internal EMap memory file. The Put NDO File command actually delivers the file to the Rules and Formatter. If the user does not include a Put NDO File command in the map or the map does not encounter and execute a Put NDO File command., the file will never be converted and delivered to the Sybase Adapter.

- *NDO File Name* – the name of both the EMap sequential file name (of NDO file type) and the name of the schema node in the NDO data tree. For the Put NDO File rule command, the entry in this text box refers to both the sequential file and the NDO data tree. For the Read NDO File and Write NDO File rule commands, the entry in this text box refers to the sequential file. When you select the up arrow, the Files/Databases window displays next to the Rule Command window. If you double-click a file on the Files/Databases window, the program automatically populates the NDO File Name text box with that file name.
- *NDO File Record* – the name of both a record in the EMap sequential file and the name of a record (or container) node in the NDO data tree. For the Put NDO File rule command, the entry in this text box is meaningless. For the Read NDO File and Write NDO File rule commands, the entry in this text box refers to a record in the sequential file. When you select the up arrow, the Records/ Tables window displays next to the Rule Command window. If you double-click a record on the Records/Tables window, the program populates the NDO File Record text box with that record name and the NDO File Name text box with the file containing that record.

Perform Rule

The Perform Rule command executes a specified rule. By allowing one rule to call, or perform, another rule, the number of commands that need to be entered can be reduced.

Variables

- *Perform Rule Action* – choose from:
 - Perform Rule – performs the rule that is called exactly once. (All the text boxes except the Rule text box are grayed out.)
 - Perform While Rule – performs the rule that is called at least one time, and possibly more, based on when the specified condition becomes false. (Information must be entered in all the text boxes.)
- *Rule* – the actual rule that is performed either unconditionally or “while” a specified condition exists. You can type the rule number in the Rule text box, or you can select the up arrow to search for it. If you select the up arrow, the Rule Definitions window displays. On this window, you can select a rule or add a new one.
- *Left* – in the “while” condition can be:

- Memory Variable – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the bottom text box below the Left text box.
- Record/Field – displays next to the Rule Command window. Double-click a field and the program automatically enters it in the bottom text box below the Left text box.
- System Variable – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the bottom text box below the Left text box.
- *Logical* operator – specifies the relationship between the Left and Right variables in the “while” condition. Choose from:
 - EQ – equal to.
 - NE – not equal to.
 - GT – greater than.
 - GE – greater than or equal to.
 - LT – less than.
 - LE – less than or equal to.
- *Right* – in the “while” condition can be:
 - Memory Variable – displays next to the Rule Command window. Double-click a memory variable and the program automatically enters it in the bottom text box below the Right text box.
 - Record/Field – displays next to the Rule Command window. Double-click a field and the program automatically enters it in the bottom text box below the Right text box.
 - System Variable – displays next to the Rule Command window. Double-click a system variable and the program automatically enters it in the bottom text box below 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 automatically entered in the bottom text box below the Right text box.

- **Error Condition** – choose for the drop-down list of error conditions. (Error conditions are represented internally in the system by 2-digit codes. You can use an **If Condition** to compare one of these internal error codes to the value in a record field, memory variable, or system variable.) The 2-digit codes and explanations for each error condition are shown below:

Number	Code	Explanation
	No Error	Normal completion, with no errors detected.
02	End of File	The program reached the end of the file being processed.
03	Duplicate Key	You are attempting to add a record that already exists.
04	Record Missing	The requested record was not found.
05	File Full	The file space allocated has been exhausted.
06	Device Error	A physical device error has occurred.
07	Open Error	You are attempting to open a file that is already open or an I/O command was issued before a file was opened.
08	Out of Memory	There is not enough memory available.
09	File Conflict	There is a conflict between file attributes.
10	File Missing	The operating system cannot find a file.
	Other	An unspecified error condition has occurred.

Put File to Queue

The Put File to Queue command designates file as a queue file for use with the Sybase Adapter for EDI.

While the file is held in memory until it has been written, then it is placed in the queue as specified in the Adapter configuration file.

Parameters

File Name – choose the up arrow and the Directories/Mailboxes window displays. Drag a file and drop it onto the File Name text box.

Sequential I/O

The Sequential I/O command reads and writes to ASCII text files, also called flat files or sequential files. The Application Directory, Application File, and File Record identify the specific record and file to be read or written. The ability to select the directory, file, and record enables you to maintain the same records in multiple files, or the same files in multiple directories.

Variables

- *I/O Command* – choose from:
 - *Read Record* – reads the next record in the file. This command is executed each time it is encountered in a map execution.
 - *Read Once* – reads the next record in the file. This command is executed only the first time it is encountered in a map execution.
 - *Write Record* – writes or appends the record to the file. The file is created if it does not exist. If it does exist, new records are added to those already in the file.
 - *Rewrite Record* – rewrites the first record in the file. This command is used for one-record files such as counter type records.
- *Application Directory* – the name of the directory in which the file containing the record to be read or written is located. When you select the up arrow next to the Application Directory text box, the Directories (Mailboxes) window displays. If all three location-related text boxes are empty, you can open the directory and file containing the record and double-click the record. The program automatically enters the names of the directory, file, and record in the three text boxes. Otherwise, you must enter information in the text boxes or use drag-and-drop actions.
- *Application File* – the name of the file in which the record to be read or written is located.
- *File Record* – the name of the record to be read or written.

SQL

SQL is a computer language that is used to organize, manage, and retrieve data that is stored in a relational database. SQL is the acronym for Structured Query Language.

ODBC, or Open Database Connectivity, provides standardized access to a relational database. A DSN (Data Source Name) points to the database and links it with a specific ODBC driver.

Within EMap, the DSN is associated with an internal program file name for the purpose of establishing a connection to the database. SQL commands are used to access and manipulate the data in the database.

Below is a brief explanation of terms that are used in EMap SQL rule commands or instructions:

- **close** – ends a query and terminates the program’s access to the results of the query.
- **commit** – signals the successful end of a transaction. It notifies the database that all of the transaction statements have been executed and the transaction is complete - leaving the database self-consistent.
- **cursor** – a buffer that holds the result of a select statement. Each cursor defines a current row position in a set of result rows returned by an SQL query.
- **fetch** – advances the cursor to the next available row of query results if one exists – making this new row of query results the current row of the cursor. (If the cursor moves beyond the last row of query results, the fetch statement returns a NOT FOUND warning.)
- **insert** – adds new rows of data to a table.
- **open** – starts executing an SQL query and generating the query results.
- **rollback** – signals the unsuccessful end of a transaction. It tells the database not to complete the current transaction, but instead to back out any changes that were made to the database during the transaction – restoring the database to its state before the transaction began.
- **select** – retrieves data from a database and returns it in the form of query results.
- **transaction** – a logical unit of work consisting of a sequence of one or more SQL statements that perform independent actions and are usually closely related. Each statement performs a portion of a total task that is not complete until all the statements have been successfully executed.
- **update** – modifies data in a database.
- **where clause** – tells the SQL to include in the query results only those rows of data that meet a specified “search” condition.

Close Connection for File

The Close Connection for File command ends access to the query results and breaks the association between the database and the cursor.

Variables

- *SQL Application File Name* – the internal EMap name that provides the link to the database which is being queried and to which the connection no longer exists.

Close Cursor for File Record

The Close Cursor for File Record command ends access to the query results and breaks the association between the database table and the cursor.

Variables

- *SQL Application File Name* – the internal EMap name that provides the link to the database being queried and containing the records/tables to which the connection is being broken.
- *SQL Application File Name – Record Only* – the internal EMap name for the specific record, or table, (in the file/database specified above) which is being queried and to which the connection is being broken.

Commit All Files

Commit All Files inserts or updates records for all files, or databases, that are currently open. New records (tables) and modified records (tables) are physically inserted into the database. Once a commit is executed, the database cannot be restored to its condition before the commit.

Commit One File

Commit One File inserts or updates records for a specified file, or database, that is currently open. New records (tables) and modified records (tables) are physically inserted into the database. Once a commit is executed, the database cannot be restored to its condition before the commit.

Execute SQL Command

Execute SQL Command performs a SQL statement that is stored in a memory variable.

Because the SQL statement is stored in a memory variable, it can be constructed dynamically, based on results obtained while reading the data. For example, an SQL statement (such as select, insert, or update) can be assigned to the memory variable initially and SQL clauses (such as where or order by) can be added based on conditional logic applied to the data. The advantages of this command are the potential of eliminating the need for unnecessary nested condition testing and the ability to dynamically construct SQL statements based on data conditions (such as information passed from a Web browser to the server).

Variables

- *SQL Application File Name* – the name of the file associated with the DSN (Data Source Name) that points to the database being accessed by the SQL statement. (You made the association between the file and the DSN on the File – New/Properties window.)
- *SQL Application File Record* – EMap’s internal name for the database table that contains the data being accessed by the SQL statement. (When you import an ODBC record definition, EMap creates an internal record name using the name of the database table whose layout is being used to create the record definition. The corresponding Table Name, Record Name, and File Name appear on the Table Selection window.)
- *SQL Table Name* – the actual name of the database table that contains the data being accessed by the SQL statement.
- *SQL Statement – Memory Variable* – contains the dynamically created SQL statement. You begin by assigning a standard SQL statement to the memory variable. To complete the statement, you add SQL clauses by concatenating character strings (such as Where or state = ‘) or record fields (such as state_field) to the memory variable, based on conditions that you apply to the data. The resulting memory variable contains the entire dynamically created SQL statement that is executed.

Fetch after Prior Select

The Fetch after Prior Select command moves a single row/record/table (pointed to by the cursor) to EMap’s record definition buffer, after a select statement has retrieved the data from a database. A new record is moved each time this command is invoked. This command is equivalent to Sequential I/O - Read Record.

Variables

- *SQL Application File Name* – the internal EMap name that provides the link to the database to which the data is moved.

- *SQL Application File Record* – the internal EMap name for the specific rows/record/table (in the file/database specified above) to which the data is moved.
- *SQL Table Name* – the name of the database table that corresponds to the internal EMap file name. This is the actual database table from which the data is retrieved.

Fetch after Procedure Call

The Fetch after Procedure Call command retrieves the next row of query results for use by the application program after a specified stored procedure has been called and has returned a cursor.

Variables

- *SQL Application File Name* – the internal EMap name that provides the link to the database from which the query results are retrieved.
- *SQL Application File Record* – the internal EMap name for the specific record, or table, that is being retrieved.
- *Procedure Call Storage Record* – the record in the database that holds the stored procedure to be performed. This procedure returns a cursor value to be used by the fetch statement.

Fetch Once after Prior Select

The Fetch Once after Prior Select command moves a single row/record/table (pointed to by the cursor) to EMap's record definition buffer, after a select statement has retrieved the data from a database. A new record is moved only one time during map execution. This command is equivalent to Sequential I/O - Read Record.

Variables

- *SQL Application File Name* – the internal EMap name that provides the link to the database to which the data is moved.
- *SQL Application File Record* – the internal EMap name for the specific rows/record/table (in the file/database specified above) to which the data is moved.
- *SQL Table Name* – the name of the database table that corresponds to the internal EMap file name. This is the actual database table from which the data was retrieved.

Insert Record into Table

The Insert Record into Table command adds new rows of data to a table – the equivalent of writing a specified record/table to a file/database. Although this command temporarily places the record/table into the file/database, the action is not permanent until a COMMIT statement has been executed. Until then, the temporary insertion can be reversed by a ROLLBACK statement.

Variables

- *SQL Application File Name* – the internal EMap name that provides the link to the database containing the record/table into which data is being inserted.
- *SQL Application File Record* – the internal EMap name for the specific record, or table, (in the file/database specified above) into which data is written or inserted.
- *SQL Table Name* – the database name for the table that corresponds to the internal EMap file name.

Rollback All Files

The Rollback All Files command restores the data in all open files, or databases, to their condition before the current transaction was executed. After the rollback statement has been executed, the files, or databases, are restored to their condition after the last commit statement.

Rollback One File

The Rollback One File command restores the data in a specified open file, or database, to its condition before the current transaction was executed. After the rollback statement has been executed, the file, or database, are restored to its condition after the last commit statement.

Variables

- *SQL Application File Name* – the internal EMap name that provides the link to the database from which the data is being backed out.

Select and Update

The Select and Update command updates multiple rows in a database based on conditions specified in “where” clauses. A select statement retrieves data from a database and returns it in the form of query results. (When using this command, any fields/columns that have the auto-increment flag checked are not updated. The auto-increment flag is set on the Fields tab of the New Field or Field Properties window. See Adding a New Field in the Records/Tables chapter for instructions on setting this attribute.)

When you select the up arrow next to the Select and Update text box, the Select and Update FROM Application Record window displays.

Variables

- *SQL User File* – the internal EMap name that provides the link to the database containing the rows/records that are updated.
- *SQL User Record* – the internal EMap name for the specific rows/records/tables (in the file/database specified above) that are updated.
- *SQL Table Name* – the name of the database table that corresponds to the internal EMap file name. This is the actual database table that is updated.

You can use one or more where clauses to specify conditions. On each Where Clause tab, enter the following:

- *Record Field* – the application record field that holds the value that is used as the left side of the conditional statement defining which rows in the database are updated. If you choose Select, the Records/Tables window displays. Double-click a field or drag it and drop it onto the Record Field text box.
- *Operator* – logical operator that relates the left and right sides of the conditional statement defining which rows in the database are updated. The choices in the drop-down list include:
 - EQ – equal to.
 - NE – not equal to.
 - GT – greater than.
 - GE – greater than or equal to.
 - LT – less than.
 - LE – less than or equal to.
- *Variable* – holds the value that is used as the right side of the conditional statement defining which rows in the database are updated. Choose from:

- **Memory Variable** – appears in the first text box, and the Memory Variables window displays next to the Select and Update FROM Application Record window. Double-click a memory variable and the program automatically enters it in the second text box.
- **Record/Field** – appears in the first text box, and the Records/Tables window displays next to the Select and Update FROM Application Record window. Double-click a field and the program automatically enters it in the second text box.
- **String Constant** – appears in the first text box, and the WinMap window displays next to the Select and Update FROM Application Record window. Enter a value in the text box on the WinMap window and select OK. The value is automatically entered in the second text box.

When you finish entering all the information, select Compile Statement and the program constructs the SQL statement in the text box at the bottom of the window.

Select Test Statement to make sure that the program can connect with the database to execute the statement.

Select for Retrieval

The Select for Retrieval command queries the database, retrieves data, creates a cursor (buffer), and places in it the rows which match the “where” criteria in the select statement. A record is selected each time the rule is invoked.

When you select the up arrow next to the Select for Retrieval text box, the Select From Table For Retrieval window displays.

Variables

- *SQL User File* – the internal ECTMap name that provides the link to the database which is queried and from which data is retrieved.
- *SQL User Record* – the internal ECTMap name for the specific rows/record/table (in the file/database specified above) which is queried and from which data is retrieved.
- *SQL Table Name* – the name of the database table that corresponds to the internal ECTMap file name. This is the actual database table which is queried and from which data is retrieved.

You can use one or more “where clauses” to specify conditions. On each Where Clause tab, enter the following information:

- *Record Field* – the application record field that holds the value that is used as the left side of the conditional statement defining which rows in the database are updated. If you choose Select, the Records/Tables window displays. Double-click a field or drag it and drop it onto the Record Field text box.
- *Operator* – logical operator that relates the left and right sides of the conditional statement defining which rows in the database are updated. The choices in the drop-down list include:
 - EQ – equal to.
 - NE – not equal to.
 - GT – greater than.
 - GE – greater than or equal to.
 - LT – less than.
 - LE – less than or equal to.
- *Variable* – holds the value that is used as the right side of the conditional statement defining which rows in the database are updated. Choose from:
 - Memory Variable – appears in the first text box, and the Memory Variables window displays next to the Select from Table for Retrieval window. Double-click a memory variable and the program automatically enters it in the second text box.
 - Record/Field – appears in the first text box, and the Records/Tables window displays next to the Select from Table for Retrieval window. Double-click a field and the program automatically enters it in the second text box.
 - String Constant – appears in the first text box, and the WinMap window displays next to the Select from Table for Retrieval window. Enter a value in the text box on the WinMap window and select OK. The value is automatically entered in the second text box.

When you finish entering all the information, select Compile Statement and the program constructs the SQL statement in the text box at the bottom of the window.

Select Test Statement to make sure that the program can connect with the database to execute the statement.

Select Once for Retrieval

The Select Once for Retrieval command queries the database and creates a cursor (buffer) containing the rows that match the “where” criteria. Does not reselect the record the next time the rule is invoked.

When you select the up arrow next to the Select Once for Retrieval text box, the Select From Table Once For Retrieval window displays.

Variables

- *SQL User File* – the internal EMap name that provides the link to the database which is queried and from which data is retrieved.
- *SQL User Record* – the internal EMap name for the specific rows/record/table (in the file/database specified above) which is queried and from which data is retrieved.
- *SQL Table Name* – the name of the database table that corresponds to the internal EMap file name. This is the actual database table which is queried and from which data is retrieved.

You can use one or more “where clauses” to specify conditions. On each Where Clause tab, enter the following information:

- *Record Field* – the application record field that holds the value that is used as the left side of the conditional statement defining which rows in the database are updated. If you choose Select, the Records/Tables window displays. Double-click a field or drag it and drop it onto the Record Field text box.

Operator – logical operator that relates the left and right sides of the conditional statement defining which rows in the database are updated. The choices in the drop-down list include:

- EQ – equal to.
- NE – not equal to.
- GT – greater than.
- GE – greater than or equal to.
- LT – less than.
- LE – less than or equal to.
- *Variable* – holds the value that is used as the right side of the conditional statement defining which rows in the database are updated. Choose from:

- Memory Variable – appears in the first text box, and the Memory Variables window displays next to the Select from Table Once for Retrieval window. Double-click a memory variable and the program automatically enters it in the second text box.
- Record/Field – appears in the first text box, and the Records/Tables window displays next to the Select from Table Once for Retrieval window. Double-click a field and the program automatically enters it in the second text box.
- String Constant – appears in the first text box, and the WinMap window displays next to the Select from Table Once for Retrieval window. Enter a value in the text box on the WinMap window and select OK. The value is automatically entered in the second text box.

When you finish entering all the information, select Compile Statement and the program constructs the SQL statement in the text box at the bottom of the window.

Select Test Statement to make sure that the program can connect with the database to execute the statement.

Stored procedure call

The SQL command invokes specialized routines that are stored inside a database.

A procedure is a set of SQL statements grouped together as an executable unit to perform a specific task. Stored procedures give the SQL language the ability to perform database-related application processing within the database itself - a capability normally associated with programming languages. The application processing capabilities include conditional execution, looping, block structure, named variables, and named procedures. Prior to the advent of relational database systems, databases were essentially used only for the storage and retrieval of data.

The SQL *Stored Procedure Call* in EMap allows a map to issue a call to a stored procedure from within EMap. However, not all databases support the entire range of options offered by EMap. As a result, mappers must be aware of any limitations imposed by the specific ODBC database drivers that they are using. In addition, some databases limit the number of Select statements that are permitted. (In such a case, you would create rules to eliminate unnecessary Select statements.)

On the SQL Procedure Call window, you enter information about the stored procedure itself, the database in which it is stored, and the values passed into and out of the stored procedure.

Variables

Enter the following on the SQL Procedure Call window:

- *Procedure Name* – the name that was given to the procedure when it was created and under which it is stored in the database.
- *User File Name* – the name of the table that contains the data on which you are performing a Fetch After Procedure Call.

You create the association between the file name and the DSN on the New File or File Properties window. When you choose SQL DATABASE as the File Type, a Data Source Name text box appears. You enter the DSN in this text box and the program creates and displays the Driver Connect String in the text box at the bottom of the window.

- *User File Record* – the name of the record associated with the cursor (or h statement) that points to a location in the database specifying the table in the database to be used. A stored procedure can return no value or it can return either a value or a specific location in the database. When it returns a location in the database, the program accesses the User File Record to find the cursor that points to the specific row in the database that was returned by the stored procedure.

You create the association between the file (containing the DSN pointing to the database) and the record (containing the cursor pointing to the location in the database) by dragging the record from the Records/Tables window and dropping it on the file on the Files/Databases window.

- *Record Field Parameters* – in this section of the SQL Procedure Call window, indicate whether you pass parameters to the stored procedure, and if so, which fields in the record hold the parameters.
 - Select No Record Field Parameters – if no parameters are passed to the stored procedure.
 - Select All Fields in Rec Are Parameters – if all fields in the Parameter Record (specified in the following Parameters section) are passed as parameters to the stored procedure.
 - Select Marked Fields in Rec Are Parameters – if only specific fields in the Parameter Record (specified below) will be passed as parameters to the stored procedure.

When you define the fields in a record, you may mark a field as an SQL Stored Procedure Parameter on the Options tab of the New Field or Field Properties window.)

- *Parameters* – in this section of the SQL Procedure Call window, enter information about data being passed into or out of the stored procedure.
 - Parameter Record – contains the data being passed into the stored procedure.
 - Return Memory Variable – contains the data that is returned by the stored procedure. The stored procedure can return either a value or a location. (If the data returned is a location, that location is specified by the cursor linked to the User File Record.)
 - Return SQL Type – indicates the type of value that is returned by the stored procedure. You can choose from the drop-down list of eighteen possible SQL types.
- *Stored Procedure Call* – displays the SQL stored procedure call statement after it has been compiled.
 - Select the Compile Statement button to compile the SQL stored procedure call statement.
 - Select the Test Statement button to test whether you can connect properly to the ODBC database containing the stored procedure.

Stop Run

The Stop Run command stops the running of the run-time engine.

All activity in the current map stops, and messages are written to the trace file and the log file.

If multiple maps are being run, the runtime begins processing the next map.

