



THE ENTERPRISE MIDDLEWARE SOLUTION

# BEA eLink Mercator Integrator for DIO

## User Guide

BEA eLink Mercator Integrator for DIO Version 1.2  
Document Edition 1.2  
October 1999

# Copyright

Copyright © 1999 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, ObjectBroker, TOP END, TUXEDO, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Connect, BEA eLink, BEA Manager, BEA MessageQ, Jolt and M3 are trademarks of BEA Systems, Inc.

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

### **BEA eLink Mercator Integrator for DIO User Guide**

<b>Document Edition</b>	<b>Part Number</b>	<b>Date</b>	<b>Software Version</b>
1.0	870-001001-001	May 1999	BEA eLink Mercator Integrator for DIO 1.0
1.1	870-001001-001	July 1999	BEA eLink Mercator Integrator for DIO 1.1
1.2	870-001001-002	October 1999	BEA eLink Mercator Integrator for DIO 1.2

---

# Contents

## Preface

Purpose of This Document .....	vii
Who Should Read This Document .....	vii
How This Document Is Organized .....	viii
How to Use This Document .....	viii
Opening the Document in a Web Browser .....	ix
Document Conventions .....	ix
BEA eLink Data Integration Option Documentation .....	x
BEA Publications .....	xi
Other Publications .....	xi
Contact Information .....	xi
Documentation Support .....	xii
Customer Support .....	xii

## 1. Understanding the Data Integration Option

Using the BEA Solution for Data Mapping .....	1-2
Using Data Mapping with a Mainframe Environment .....	1-2
Using Data Mapping with SAP R/3 IDOCs .....	1-4
FML32 Value Buffers with SAP R/3 .....	1-5
Using Data Mapping with Other eLink Integrators .....	1-6
Requirements for Data Mapping .....	1-7
BEA eLink Integrator Components .....	1-7
TUXEDO elinkmerc Server .....	1-7
FML and VIEW Importer .....	1-8
Mercator Components .....	1-8
Importers .....	1-8
Type Editor .....	1-9

---

Map Editor.....	1-9
Overview of the eLink Integrator Data Mapping Process .....	1-9
Development Phase .....	1-11
Using Importers to Create Mercator Type Tree Files .....	1-12
Compiling Type Tree Files into Mercator Map Files .....	1-12
Execution Phase.....	1-12
Setting Up the Execution Environment.....	1-13
Initiating a Service Request for Data Mapping.....	1-14

## **2. Installing BEA eLink Integrator**

Installation Prerequisites.....	2-1
Installing on the Development Platform (Windows NT) .....	2-2
Installing on the Execution Platform .....	2-6
Installing for Execution on Windows NT .....	2-7
Installing for Execution on a UNIX Platform (HP-UX, AIX, or SUN Solaris) .....	2-7
Distribution Libraries and Executables .....	2-10
HP-UX.....	2-10
SUN Solaris .....	2-11
IBM AIX .....	2-11
Windows NT 4.0 .....	2-11
Uninstalling eLink Integrator on Windows NT.....	2-12

## **3. Configuring the Environment for eLink Integrator**

Configuring the elinkmerc Server to Run with eLink Integrator .....	3-1
Define the elinkmerc Server.....	3-2
Advertise the Services for Data Mapping .....	3-3
Sample UBBCONFIG File for Data Mapping.....	3-4
Configuring the eLink for Mainframe Gateway to Run with eLink Data Integration Option .....	3-5

## **4. Data Mapping Conversion**

Preparing for Data Mapping Conversion.....	4-1
Creating Files for Conversion .....	4-2
Creating an FML Group Format File .....	4-2
Using the mti2fgf Utility .....	4-4