String Operations

The String Operations command is used to determine whether inbound data is alpha or numeric; to perform block moves or move a record; to determine the length of a string; to perform find and replace options; to locate a specific string within another string; or to convert the case of a string. The String Action Type can be one of the following:

String Case Conversion

Variables

- *String Action Type* of String Case Conversion – used to convert a field or variable string to upper or lower case.
- *Source* – contains the data whose case is converted. The converted data remains in the Source. Choose from:
 - *Memory Variables* – displays next to the Rule Command window. Drag a memory variable and drop it in the Source text box.
 - *Record/Field* – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it in the Source text box.
 - *Record Only* – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it in the Source text box.
- *Conversion Type* – choose from:
 - *Upper* – uppercase data expressed in capitals.
 - *Lower* – lowercase data expressed in small letters.

Select OK on the String Case Conversion window to return to the Rule Definitions window.

String Find

The *String Action Type* of String Find is used to find a specific string within another string.

Under Parameters, select the up arrow next to the String Action Type text box and choose String Find from the drop-down list. The program enters String Find in the String Action Type text box and enters the label String Find over the text box below.

Select the up arrow next to the String Find text box, and the String Find window displays.

Variables

- *Find What* – contains information about the data for which the system will search. The first Find What text box specifies the type of data to be found. Choose from:

- **Memory Variable** – the program enters Memory Variable into the first Find What text box. The Memory Variables window displays next to the Rule Command window. Double-click a memory variable, and it is entered automatically in the second Find What text box.
- **Record/Field** – the program enters Record/Field into the first Find What text box. The Records/ Tables window displays next to the Rule Command window. Navigate down through a record to reveal the fields. Double-click a field, and it is entered automatically in the second Find What text box.
- **Constant** – the program enters Constant into the first Find What text box, and the EMap window displays. Enter a value in the text box on the EMap window and select OK. The value is automatically entered in the second Find What text box.

In the string for which the system is searching, choose one of the following:

- **Ignore Trailing Space**
- **Keep Trailing Space**
- **Search** – the drop-down list of places To Search includes the following choices:
 - **Memory Variable** – the program enters Memory Variable into the first To Search text box. The Memory Variables window displays next to the Rule Command window. Double-click a memory variable, and it is entered automatically in the second To Search text box.
 - **Record/Field** – the program enters Record/Field into the first To Search text box. The Records/ Tables window displays next to the Rule Command window. Navigate down through a record to reveal the fields. Double-click a field, and it is entered automatically in the second To Search text box.
- **Start Position** – contains the name of the memory variable in which the value returned by the String Find command is stored. (The value returned is the position where the string you are searching for is found in the string being searched.) Choose Select and the Memory Variables window displays. Either select a memory variable or create a new one. Double-click your choice, and the program enters it in the Start Position text box.

- *Occurrence* – used to specify which occurrence of the string should be found. (such as first, second, third, and so on.) The Occurrence drop-down list includes the following choices:
 - *Memory Variable* – the program enters Memory Variable into the first Occurrence text box. The Memory Variables window displays next to the Rule Command window. Double-click a memory variable, and it is entered automatically in the second Occurrence text box.
 - *Record/Field* – the program enters Record/Field into the first Occurrence text box. The Records/ Tables window displays next to the Rule Command window. Navigate down through a record to reveal the fields. Double-click a field, and it is entered automatically in the second Occurrence text box.
 - *Constant* – the program enters Constant into the first Occurrence text box, and the EMap window displays. Enter a value in the text box on the EMap window and select OK. The value is automatically entered in the second Occurrence text box.

Select OK on the String Find window to return to the Rule Command-New window.

String Justification

Use the *String Action Type* of String Justification to left-justify or right-justify a field or variable.

Variables

- *Source* – contains the data that is to be justified. The justified data remains in the Source. Choose from:
 - *Memory Variable* – displays next to the Rule Command window. Drag a memory variable and drop it in the Source text box.
 - *Record/Field* – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it in the Source text box.
- *Justification* – choose from:
 - *Left* – Left-justified data begins in the left-most byte, or character position, in the data field.
 - *Right* – Right-justified data ends in the right-most byte, or character position, in the data field.

Select OK on the String Justification window to return to the Rule Definitions window.

String Length

Variables

- *String Action Type* of String Length – use to return the length of the string in a source field or variable.
- *Source* – contains the data whose length is being checked. Choose from:
 - *Memory Variable* – displays next to the Rule Command window. Drag a memory variable and drop it on the Source text box.
 - *Record/Field* – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the Source text box.
- *Length Result – Memory Variable* – the memory variable that contains the actual length of the string (the result of the length query). The Memory Variables window displays next to the Rule Command window. Drag a memory variable and drop it on the Length Result – Memory Variable text box.

Select OK on the String Length window to return to the Rule Definitions window.

String Move

Use the *String Action Type* of String Move to move a field, variable, or record to a destination field, variable or record, based on start position and length. (This rule command can be used to copy one record to another.)

Under Parameters, select the up arrow next to the String Action Type text box and choose String Move from the drop-down list. The program enters String Move in the String Action Type text box and enters the label String Move over the text box below.

Select the up arrow next to the String Move text box, and the String Move window displays.

The Source section contains information about the location and size of the data that is to be moved.

Variables

- The first *Source* text box specifies the type of data to be moved. Choose from:
 - *Memory Variable* – the program enters Memory Variable into the first Source text box. The Memory Variables window displays next to the Rule Command window. Drag a memory variable and drop it on the second Source text box.

- **Record/Field** – the program enters Record/Field into the first Source text box. The Records/Tables window displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the second Source text box.
- **Record Only** – the program enters Record Only into the first Source text box. The Records/Tables window displays next to the Rule Command window. Drag a record and drop it on the second Source text box.
- **Start Position** – a memory variable that contains the value representing the byte, or character position, at which the program begins searching the object specified above.
- **Length** – enter the number of bytes, or characters, to be moved.
- **Destination** – contains the location to which the data is moved. Choose from:
 - **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the Source text box.
 - **Record/Field** – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the Source text box.
 - **Record Only** – displays next to the Rule Command window. Drag a record and drop it on the Source text box.
- **Start Position** – a memory variable that contains the value representing the byte, or character position, at which the program begins searching the object specified above.

Select OK on the String Move window to return to the Rule Command-New window.

String Replace

The *String Action Type* of String Replace performs search and replace functions within a field or variable. (Use this to find the positions of a group of characters in a string.)

Select the String Action Type up arrow and choose String Replace from the drop-down list. Select the String Replace up arrow and the String Replace window displays.

Variables

- **Find What** – enter information about the string that you want to replace. Choose from:

- Memory Variable – displays next to the Rule Command window. Drag a memory variable and drop it on the second text box.
- Record/Field – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the second text box.
- Constant – Enter a value in the text box on the EMap window that displays and select OK. The value is automatically entered in the second text box.
- Select *Ignore Trailing Space* if you want the program to disregard blank spaces that may follow the value you are trying to replace.
- Select *Keep Trailing Space* if you want the program to take into account blank spaces that may follow the value you are trying to replace.
- Under *Replace With*, enter information about the string that will be used as the replacement. The choices include:
 - Memory Variable – displays next to the Rule Command window. Drag a memory variable and drop it on the second text box.
 - Record/Field – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the second text box.
 - Constant – Enter a value in the text box on the WinMap window that displays and select OK. The value is automatically entered in the second text box.

Note You cannot enter a space as a replacement value. The program will appear to accept the value, but then give you an error on that command when you try to generate the map. You must define a memory variable with a fixed value of space and then use that memory variable as the replacement value.

- Select *Ignore Trailing Space* if you want the program to disregard blank spaces that may follow the value you are using as the replacement.
- Select *Keep Trailing Space* if you want the program to take into account blank spaces that may follow the value you are using as the replacement.

- In the *Search* section, tell the program in what object to search and replace the string, where in the object to begin looking for the string to replace, and whether to search for all occurrences or a specific one. If you want to search for a specific occurrence (for example, the third occurrence), that value (in this example, 3) would be expressed as a constant or be found in a memory variable or a record/field. If you want to search for all occurrences, you would choose the Replace All option.
- The choices for the object *To Search* include:
 - Memory Variable – displays next to the Rule Command window. Drag a memory variable and drop it on the second text box.
 - Record/Field – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the second text box.
- In *Start Position*, enter the variable name to store the returned value after a search.
 - Memory Variable – displays next to the Rule Command window. Drag a memory variable and drop it in the Start Position text box.
- For the Occurrence, you can either search for all occurrences or the choices include:
 - Memory Variable – displays next to the Rule Command window. Drag a memory variable and drop it on the second text box.
 - Record/Field – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the second text box.
 - Constant – Enter a value in the text box on the WinMap window that displays and select OK. The value is automatically entered in the second text box.
 - Replace All – REPLACE ALL is automatically entered in the second text box.

String Type

Use the *String Action Type* of String Type to identify whether a field or variable is alpha or numeric.

Variables

- *Source* – contains the data whose type is being checked. Choose from:

- **Memory Variable** – displays next to the Rule Command window. Double-click a memory variable and it is automatically entered in the Source text box.
- **Record/Field** – displays next to the Rule Command window. Double-click a field and it is automatically entered in the Source text box.
- **Variable Type** – choose from:
 - **Alpha** – the Source is checked to see if it contains alpha data.
 - **Numeric** – the Source is checked to see if it contains numeric data.
- **Result Code – Memory Variable** – the memory variable in which the results of the type query are placed. The Memory Variables window displays next to the Rule Command window. Double-click a memory variable and it is entered in the Result Code – Memory Variable text box. The memory variable contains a 1 if the data in Source matches the Variable Type selected. You can use an If Condition command to evaluate the results.

Select OK on the String Type window to return to the Rule Definitions window.

Substring

The Substring command allows you to extract strings from strings. Data can be moved to and from memory variables, record/fields and system variables. You can define the start (beginning position) and length of the string to be extracted by delimiters, a fixed position, or a variable position.

Variables

- **Source String** – the object from which a string is extracted. Choose from:
 - **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the Source String text box.
 - **Record/Field** – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the Source String text box.
 - **System Variable** – displays next to the Rule Command window. Drag a system variable and drop it on the Source String text box.
- **Substring Start** – specifies the starting position of the first character of the substring.

- **Delimiter Based** – The Select a Delimiter window displays next to the Rule Command window. Drag a delimiter and drop it on the Substring Start text box. (See the appendices for a list of ASCII characters that can be used as delimiters.)

If a delimiter was chosen, select the character from the displayed list and enter the number of the delimiter in the additional text box that displays. (For example, enter 1 for the first occurrence of a delimiter in the string.)
- **Constant Length** – Enter the value of the starting position (first character = 1) in the text box on the Constant Value window that displays and select OK. The value is automatically entered in the Substring Start text box.
- **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the Substring Start text box.
- **Substring Length** – specifies the length of the substring.
 - **Constant Length** – enter the value of the length in the text box on the Constant Value window that displays and select OK. The value is automatically entered in the Substring Length text box.
 - **Memory Variable** – the Memory Variables window displays next to the Rule Command window. Drag a memory variable and drop it on the Substring Length text box.
 - **Delimiter Based** – the Select a Delimiter window displays next to the Rule Command window. Drag a delimiter and drop it on the Substring Length text box. (See the appendices for a list of ASCII characters that can be used as delimiters.)

If a delimiter was chosen, select the character from the displayed list and enter the number of the delimiter in the additional text box that displays. The delimiter number is the number since the start position. (For example, to extract characters between the first and second comma, enter 1 as the start position for the first comma and one as the length delimiter count.)
- **Destination String** – the object in which the extracted string is placed. Choose from:
 - **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the Destination String text box.

- **Record/Field** – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the Destination String text box.
- **System Variable** – displays next to the Rule Command window. Drag a system variable and drop it on the Destination String text box.

Table Conversion

The Table Conversion command allows data to be passed through a cross-reference table for conversion in a rule. (See the Cross Reference Tables chapter for more detailed information.)

Variables

- **Conversion Table** – the name of the cross-reference table to be used for the data conversion.
- **Source Variable** – contains the value to be looked up in the cross-reference table. Choose from:
 - **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the Source Variable text box.
 - **Record/Field** – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the Source Variable text box.
 - **System Variable** – displays next to the Rule Command window. Drag a system variable and drop it on the Source Variable text box.
- **Destination Variable** – holds the value that is returned by the cross-reference table lookup.
 - **Memory Variable** – displays next to the Rule Command window. Drag a memory variable and drop it on the Destination Variable text box.
 - **Record/Field** – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the Destination Variable text box.
 - **System Variable** – displays next to the Rule Command window. Drag a system variable and drop it on the Destination Variable text box.
- **Status Variable** – a memory variable that holds a value indicating whether the lookup was successful. If the source variable was found in the table, a value of true (No Error = 01) is returned. If the source variable was not found, a value of false is returned.

- *Failure, Return* – allows you to return a default value into the destination if the source variable was not found in the table. That default value may be returned to a:
 - *Memory Variable* – the Memory Variables window displays next to the Rule Command window. Drag a memory variable and drop it on the On Failure, Return text box.
 - *Record/Field* – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the On Failure, Return text box.
 - *Constant* – enter a value in the text box on the Constant Value window that displays and select OK. The value is automatically entered in the On Failure, Return text box.

Or it may leave the destination empty:

- *Empty Value*
- *No default value is returned.*
- *Failure, Generate* – allows you to choose whether or not to generate an error message if the variable was not found in the table. If you choose to generate an error message, the program writes the message to the trace file and the log file. If you choose not to generate an error message, nothing is written to these files. Choose from:
 - *Error Message*
 - *No Error Message*

User Exit

The User Exit command allows you to call a routine (that is performed outside of ECMap) from inside a map, pass parameters to the routine, and receive parameters back from it.

- When the User Exit command is executed on a Windows PC, the RTP calls *userex32.dll*. This .dll has one main function to which the RTP can pass parameters and which can return parameters to the RTP.
- When the User Exit command is executed on a Unix machine, the RTP calls a shared library (*userex32.lib*).

Enter the following information on the Rule Command window for the User Exit window and it is passed as parameters to the .dll or shared library:

Variables

- *User Routine Name* – the name of the routine that is passed to the .dll or shared library.
- The *User Routine Input* identifies whether data is passed to the .dll or shared library, and if so, the location from which it is passed. Choose from:
 - *Memory Variable* – displays next to the Rule Command window. Drag a memory variable and drop it on the User Routine Input text box.
 - *Record/Field* – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the User Routine Input text box.
 - *Record Only* – displays next to the Rule Command window. Drag a record and drop it on the User Routine Input text box.
 - *None* – No input is passed to the routine. None is entered in the User Routine Input text box.
- The *User Routine Output* identifies whether data is returned from the .dll or shared library, and if so, the location to which it is passed. Choose from:
 - *Memory Variable* – displays next to the Rule Command window. Drag a memory variable and drop it on the User Routine Output text box.
 - *Record/Field* – displays next to the Rule Command window. Navigate down through a record to reveal the fields. Drag a field and drop it on the User Routine Output text box.
 - *Record Only* – displays next to the Rule Command window. Drag a record and drop it on the User Routine Output text box.
 - *None* – No input is returned from the routine. Enter None in the User Routine Output text box.
- The *Status Variable* is a memory variable that contains a user-generated value returned by the .dll or shared library. The generated value must be a 2-digit numeric value. (If you enter non-numeric data, the program converts it to “00”.) The value in this status variable can be used to test for special conditions.
- The *User Language* identifies the language in which the user exit routine is written. Choose from:
 - C
 - COBOL

You must start out writing C code in the .dll. From the .dll, you can call any other executable using a “System” call or a “CreateProcess” call. With Unix, the user exit is a shared library. EMap provides you with a stub, written in C - for this shared library. Once you have entered this stub, you can write code to call other code that may be written in COBOL or other languages.

In addition to entering values on the Rule Command window, users must modify the *userexit.c* file provided by EMap and recompile it to create a new *userex32.dll* or *userex32.lib*. The *userexit.c* file included with EMap is printed below.

Add the code for your routine where it is specified (in bold) at the end of the file and then recompile the file to produce a new *userex32.dll* or shared library. When the RTP is run, the values you entered on the Rule Command window are passed in as parameters, the routine is executed, and any values you specified to be passed back are returned.

Since there can be only one user exit .dll or shared library, you must imbed the logic and the resulting actions within the code you enter in the *userexit.c* file, to perform different actions from the User Exit command.

```
#ifndef UNIX
#include <windows.h>
#endif
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#ifdef UNIX
#include <unistd.h>
#endif
#ifndef UNIX
extern "C" __declspec (dllexport)
void WINAPI USEREXIT(char *, short, char *, char *,
char *);
#endif
/*-----*
UserExit - Windows DLL and Unix shared library.* This file and userdll.c and
userexit.def are used for userexit.dll.
* This file alone is used for UNIX shared library userexit.sl
*-----
* USEREXIT function.
* Parameters:
* char * cpuExitName - pointer to dynamically
*   allocated storage which contains the routine name.
*   This pointer should not be written to. This value
*   can * be used to determine what action should be done
```

```
* by USEREXIT() function. short sLanguage - will
* contain a 1 for Cobol or a 2 for 'C' language.
* char * cpRegBuf - pointer to input buffer which has
* been loaded with the value of a Memvar, Record Field
* or Record Buffer. For 16 bit program the maximum
* record buffer length is 3200 characters. For 32 bit
* program the maximum record buffer length is 10000 .
* characters
* char * cpRetBuf - pointer to output buffer, where the
* routine output of Memvar, Record Field or Record
* Buffer should be placed.
* * Note cpRegBuf and cpRetBuf have been set to point to
* the same large buffer. Empty input cpRegBuf before
* writing to output cpRetBuf.
* char * cpStatus - pointer to dynamically allocated
* storage which has been space filled and null
* terminated to actual length of memvar of status
* memvar. The user exit routine should be careful not
* to store more information in the field than it can .
* hold parameter returns: cpRetbuf, and cpStatus.
* return value: none
*-----*/
#endif UNIX
__declspec (dllexport)
void
WINAPI
USEREXIT(char *cpUExitName, short sLanguage,
          char *cpRegBuf, char *cpRetBuf, char *cpStatus)
#else
void
#endif NO_PROTO
USEREXIT(char *cpUExitName, short sLanguage,
          char *cpRegBuf, char *cpRetBuf,
          char *cpStatus)
#else
USEREXIT(cpUExitName, sLanguage, cpRegBuf, cpRetBuf,
          cpStatus)
char *cpUExitName;
short sLanguage;
char *cpRegBuf, char *cpRetBuf, char *cpStatus;
#endif
#endif
{
#endif UNIX
MessageBeep(-1); /* ring bell - user will add code here */
#else
```

```
write(0, "\007", 1); /* ring bell - user will add code here */
#endif
}
```

Write Log

The Write Log command writes messages to the transaction log (Translog) files. (See the Trading Partner and Log Database Formats chapter of the *ECMap Reference Guide* for the format of the Translog table.) These messages are user-initiated log file entries, as opposed to the system-generated messages. User-initiated messages include the same information that would be written to a system-generated message written at the same point. There are three required fields for user-initiated messages and seven optional ones. Entries in the optional fields override system-generated information in these fields.

The Write Log rule is used in both error reporting and transaction reporting requirements. It may also be used during debugging to identify particular values, check progress, etc.

Select the up arrow next to the Write To Log text box, and the Write Log File window displays.

If the current map is an inbound or outbound transaction map, the window below appears.

On the Write Log File window, you must enter information in the first three required fields – Msg No, Sev Code, and Message.

Variables

- *Msg No* – a five-character user-assigned message number, which can be used to locate specific messages that you have written to the log file. Choices are Memory Variable and Constant. ECMap has a pre-assigned set of error message numbers that are used for system-generated messages. (see the *ECMap Reference Guide* for a full listing of error messages and numbers.) Users often begin their message numbers at 10000 so that they do not duplicate the error message numbers used by the system. Msg No corresponds to the MSG_NO field in the Translog table.
- *Sev Code* – a two-character user-assigned severity code, which can be used to classify errors that have occurred and quickly locate classes of errors in the log file. The choices are Memory Variable and Constant. ECMap uses the following set of severity codes to flag errors in system-generated messages:

Code Number	Meaning	Code
1	No errors	[W## - ##]
2	Errors but no transaction skipped	[BADTRAN W##]
3	Transactions skipped with ## errors	[UABORT W##]
4	User Abort rule and ## errors	[USTOP W##]
5	User Stop rule	
6	Fatal error stop and ## errors	[EFATAL W##]

Users have several choices. They can use only these same system severity codes, use the system codes and supplement them with additional codes, or create their own set of codes. Sev Code corresponds to the SEV_CODE field in the Translog table.

Message is a brief textual explanation of the message. The maximum length of a message is thirty (30) characters. The choices are Memory Variable, Record/Field, System Variable, and Constant. If you choose Constant, the information you enter in the Constant box is entered in the Message field in the log. Otherwise, depending on the choice you make, the contents of the Memory Variable, Record/Field, or System Variable is entered in the Message field in the log. Message corresponds to the MSG_TXT field in the Translog table.

The following seven fields are optional. For each of these seven fields:

- If an entry is made on this window, it overrides any system-generated information and the field in the user-initiated log file contains the user-entered information.
- If no entry is made on this window and there is no system-generated information, the field is blank in the user-initiated log file.
- If no entry is made on this window and there is system-generated information, the field in the user-initiated log file contains the system-generated information.
- In the top section of the Write Log File window, there is one optional field that you can use to override system-generated information.

The *Record Seq No* is a 6-character field that is generally used with outbound maps. For files containing multiple record types, the record sequence number indicates how many times you have read the record type that is currently being processed. For files with a single record type, the record sequence number indicates how many times you have read the file. Record Seq No corresponds to the RECORD_NO field in the Translog table.

- In the middle section of the Write Log File window, there are four optional fields that you can use to override system-generated EDI Standards-related information. When you are writing entries to the log file while you are processing EDI Standards data, it is often helpful to know what segment, element, or sub-element was being processed when a particular action or error occurred.
 - *Segment Name* – the 3-character name of an EDI segment. It is generally the segment that was being processed when the log file entry was written or that is relevant to an action or error that occurred. You must enter the value. Segment Name corresponds to the SEGMENT field in the Translog table.
 - *Element No* – the 2-character code of an EDI element. It is generally the element that was being processed when the log file entry was written or that is relevant to an action or error that occurred. You can use an element number that was stored in a Memory Variable, or you can enter it as a Constant. To clear the entry in this text box, choose Clear. Element No corresponds to the ELEMENT field in the Translog table.
 - *Subelement No* – the 2-character code of an EDI sub-element. It is generally the sub-element that was being processed when the log file entry was written or that is relevant to an action or error that occurred. You can use a sub-element number that was stored in a Memory Variable, or you can enter it as a Constant. To clear the entry in this text box, choose Clear. Subelement No corresponds to the SUBELEM field in the Translog table.
 - *Segment Count* – the number (long integer) of segments that have been processed in the current transaction when the log file entry is written. You can accumulate this value in a Memory Variable from which it can be retrieved. To clear the entry in this text box, choose Clear. Segment Count corresponds to the SEG_COUNT field in the Translog table.

- In the bottom section of the Write Log File window, there are two optional fields that you can use to override system-generated information.

You can identify a record name and a field name that you would like to have written to the log file.

You can enter the Record Name and Field Name or you can Browse for them on the Records/Tables window. If you Browse for the names, when you double-click the field name, both the record name and field name is entered in the text boxes. Record Name corresponds to the RECORD_NAM field in the Translog table, and Field Name corresponds to the FIELD_NAME field.

If the current map is an any-to-any map, the Write Log File window below displays.

The Write Log window for any-to-any maps has the same top and bottom sections as the Write Log window for inbound and outbound transaction maps. However, it does not have the middle section because this section contains only standards-related data.

XML I/O

The XML I/O commands allow you to read and write XML data. There is only one choice for reading XML data. There are two choices for writing XML data. For simple records, you can use one inclusive command that includes writing the start-of-record tag, the data, and the end-of-record tag. However, for more complex records that include nested data, you need to use two separate commands. The first command writes the start-of-record tag and the data; the second command writes the end-of-record tag. In addition, there is a special write XML command that allows you to write processing instructions, doctypes, comments, etc. (for example, the preamble)

- The I/O command can be:
 - Write XML – Writes XML-coded data (including both the start-of-record tag and the end-of-record tag).
 - Write XML w/ No End Record Tag – Writes XML-coded data without an end-of-record tag.
 - Write XML End Record Tag Only – Writes only the XML end-of-record tag.
 - Write XML Text – Writes XML processing instructions, doctypes, comments, etc.

- Read XML – Reads XML-coded data.
- Application Directory – the name of the directory containing XML data to be read or the directory to which XML data will be written. Select the up arrow next to the Application Directory text box, and the Directories (Mailboxes) window displays. Navigate down into the records in the directory and double-click the record containing XML data. The program automatically populates the Application Directory, Application File, and File Record text boxes.
- Application File – the name of the file containing the XML data to be read or to which the XML data will be written.
- File Record – the name of the record containing the XML data to be read or to which the XML data will be written

For the Write XML Text rule command, the window displays as shown below:

- XML designates the location (directory/file) to which XML processing instructions, doctypes, comments, etc. will be written. This is the same directory and file that you entered for the other Write XML rule commands. Select the up arrow after the XML text box, and you have two choices:
 - If you choose Memory Variable, the Memory Variables window displays. Either double-click the memory variable in which the location (directory/file) is stored or drag it from the Memory Variable window and drop it on the XML text box. The caption over the text box changes to XML – Memory Variable.
 - If you choose Select File and Location, the Directories (Mailboxes) window displays. Drill down into the directory that contains the file to which the XML data will be written. Either double-click the file or drag it from the Directories (Mailboxes) window and drop it on the XML text box. The caption over the text box changes to XML – Directory / File.
- XML Text contains the actual XML processing instructions, doctypes, comments, etc. Select the up arrow after the XML Text text box, and you have the following three choices:
 - If you choose Memory Variable, the Memory Variable window displays. Either double-click the memory variable containing XML text or drag it from the Memory Variable window and drop it on the XML Text text box. The caption over the text box on the Rule Command window changes to XML Text – Memory Variable.

- If you choose Record/Field, the Records/Tables window displays. Drill down into the record that contains the field in which the XML text is stored. Either double-click the field containing XML text or drag it from the Records/Tables window and drop it on the XML Text text box. The caption over the text box on the Rule Command window changes to XML Text – Record / Field Variable.
- If you choose Constant, the Constant Value window displays. Enter the XML text in the text box on the Constant Value window and select OK. The caption over the text box on the Rule Command window changes to XML Text – Constant.

Using the View Menu

About this chapter

This chapter describes EMap's View menu.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	466
Viewing standards	466
Viewing work space	467
Viewing toolbars	467
Viewing small or large icons	468

Introduction

ECMap gives you several choices for how information is viewed. If the current map is a transaction map, the View menu on the main ECMap window includes the following choices:

- Transaction Table
- Transaction Tree
- Work Space
- Toolbars
- Small (Large) Icons

If the current map is an any-to-any map, the View menu on the main ECMap window includes only the following choices:

- Work Space
- Toolbars
- Small (Large) Icons

Transaction Table and Transaction Tree, the choices related to standards, do not appear on the View menu for an any-to-any map.

Viewing standards

If you choose Transaction Table from the View menu on the main ECMap window, the Create Transaction: <transaction> window displays. On this window, the standards information is displayed in a tabular format for the transaction mapped in the current map. (When the window opens, it is covered by a Tip box that tells you that you must right-click a segment to select it.)

If you choose Transaction Tree from the View menu, the Transaction and <transaction> windows display side by side. On the <transaction> window, the standards information is displayed in a tree presentation for the transaction mapped in the current map. (You can select a different transaction on the Transaction window, and the information for the new transaction is displayed on the <transaction> window. Select a transaction by double-clicking it.)

Whenever you view standards information from anywhere in EMap, the program displays the information in the Standard Table or Standard Tree format that you have selected on this menu. Standard Table is the default selection.

Viewing work space

If you choose Work Space from the View menu on the main EMap window, a vertical Work Space window appears on the left side of the main EMap window. The menu options appear as icons in a tree presentation. They are grouped differently than they are in the drop-down lists at the top of the window.

When the Work Space is first displayed, the following main icons appear:

- Project
- Map
- Application Files
- Build
- Tools
- Help

Select the main icons to display the options available from each.

Viewing toolbars

When EMap is installed, the default is to have the icons appear on a toolbar across the top of the window, underneath the menu options.

To have these icons not display, choose Toolbar from the View menu. Toolbar is cleared, and the icons disappear.

If you choose Toolbar again, the option is reselected and the icons once again appear at the top of the window.

The icons that appear differ, depending on whether the current map is a transaction map or an any-to-any map. If the current map is an any-to-any map, there are no company or trading partner icons.

Viewing small or large icons

When EMap is installed, the default is to use small icons across the top of the window.

If you choose Large Icons from the View menu on the main EMap window, a larger version of these icons is displayed and the selected option on the View menu changes to Small Icons.

If you choose Small Icons, the smaller icons again display and the selected option on the View menu changes back to Large Icons. Use this option to toggle back and forth between displaying large and small icons at the top of the main EMap window. This option only affects the size of the icons at the top of the window and does not affect the size of the icons on the Work Space. The icons on the Work Space are always small icons.

When you select a preference on the View menu, EMap remembers that preference each time the program is opened. Your preferences affect:

- the manner in which EDI standards information display throughout the program.
- whether Work Space displays on the EMap desktop.
- whether program icons display at the top of the main EMap window.
- whether the icons at the top of the window are large or small.

In order to change a preference, you must go back to the View menu and make a new selection.

Working with Cross-Reference Tables

About this chapter

This chapter discusses cross-reference tables in ECMap.

Topics

This chapter includes the following topics:

Topic	Page
About cross-reference tables	470
Accessing cross-reference tables from Tools	471
Accessing cross-reference tables from Mapping	483
Importing Cross-reference tables	486

About cross-reference tables

ECMap's cross-reference tables serve multiple purposes and provide an easy way to convert data during mapping. You can use them to produce pre-assigned EDI codes when application data does not meet the EDI standard format or for any other one-to-one conversion. If an EDI data element has an associated code list in the Standards Tables, you can view the full code list in ECMap.

Use cross-reference tables to convert one code to another code, such as when your trading partner has a code that must be translated to or from your code or when a code in the application data must be converted to or from an EDI code.

An invoice's Unit of Measure (UOM) field provides a good example of how cross-reference tables convert codes. In the X12 standard, the UOM element is defined as a 2-character ID field with an associated list of valid ID codes, such as EA for each and DZ for dozen. If the application data contains the digit 1 for dozen and the digit 2 for each, you can create a cross-reference table to convert the digit 1 to DZ for dozen and convert the digit 2 to EA for each.

You can also use cross-reference tables to convert codes to text or text to codes. For example, 01 might be the code for January, 02 for February, and so forth. A cross-reference table can look up a value in a list, such as a Weekday list: Monday, Tuesday, Wednesday, and so on.

In addition to simple data transformation, cross-reference tables offer a much wider range of possibilities. For example, EDI-based data dictionaries have code lists with description fields. These codes, called qualifiers in X12 and EDIFACT, tell the user (and the program) what kind of data is associated with that code. You can map an element that has a code list to a field, invoke the cross-reference function, and have the cross-reference function automatically build the table for you, by requesting the option to create entries in the table by filling them with the code description.

Using this method, ECMap can build a cross-reference table for PER-01 and PER-03, and map EDI data PER~EX~J.DOE~TE~9735551111 as follows:

ADMINISTRATIVE COMMUNICATIONS	
Expeditor	J.DOE
Telephone	973-555-1111

In EMap, you can add, delete, modify or find cross-reference tables. You can also display a list of cross-reference tables that are in the current map directory but are not defined in the map.

Note Cross-reference tables are often kept in a common directory so multiple maps can access them. You can select tables from this common directory, and EMap automatically defines them in your map.

Within a cross-reference table, you can add, delete, modify and find entries. You can also automatically populate field values in a cross-reference table entry for EDI-based cross-reference tables.

Transaction maps and any-to-any maps

The cross-reference table windows for transaction maps are slightly different from the windows for any-to-any maps. While the two sides of an entry in a cross-reference table for a transaction map are Standard data and Application data, the two sides of an entry for an any-to-any map are Source data and Destination data. The terminology on certain cross-reference table windows reflects this difference. In addition, less information is required to set up a cross-reference table for an any-to-any map, and there is one less menu option on the Cross Reference Table Properties window for any-to any maps.

Tools and Mapping windows

You can access the cross-reference table function from the Tools menu EMap's main menu or from the Mapping window (through the Build menu selection). While the Tools menu option lets you build cross-reference tables inside or outside of mapping, the Mapping window option lets you build and use cross-reference tables *while* you are mapping.

Accessing cross-reference tables from Tools

This section describes how to perform cross-reference table actions from the Tools menu.

Cross-reference tables for transaction maps

This section provides instructions specific to EDI transaction maps.

❖ **Adding a new cross-reference table**

- 1 Launch EMap and select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 On the Cross Reference Tables window, select File > New from the menu. The New Cross Reference Table window displays.

The program automatically populates the Project and Map text boxes.

- 3 In the **File Name** text box, type the name of the new cross-reference table, using 8 characters or fewer. This field is required.
- 4 In the **Standard Field Number** group, select *one* of the following options listed in Steps 4a and 4b:

- a Select the **No Field Number** option if you are creating a cross-reference table that does not use EDI values. Type the following required information in the appropriate text boxes:

- 1 Type a value in the **Standard Field Length** text box. This value is the source field length in an inbound map and the destination field length in an outbound map.
- 2 Type a value in **Application Field Length** text box. This value is the destination field length in an inbound map and the source field length in an outbound map.

The maximum field length of a cross-reference table entry is 100 characters. If an application field length is greater than 100 characters, you receive the following warning:

```
Standard Field Length Must Be <=100
```

- b Select the **Select Field Number** option if you are creating a cross-reference table that uses EDI values. When you select this option, the Select Element for Cross Reference window displays.

- 1 On the Select Element for Cross Reference window, highlight the line containing the element you want to add to the cross-reference table and click OK.

If there are values associated with the element, EMap inserts them into the Standard Field Length, Application Field Length, and Field Number text boxes.

- 2 Type the File Name.

5 In the Description text box, optionally type a brief description of the EDI data that is associated with the EDI element number or a brief description of the data elements being cross-referenced in the table.

6 Click Create.

If you are missing the file name or if a field length exceeds 100, you get an error message.

7 Correct errors and click Create again.

If you have no errors, EMap creates the cross-reference table, and the Cross Reference Table Properties window displays.

You have now created and defined a cross-reference table, but you have not yet created the entries in the table. You can enter the values for the entries in the cross-reference table now, or you can enter them later. See “Modifying an entry in a cross-reference table” on page 476.

For EDI elements with an ID data type, EMap has the capability to automatically create the entries in a cross-reference table. You must be in the Mapping window to use this feature. Refer to Chapter 13, “Mapping” for detailed instructions.

❖ **Deleting a cross-reference table**

1 On the Cross Reference Tables window, select Edit > Delete from the menu.

The program asks you to confirm that you want to delete the table (file) before it actually removes it.

2 Click Yes to remove and Cancel to exit the dialog box.

Warning! If you accidentally remove a table, you must create it again.

❖ **Finding a cross-reference table**

1 On the Cross Reference Table window, select Edit > Find or Find Next from the menu. The Find window displays.

2 In the Search String text box, type a character string and click Find.

3 Each time the program finds the character string, click Find Next to search for the next occurrence of the string.

❖ **Displaying unreferenced cross-reference tables**

Cross-reference tables are often kept in a common directory where multiple maps can access them. This practice saves mappers the unnecessary work of recreating cross-reference tables and entries. If all cross-reference tables are stored in a common directory, the Directory Scan utility lets mappers view all of the cross-reference tables in the cross-reference directory (for example, the common directory) that the current map is not using. Mappers can then select any of these unreferenced (currently unused) cross-referenced tables and have ECTMap automatically define the selected cross-reference tables in the current map.

Unreferenced tables can also occur accidentally when

- You use a common directory for all cross-reference tables
- You copy maps from one directory to another without using the Copy Map utility
- You move files for a particular cross-reference table from one map to another

In these cases, the *mxref* table for the project/map is not updated, leaving the tables undefined in the map. Follow the steps in the next procedure to display all cross-reference tables in the current map directory that are not linked to an element or rule.

- 1 On ECTMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 On the Cross Reference Tables window, select File > Directory Scan from the menu.
 - If there are no unused cross-reference tables, you receive a message telling you that no unreferenced cross-reference tables were found in the map directory.
 - If there are unused tables, the Cross Reference Directory Tables window displays.
- 3 To have ECTMap automatically define a cross-reference table in the current map, highlight the table on the Cross Reference Directory Tables window and click OK.

You can select multiple tables, using the SHIFT and CTRL keys:

- a To select multiple contiguous tables, click to select the first table, press SHIFT, and click the last table in the list. The entire group of tables is highlighted.

- b To select non-contiguous tables, click to select the first table, press CTRL, select the next table, press Ctrl, and so on, until you have highlighted all the tables you want to select.
- 4 The selected table or tables are defined and you return to the Cross Reference Tables window.

❖ **Adding entries to a cross-reference table**

- 1 On EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 On the Cross Reference Table window, highlight a table, right-click it and select Properties from the submenu. The Cross Reference Table Properties window displays.



If this is an existing cross-reference table, there is a list of all the entries in the highlighted table. If this is a new table, there are no entries.

- 3 On the Cross Reference Table Properties window, select File > New from the menu. The New Cross Reference Entries window displays.
- 4 In the **Table** text box, EMap inserts the name of the current cross-reference table. You cannot change this value; however, you must type entries in the remaining text boxes:

- a In the **EDI Value** text box, type a valid EDI code from the code list for the standard used in the current map. For transaction maps, this value is the EDI side of the EDI/application code conversion performed by the cross-reference table. For non-EDI conversions, this is the source or input value.
 - b In the **Application Value** text box, type the value that corresponds to the EDI value. For transaction maps, this value is the application side of the EDI/application code conversion performed by the cross-reference table. For non-EDI code conversions, this is the destination or output value.
 - c In the **Description** text box, type a brief description of this entry in the cross-reference table. This is a required field.
- 5 Click Next to add another cross-reference entry or OK to return to the Cross Reference Table Properties window. The new entries display in the list of table entries.

❖ **Modifying an entry in a cross-reference table**

- 1 On ECMaP's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 In the Cross Reference Table window, right-click a table and select Properties from the submenu. The Cross Reference Table Properties window displays.
- 3 In the Cross Reference Table Properties window, right-click a table entry and select Properties from the submenu. The Cross Reference Entry Properties window displays.
- 4 In the **Table** text box, ECMaP inserts the name of the current cross-reference table. You cannot change this value; however, you will edit the remaining text boxes:
 - a In the **EDI Value** text box, type a valid EDI code from the code list for the standard used in the current map. For transaction maps, this value is the EDI side of the EDI/application code conversion performed by the cross-reference table. For non-EDI conversions, this is the source or input value.
 - b In the **Application Value** text box, type the value that corresponds to the EDI value. For transaction maps, this value is the application side of the EDI/application code conversion performed by the cross-reference table. For non-EDI code conversions, this is the destination or output value.

- c In the **Description** text box, type a brief description of this entry in the cross-reference table. This is a required field.
- 5 Click Next to edit another entry or select OK to return to the Cross Reference Table Properties window.

❖ **Deleting an entry in a cross-reference table**

- 1 On EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 In the Cross Reference Table window, right-click a table and select Properties from the submenu. The Cross Reference Table Properties window displays.
- 3 Right-click a table entry and select Delete from submenu. The program asks you to confirm your decision before the entry is permanently removed.

Warning! If you delete an entry in a cross-reference table by mistake, you must re-enter it as a new entry.

- 4 Click Yes to delete the table or No to return to the Cross Reference Table Properties window.

❖ **Finding an entry in a cross-reference table**

- 1 On EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 In the Cross Reference Table window, right-click a table and select Properties from the submenu. The Cross Reference Table Properties window displays.
- 3 On the Cross Reference Table Properties window, select Edit > Find from the menu. The Find window displays.
- 4 In the Search String text box, type the character string you want the program to search for and click Find. Each time the program finds the character string, you can click Find Next to search for the next occurrence of the string.

❖ **Populating the fields in a cross-reference table**

To quickly create a cross-reference table, you can automatically populate blank application fields in a cross-reference table entry - with either the description or the EDI code value.

- 1 On EMap's main menu, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 In the Cross Reference Table window, right-click a table and select Properties from the submenu. The Cross Reference Table Properties window displays.
- 3 In the Cross Reference Table Properties window, select Options > Fill Fields From Description or Fill Fields From EDI Value.
 - a Select **Description** and EMap populates any blank cross application fields in the Cross Reference column with values from the Description column.
 - b Select **EDI Value** and EMap populates any blank application fields in the Cross Reference column with values from the EDI Field Value column.

If you see the warning, One or more descriptions were truncated when filling user value, it means that some of the field lengths you specified on the New Cross Reference Table window exceeded the length of application field being populated.

❖ **Toggling between the full EDI list and an abbreviated list**

- 1 On EMap's main menu, select Tools > Cross Reference Tables from the window. The Cross Reference Tables window displays.
- 2 In the Cross Reference Table window, right-click a table and select Properties from the submenu. The Cross Reference Table Properties window displays.
- 3 From the menu, select Options > Modify View to toggle back and forth between the full EDI code list and an abbreviated list of only those EDI elements that have corresponding application values in the cross-reference table.

Look at the **View** text box to see which view you are in. For example, ALL RECORDS shows the full EDI code list, while REFERENCE ONLY displays an abbreviated list.

Cross-reference tables for any-to-any maps

This section provides instructions on using cross-reference tables with any-to-any maps.

❖ Adding a new cross-reference table

- 1 On EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 On the Cross Reference Tables menu, select File > New from the menu. The New Cross Reference Table window displays.
- 3 EMap inserts the name of the current project and map, but you type values in the following text boxes:
 - a In the **File Name** text box, type the name of the new cross-reference table, using 8 characters or fewer. This field is required.
 - b In the **Source Field Length** text box, type the length of the application data field upon which the table lookup is being performed.
 - c In the **Destination Field Length** text box, type the length of the application data field that is being provided by the table lookup.
 - d In the **Description** text box, optionally type a brief description of the data elements being cross-referenced in the table.
- 4 Click Create.

If you are missing the file name or if a field length exceeds 100 characters, you get an error message.
- 5 Correct errors and click Create again.

If you have no errors, EMap creates the cross-reference table, and the Cross Reference Table Properties window displays.

At this point you have created and defined a cross-reference table, but you have not yet created the entries in the table. You can enter the values for the entries in the cross-reference table now, or you can enter them later. See “Modifying an entry in a cross-reference table” on page 476.

❖ Deleting a cross-reference table

- 1 In EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 Highlight a table from the list, right-click it and select Delete from submenu. The program asks you to confirm your decision before it actually removes the table (file).

Warning! If you accidentally remove a table, you need to create it again.

- 3 Click Yes to delete the table or No to return to the Cross Reference Table window.

❖ **Finding a cross-reference table**

- 1 In EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 Select Edit > Find from the menu. The Find window displays.
- 3 In the Search String text box, type the character string you want the program to search for and click Find. Each time the program finds the character string, you can click Find Next to search for the next occurrence of the string.

❖ **Displaying unreferenced cross-reference tables**

Cross-reference tables are often kept in a common directory where multiple maps can access them. This practice saves mappers the unnecessary work of recreating cross-reference tables and entries. If all cross-reference tables are stored in a common directory, the Directory Scan utility lets mappers view all of the cross-reference tables in the cross-reference directory (for example, the common directory) that the current map is not using. Mappers can then select any of these unreferenced (currently unused) cross-referenced tables and have EMap automatically define the selected cross-reference tables in the current map.

- 1 On EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 On the Cross Reference Tables window, select File > Directory Scan from the menu.
 - If there are no unused cross-reference tables, you receive a message telling you that no unreferenced cross-reference tables were found in the map directory.
 - If there are unused tables, the Cross Reference Directory Tables window displays.
- 3 To have EMap automatically define a cross-reference table in the current map, highlight the table on the Cross Reference Directory Tables window and click OK.

You can select multiple tables, using the SHIFT and CTRL keys:

- a To select multiple contiguous tables, click to select the first table, press SHIFT, and click the last table in the list. The entire group of tables is highlighted.

- b To select non-contiguous tables, click to select the first table, press CTRL, select the next table, press Ctrl, and so on, until you have highlighted all the tables you want to select.
- 4 The selected table or tables are defined and you return to the Cross Reference Tables window.

❖ **Adding entries to a cross-reference table**

- 1 To add entries to or modify entries in a cross-reference table, highlight the table and choose Properties from the Edit menu on the Cross Reference Tables window.

The Cross Reference Table Properties window displays. If this is an existing cross-reference table, there is a list of all the entries in the highlighted table. If this is a new table, there are no entries.

- 2 From the Cross Reference Table Properties window, you perform actions on the entries in a cross-reference table. Whether you are creating a new cross-reference table or modifying one, you can add a new entry to the table, edit or delete an entry, or search for an entry.

❖ **Adding an entry to a cross-reference table**

- 1 On EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 On the Cross Reference Table window, highlight a table, right-click it and select Properties from the submenu. The Cross Reference Table Properties window displays.

If this is an existing cross-reference table, there is a list of all the entries in the highlighted table. If this is a new table, there are no entries.

- 3 On the Cross Reference Table Properties window, select File > New from the menu. The New Cross Reference Entries window displays.
- 4 In the **Table** text box, EMap inserts the name of the current cross-reference table. You cannot change this value; however, you must type entries in the remaining text boxes:
 - a In the **Source Value** text box, type the input value which EMap uses to look up another value.
 - b In the **Destination Value** text box, type the output value, which is generated by the table lookup.
 - c In the **Description** text box, type a brief description of this entry in the cross-reference table. This is a required field.

- 5 Click Next to add another cross-reference entry or OK to return to the Cross Reference Table Properties window. The new entries display in the list of table entries.

❖ **Modifying an entry in a cross-reference table**

- 1 On EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 In the Cross Reference Table window, right-click a table and select Properties from the submenu. The Cross Reference Table Properties window displays.
- 3 In the Cross Reference Table Properties window, right-click a table entry and select Properties from the submenu. The Cross Reference Entry Properties window displays.
- 4 In the **Table** text box, EMap inserts the name of the current cross-reference table. You cannot change this value; however, you must type entries in the remaining text boxes:
 - a In the **Source Value** text box, type the input value which EMap uses to look up another value.
 - b In the **Destination Value** text box, type the output value, which is generated by the table lookup.
 - c In the **Description** text box, type a brief description of this entry in the cross-reference table. This is a required field.
- 5 Click Next to edit another entry or select OK to return to the Cross Reference Table Properties window.

❖ **Deleting an entry in a cross-reference table**

- 1 On EMap's main window, select Tools > Cross Reference Tables from the menu. The Cross Reference Tables window displays.
- 2 In the Cross Reference Table window, right-click a table and select Properties from the submenu. The Cross Reference Table Properties window displays.
- 3 Right-click a table entry and select Delete from submenu. The program asks you to confirm your decision before the entry is permanently removed.

Warning! If you delete an entry in a cross-reference table by mistake, you must re-enter it as a new entry.

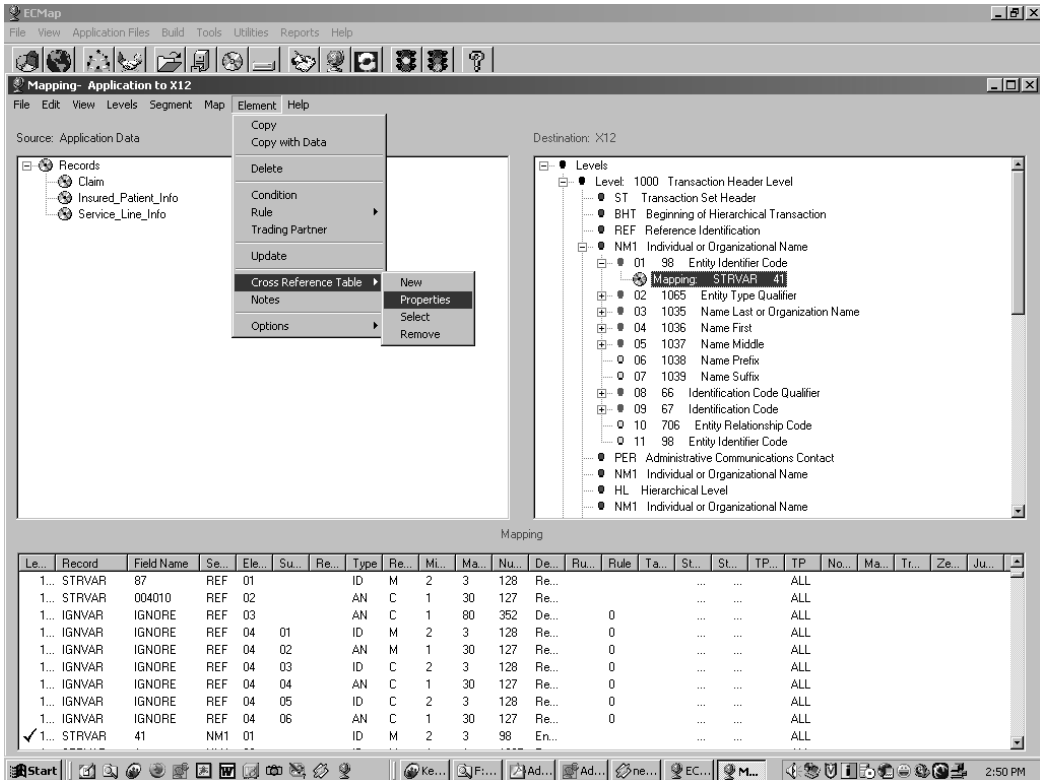
- 4 Click Yes to delete the table or No to return to the Cross Reference Table Properties window.
- ❖ **Finding an entry in a cross-reference table**
- 1 Choose Find or Find Next from the Edit menu on the Cross Reference Table Properties window. The Find window displays.
 - 2 In the Search String text box, enter the character string you want the program to search for and select Find. Each time the program finds the character string, you can select Find Next to search for the next occurrence of the string.
- ❖ **Toggling between ALL RECORDS and REFERENCE ONLY**
- 1 On EMap's main menu, select Tools > Cross Reference Tables from the window. The Cross Reference Tables window displays.
 - 2 In the Cross Reference Table window, right-click a table and select Properties from the submenu. The Cross Reference Table Properties window displays.
 - 3 From the menu, select Options > Modify View to toggle back and forth between the full EDI code list and an abbreviated list of only those EDI elements that have corresponding application values in the cross-reference table.
- Look at the **View** text box to see which view you are in. For example, ALL RECORDS shows the full EDI code list, while REFERENCE ONLY displays an abbreviated list.

Accessing cross-reference tables from Mapping

From the Mapping window, you can perform the same actions related to cross-reference tables that you performed from the Tools menu on the main EMap window. The difference is that from the Mapping window, you can build and use cross-reference tables on the fly while you are mapping. You can also create and remove associations between cross-reference tables and elements. (You get to the Mapping window by selecting the globe icon or by choosing Mapping from the Build menu on the main EMap window.)

❖ **Accessing the Cross Reference Tables window from the Mapping window**

- 1 On EMap's main window, select Build > Mapping from the menu. The Mapping window displays.
- 2 Highlight an element and expand it.



3 Choose one of the following:

- From the menu, select Element > Cross Reference Table. A submenu provides options for New, Properties, Select and Remove.
- Right-click an element in the EDI standards data pane and select Cross Reference Table from the submenu. The Cross Reference Tables pane is superimposed over the application data pane.

4 From either submenu, select either New, Properties, Select, or Remove. If you choose:

- **New** – you can add a new cross-reference table. The New Cross Reference Table window displays. See “Adding a new cross-reference table” on page 472.
- **Properties** – you can modify a cross-reference table. The Cross Reference Table Properties window displays. From this window, you can add, delete, modify or find an entry in the cross-reference table associated with the highlighted element. See “Modifying an entry in a cross-reference table” on page 476.
- **Select** – you can associate the highlighted element with a cross-reference table. This action is available only from the Mapping window. See the procedure, Associating a cross-reference table with an element.
- **Remove** – you can remove the association between a cross-reference table and the highlighted element. This action is available only from the Mapping window. If you choose Remove and there is no cross-reference table associated with the element, you receive an error message. See the procedure, “Removing the association between a cross-reference table and an element” on page 485.

❖ **Associating a cross-reference table with an element**

This procedure is for transaction maps.

- 1 On EMap’s main window, select **Build > Mapping** from the menu. The Mapping window displays.
- 2 Highlight an element and expand it.
- 3 From the menu, select **Element > Cross Reference Table > Select**.

This step, available only from the Mapping utility, lets you create an association between a cross-reference table and the highlighted element.

- 4 Drag a table from the list of available tables on the Cross Reference Tables pane and drop it onto the element.

❖ **Removing the association between a cross-reference table and an element**

This procedure is for transaction maps.

- 1 On EMap’s main window, select **Build > Mapping** from the menu. The Mapping window displays.
- 2 Highlight an element and expand it.
- 3 From the menu, select **Element > Cross Reference Table > Remove**.

This option removes the association between the highlighted element and the cross-reference table to which it is linked, but it does not delete the cross-reference table.

Note The program does not ask you to confirm your choice before it performs the action.

- 4 If you make a mistake and want to restore the link between the element and the cross-reference table, simply choose **Select** to drag and drop the cross-reference table back onto the element from which you removed it.

This is a different action from **Delete**, which permanently removes the cross-reference table.

Importing Cross-reference tables

You can import cross-reference tables from one map to another. This enables you to create or revise a cross-reference table once and then copy it into other maps.

❖ Importing cross-reference tables from another map

- 1 Select **Utilities>Import Cross-Reference Tables**.

The **Select Cross-Reference Tables** to Import window displays

- 2 Browse to the directory where the map you will be imported from resides.
- 3 Double-click the directory and a list of cross-reference tables defined for that map displays.
- 4 Check the tables you want to import and click **Next**.

The **Select Map for Table Import** screen displays a list of your projects and maps.

- 5 Open a project and select your destination map (where the tables will be imported *to*).
- 6 Click **Finish** to import the tables. The import process runs.

If the import table already exists in the destination map, a prompt displays asking if the import table should overwrite the current table. Click **Yes** to overwrite.

- 7 When the window opens that indicates the process is complete, click **OK** to continue.

What to do next

You can begin working with your new map or you can do one of the following:

- Import the current set of tables to another map by selecting that map on the Select Map for Table Import screen.
- Select a new set of tables from the Select Cross-Reference Tables to Import window and returning to the Select Map for Table Import Window.
- Select a different source and import those tables following the same steps.

If you import a table to a map and the table did not exist previously, update the map to use the new table by linking it to the appropriate element or rule.

Copying Maps

About this chapter

This chapter describes how to use the copy map utility.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	490
Working with the copy map utility	490

Introduction

ECMap lets you copy maps between projects and to and from directories. The copy map utility serves a number of different functions:

- Create backup copies of your maps by creating a backup directory and copying maps into it. You can restore maps from these backups.
- Use the same map in more than one project by copying a map and all related files from one project to another.
- Reuse similar maps by modifying the copies, saving time that you would have spent developing and testing parts of the map that remain unchanged.
- Send and receive maps by copying the map to a directory, compress it, and send it. If you receive a map, unzip it to a directory and then copy it from the directory to a map.

Working with the copy map utility

This section describes how to use the copy map utility, including copying from map to map, map to directory, confirming the copy command, and viewing the copy map log.

❖ Copying a map

- 1 In EMap's main menu, select Utilities > Copy Map. The Copy Map window displays.



On the Copy Map window, the program populates the Map Name and Project Name text boxes with the names of the currently selected map and the project in which it is stored.

- 2 You can accept the project and map names entered by the program or you can change them.
 - a To accept the current settings click Run.
 - b To change the project and its associated map do the following
 - 1 Click the up arrow to the right of the Project Name text box. The Select Project and Map window displays.
 - 2 Double-click the appropriate project and its associated map. The Select Project and Map window closes and populates the Project Name and Map Name text boxes in the Copy Map window.
- 3 In the Trade Partner Option group, choose one of the following:
 - a Click Include Trade Partner Tables to copy all map-related files and databases, including trading partner tables (tp, tradstat, and wixset).
 - b Click Exclude Trade Partner Tables to copy map-related files and databases, excluding trading partner tables.

You usually include the trading partner tables when you copy a map, but you may want to exclude them for various reasons. For example, if you are copying a map to use in a different project, that project might use an entirely different set of trading partner tables.

- 4 In the Map Copy Direction group, choose from one of four options and then click Run.

See the following procedures for information on choosing map direction:

- See “Using the This Map to Map” to copy the map specified in the Project Name and Map Name text boxes at the top of the window to a different map in the same project or to a map in a different project
- See “Using the This Map from Map option” on page 493 to copy a different map in the same project or a map in a different project to the map specified in the Project Name and Map Name at the top of the window.
- See “Using the This Map to Directory option” on page 493 to copy the map specified in the Project Name and Map Name at the top of the window to a specified directory. You usually choose this option when you are creating a backup or sending a map to someone.
- See “Using the This Map from Directory option” on page 494 to copy from the map specified in the Project Name and Map Name at the top of the window to a different map in the same project or to a map in a different project.

Note You usually choose this option when you are restoring from a backup or receiving a map from someone.

❖ Using the This Map to Map

This option, which is the default, copies the map specified in the Project Name and Map Name text boxes at the top of the window to a different map in the same project or to a map in a different project.

Clicking this option activates a group in this window called Copy Map to. This is where you specify your destination.

- 1 In the Copy Map To group, type a name in the Project Name text box. If you want your map copy to go into the current project, simply accept the default.
- 2 In the Map Name text box, either type a new name or click the up arrow to search for the names in the Select Project and Map window.

- 3 On the Select Project and Map window, double-click the project and map to which you want to copy the map and select OK. When you return to the Copy Map window, you will notice that EMap automatically populated the Project Name and Map Name text boxes with your choice.

❖ **Using the This Map from Map option**

If you choose to copy This Map from Map, the map you specified in the Project Name and Map Name text boxes at the top of the window is copied from another map, which you specify in the Copy Map From group.

- 1 In the Copy Map From group, type the Project Name and Map Name from which you want to copy the map or select the up arrow next to the Map Name text box to search for the names. The Select Project and Map window displays.
- 2 On the Select Project and Map window, double-click the project and map from which you want to copy the map and click OK. When you return to the Copy Map window, you will notice that EMap automatically populated the Project Name and Map Name text boxes with your choice.

❖ **Using the This Map to Directory option**

If you choose to copy This Map to Directory, the map you in the Project Name and Map Name text boxes at the top of the window is copied to a directory, which you specify in the Copy Map To group.

- 1 In the Copy Map To group, type the name of the directory to which you are copying the map in the Directory text box. Alternatively, click Browse to navigate to the directory. The Select a Directory window displays.
- 2 On the Select a Directory window, double-click the directory to which you will copy the map and click OK. When you return to the Copy Map window, you will notice that EMap automatically populated the Copy Map To text box with your choice.

If the program finds existing map files with the same names in the directory you select, the Databases Already Exist window displays.

- a To stop, click Cancel to return to the Copy Map window.
 - b To proceed, click OK and EMap asks you to confirm that you want to overwrite the existing databases. If you click Yes, EMap copies the map files to the directory, overwriting the existing files.
- 3 If EMap finds no existing files, a confirmation dialog displays.
 - a Click No to return to the Copy Map window.

- b Click Yes and EMap program copies the map files to the directory.

❖ **Using the This Map from Directory option**

If you choose to copy This Map from Directory, the map you specified in the text boxes at the top of the window is copied from a directory that you specify in the text boxes on the top right of the window. The Copy Map window displays as shown below:

- 1 In the Copy Map From group, type the name of the directory from which you are copying the map in the Directory text box. Alternatively, click Browse to navigate to the directory. The Select a Directory window displays.
- 2 On the Select a Directory window, double-click the directory from which you will copy the map and click OK. When you return to the Copy Map window, you will notice that EMap automatically populated the Copy Map From text box with your choice.

When you choose to copy This Map from Directory, you have another option. The Create This Map Definition from Directory Copy button lets you automatically create a map definition, which is important if you share maps with other developers.

Note When using this button, be sure to identify one default directory by selecting Change All on the Map Directories tab of the New Map Definition window. This ensures that the map definition files are all stored together.

❖ **Confirming the copy command**

Now that you have set all the options, you are ready to create the copy of your map.

- 1 Click Run. The program displays a Confirm dialog box that restates the details of the copy map option you have chosen.
- 2 The program asks for your confirmation before the map is actually copied.
 - a If you do not want to proceed or you want to change something, click No to return to the Copy Map window.
 - b If you want to copy the map, click Yes. The map is copied and you return to the main EMap window.

❖ Viewing the copy map log

- 1 If you want to see a listing of the actions that took place during the copy process, click View Log. The View Log window displays.
- 2 The information on this window lists each of the databases that were copied. If any problems were encountered, it issues warnings and errors. You should look at all warnings, but they do not necessarily prevent the copied map from running correctly. You must look at errors since they indicate a condition that would prevent the map from running.

Importing the SEF Standard

About this chapter

This chapter describes the SEF (Standard Exchange Format) Import utility, which works in conjunction with ECTMap.

Topics

This chapter discusses the following topics:

Topic	Page
About the SEF Import utility	498
Working with the SEF Import utility	498

About the SEF Import utility

The SEF Import utility is a separately purchased utility that works in conjunction with ECMaP. It lets you import subsets of EDI Standards, called SEF (Standard Exchange Format) Standards, to ECMaP and use them just as you would use any of the standards bundled with ECMaP.

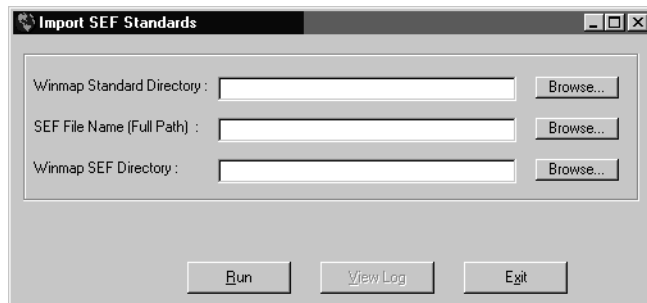
Note If you have not purchased this utility, you can skip this chapter.

Working with the SEF Import utility

This section provides instructions on how to use the SEF Import utility.

❖ Using the SEF Import utility

- 1 In ECMaP's main menu, select Utilities > Import > SEF Standards. The Import SEF Standards window displays.



- 2 Populate all fields in the SEF Standards window by typing fully qualified paths or by clicking Browse to navigate to the appropriate locations:
 - a In the **EDI Standard Directory** text box, browse to the location that contains the version of the bundled X12 standard corresponding to the version of the SEF Standard to import.
 - b In the **SEF File Name (Full Path)** text box, browse to the (source) directory from which the SEF Standard is imported.
 - c In the **SEF Output Directory** text box, browse to the directory to which the SEF Standard is imported.

3 Click Run.

If you specified a directory that does not exist, a dialog box appears, telling you that the directory does not exist and asking whether you want EMap to create it for you.

4 Choose one of the following:

- a Click No in order to browse to an existing directory or type one that you want to create.
- b Click Yes and EMap creates the directory. The View Log button on the Import SEF Standards window is active.
- c Click View Log and the <date> Standards Creation Log window appears with a list of all the log messages created while the SEF Standard was being imported.

5 Click Cancel to exit the <date> Standards Creation Log window.

6 Click Exit on the Import SEF Standards window to return to the main EMap window.

You can now use the SEF Standards that you imported just as you would use any other standard. If you want to use a SEF Standard, simply point to the directory containing the SEF standard you want to use when you define your map on the Map Directories tab of the New Map Definition window.

Working with Print Maps

About this chapter

This chapter discusses how to create print maps and convert EDI transactions to readable formats.

Topics

This chapter includes the following topics:

Topic	Page
About print maps	502
Working with print maps	502
Viewing log files	503
Associating a print map with a trading partner	505
Running a print map	505
Modifying and regenerating a print map	506

About print maps

Print maps are inbound transaction maps that generate printed output, rather than an application file. The standard print map makes it easy to see the EDI data by listing every segment, element, and code with their descriptions.

Note The Generate Print Map option works for inbound maps only.

While the printed output is often an EDI-like rendering of the EDI message the map is processing, there are no restrictions on the print format. For example, you can make minor alterations to the standard print map to produce output that lets you see the message more easily, rather than the EDI data format.

Working with print maps

This section discusses how to work with print maps.

❖ Creating a print map

- 1 Define your print map just as you define any other inbound transaction map. Refer to Chapter 3, “Working with Maps” for instructions on creating a map.
- 2 Specify the EDI standard to use (X12, EDIFACT, or HL7), the transaction set or message to be mapped, and the version of the standard.
- 3 Select the year to use as the “Century Minimum” when 6-digit date fields are mapped to 8-digit date fields, specify whether 8-digit dates are used in X12 envelopes (if it applies), and enter a description of the map. The map direction must be IN.

❖ Converting an EDI transaction to a readable format

- 1 Open an inbound map
- 2 In EMap’s main menu, select Utilities > Generate Print Map.

The Generate Print Map window displays. The names of the current Project and Map appear in the Map group.

When you generate a print map, EMap automatically creates the transaction (a transaction that includes all of the segments defined in the selected transaction set or message), does the mapping, and creates the rules and flow.

3 Click Run.

The program attempts to generate the map. A text box appears in the Options group and produces messages. The messages reflect the map generation activity taking place, such as Creating map transaction files, creating and mapping record fields, creating cross reference files, creating print map rules, creating flow, processing application files, processing rules, processing master element and condition tables, and processing reference files.

When all activity is finished, the message in the text box displays the final status of the map generation process.

4 Based on the results of the map run, do one of the following:

- a If the map run succeeded, proceed to Run Map.
- b If the map run failed, find and fix the problems that caused the failure and regenerate the map.

Viewing log files

While the program attempts to generate the map, it is creating a log file. This log file includes a list of all the Warnings, “Not Referenced” messages, or Errors that were found during map generation. For reference, the first line contains the date and time and the names of the project and map. At the end of the log, there is a summary of the problems encountered and a message indicating whether or not the map generated successfully.

- Errors cause map generation to fail. All Errors must be checked and fixed before the map is generated again.
- Warnings and “Not Referenced” messages indicate potential problems. While they should be checked, they do not prevent your map from being generated, nor do they necessarily have to be corrected.

For example, you receive a “Not Referenced” message if your data contains records that are not used in the map. If you intentionally excluded these records from your map, ignore the message. However, if you do need information from the records and simply did not include the correct instructions in your mapping, you need to change your mapping instructions and regenerate the map.

If the map generated successfully, the text box in the Options section of the screen displays the message “Map Generation Completed Successfully”.

Even when a map generates successfully, the log may contain Warnings or “Not Referenced” messages, which should be checked. If they indicate any real problems, make the necessary changes to your mapping instructions and generate your map again.

Refer to the *ECMap Reference Guide* for a description of messages that you may receive while generating a map.

❖ **Viewing a log and correcting errors**

- 1 Select View Log to see any potential error conditions that were found. The Generate Log screen displays.
- 2 Review the entries in the log.
 - a If your map is correct, click Cancel to return to the Generate Print Map screen.
 - b If your map did not generate successfully, click View Log and the Generate Log screen displays. Review Errors, Warnings, and “Not Referenced” messages that ECMap encountered during map generation.

Note You will know your map did not generate successfully when the message, No Map Generated. All Errors Need To Be Corrected, appears in the text box. Unless you modified the print map after generating it, you should not receive this message. If you changed the map, preventing the map from generating, you must correct the problems.

- 3 Click Cancel on the Generate Log screen to return to the Generate Print Map screen.
- 4 Click Exit screen to return to the main ECMap screen.
- 5 Correct the print map.
- 6 Generate your print map again.

Associating a print map with a trading partner

After the print map has been generated, you must create a trade agreement associating the map with a trading partner or with the ALL TradePartner.

❖ Associating a print map with a trading partner

- 1 In ECMaP's main menu, select File > Address Book > Contacts (Trading Partner). The Trading Partners screen displays.
- 2 Highlight the trading partner with whom the map is associated and select File > Trade Agreements. The Trade Agreements with Trade Partner:<trading partner> screen displays.
- 3 Select File > Add Map. ECMaP automatically creates a trade agreement linking the print map with the trading partner you selected and displays it on the screen.
- 4 Right-click the map to highlight it and choose Properties from the submenu.
- 5 Click the General tab to make it active and, in the Map Information group, click the arrow next to Map Type and change from IN (Inbound) to PRT (Print).
- 6 Click OK to close the Trade Agreements screen.
- 7 Close the Trade Agreements With Trade Partner: <trading partner> screen and the Trading Partners screen to return to the main ECMaP screen.

Running a print map

❖ Running a print map

- 1 In ECMaP's main menu, select Build > Run Map. The Run Inbound Map screen displays.
- 2 Enter the information on the tabs of the Run Inbound Map screen just as you would for any other inbound transaction map. The only requirement is that you must check the Print EDI check box on the Required tab. Refer to Chapter 17, "Running Maps" for instructions on running a map.

Modifying and regenerating a print map

You modify print maps the same way you modify and regenerate other maps. Refer to Chapter 3, “Working with Maps” for instructions on modifying a map. There are, however, two ways to regenerate a print map, and they produce two different results. It is important to understand the difference so that you can make the correct choice.

❖ Regenerating a print map using Generate Map

The following steps assume that you modified and regenerated your map.

- 1 In EMap’s main window, select Build > Generate Map. The Generate Map screen displays.

Note This is the preferred method of regenerating a print map.

All the changes you made to the print map after it was originally generated are now included.

- 2 Click Run at the bottom of the screen to regenerate the map.

❖ Regenerating a print map using Generate Print Map

The following steps assume that you modified and regenerated your map.

- 1 In EMap’s main menu, select Utilities > Generate Print Map. The Generate Print Map screen displays.

Note If you use this method to generate a print map, it is possible for you to lose the changes that you made to the map after it was originally generated.

- 2 On the Generate Print Map screen, click Run. A generate message box displays.

- 3 Choose one of the following:

- a Cancel – you do not generate the print map.
- b No – you generate a new print map that includes all changes you have made.
- c Yes – you revert to the original print map you generated and all changes you made since the map was originally generated are lost. Before the map is generated, however, you receive a message asking you to confirm overwriting current files.

- 4 After you make your selection, EMap attempts to regenerate the print map. The same actions that occurred when you originally generated your print map take place again. Messages flash across the text box and, finally, a message appears either saying that your map was successfully generated or that no map was generated.
- 5 If generation was not successful, correct the problems and regenerate the map until it generates successfully.
- 6 Click Exit and close all windows to return to EMap's main screen.

Working with Compliance Maps

About this chapter

This chapter describes how to create and run compliance maps.

Topics

This chapter includes the following topics:

Topic	Page
The EDI environment	510
Compliance checking and data mapping	511
Creating and running a compliance map	514

The EDI environment

The Electronic Data Interchange (EDI) environment uses standardized formats when data is exchanged between trading partners. These standard formats, or standards, provide the format and data content of EDI messages. While standards allow trading partners with diverse applications to exchange data, implementing these standards brings with it the hazard of exchanging data erroneously.

To prevent bad data exchange, each standard includes detailed documentation of compliance functionality, both from a business perspective and from a standards perspective. These functional definitions are documented in the standard in the form of segment notes, syntax notes, and semantic notes for each segment. These notes ensure that the segments and elements within a specific EDI message conform to the rules and regulations that were established for that specific transaction in that specific version of that specific standard.

Typical EDI transactions

The following represents a typical flow in a typical EDI transaction:

- 1 A trading partner sends another trading partner a business document. For example, a physician's office sends a medical claim to the patient's insurance agency.
- 2 The receiving trading partner's EDI translation software both maps the data and verifies that the EDI transaction conforms to an agreed-upon standard.

A number of different ways exist to verify both incoming and outgoing EDI data to determine its compliance with both the trading partner's requirements and the standard being used.

Incoming EDI documents

The following is true for incoming EDI documents:

- The EDI data is posted directly to the receiving application and checked for syntactical errors as it is posted. Both good (syntactically correct) and bad (syntactically incorrect) data are posted. The bad data is listed on an error report and posted to the log.

Outgoing EDI documents

- The EDI data is checked for syntactical errors before posting. Only the good data is passed to the conversion and posting routines. The bad data is listed on an error report and posted to the log.
- The EDI data is converted and posted to the application without any syntactical error checks. The application performs a compliance check on the data and returns an application acknowledgement indicating whether the data was good or bad.

The following is true for outgoing EDI documents:

- The EDI data is verified for syntactical correctness as it is generated, and the results are posted to the log.
- The EDI data is generated and the resulting EDI file is verified for syntactical correctness. The results are posted to the log.

In each case, a functional acknowledgement can be generated from the log and sent back to the trading partner.

With EMap, you can automatically create compliance maps to validate that an X12 message being sent or received is compliant with the definition specified in the EDI Standard. Compliance maps can be created for the full X12 Standard or any subsets of the Standard, including subsets defined by imported SEF Standards (Implementation Guides). In addition, EMap lets you combine user-generated compliance checking with data mapping. When you set up your compliance map, you can tailor the compliance checking to fit data verification scenarios suited to your needs.

Compliance checking and data mapping

Table 26-1 illustrates the compliance checking and data mapping processes that EMap performs.

Note The switches are for running the RTP from a batch command file script.

Table 26-1: Compliance checking and data-mapping processes

Type of verification	Description of verification activity	Checked automatically
Valid Data Type	Checks datatype attribute of both fields to verify compatibility when source is mapped to destination.	RTP

Type of verification	Description of verification activity	Checked automatically
Valid Data Size	<ul style="list-style-type: none"> • Checks to see if an EDI element is mapped to a data field smaller than the element's maximum width. • Generates a warning message that can be suppressed by selecting the Truncate box on the Mapping – Element window. 	RTP
Matching Control Counts	<ul style="list-style-type: none"> • Checks control counts for mismatches on inbound maps. • Generates a warning message. 	RTP
Mandatory Segments	<ul style="list-style-type: none"> • Checks for mandatory segments. • On inbound maps, segments that cause a level change are usually mandatory. • On outbound maps, mandatory segments are typically associated with mandatory records. • Generates a warning message. 	Compliance Map [-k switch or defined flow] [user override allowed]
Mandatory Elements	<ul style="list-style-type: none"> • Checks for mandatory elements. • Elements are mandatory unless the Mandatory check box is unchecked on the Mapping – Element window. 	RTP [user override allowed]
Valid Standards Code List Value	Verifies that elements defined as ID data types have valid code values using a cross reference table.	Compliance Map
Correct Segment Sequence	Checks to see whether the segments are in the correct sequence in inbound transaction maps.	Compliance Map [-k switch]
Correct Number of Segments	<ul style="list-style-type: none"> • Verifies that the transaction contains the correct number of segments. • This can also be checked by attaching to the flow a rule command that determines the number of included segments by querying the SE01 value. 	Compliance Map [-k switch]
Correct Element Usage According to Syntax Rules	Verifies that elements conform to the syntax rules of the standards with user-created rules in the compliance map. Once the rules are created, the rule definitions can be imported into other maps.	Compliance Map
Correct Number of Elements in a Segment	Verifies that the number of elements in the segment does not exceed the EDI Standards definition.	Compliance Map [-k switch]
Correct Number of Loop Counts	Verifies that the number of loops for the segment does not exceed the EDI Standards definition.	Compliance Map [-k switch]

Table 26-2 illustrates the compliance checks that you can perform at the application level.

Table 26-2: Compliance checks performed at the application level

Type of verification	Description of verification activity	Checked automatically
Valid Data	<p>ECMap provides the flexibility for the user to create “programming logic” that defines unique editing criteria for compliance and data validation using the extended command set provided in the rule function. Once you have developed the rules, performing them at precise stages of map execution can achieve the results dictated by user requirements.</p> <p>Myriad map components can be used to develop “programming logic” for compliance and data validation, including application record and file definitions, segment and element map assignments, rule definitions, flow definitions and rule assignments, and runtime switch and option settings.</p>	RTP [user override allowed]
Valid Trading Partner	<p>For both inbound and outbound transactions, you must associate a valid trading partner with the map by setting up a trade agreement that links the trading partner with the map.</p> <p>To override the assigned trading partner, select the ALL TradePartner default box on the Run Inbound or Outbound Map window.</p>	RTP [user override allowed]

Creating and running a compliance map

This section describes how to create and run a compliance map, including how to run a compliance map with a batch command file.

Defining a compliance map

Use the steps in the following procedure to define a compliance map.

❖ Defining a compliance map

To create a compliance map, you must first define the map on the New Map Definition window, just as you would any other inbound transaction map.

- 1 In EMap's main window, select File > Maps > Select to open the map record.
- 2 Right-click your map and choose Properties from the submenu. The Map Definition Properties window opens.
- 3 Click the Map Properties tab to make it active and specify the following:
 - a In the Map group, click the down arrow to the right of the Map Type text box and choose X12.
 - b In the Options group, click the down arrow to the right of the Direction text box and choose IN.
- 4 In the Options group, populate the remaining fields:
 - a In the Transaction text box, type the transaction for the compliance map to check; for example, 270.
 - b In the Version text box, click the down arrow and choose the version of the X12 standard used in the map; for example, 004010.
 - c In the Description text box, optionally type a description of the map; for example, Health care eligibility request for Acme Co.
 - d In the Century Minimum text box, type the year to be used as the century minimum for converting 6-digit dates.
 - e In the 8-Digit Date in X12 Envelope text box, click the down arrow and choose Y (yes) or N (no).
- 5 Click the Map Directories tab to make it active and specify the EDI Standards Table directory (last field), which contains the version of the standard that the compliance map checks.

- 6 To generate a compliance map using the full ASC X12 transaction set, set the EDI Standards Table directory to the directory that contains the official version of the Standards, such as *c:\Program Files\Sybase\Standards\<standard>*.

If the EMap SEF Utility was used to import a standard, EMap can generate a compliance map based on the implementation specified, but you must define the EDI Standards Table directory as the location of the imported standard, such as *c:\Program Files\Sybase\SEF850*.

Creating the transaction to be checked by the compliance map

Use the steps in the following procedure to create a transaction that the compliance map checks.

❖ Creating the transaction to be checked by the compliance map

After you define the compliance map, create the transaction that is checked for compliance.

- 1 In EMap's main menu, select **Build > Create Transaction**. The Transaction Tree windows display.
- 2 On the Create Transaction window, select **Segments > Check All** if you want the compliance map to include all of the segments in the transaction. Otherwise, select only those segments that you want the compliance map to check by clicking the box adjacent to each segment.

Note You can remove checkmarks by selecting the **Erase Checkmarks** option on the Segment menu.

- 3 When you have finished selecting the segments, you do not need to do anything else. EMap automatically creates the transaction.

Associating the compliance map with a trading partner

Use the steps in the following procedure to associate a compliance map with the trading partner.

❖ **Associating the compliance map with a trading partner**

After you define the compliance map and select the segments to include in the map, associate the compliance map with either a specific trading partner or with the ALL TradePartner.

The ALL TradePartner name is generally used as the default trading partner assigned to compliance maps. Using the ALL TradePartner as the default ensures that the compliance map has a generic association for a trading partner definition so the map processes transactions from any sender address. In this way, the same compliance map can be used to run both inbound and outbound transactions. The value for the sender address in the ISA and GS does not matter since the map is associated with the ALL TradePartner.

- 1 In ECTMap's main window, select File > Address Book > Contacts (Trading Partner). The Trading Partners window displays.
- 2 On the Trading Partners window, highlight the ALL trading partner.
- 3 Right-click the ALL trading partner and choose Trade Agreement from the submenu. The Trade Agreement with Trade Partner:ALL displays.
- 4 Select File > Add Map. The program automatically creates a trade agreement linking the current map with this trading partner. The map information displays on the window and highlighted.
- 5 The map is highlighted. From the menu, select Edit > Properties. If you have not selected the standard for your map, the Standard Type for Map window displays.
- 6 Select the appropriate standard for your map. The Trade Agreements window displays.

The program automatically enters the information, but you must make these changes.

- 7 Click the General tab to make it active and in the Map Information group, click the down arrow in the Map Type text box and change the map type from IN (inbound) to CMP (compliance).
- 8 Click the Overrides tab to make it active and in the Max Override Values group; then check the Map and route inbound EDI to Trade Agreement mailbox check box so that the bad data is separated from the good data.
- 9 After you enter all of the required information, click OK to close the Trade Agreements – Properties window and close all remaining windows to return to the main ECTMap window.

Generating a compliance map

Use the steps in the following procedure to generate your compliance map.

❖ **Generating a compliance map**

After defining the compliance map, creating the transaction to be checked, and setting up the trade agreement that links the map to a trading partner, you generate the compliance map.

- 1 In EMap's main window, select Utilities > Generate Compliance Map.
The Generate Map window displays three options and four buttons: Run, View Log, Exit, and Help.
- 2 Use the Generate Error for Ignored Elements check box under Options when compliance maps have been generated from implementation guides. If this option is checked, the compliance map generates an error if elements that were designated as not used contain data.
- 3 Click Run.

A text box appears in the Generate Map window containing messages about the generate process; for example, creating map transaction files, creating and mapping record files, creating compliance rules, creating flow, generating map – processing application files, processing rules, processing master element and condition tables, and processing reference files.

When your compliance map is finished, the message, Map Generation Completed Successfully, displays.

If the map did not generate successfully, you can view the log file.

- 4 Click View Log and the Generate Log window displays.
- 5 After you review the information in the log, click Cancel on the Generate Log window to return to the Generate Map window.
- 6 Click Exit on the Generate Map window to return to the main EMap window.

Running the compliance map

Use the steps in the following procedure to run your compliance map.

❖ **Running the compliance map**

After successfully generating your compliance map, you are ready to run it.

- 1 In EMap's main menu, select Build > Run Map or click the Run icon.
The Run Inbound Map window displays.
- 2 Enter all of the required information on the Run Inbound Map window. Refer to Chapter 17, "Running Maps" for detailed instructions on how to run inbound maps.
- 3 Click the Required tab of the Run Inbound Map window to make it active:
 - a In the Inbound EDI File text box, type the full path and file name or browse to the location of the file to be checked.
 - b Click the Compliance Check box to indicate that the data should run through compliance checking. Checking this box invokes the -k runtime switch. See the Runtime Technical Notes in EC RTP Reference Guide for information on how this switch operates.
 - c In the Log Type text box, click the down arrow and choose either Expanded Text Log or ODBC Log as the Log Type. Compliance checking will not work correctly without a log or with a non-expanded text log.
- 4 Click the Option 1 tab of the Run Inbound Map window to make it active and do the following:
 - a Click the All Trading Partner Default check box to indicate that the ALL TradePartner definition be used. This allows the compliance map to be run with any sender.
 - b Check the Ignore Trading Partner Mailbox and Ignore Trade Agreement Mailbox to test maps in the EMap MDP (Map Development Program).
 - c Check the Create Bad Transaction Log check box to set up an override file location for data which fails the compliance checks.
 - d In the Company Identification text box, type the information or browse to the location of the company ID information. This field is required.
- 5 Click the Option 2 tab of the Run Inbound Map window to make it active:
 - Check the Zero Fill EDI Non-Null Numbers check box.
- 6 After you enter the required information, click Run Map to execute the compliance map.

Refer to “Mailbox chart” on page 84 to see where the program places the good and the bad data. The location depends on choices that you made on the tabs of the Run Inbound Map window (or the equivalent switches that were set if the map is being run from a command line).

Modifying and regenerating the compliance map

Use the steps in the following procedure to modify and regenerate your compliance map.

❖ **Modifying and regenerating the compliance map**

After you have generated and run the compliance map, you may need to modify the map. If you do edit the map, you must regenerate it for the changes to take effect. When you regenerate a compliance map, you must use Generate Compliance Map on the Utility menu and not Generate Map on the Build menu.

- 1 In ECTMap’s main menu, select **Build > Generate Map** or click the **Generate** icon. Because the compliance map has been generated before, you are prompted with the following message:

```
Overwrite Existing Files? Generate Map and new MDP
files, Select 'Yes'.
Generate Map with Existing MDP Files, Select 'No'.
To Cancel Run, Select Cancel.
```

- 2 Click one of the following choices:
 - Cancel to exit the process
 - No to generate a new map that includes the changes you made
 - Yes to revert to the original map you generated, which overwrites any changes you made.

Running a compliance map with a batch command file

Use the steps in the following procedure to run your compliance map with a batch command file.

❖ **Running a compliance map with a batch command file**

ECMap includes a utility that creates a batch command file to automatically execute the compliance map. This utility saves all of the specified switches and options that you configured for the compliance map and writes them to the command file. These configurations include all the choices you made on the tabs of the Run Inbound Map window.

- 1 In ECMap's main window, select **Build > Run Map**. The Run Inbound Map window displays.
- 2 Click the **Option 2** tab to make it active.
- 3 Click **Create Batch Command File**. The Batch File Name window displays.

The **Create Batch Command File** button lets you create a DOS batch command file to automatically run the compliance map from a command line.

- 4 After you have selected all of the options for running the compliance map, type the full path file name in the text box. This is the location where you store the runmap script
- 5 Click **OK**. The program automatically creates a file that includes remarks and a line containing the runmap executable followed by the switches that correspond to the options selected on the tabs of the Run Inbound Map window.

A sample batch command file for a map appears:

```
REM Windows/UNIX Inbound Run Command
REM Inbound Trace file will be in
c:\maps\megaproject\bigcompany
REM Log Messages will be in ODBC With Connection
String
"DSN=ODBCLog"
wrmi32 c:\maps\data\hlthing.x12 -sl "DSN=ODBCLog"
-dg c:\maps\megainsurance\biglab
-dt "c:\maps\megainsurance\biglab" -it -o -b -wx 0
```

An explanation of runmap switches is included in the *ECRTP Reference Guide*. If you are using the ECMap batch command file in a UNIX environment, you must replace *wrmi32* (the runmap executable) with *mapinrun*.

- 6 You can add *.bat* to the file name and run the map from a command line. Then you can do one of the following:

- a You can edit the switches in the file and then run the map.

If you edit the file from Windows Explorer, do not double-click the file to open it or the batch program will execute. Instead, open it with a text editor or temporarily give the file a different extension.

- b You can cut and paste information from the file, such as the runmap switches, for use elsewhere.

The batch command file can be executed separately from EMap, incorporated in another batch command file, or called from another application.

Alternate scenarios

If you are using the EMap batch command file in a Windows NT environment and need the map to execute before continuing with other processing, insert the command `&&` before the `wrmi32` command line.

If you require the batch file to pause or wait for an action to take place before proceeding, consider using EC/EDI Gateway for centralized runs.

EDI to XML Generate Map Utility

About this chapter

This chapter describes how to use the EDI to XML Generate Map Utility.

Topics

This chapter includes the following topics:

Topic	Page
Introduction	524
Working with the EDI to XML utility	524

Introduction

The Generate EDI to XML Map utility provides a largely automated mechanism for converting an EDI transaction to an XML representation. The utility uses the EDI standard as input to create a map that will convert EDI transactions into the corresponding XML representation.

This utility derives the XML tag names from the segment and element descriptions found in the standards X12 transaction definition. When you run the map on a given transaction, missing X12 segments will not be generated nor will empty X12 elements be generated in the XML data output.

Working with the EDI to XML utility

❖ Prerequisites to using the EDI to XML Map utility

- 1 Define your map just as you would define any other inbound EDI X12 map. Refer to Chapter 3, “Working with Maps” for instructions.
- 2 Specify the X12 EDI standard or version and transaction to be mapped.
- 3 Select the year to be used as the Century Minimum when 6-digit date fields are mapped to 8-digit date fields, specify whether 8-digit dates are used in X12 envelopes, and optionally enter a description of the map. The map direction must be IN.

❖ Creating an XML map

When you create an XML map, EMap automatically creates a complete map based on the X12 transaction, including XML elements and attributes, the mapping, and the rules and flow.

- 1 In EMap’s main menu, select Utilities > Generate EDI to XML Map.
The Generate EDI to XML Map window displays. The names of the current Project and Map appear in the Map group.
- 2 In the Options group, click Run.

The program converts your EDI map into an XML map. A text box appears in the Options group, which processes visible messages. The messages reflect the map generation activity taking place; for example, actions such as creating map transaction files, creating and mapping record fields, creating cross-reference files, creating XML map rules, creating flow, processing application files, processing rules, processing master element and condition tables, and processing reference files. When all activity is finished, the message in the text box gives the final status of the map generation process. The initial generation of the XML map will be successful.

- 3 Click View Log, and the Generate Log window displays results similar to the following:

```
7/7/2004 5:49:56 PM Log For Project: My
Map:HIPAA_270_4010_50
Map Generation Completed Successfully
No Errors and No Warnings
```

- 4 Click Cancel to return to the EDI to XML Map window.

❖ **Generating the map and troubleshooting the results**

Under certain circumstances, such as when an EDI transaction contains HL segments, the map may require modifications if it was based on the native X12 transaction definition.

Modify the map just like any other data transformation map, and then use the Generate utility to generate a new *.map* file. Refer to “Modifying a map” on page 34.

- 1 In EMap’s main window, select Build > Generate Map or click the Generate Map icon.

While the program generates the map, it creates a log file. This log file includes a list of all the warnings, not referenced messages, or errors found during map generation. For reference, the first line contains the date and time and the names of the project and map. At the end of the log, there is a summary of the problems encountered and a message indicating whether or not the map generated successfully.

- 2 Click View Log, and the Generate Log window displays results similar to the following:

```
7/7/2004 5:49:56 PM Log For Project: My
Map:HIPAA_270_4010_50
Map Generation Completed Successfully
No Errors and No Warnings
```

- If the map generation processes succeeded, proceed to Run Map.
- If the map generation process failed, find and fix the problems that caused the failure and regenerate the map until no errors remain.

Warnings and “Not Referenced” messages indicate potential problems. They do not prevent your map from being generated, nor do they necessarily have to be corrected, although you should check them. For example, you receive a “Not Referenced” message if your data contains records that are not used in the map. If you intentionally excluded these records from your map, ignore the message. However, if you do need information from the records and simply did not include the correct instructions in your mapping, you need to change your mapping instructions and regenerate the map.

Refer to the *ECMap Reference Guide* for a description of messages that you might receive while generating a map.

- 3 Click Cancel to return to the EDI to XML Map window.

❖ **Associating an XML map with the ALL trading partner**

After generating the XML map, you must create a trade agreement associating the map with the ALL TradePartner. Unless you are going to build specific versions based on the Trade Partner, choose ALL TradePartner.

- 1 In ECMap’s main window, select File > Address Book > Contacts (Trading Partner). The Trading Partners window displays.
- 2 On the Trading Partners window, highlight the ALL trading partner.
- 3 Right-click the ALL trading partner and choose Trade Agreement from the submenu. The Trade Agreement with Trade Partner:ALL displays.
- 4 Select File > Add Map. The program automatically creates a trade agreement linking the current map with this trading partner. The map information displays on the window and highlighted.
- 5 The map is highlighted. From the menu, select Edit > Properties. If you have not selected the standard for your map, the Standard Type for Map window displays.
- 6 Select the appropriate standard for your map. The Trade Agreements window displays.
- 7 The program automatically enters the information.
- 8 Click OK on the Trade Agreements window.

- 9 Close the Trade Agreements With Trade Partner:ALL and Trading Partner windows to return to the main EMap window.

Update Database Structures

About this chapter

This chapter describes how to update database structures.

Topics

This chapter includes the following topics:

Topic	Page
About your databases	530
Updating the database	530
Upward-compatibility	531
Backing up maps	531

About your databases

When the version of the software you use to run a map is different from the version of the software you used to create the map, EMap alerts you that you need to update your database structures.

Changes to the software often require changes to the databases, usually to add additional fields or to modify existing ones. You must change the databases associated with maps that were created with previous versions of the software to reflect new formats.

EMap provides a utility to automatically synchronize the older maps with the requirements of the newer software.

Updating the database

When EMap updates database structures, it first moves the data in the map's original database structure to a temporary area. Next, it replaces the original database structures with the new ones. Then, it takes the data from the temporary area and puts it into the new structure. Finally, it clears the temporary area.

This following procedure describes the steps you take to start the update process.

❖ Updating the database structure

- 1 In EMap's main window, select Utilities > Update Database Structures.

The program begins updating your database, which you can see if you closely watch the status bar.

When finished, a window opens that reads "The Database Update has Completed."

- 2 Click OK to close the window and return to EMap's main window.

The databases associated with the current map have been updated, and the map can now run with the new software.

Upward-compatibility

ECMap is upward-compatible so that you are always able to use maps created in an earlier version of the software. However, you must use this utility every time you upgrade your software or whenever you run a map that you created with an older version of the software. For instance, if you copy a map or import a project with all of its associated maps, you might need to use this utility.

As with most software, ECMap is not guaranteed to be downward-compatible. If you attempt to run a map created with newer software than the version that you are running, you are alerted that you need to update your database structures. You can use the Update Database Structures utility to do this, but there is no guarantee that it works correctly.

Backing up maps

It is always good policy to make backup copies of your data files before you perform any sort of upgrade or update. You can use the Copy Map utility to make copies of your maps. If you experience any problems, you can restore your original maps from these copies.

See Chapter 23, “Copying Maps” for instructions on using ECMap’s Copy Map utility.

Compact Databases

About this chapter

This chapter describes the Compact Databases utility.

Topics

This chapter includes the following topics:

Topic	Page
About the Compact Database utility	534
Working with the utility	534

About the Compact Database utility

ECMap uses a Microsoft® Access databases to store the internal data that ECMap creates and uses. Because Access databases are designed to optimize interactive performance at the expense of recoverable disk space, Sybase recommends that you periodically compact the databases to improve program performance.

ECMap includes an effortless utility that performs this activity.

Working with the utility

In ECMap's main window, select Utilities > Compact Databases. The following actions occur:

- Unused space created by deletions is reclaimed. When rows are deleted from the database, Access marks them as available but does not actually release the space. The resulting database size is not reduced until the database is compacted.
- Table pages are placed adjacent to one another. This makes processing more efficient because the parts of each table are contiguous and not scattered across the database.
- Table pages are ordered by the primary key. Access does not maintain the primary key order when records are added, deleted, or modified after the database is compacted. Reordering the table pages allows the program to read ahead more efficiently.
- Table statistics used in the query optimization process are regenerated. This removes any problems with out-of-date statistics that are the result of rollbacks or improper closing of the database.
- All queries are flagged to be recompiled the next time the query runs. Because database statistics can change over time, previously compiled queries can contain inaccurate optimization information.

Update Dates

About this chapter

This chapter describes an ECMaScript utility that updates dates.

Topics

This chapter includes the following topics:

Topic	Page
About the date utility	536
Working with the date utility	536

About the date utility

ECMap lets you use a variety of date formats, including both 6-digit and 8-digit dates. Most software applications now require dates with 4-digit years, but many older applications still have date fields with 2-digit years. As a result, you may find yourself mapping a 6-digit date field to an 8-digit date field. ECMap provides a utility to take care of such situations.

Field century logic

ECMap includes century logic to update date formats on a field-specific basis or globally. This century logic is available when you are defining date fields in application files and when you are defining memory variables as date fields. In each case, you specify a *field century minimum* or a *global century minimum* that is used to convert 6-digit dates to 8-digit dates. You choose a specific year as the century minimum, and ECMap uses this year to determine whether a 2-digit year belongs to the 20th century or to the 21st century. Every 2-digit year greater than the year you specify as the century minimum is considered 20th century; for instance, the 1900s. Every year that is less than or equal to the century minimum is considered 21st century; for instance the 2000s.

For example, if you specify your century minimum as 40:

- The year 10 changes to 2010
- The year 39 changes to 2039
- The year 40 changes to 2040
- The year 41 changes to 1941
- The year 60 changes to 1960

Global century logic

ECMap also includes a general utility that applies the same century logic globally. When you use this utility, ECMap automatically updates the date format of any 6-digit date fields when they are mapped to 8-digit date fields. You can choose to apply a global century minimum to all data fields or only those for which you have not set a field century minimum.

Working with the date utility

This section provides instructions for how to use the utility.

❖ Using the utility

- 1 In ECMap's main window, select Utilities > Update Date. The Update Dates window displays.

- 2 In the Date Type group, click the down arrow to select the Date Update Type:
 - If you choose Change All Dates, century logic is applied to all date fields during mapping. The year defined as the century minimum is used to assign a 2-digit century to 6-digit dates.
 - If you choose Change Dates Not Already Set, century logic is applied to any date fields to which century logic has not already been applied during mapping. For these fields, the year defined as the century minimum is used to assign a 2-digit century to 6-digit dates.
- 3 In the Date Type group, click the down arrow to select the Century Minimum Type:
 - If you choose “Use global century minimum,” the year defined as the global century minimum is used to determine the century to be assigned to a 2-digit year. This is the only choice that is applicable.
 - If you choose “Use field century minimum,” the year defined as the field century minimum is used to determine the century to be assigned to a 2-digit year. This choice is not applicable.
 - If you choose “Use no century minimum,” no year is used to determine the century to be assigned to a 2-digit year. This choice is not applicable.
- 4 In the Year group, type a 2-digit year in the Global Century Minimum text box. This figure is used to determine the century applied to the date.

Whenever this map runs, the program performs date logic, where applicable.
- 5 Click Continue to close the Update Dates window and return to the main EMap window.

Generating Reports

About this chapter

ECMap includes pre-designed reports for 44 transaction maps and 35 any-to-any maps.

You can use the reports for a variety of purposes, but they are particularly useful for debugging during map development. They also provide thorough documentation for completed maps.

Topics

This chapter includes the following topics:

Topic	Page
Alphabetic listing of all ECMap reports	540
Transaction map reports	543
Any-to-any map reports	547

Alphabetic listing of all EMap reports

The following table provides an alphabetic listing and brief description of all EMap reports.

Report	Description
Alphanumeric Dates for Memory Variables	A listing of all memory variables that contain the word Date in the name
Alphanumeric Dates for Record Fields	A listing of all application fields that contain Date in the name and/or have DT as the data type
Application Files Record Listing	A listing of the names and descriptions of all the files and the records
Application Record Cross Reference	A listing of alphabetically each record name/field name and the segment, element number description and sequence number description where it is used
Application Records Detail Listing	A listing of the directories, files, records and field definitions and all field details for the selected map
Application Records Summary Listing	A listing of the directories, files, records and field definitions for the selected map
Company Identification	A listing of company identification information
Cross Reference Tables	A listing of all entries for a specific cross-reference table
Date Cross Reference (Memory Variables)	A listing of all memory variables, showing where they are mapped in the transaction
Element Rule Summary	A summary of the rules for each element contained in this map
Date Field Cross Reference (Fields)	A listing of all fields defined as DT, showing the elements they are mapped to and the rules in which they are used
Element Values	Prints a listing of all the EDI codes associated with a particular element
Field Cross Reference	A listing of all the application fields, showing where they are used in rules
Flow Definition	A listing of the levels of the transaction in numeric order, along with the segment name, new level and description, level description and rules applied
Flow Detail	A comprehensive report on the map, including the flow, rules, segments, elements and all mapping details

Report	Description
Flow Rule Summary	A listing of all rules in the flow for this map
Level List	A listing of all defined levels and descriptions for this map
Make Transaction	Prints out all the segments that would be created if the Make Transaction option were to be executed. The report prints out each segment and the elements contained within each segment
Map Detailed Definition	A comprehensive report on the element mapping within the selected map
Map Implementation Guide	This report includes notes contained in the segments and elements of the selected map. As you map, include notes specific to the selected segment or element. Refer to Chapter 5 Inbound Segment/Element Mapping and Outbound Segment/Element Mapping for more information about the Notes field
Map Quick Reference	A listing of all notes and record/fields mapped.
Map Segment Summary	A summary report of this transactions segments
Memory Variable Cross Reference	A listing of all memory variables, showing where they are used in rules and mapping
Memory Variables	A listing of the defined memory variables for the selected map
Memory Variable Dates	A listing of all memory variables that contain Date in the name or are defined as DT data type
Memory Variable Dates of length 8 or longer	A listing of all memory variables that are defined as DT data type with YYYY in the format as the year, showing where they are used
Memory Variable Dates w/ 2 digit Year	A listing of all memory variables that are defined as DT data type with YY in the format as the year, showing where they are used
Performed Rules	A listing of rules that are performed by other rules
Possible Date Memory Variables	A listing of all memory variables that contain the word Date in the name or are defined as DT data type and used to quickly identify what memory variables may be affected by Y2K logic
Possible Record Field Dates	A listing of all record fields that contain the word Date in the name or are defined as DT data type and used to quickly identify what fields in the application data may be affected by Y2K logic

Report	Description
Project	A listing of all the maps and directories in the project
Record Field Dates of length 8 or longer	A listing of all record fields that are defined as DT data type with YYYY in the format as the year, showing where they are used
Record Field Dates w/ 2 digit Year	A listing of all record fields that are defined as DT data type with YY in the format as the year, showing where they are used
Rule Cross Reference	A listing of all the rules, showing where they are invoked
Rule List	A listing of all the defined rule numbers and their descriptions
Rule Definition	A listing of the command contents of each rule by rule number
Selected Segment	Prints out the details for the highlighted segment. These details include each element - with its minimum and maximum length, the requirement of the segment (mandatory, optional, or conditional), the element number and description, and the element type
Selected Transaction Template	Prints out the detail of all segments contained within the selected transaction
System Variable Cross Reference	A listing of all the system variables used in a map, showing where they are used
System Variables	A listing of all system variables - showing variable name, description, length and variable type
Trading Partner Detail	A listing of all the information from the Trade Partner database including default overrides, contact information, and transaction information
Trading Partner List	A listing of all the Trading Partners - including GS code, ISA qualifiers and transaction status
Transaction Listing	Prints a list of all the transactions contained within a particular standard

Transaction map reports

You access most reports for transaction maps from the Reports menu on the main EMap window. You can access other reports from the View (Transaction Table or Transaction Tree) menu, which you open using the View and Build menus on the main EMap window.

Using the View menu

Accessing the Transaction Table or Transaction Tree using the View menu option lets you see one window at a time; for example, you can see either the transaction table elements or the entire transaction tree. However, you can toggle back and forth by selecting View > Transaction Table or View > Transaction Tree as many times as you want.

- In the Main EMap window, select View > Transaction Table and the standard table for the current map opens.
- In the Main EMap window, select View > Transaction Tree and two windows open: the current map's transaction table on the left and transaction tree on the right (for example, 837 Health Care Claim).

Using the Build menu

You can access either the Transaction Table or Transaction Tree, depending on which you last viewed using EMap's View menu.

- In the main EMap window, select Build > Create Transaction and the either the standard Transaction Table window or Transaction Tree window opens.

Accessing transaction map reports from the Reports menu

The following reports, listed by functional group, are available from the Reports menu on EMap's main window.

Note In order to see all the reports in the following list, the current map must be a transaction map. If it is not a transaction report, some of the report options will be dimmed.

- A) Project
- B) Company Identification
- C) Trading Partner List
- D) Trading Partner Detail
- E) Application Records Summary Listing

- F) Application Records Detail Listing
- G) Application Files Record Listing
- H) Memory Variables
- I) System Variables
- J) Application Record Cross Reference
- K) Additional Cross Reference Reports:
 - Referenced Items
 - Not Referenced Items
- L) Level List
- M) Flow Definition
- N) Flow Detail
- O) Map Implementation Guide
- P) Map Quick Reference
- Q) Map Segment Summary
- R) Map Detailed Definition
- S) Performed Rules
- T) Flow Rule Summary
- U) Element Rule Summary
- V) Rule List
- W) Rule Definition
- X) Year 2000 Date Cross Reference:
 - 1) Record Field Dates w/ 2-Digit Year
 - 2) Record Field Dates of length 8 or longer
 - 3) AlphaNumeric Dates for Record Fields
 - 4) Possible Record Field Dates
 - 5) Memory Variable Date w/ 2-Digit Year
 - 6) Memory Variable Date of length 8 or longer
 - 7) AlphaNumeric Dates for Memory Variables

- 8) Possible Date Memory Variables
- Y) Year 2000 Date Listings:
 - Record Field Dates
 - Memory Variable Dates

Accessing transaction map reports from the View menu

This section describes the transaction map reports accessed from the View menu in EMap's main window. Which reports are available depends on your view. For example:

- In the Transaction Table view, you have access to the full range of reports depending on which tab is active in the Create Transaction window.
- In the Transaction Tree view, only the segment-related reports are available.

For more information on accessing reports using the Transaction Table and Transaction Tree views, see the following procedures.

❖ Accessing transaction-related reports from the Transaction Table view

- 1 In EMap's main window, select View > Transaction Table. The Create Transaction window opens.
- 2 Highlight the Transaction tab to make it active.
- 3 On the Create Transaction window menu, point to Reports and click to choose one of the following:
 - *Selected Transaction Template* prints the details of all segments contained within the selected transaction.
 - *Transaction Listing* prints a list of all the transactions contained within a particular standard.

Note If you access reports from the Transaction Tree view, you must select a report from the Reports menu in the Transaction window on the left. If you point to the Reports menu in the <Transaction> window on the right, you can only print segment-related reports.

❖ **Accessing segment-related reports from the Transaction Table view**

- 1 In EMap's main window, select View > Transaction Table. The Create Transaction window opens.
- 2 Highlight the Segment tab to make it active.
- 3 On the Create Transaction window menu, point to Reports and click to choose one of the following:
 - *Selected Segment* prints the details for the highlighted segment. These details include each element, with its minimum and maximum length, the requirement of the segment (for example, mandatory, optional, or conditional), the element number and description, and the element type.
 - *Make Transaction* prints the segments and the elements contained within the segments that would be created if the Make Transaction option were to be executed.

❖ **Accessing segment-related reports from the Transaction Tree view**

- 1 In EMap's main window, select View > Transaction Tree. Two Transaction windows open.
- 2 Click the <Transaction> window on the right, where <Transaction> represents the name of your current map, such as 837 Health Care Claim
- 3 In the <Transaction> window menu, point to Reports and click to choose one of the following:
 - *Selected Segment* prints the details for the highlighted segment. These details include each element, with its minimum and maximum length, the requirement of the segment (for example, mandatory, optional, or conditional), the element number and description, and the element type.
 - *Make Transaction* prints the segments and the elements contained within the segments that would be created if the Make Transaction option were to be executed.

❖ **Accessing code-related reports in the Transaction Table view**

- 1 In EMap's main window, select View > Transaction Table. The Create Transaction window opens.
- 2 Highlight the Segment tab to make it active.
- 3 On the Create Transaction window menu, point to Reports and click to choose one of the following:

- *Element Values* prints a full listing of all the EDI codes associated with a particular element.

To access code-related reports in the Standard Table view, make either the Full Code tab or the Implementation Code tab active and choose a report above from the Reports menu at the top of the Create Transaction window.

❖ **Accessing code-related reports in the Standard Tree view**

- 1 In EMap's main window, select View > Transaction Tree. Two Transaction windows open.
- 2 Click the <Transaction> window on the right, where <Transaction> represents the name of your current map, such as 837 Health Care Claim
- 3 In the <Transaction> window menu, highlight an element and choose Full Code List or Implementation Code List. The Full Code List or Implementation Code List window displays.
- 4 Choose a report from the Reports menu on the Full Code List or Implementation Code List window.

If you choose Full Code List, the Full Code List window displays – regardless of whether there are any codes associated with the highlighted element. If you choose Implementation Code List and no Implementation Code list exists, you receive the message that no implementation codes are available.

Any-to-any map reports

List of any-to-any map reports

The following reports, listed in functional groups, are available from the Reports menu on the main EMap window, when the currently selected map is an any-to-any map:

- A) Project
- E) Application Records Summary Listing
- F) Application Records Detail Listing
- G) Application Files Record Listing

- H) Memory Variables
- I) System Variables
- K) Additional Cross Reference Reports:
 - Referenced Items
 - Not Referenced Items
- L) Level List
- M) Flow Definition
- N) Flow Detail
- S) Performed Rules
- T) Flow Rule Summary
- V) Rule List
- W) Rule Definition
 - 1) Record Field Dates w/ 2-Digit Year
 - 2) Record Field Dates of length 8 or longer
 - 3) AlphaNumeric Dates for Record Fields
 - 4) Possible Record Field Dates
 - 5) Memory Variable Date w/ 2-Digit Year
 - 6) Memory Variable Date of length 8 or longer
 - 7) AlphaNumeric Dates for Memory Variables
 - 8) Possible Date Memory Variables
- Y) Year 2000 Date Listings:
 - Record Field Dates
 - Memory Variable Dates

Accessing any-to-any reports from the Reports menu

You access reports for any-to-any maps from the Reports menu on EMap's main window. The reports for any-to-any maps are the same as those for transaction maps, excluding reports related to company, trading partner, or standards. See "Accessing transaction map reports from the Reports menu" on page 543.

Index

A

- Abort Transaction** rule command 411
- accessing
 - Cross Reference Tables window from Mapping window 484, 485
 - Element Library window 166
 - full code list 168
 - implementation code list 168
 - reports for any-to-any maps 549
 - Rule Definitions window 403
 - Segment Library window 164
 - segment-related mapping actions 201
 - the standards library 153
- acknowledgements, functional
 - building 314
 - viewing transaction logs for 323
- adapter configuration files, creating 302
- adding
 - action buttons to HTML templates 113
 - adapter configuration file 302
 - any-to-any map flow entries 272
 - codes 160
 - codes to code list 169
 - codes to the full code or implementation code list 150
 - company profiles 49
 - copies of files, including record associations 137
 - copies of segments with related mapping actions 181
 - cross-reference table entries, any-to-any maps 481
 - cross-reference table entries, transaction maps 475, 481
 - cross-reference tables, transaction maps 472
 - databases 133
 - element copy in Mapping window 187
 - elements 148, 158, 166
 - elements, outbound transaction mapping 209
 - entries for I/O options 309, 321, 342, 356
 - fields based on EDI elements 198
 - fields to records 123
 - files 133
 - form methods to HTML templates 113
 - inbound map flow entries 264
 - levels 224
 - map parameter values 307, 320, 341, 355
 - maps 25
 - memory variables 363
 - new entries to drop-down lists 110
 - new levels to EDI segments 253
 - notes to elements 192
 - outbound map flow entries 256
 - parameter values for a map 307, 320, 341, 355
 - print maps 502
 - projects 16
 - radio buttons 111
 - records 121
 - rule commands 410
 - rules 404
 - segments 165
 - segments for customized transactions 147
 - segments to the standards library 156
 - style sheets 109
 - trade agreements 67
 - trading partners 54
 - transactions 163
 - transactions to the standards library 154
 - users 89
- address book 48
- administrative password, changing 88
- administrative tasks 89
- Advanced tab, Any-to-Any Record Flow window 276
- ALL trading partners, associating XML maps with 526
- alphabetical list
 - of EMap reports 540
 - of system variables 376
 - rule commands 399
- Alphanumeric Dates for Memory Variables report 540

Index

- Alphanumeric Dates for Record Fields report 540
 - any-to-any map flow entries 276
 - adding 272
 - deleting 276
 - multiple files option 279
 - viewing 277
 - Any-to-Any Map window 176
 - using system variables on 375
 - any-to-any mapping 225
 - any-to-any maps 24, 359
 - adding cross-reference table entries to 481
 - defining 27
 - deleting cross-reference tables from 479
 - finding cross-reference entries 483
 - manually creating map flow 270
 - reports 547
 - reports, accessing 549
 - running 347
 - running, File Alias tab 353
 - running, I/O Redirect tab 356
 - running, ODBC Alias tab 354
 - running, Option 1 tab 349
 - running, Option 2 tab 351
 - running, Parameters tab 355
 - running, Required tab 347
 - running, Web Script tab 357
 - viewing fields and records associated with 277
 - viewing levels used in 277
 - viewing the rules used in 277
 - viewing trace files for 358
 - Any-to-Any Record Flow window
 - Advanced tab 276
 - Required tab 272
 - application data
 - inbound transaction mapping 194
 - mapping EDI elements to in inbound transaction mapping 194
 - outbound transaction mapping 216
 - Application File Record Listing report 540
 - Application Record Cross Reference report 540
 - Application Records Detail Listing report 540
 - Application Records Summary Listing report 540
 - applying options for elements
 - outbound transaction mapping 214
 - applying options, elements
 - inbound transaction mapping 192
 - archiving maps 38
 - Arithmetic** rule command 371, 413
 - assigning levels to EDI segments 223, 253
 - adding new levels 253
 - Assignment** rule command 372, 415
 - associating
 - compliance maps with trading partners 516
 - cross-reference tables with elements 191, 213
 - print maps with trading partners 505
 - XML maps with ALL trading partners 526
 - associations
 - between cross-reference tables and elements, removing 486
 - between directories and files, creating 142
 - between directories and files, removing 137, 142
 - between files and records, creating 129
 - between files and records, removing 129, 137
 - attaching
 - conditions to elements, inbound transaction mapping 188
 - conditions to elements, outbound transaction mapping 210
 - conditions to segments 204
 - rules to elements, inbound transaction mapping 189
 - rules to elements, outbound transaction mapping 212
 - automatic mapping of EDI to and from serialized NDO 39
 - automatically populating fields in a cross-reference table, transaction maps 477
- ## B
- Bad mailboxes 85
 - batch command file, running compliance maps with 520
 - Binary Data Placement** rule command 416
 - Build Acknowledgement window
 - File Alias tab 318
 - I/O Redirect tab 321
 - ODBC Alias tab 319
 - Option 1 tab 316
 - Option 2 tab 317
 - Parameters tab 320

Required tab 314
 building functional acknowledgements 314

C

Change Password function 92

changing

- codes 160
- codes in the code list 150
- compliance maps 519
- databases 136
- directory names 141
- elements 149, 159, 167
- fields in records 129
- files 136
- levels 224, 254
- memory variables 367
- parameter values for a map 307, 320, 341, 355
- passwords 90
- records 122
- rules 405
- segments 165
- segments in customized transactions 148
- segments in the standards library 157
- the administrative password 88
- transactions 163
- view type of standards library 156

check boxes 112

Check Point rule command 416

Clear rule command 418

Close Connection for File rule command 434

Close Cursor for File Record rule command 434

COBOL copybook, importing record definitions from 94

code list

- adding codes to 150, 169
- deleting codes from 170
- displaying for an element 219
- displaying for inbound transaction maps 197
- finding codes in 170
- modifying codes in 150, 169

code-related report, producing 161, 171

codes

- adding 160
- adding to code list 169

deleting 161

deleting from code list 170

finding 161

finding in code list 170

locating 222

modifying 160

modifying in code list 169

modifying in the code list 150

command lines, removing from mappings 228

commands

Abort Transaction 411

Arithmetic 413

Assignment 415

Binary Data Placement 416

Check Point 416

Clear 418

Close Connection for File 434

Close Cursor for File Record 434

Comment 418

Commit All Files 434

Commit One File 434

Concatenate Strings 418

Do Nothing 421

Execute SQL Command 434

Fetch after Prior Select 435

Fetch after Procedure Call 436

Fetch Once after Prior Select 436

File Management 421

HTML I/O 422

If Condition 423

Insert Record into Table 437

Keyed Record I/O 426

Map Level 427

NDO File Command 428

Perform Rule 429

Put File to Queue 431

Rollback All Files 437

Rollback One File 437

Select and Update 438

Select for Retrieval 439

Select Once for Retrieval 441

Sequential I/O 432

SQL 432

Stop Run 444

String Operations 444

Substring 452

- Table Conversion** 454
- User Exit** 455
- Write Log** 459
- XML I/O** 462
- Comment** rule command 418
- Commit All Files** rule command 434
- Commit One File** rule command 434
- Compact Databases option 534
- company ID 49
- Company Identification report 540
- company profiles
 - adding 49
 - deleting 52
 - modifying 52
- compiling maps 282
- compliance checking and data mapping processes 511
- compliance maps 24
 - associating with trading partners 516
 - creating and running 514
 - creating transactions to be checked by 515
 - defining 514
 - generating 517
 - modifying 519
 - regenerating 519
 - running 518
 - running with batch command file 520
- Concatenate Strings** rule command 372, 418
- conditional mapping 229
 - elements 243
 - inbound 231
 - inbound, elements, based on data criteria 233
 - inbound, elements, based on trading partner 232
 - inbound, segments, based on trading partner 231
 - outbound 236
 - outbound, elements, based on data presence 240
 - outbound, elements, based on trading partner 239
 - outbound, segments, based on data criteria 238
 - outbound, segments, based on trading partner 237
 - segments 242
 - trading partners 242
- conditional variables
 - definition of 196
 - mapping elements to 195, 218
- conditions
 - attaching to elements, inbound transaction mapping 188
 - attaching to elements, outbound transaction mapping 210
 - attaching to segments 204
- configuration files
 - adapter, creating 302
- constants, mapping to elements 219
- contacts 53
- converting EDI transactions
 - to readable forms 502
 - to XML 524
- copies
 - of elements, creating 209
 - of files, creating 137
- copy map utility
 - copy map log, viewing 495
 - This Map from Directory option 494
 - This Map from Map option 493
 - This Map to Directory option 493
 - This Map to Map option 492
- copying
 - maps 35
 - records 123
 - rules 405
 - segments 181
 - trading partners 65
 - trading partners database tables 65
- Create Transaction window, accessing transaction map reports from 545
- creating
 - adapter configuration file 302
 - associations between directories and files 136, 142
 - associations between files and records 129
 - compliance maps 514
 - copies of directories 142
 - copies of elements 209
 - copies of files, including record associations 137
 - copies of inbound map flow 269
 - database tables that contain a project definition 20
 - databases 133
 - fields based on EDI elements 220
 - files 133
 - inbound map flow, automatically 263
 - inbound map flow, manually 263
 - map flow, any-to-any maps 270
 - ODBC tables 106
 - outbound map flow 254

- outbound map flow, automatically 255
- outbound map flow, manually 255
- print maps 502
- records 121
- transactions to be checked by compliance maps 515
- XML map, automatically 524
- creating ODBC tables 106
- creating templates for an HTML form 107
- Cross Reference Tables report 540
- Cross Reference Tables window, accessing 484, 485
- cross-reference tables
 - adding entries, transaction maps 475, 481
 - adding, transaction maps 472
 - associating with elements 191, 213
 - automatically populating fields, transaction maps 477
 - definition of 470
 - deleting entries, transaction maps 477, 482
 - deleting, any-to-any maps 479
 - deleting, transaction maps 473
 - entries, adding, any-to-any maps 481
 - entries, finding, any-to-any maps 483
 - finding for transaction maps 473
 - locating 223
 - modifying entries, transaction maps 476, 477, 478, 482, 483
 - removing associations between elements and 486
 - unreferenced, displaying, transaction maps 474, 480

D

- data
 - criteria, inbound conditional element mapping based on 233
 - criteria, outbound conditional element mapping based on 238
 - presence, outbound conditional element mapping based on 240
- database structures, updating 529
- database tables, creating for project definitions 20
- database/table structure 140
- databases
 - creating 133

- modifying 136
- Date Cross Reference (Memory Variables) report 540
- Date Field Cross Reference (Fields) report 540
- dates
 - updating 536
- defining
 - any-to-any maps 27
 - compliance maps 514
 - transaction maps 27
- definitions
 - conditional variable 196
 - cross-reference tables 470
 - flow 2
 - level 250
 - map 24
 - map flow 250
 - memory variables 362
 - project 16
 - rule 2
 - rules 398
 - system variables 370
- deleting 367
 - any-to-any map flow entries 276
 - associations between directories and files 137, 142
 - associations between files and records 129, 137
 - codes 161
 - codes from code list 170
 - company profiles 52
 - cross-reference table entries, transaction maps 477, 482
 - cross-reference tables from transaction maps 473
 - cross-reference tables, any-to-any maps 479
 - directories 142
 - elements 159, 167, 187
 - elements, outbound transaction mapping 210
 - fields in records 128
 - files and associated records 136
 - inbound map flow entries 268
 - levels 224, 254
 - map parameter values 308, 320, 341, 355
 - maps 35
 - memory variables 367
 - outbound map flow entries 259
 - parameter values for a map 308, 320, 341, 355
 - projects 18
 - radio buttons 112

Index

- records 122
- rule commands 411
- rules 405
- segments 165, 182
- segments from the standards library 157
- segments, outbound transaction mapping 203
- style sheets 109
- trade agreements 83
- trading partners 64
- transactions 163
- transactions from the standards library 155
- users 89
- Delimiter/Terminator tab of the Trading Partner – New window 62
- Delimiter/Terminator tab of Trading Partner window 54
- destination fields
 - mapping system variables to 227
 - mapping to source fields 226
- detail view
 - segments 183, 207
- directories 140
 - and files, creating associations between 136
 - and files, removing associations between 137
 - creating copies of 142
 - deleting 142
 - modifying names of 141
- directories and files
 - removing associations between 137
- directory/file/record structure 140
- displaying
 - code list for an element 219
 - unreferenced cross-reference tables, transaction maps 474, 480
- Do Nothing** rule command 421
- drop-down lists
 - adding new entries to 110
 - field type 110, 111
- E**
- e-Biz 2000, exporting schemas to 114
- e-Biz Integrator, exporting schemas to 115
- EDI
 - elements, creating fields based on 198, 220
 - elements, mapping to application data 194
 - environment 510
 - files, viewing, inbound maps 313
 - files, viewing, outbound maps 346
 - identifier values 45
 - messages, selecting segments to include in 151
 - segments, adding levels to 253
 - segments, assigning levels to 253
 - standards 2
 - standards, incoming documents 510
 - standards, outgoing documents 511
 - transactions, converting to readable forms 502
 - transactions, converting to XML 524
- EDIFACT standard
 - Map Properties tab 29
- Electronic Data Interchange. *See* EDI
- electronic signature (company ID) for outgoing messages 49
- Element Library window, accessing 166
- Element Rule Summary report 540
- Element Values report 540
- elements
 - adding 148, 158, 166
 - adding copies in Mapping window 187
 - adding notes 192
 - adding notes, outbound transaction mapping 214
 - adding, outbound transaction mapping 209
 - applying options for, outbound transaction mapping 214
 - applying options, inbound transaction mapping 192
 - associating cross-reference tables with 213
 - associating with cross-reference tables 191
 - attaching conditions to 188
 - attaching rules to, inbound transaction mapping 189
 - attaching rules to, outbound transaction mapping 212
 - conditional mapping 243
 - copying, with all related mapping actions 187
 - creating copies, outbound transaction mapping 209
 - deleting 159, 167, 187
 - deleting, outbound transaction mapping 210
 - displaying the code list for 219
 - finding 159, 168
 - ignoring (not mapping) 197, 220

- inbound transaction mapping 185
- inserting new in Mapping window 187
- linking to trading partners 190, 212
- mapping constants to 219
- mapping to a field in a record 217
- mapping to a memory variable 217
- mapping to conditional variables 195, 218
- mapping to fields in a record 195
- mapping to memory variable 195
- modifying 149, 159, 167
- outbound transaction mapping 208
- removing associations between cross-reference tables and 486
- repeating the mapping of 196, 218
- unmapped, ignoring 182, 204
- updating mapping for 191
- updating mapping for, outbound transaction mapping 213
- entering a user name and password 88
- entries
 - adding to inbound map flow 264
 - deleting from inbound map flow 268
 - in drop-down lists, adding 110
 - modifying in inbound map flow 269
- examples
 - inbound conditional element mapping, trading-partner-specific 245
 - inbound conditional segment mapping, trading-partner-specific 244
 - outbound conditional element mapping, trading-partner-specific 246
 - outbound conditional segment mapping, trading-partner-specific 246
- Execute SQL Command** rule command 434
- exporting
 - projects 19
 - schemas, to e-Biz 2000 114
 - schemas, to e-Biz Integrator 115
 - schemas, to MQSI 115

F

- Fetch after Prior Select** rule command 435
- Fetch after Procedure Call** rule command 436
- Fetch Once after Prior Select** rule command 436

- Field Cross Reference report 540
- field labels associated with a field, modifying 109
- field types
 - check boxes 112
 - drop-down list 110, 111
 - radio buttons 111
 - text box 110
- fields
 - adding to records 123
 - associated with any-to-any maps, viewing 277
 - creating based on EDI elements 198, 220
 - finding 222
 - in records, deleting 128
 - in records, modifying 129
 - mapping elements to, in a record 195
- File Alias tab 339
 - Build Acknowledgement window 318
 - Run Any-to-Any Map window 353
 - Run Inbound Map window 303
 - Run Outbound Map window 339
- file layouts 94
- File Management** rule command 421
- files
 - and associated records, removing 136
 - and directories, creating associations between 136
 - and records, creating associations between 129
 - and records, deleting associations between 129
 - and records, removing associations between 137
 - creating 133
 - creating copies of, including record associations 137
 - importing record definitions from 102
 - modifying 136
- find function, outbound transaction mapping 221
- finding
 - codes 161, 222
 - codes in code list 170
 - cross-reference table entries, any-to-any maps 483
 - cross-reference tables 223
 - cross-reference tables, transaction maps 473
 - elements 159, 168
 - fields 222
 - mappings 222
 - memory variables 222, 368
 - records 222
 - rules 223

Index

- segments 165
- segments in the standards library 157
- transactions 164
- transactions within the standards library 156
- Flow Definition report 540
- Flow Detail report 540
- Flow Rule Summary report 541
- flow, definition of 2
- folders 140
- form methods, adding to HTML templates 113
- full code list, accessing 168
- functional acknowledgements
 - building 314
 - viewing trace files for 322
 - viewing transaction logs for 323

G

- Generate EDI to XML Map utility 524
 - automatically creating an XML map 524
 - generating the XML map 524
- generating
 - compliance maps 517
 - maps 282
 - multiple maps in a project 38
- getting to the standards library 152
- Good mailboxes 84

H

- HL7 health care standard
 - Map Properties tab 28
- HTML forms
 - creating templates 107
 - importing record definitions from 100
 - naming the template 107
 - selecting style sheets to apply to 109
- HTML I/O** rule command 422
- HTML templates
 - adding action buttons to 113
 - adding form methods to 113

I

- I/O options, adding entries for 309, 321, 342, 356
- I/O Redirect tab
 - Build Acknowledgement window 321
 - Run Any-to-Any Map window 356
 - Run Inbound Map window 308
 - Run Outbound Map window 342
- icons, viewing 468
- identifier values for EDI 45
- If Condition** rule command 373, 423
- ignoring unmapped elements 182, 197, 204, 220
- implementation code list, accessing 168
- importing
 - memory variables 367
 - ODBC record definitions 94, 96
 - projects 20
 - record definitions from a COBOL copybook 94
 - record definitions from files 102
 - record definitions from HTML 100
 - record definitions from maps 102
 - record definitions from XML 98
 - rules from other projects 405
- In mailboxes 84
- inbound map flow
 - creating automatically 263
 - creating copies of 269
 - creating manually 263
 - viewing 269
 - viewing levels associated with 269
 - viewing rules used 269
 - viewing segment detail for 270
- inbound map flow entries
 - adding 264
 - deleting 268
 - modifying 269
- inbound maps
 - conditional mapping 231
 - print maps 502
 - running 287
 - running, File Alias tab 303
 - running, I/O Redirect tab 308
 - running, ODBC Alias tab 305
 - running, Option 1 tab 291
 - running, Option 2 tab 294
 - running, Parameters tab 306
 - running, Required tab 287

- running, Web Script tab 310
- viewing the EDI file for 313
- viewing the trace files for 312
- viewing transaction logs 313
- inbound transaction mapping 180
 - application data 194
 - conditional, element, example of 245
 - conditional, elements, based on data criteria 233
 - conditional, elements, based on trading partner 232
 - conditional, segment, example of 244
 - conditional, segments, based on trading partner 231
 - displaying code lists 197
 - elements 185
 - mapping EDI elements to application data 194
- Insert Record into Table** rule command 437

K

- Keyed Record I/O** rule command 373, 426

L

- large icons, viewing 468
- Level List report 541
- levels
 - adding 224
 - assigning new to EDI segments 253
 - assigning to EDI segments 223, 253
 - associated with inbound map flow 269
 - definition of 250
 - deleting 224, 254
 - in any-to-any maps, viewing 277
 - modifying 224, 254
 - viewing (outbound map flow) 260
- linking elements to trading partners 190, 212
- list of rule commands 399
- locating
 - codes 161, 222
 - cross-reference tables 223
 - cross-reference tables, transaction maps 473
 - elements 159, 168
 - fields 222

- mappings 222
- memory variables 222, 368
- records 222
- rules 223
- segments 165
- transactions 164
- transactions within the standards library 156
- log files for print maps 503
- logging in to EMap 88
- loop ID, viewing 184, 207

M

- mailboxes 140
 - Bad 85
 - Good 84
 - In 84
 - Out 84
- mailboxing chart 84
- Make Transaction option 151
- Make Transaction report 541
- making
 - a segment trader-partner-specific 206
 - transactions 151
- manually creating map flow in any-to-any maps 270
- Map Detailed Definition report 541
- Map DSN tab 33
- map flow
 - definition of 250
 - manually creating, any-to-any maps 270
- map flow, inbound
 - adding entries 264
 - adding entries, any-to-any mapping 272
 - creating 263
 - creating copies of 269
 - creating manually 263
 - deleting entries 268
 - modifying entries 269
- Map Implementation Guide report 541
- Map Level** rule command 427
- map name extension 287, 288, 330, 347
- map parameter values
 - adding 307, 320, 341, 355
 - deleting 308, 320, 341, 355
 - modifying 307, 320, 341, 355

Index

- Map Properties tab 29
- Map Quick Reference report 541
- Map Segment Summary report 541
- mapping
 - an element to a field in a record 217
 - an element to a memory variable 217
 - any-to-any 225
 - automatic to and from serialized NDO 39
 - conditional 229
 - conditional, elements 243
 - conditional, inbound maps 231
 - conditional, outbound 236
 - conditional, segments 242
 - constants to elements 219
 - EDI data to application data (outbound transaction mapping) 217
 - elements to conditional variables 195, 218
 - elements to fields in a record 195
 - elements to memory variables 195
 - inbound maps, viewing 270
 - refreshing mapping commands 227
 - repeating the mapping of an element 218
 - repeating the mapping of elements 196
 - source fields to destination fields 226
 - system variables to destination fields 227
 - trading partner conditionals 242
 - updating for an element 191
 - updating for elements, outbound transaction mapping 213
- mapping actions
 - creating a segment copy with related 181
 - creating element copies with all related 187
 - segment-related 201
- Mapping window
 - accessing the Cross Reference Tables window from 484, 485
 - adding element copy 187
 - inserting elements 187
 - Memory Variable pane 368
 - refreshing 228
- mappings
 - locating 222
 - removing command lines 228
- maps
 - adding 25
 - any-to-any 24
 - any-to-any, defining 27
 - archiving 38
 - compiling 282
 - compliance 24
 - copying 35
 - definition of 24
 - deleting 35
 - generating 282
 - generating multiple 38
 - importing record definitions from 102
 - inbound, running 287
 - modifying 34
 - outbound transaction 174
 - print 24
 - selecting 34
- Memory Variable Cross Reference report 541
- Memory Variable Dates of Length 8 or Longer report 541
- Memory Variable Dates report 541
- Memory Variable Dates with 2-Digit Year report 541
- Memory Variable pane of the Mapping window 368
- memory variables 367
 - adding 363
 - definition of 362
 - finding 368
 - importing 367
 - locating 222
 - mapping an element to 217
 - mapping elements to 195
 - modifying 367
- Memory Variables report 541
- Memory Variables window 363
- menus, View 466
- modifying 276
 - administrative password 88
 - any-to-any map flow entries 276
 - codes 160
 - codes in code list 169
 - codes in the code list 150
 - company profiles 52
 - compliance maps 519
 - cross-reference table entries, transaction maps 476, 477, 478, 482, 483
 - databases 136
 - directory names 141
 - elements 149, 159, 167

- fields in records 129
- files 136
- inbound map flow entries 269
- levels 224, 254
- map parameter values 307, 320, 341, 355
- maps 34
- memory variables 367
- outbound map flow entries 260
- parameter values for a map 307, 320, 341, 355
- passwords 90, 92
- projects 18
- radio buttons 112
- records 122
- rule commands 410
- rules 405
- segments 165
- segments in customized transactions 148
- segments in the standards library 157
- trade agreements 83
- trading partners 64
- transaction 163
- transactions in the standards library 155
- view type of standards library 156
- Web parameter or field label associated with a field 109
- mproject** table 16
- MQSI, exporting schemas to 115
- mtable** table 16
- multiple files option
 - any-to-any map flow 279
 - outbound map flow 262
- mxref** table 16

N

- NDO File Command** rule command 428
- New Map Definition window 25
 - Map Directories tab 29
 - Map DSN tab 33
 - Map Properties tab, EDIFACT standard 29
 - Map Properties tab, HL7 standard 28
 - Map Properties tab, X12 standard 27
- not mapping (ignoring) elements 197
- notes
 - about segments, entering 184

- adding to elements 192
- adding to elements, outbound transaction mapping 214
- segment, entering, outbound transaction mapping 207

O

- ODBC
 - record definitions, importing 94, 96
 - tables, creating 106
- ODBC Alias tab
 - Build Acknowledgement window 319
 - Run Any-to-Any Map window 354
 - Run Inbound Map window 305
 - Run Outbound Map window 340
- Option 1 tab
 - Build Acknowledgement window 316
 - Run Any-to-Any Map window 349
 - Run Inbound Map window 291
 - Run Outbound Map window 334
- Option 2 tab
 - Build Acknowledgement window 317
 - Run Any-to-Any Map window 351
 - Run Inbound Map window 294
 - Run Outbound Map window 336
- options
 - applying for elements, inbound transaction mapping 192
 - applying for elements, outbound transaction mapping 214
- Out mailboxes 84
- outbound map flow
 - adding entries 256
 - creating automatically 255
 - creating manually 255
 - deleting entries 259
 - modifying entries 260
 - selecting multiple files option 262
 - viewing 260
 - viewing fields and records 260
 - viewing levels 260
 - viewing rules used 260
- outbound maps
 - ODBC Alias tab 340

Index

- running 330
 - running, File Alias tab 339
 - running, I/O Redirect tab 342
 - running, Option 1 tab 334
 - running, Option 2 tab 336
 - running, Parameters tab 341
 - running, Required tab 330
 - running, Web Script tab 343
 - viewing EDI files for 346
 - viewing trace files for 344
 - viewing transaction logs for 345
 - outbound transaction mapping 174
 - adding elements 209
 - adding notes to elements 214
 - application data 216
 - attaching conditions to elements 210
 - conditional 236
 - conditional, element, based on data presence 240
 - conditional, element, example of 246
 - conditional, elements, based on trading partner 239
 - conditional, segment, example of 246
 - conditional, segments, based on data criteria 238
 - conditional, segments, based on trading partner 237
 - creating copies of elements 209
 - creating copies of segments 203
 - elements 208
 - entering notes about segments 207
 - find function 221
 - mapping EDI data to application data 217
 - segments 201
 - selecting and inserting segments 202
- P**
- parameter values for a map
 - adding 307, 320, 341, 355
 - deleting 308, 320, 341, 355
 - modifying 307, 320, 341, 355
 - Parameters tab
 - Build Acknowledgement window 320
 - Run Any-to-Any Map window 355
 - Run Inbound Map window 306
 - Run Outbound Map window 341
 - passwords
 - changing 90, 92
 - entering 88
 - Perform Rule** rule command 374, 429
 - Performed Rules report 541
 - physical organization of EMap information 140
 - Possible Date Memory Variables report 541
 - Possible Record Field Dates report 541
 - print maps 24, 502
 - associating with trading partners 505
 - converting EDI transactions to readable forms 502
 - creating 502
 - log files 503
 - regenerating using Generate Map 506
 - regenerating using Generate Print Map 506
 - running 505
 - Project report 542
 - projects
 - adding 16
 - definition of 16
 - deleting 18
 - exporting 19
 - importing 20
 - importing rules from 405
 - modifying 18
 - selecting 19
 - projects.mdb** file 16
 - Put File to Queue** rule command 431
- R**
- radio buttons 111
 - adding 111
 - deleting 112
 - modifying 112
 - Read HTML Data** rule command 422
 - recalculating the starting column of fields in a record 129
 - reconciliation maps 328
 - record definitions
 - importing from a COBOL copybook 94
 - importing from files 102
 - importing from HTML 100
 - importing from maps 102
 - importing from XML 98
 - ODBC, importing 94, 96
 - Record Field Dates of Length 8 or Longer report 542

- Record Field Dates with 2-Digit Year report 542
- records 94
 - adding 121
 - and files, creating associations between 129
 - and files, deleting associations between 129
 - and files, removing associations between 137
 - associated with any-to-any maps, viewing 277
 - copying 123
 - deleting 122
 - locating 222
 - mapping elements to fields 195
 - modifying 122
 - viewing details 123
 - viewing, outbound map flow 260
- refreshing
 - mapping commands 227
 - Mapping window 228
- regenerating
 - compliance maps 519
 - print maps using Generate Map 506
 - print maps using Generate Print Map 506
- removing
 - associations between directories and files 137, 142
 - associations between files and records 129, 137
 - files and associated records 136
- repeating the mapping of an element 196
- reports
 - accessing, for any-to-any maps 549
 - alphabetic list of 540
 - Alphanumeric Dates for Memory Variables 540
 - Alphanumeric Dates for Record Fields 540
 - any-to-any maps 547
 - Application Files Record Listing 540
 - Application Record Cross Reference report 540
 - Application Records Detail Listing 540
 - Application Records Summary Listing 540
 - code-related 161, 171
 - Company Identification 540
 - Cross Reference Tables 540
 - Date Cross Reference (Memory Variables) 540
 - Date Field Cross Reference (Fields) 540
 - Element Rule Summary 540
 - Element Values 540
 - Field Cross Reference 540
 - Flow Definition 540
 - Flow Detail 540
 - Flow Rule Summary 541
 - Level List 541
 - Make Transaction 541
 - Map Detailed Definition 541
 - Map Implementation Guide 541
 - Map Quick Reference 541
 - Map Segment Summary 541
 - Memory Variable Cross Reference 541
 - Memory Variable Dates 541
 - Memory Variable Dates of Length 8 or Longer 541
 - Memory Variable Dates with 2-Digit Year 541
 - Memory Variables 541
 - Performed Rules 541
 - Possible Date Memory Variables 541
 - Possible Record Field Dates 541
 - producing from the standards library 156
 - Project 542
 - Record Field Dates of Length 8 or Longer 542
 - Record Field Dates with 2-Digit Year 542
 - Rule Cross Reference 542
 - Rule Definition 542
 - Rule List 542
 - segment-related 166
 - Selected Segment 542
 - Selected Transaction Template 542
 - System Variable Cross Reference 542
 - System Variables 542
 - Trading Partner Detail 542
 - Trading Partner List 542
 - Transaction Listing 542
 - transaction map 543
 - transaction map, accessing from the Create Transaction window 545
 - transaction-related 164
- Required tab
 - Any-to-Any Record Flow window 272
 - Build Acknowledgement window 314
 - Run Any-to-Any Map window 347
 - Run Inbound Map window 287
 - Run Outbound Map window 330
- Rollback All Files** rule command 437
- Rollback One File** rule command 437
- rule commands
 - Abort Transaction** 411

- adding 410
- Arithmetic** 371, 413
- Assignment** 372, 415
- Binary Data Placement** 416
- Check Point** 416
- Clear** 418
- Close Connection for File** 434
- Close Cursor for File Record** 434
- Comment** 418
- Commit All Files** 434
- Commit One File** 434
- Concatenate Strings** 372, 418
 - deleting 411
- Do Nothing** 421
- Execute SQL Command** 434
- Fetch after Prior Select** 435
- Fetch after Procedure Call** 436
- Fetch Once after Prior Select** 436
- File Management** 421
- HTML I/O** 422
- If Condition** 373, 423
- Insert Record into Table** 437
- Keyed Record I/O** 373, 426
 - list of 399
- Map Level** 427
 - modifying 410
- NDO File Command** 428
- Perform Rule** 374, 429
- Put File to Queue** 431
- Read HTML Data** 422
- Rollback All Files** 437
- Rollback One File** 437
- Select and Update** 438
- Select for Retrieval** 439
- Select Once for Retrieval** 441
- Sequential I/O** 432
- SQL** 432
- Stop Run** 444
- String Operations** 444
- Substring** 452
- Substring rule** 374
- Table Conversion** 374, 454
- User Exit** 455
 - using system variables in 371
- Write HTML Form** 422
- Write Log** 459
- XML I/O** 462
- Rule Cross Reference report 542
- Rule Definition report 542
- Rule Definitions window
 - accessing 403
- Rule List report 542
- rules
 - adding 404
 - any-to-any maps, viewing 277
 - attaching to elements, inbound transaction mapping 189
 - attaching to elements, outbound transaction mapping 212
 - copying 405
 - definition of 2, 398
 - deleting 405
 - importing from other projects 405
 - importing multiple rules 405
 - locating 223
 - modifying 405
 - viewing, inbound map flow 269
 - viewing, outbound map flow 260
- Run Any-to-Any Map window 347
 - File Alias tab 353
 - I/O Redirect tab 356
 - ODBC Alias tab 354
 - Option 1 tab 349
 - Option 2 tab 351
 - Parameters tab 355
 - Required tab 347
 - Web Script tab 357
- Run Inbound Map window 287
 - File Alias tab 303
 - I/O Redirect tab 308
 - ODBC Alias tab 305
 - Option 1 tab 291
 - Option 2 tab 294
 - Parameters tab 306
 - Required tab 287
 - Web Script tab 310
- Run Map option, starting 286
- Run Outbound Map window 330, 339
 - I/O Redirect tab 342
 - ODBC Alias tab 340
 - Option 1 tab 334
 - Option 2 tab 336

- Parameters tab 341
- Required tab 330
- Web Script tab 343

running

- any-to-any maps 347
- compliance maps 518
- print maps 505

- runtime component 2

S

samples

- inbound conditional element mapping, trading-partner-specific 245
- inbound conditional segment mapping, trading-partner-specific 244
- outbound conditional element mapping, trading-partner-specific 246
- outbound conditional segment mapping, trading-partner-specific 246

schemas

- exporting to e-Biz 2000 114
- exporting to e-Biz Integrator 115
- exporting to MQSI 115

segment detail

- viewing 183, 207
- viewing, inbound map flows 270

- Segment Library window, accessing 164

- segment-related mapping actions 201

- segment-related reports, producing 166

segments

- adding 165
- adding for customized transactions 147
- adding to the standards library 156
- attaching conditions to 204
- conditional mapping 242
- copies of, adding with related mapping actions 181
- copying 181
- creating copies of, outbound transaction mapping 203
- deleting 165, 182
- deleting from the standards library 157
- entering notes about, inbound transaction mapping 184

- entering notes about, outbound transaction mapping 207

- finding 165

- finding in the standards library 157
- in customized transactions, modifying 148

- making trading-partner-specific 206

- modifying 165

- modifying in the standards library 157

- outbound transaction mapping 201

- outbound transaction mapping, deleting 203

- selecting and inserting in outbound transaction mapping 202

- selecting those to be included in EDI messages 151

- Select and Update** rule command 438

- Select for Retrieval** rule command 439

- Select Once for Retrieval** rule command 441

- Selected Segment report 542

- Selected Transaction Template report 542

selecting

- and inserting segments, outbound transaction mapping 202

- maps 34

- multiple files option, outbound map flow 262

- projects 19

- segments to include in EDI messages 151

- style sheets to apply to HTML forms 109

- Sequential I/O** rule command 432

- signature (company ID) 49

- small icons, viewing 468

- source fields, mapping to destination fields 226

- SQL** rule command 432

- Standard Exchange Format (SEF) utility 2, 498

standards

- EDI, incoming documents 510

- EDI, outgoing documents 511

- viewing 466

- standards library 152

- accessing 153

- adding segments to 156

- adding transactions to 154

- changing view type 156

- deleting segments from 157

- deleting transactions from 155

- finding segments in 157

- finding transactions within 156

Index

- getting to 152
 - modifying segments in 157
 - modifying transactions 155
 - producing transaction-related reports 156
 - starting Run Map option 286
 - Stop Run** rule command 444
 - stored procedure call 442
 - String Case Conversion** string operation 445
 - String Find** string operation 445
 - String Justification** string operation 447
 - String Length** string operation 448
 - string operations
 - String Case Conversion** 445
 - String Find** 445
 - String Justification** 447
 - String Length** 448
 - String Replace** 449
 - String Type** 451
 - String Operations** rule command 444
 - String Replace** string operation 449
 - String Type** string operation 451
 - style sheets
 - adding 109
 - deleting 109
 - Substring** rule command 374, 452
 - switching between mapping and rule functionality using the
 - View Rules button 228
 - System Variable Cross Reference report 542
 - system variables
 - alphabetical list 376
 - definition of 370
 - mapping to destination fields 227
 - using in a rule command 371
 - using on the Any-to-Any Map window 375
 - viewing list of 370
 - System Variables report 542
- ## T
- TA1 acknowledgements 324–329
 - reconciliation map 328
 - requesting 328
 - setting up 325
 - Table Conversion** rule command 374, 454
 - templates
 - naming for the HTML form 107
 - text box field type 110
 - This Map from Directory option, copy map utility 494
 - This Map from Map option, copy map utility 493
 - This Map to Directory option, copy map utility 493
 - This Map to Map option, copy map utility 492
 - toolbars, viewing 467
 - trace files
 - viewing, any-to-any maps 358
 - viewing, functional acknowledgements 322
 - viewing, inbound maps 312
 - viewing, outbound maps 344
 - trade agreements 67
 - adding 67
 - deleting 83
 - modifying 83
 - Trading Partner – New window
 - Delimiter/Terminator tab 62
 - Trading Partner Detail report 542
 - Trading Partner List report 542
 - Trading Partner window 54
 - Delimiter/Terminator tab 54
 - trading partners 53
 - adding 54
 - ALL, associating XML maps with 526
 - associating compliance maps with 516
 - associating print maps with 505
 - conditionals 242
 - copying 65
 - database tables, copying 65
 - deleting 64
 - inbound conditional element mapping based on
 - 231, 232
 - linking elements to 190, 212
 - modifying 64
 - outbound conditional element mapping based on
 - 237, 239
 - trading-partner-specific, making a segment 206
 - Transaction Listing report 542
 - transaction logs
 - viewing, any-to-any maps 359
 - viewing, functional acknowledgements 323
 - viewing, inbound maps 313
 - viewing, outbound maps 345
 - transaction map reports
 - from the Create Transaction window 545

- Transaction Mapping window 174
 - transaction mapping, inbound 180
 - application data 194
 - elements of 185
 - mapping EDI elements to application data 194
 - transaction mapping, outbound
 - adding notes to elements 214
 - application data 216
 - attaching conditions to elements 210
 - creating copies of 209
 - creating copies of segments 203
 - deleting elements 210
 - deleting segments from 203
 - elements 208
 - entering notes about 207
 - find function 221
 - mapping EDI data to application data 217
 - segments 201
 - selecting and inserting segments 202
 - transaction maps
 - adding cross-reference table entries 475, 481
 - adding cross-reference tables for 472
 - automatically populating fields in a cross-reference table 477
 - defining 27
 - deleting cross-reference table entries from 477, 482
 - deleting cross-reference tables from 473
 - displaying unreferenced cross-reference tables for 474, 480
 - finding cross-reference tables for 473
 - inbound, print maps 502
 - modifying cross-reference tables in 476, 477, 478, 482, 483
 - removing associations between cross-reference tables and elements 486
 - reports 543
 - transaction-related reports
 - producing 164
 - producing from the standards library 156
 - transactions
 - adding 163
 - adding to the standards library 154
 - creating, to be checked by compliance maps 515
 - customized, adding segments for 147
 - customized, modifying segments in 148
 - deleting 163
 - deleting from the standards library 155
 - finding 164
 - finding within the standards library 156
 - making 151
 - modifying 163
 - modifying in the standards library 155
- ## U
- unmapped elements, ignoring 182, 204
 - unreferenced cross-reference tables
 - displaying for transaction maps 474, 480
 - Update Date option 536
 - Update DB Structures option 530
 - updating
 - database structures 529
 - dates 536
 - updating mapping for elements 191
 - User Exit** rule command 455
 - user name, entering 88
 - users
 - adding 89
 - deleting 89
 - utilities
 - Compact Databases 534
 - Generate EDI to XML Map 524
 - Update Date 536
 - Update DB Structures 530
- ## V
- View menu 466
 - View Rules button, switching between mapping and rule functionality 228
 - view type of standards library, changing 156
 - viewing
 - any-to-any map flow entries 277
 - copy map log 495
 - ECMap information 466
 - EDI file, inbound maps 313
 - EDI files, outbound maps 346
 - fields and records, any-to-any maps 277
 - fields and records, outbound map flow, fields 260

- icons 468
- inbound map flow 269
- inbound map flow, rules used 269
- levels used in any-to-any maps 277
- levels, inbound maps 269
- list of system variables 370
- loop ID 184, 207
- mapping, inbound maps 270
- outbound map flow 260
- outbound map flow, levels 260
- outbound map flow, rules used 260
- record details 123
- rules used in any-to-any maps 277
- segment detail 183, 207
- segment detail, inbound map flows 270
- standards 466
- toolbars 467
- trace files, any-to-any maps 358
- trace files, functional acknowledgements 322
- trace files, inbound maps 312
- trace files, outbound maps 344
- transaction logs, any-to-any maps 359
- transaction logs, functional acknowledgements 323
- transaction logs, inbound maps 313
- transaction logs, outbound maps 345
- work spaces 467
- viewing, outbound map flow 260

W

- Web parameters associated with a field, modifying 109
- Web Script tab
 - Run Any-to-Any Map window 357
 - Run Inbound Map window 310
 - Run Outbound Map window 343
- windows
 - Any-to Any Record Flow (Advanced tab) 276
 - Any-to-Any Map 176
 - Any-to-Any Map, using system variables on 375
 - Any-to-Any Record Flow (Required tab) 272
 - Build Acknowledgement 314
 - Create Transaction, accessing transaction map reports
 - from 545
 - Cross Reference Tables 484, 485
 - Element Library 166

- Mapping, accessing the Cross Reference Tables
 - window from 484, 485
- Mapping, inserting a new element in 187
- Mapping, Memory Variable pane 368
- Mapping, refreshing 228
- Memory Variables 363
- Rule Definitions, accessing 403
- Run Any-to-Any Map 347
- Run Inbound Map 287
- Run Outbound Map 330
- Segment Library 164
- Trading Partner 54
- Transaction Mapping 174
- work spaces, viewing 467
- workbench component 2
- Write HTML Form** rule command 422
- Write Log** rule command 459

X

- XML
 - converting EDI transactions to 524
 - XML I/O** rule command 462
 - XML Internet standard
 - importing record definitions from 98
 - XML map, automatically creating 524
 - XML maps
 - associating with the ALL trading partner 526
 - using the Generate EDI to XML Map utility to generate 524