---

Transferring Files to Development Environment.....	4-4
Developing Rules for Conversions Between FML Buffers and Other Applications .....	4-5
Set System Properties for the Mercator Open Edition Client .....	4-5
Create an FML Type Tree File Using the FML and View Importer.....	4-6
FML Type Tree Fields .....	4-10
Create a non-FML Type Tree File Using Importers .....	4-10
Control Information for TUXEDO /Q .....	4-11
Create a Mercator Map File .....	4-13
Required Files for Conversion Process .....	4-14
Executing a Service Request with Data Mapping .....	4-15
Transfer Files to APPDIR in the Execution Environment .....	4-16
Set Environment Variables.....	4-16
Start BEA TUXEDO and Other Remote Applications .....	4-17
Write a Client Program to Create FML Buffers.....	4-17
Issue a Service Request for Data Mapping .....	4-17

## **A. Transaction Support**

Transactional Message Broker .....	A-2
------------------------------------	-----

## **B. Error and Informational Messages**

## **C. Data Mapping Worksheet**

Step 1: Install Products.....	C-1
Step 2: Set TUXEDO Environment Variables .....	C-2
Step 3: Put Mercator in the System Path .....	C-3
Step 4: Transfer Source Files to Windows NT.....	C-3
Step 5: UBBCONFIG File Definitions.....	C-4
Step 6: DMCONFIG File Definitions (Optional).....	C-5
Step 7: Create an FML Type Tree File.....	C-6
Step 8: Create a non-FML Type Tree File .....	C-6
Step 9: Create a Mercator Map File .....	C-7
Step 10: Port Map Files for Execution on UNIX Platforms.....	C-7
Step 11: Renaming Meta-Type Information File (Optional).....	C-7
Step 12: Transfer Files.....	C-8
Step 13: Start BEA TUXEDO and Remote Applications .....	C-8

---

Step 14: Issue a Service Request for Data Mapping.....	C-8
--	-----

---

# Preface

BEA eLink Mercator Integrator for DIO (hereafter referenced as eLink Integrator) is a component of the eLink Data Integration Option (eLink DIO) that runs with the BEA TUXEDO product as well as other BEA products.

## Purpose of This Document

This document describes the BEA eLink Integrator component and gives instructions for using the tool for data mapping. This guide explains how to install and use the eLink Integrator component and how this component fits into the BEA TUXEDO environment.

## Who Should Read This Document

This document is intended for system administrators who will configure and administer eLink Integrator as well as programmers who will map FML buffers to non-FML buffer types. This guide assumes knowledge of BEA TUXEDO, BEA eLink for Mainframe, TSI Mercator, and SAP R/3 products. This guide also assumes knowledge of the C programming language. In addition, programmers will find useful pointers for developing client programs and service routines that send data through eLink Integrator.

---

## How This Document Is Organized

The *BEA eLink Mercator Integrator for DIO User Guide* is organized as follows:

- ◆ *Understanding the Data Integration Option* introduces the eLink Integrator component and explains how eLink Integrator fits into the BEA TUXEDO environment.
- ◆ *Installing BEA eLink Integrator* explains how to install the eLink Integrator component.
- ◆ *Configuring the Environment for eLink Integrator* provides information for configuring BEA TUXEDO and BEA eLink for Mainframe to run with the eLink Integrator component to handle service requests for data mapping.
- ◆ *Data Mapping Conversion* provides information about using the eLink Integrator component to define the rules for data mapping. It includes information for developing the rules for data mapping and executing a TUXEDO service request for data mapping.
- ◆ *Transaction Support* describes support for TUXEDO transactions integrated with the existing Mercator rollback functionality.
- ◆ *Error and Informational Messages* describes error and informational messages as well as actions to resolve the errors.
- ◆ *Data Mapping Worksheet* is a planning worksheet for the data mapping installations, configurations and tasks.

## How to Use This Document

The *BEA eLink Mercator Integrator for DIO User Guide* is designed primarily as a printed document; however, a PDF version of the document is available on the product CD.



---

## Opening the Document in a Web Browser

To access the online version of this document if it is installed on a Web server, open the following HTML file:

`http://(directory path to eLink Integrator documentation)/index.htm`

**Note:** The online documentation requires a Web browser that supports HTML 3.0. Netscape Navigator 4.0 or higher or Microsoft Internet Explorer 4.0 or higher are recommended.

## Document Conventions

The following documentation conventions are used throughout this document:

<b>Item</b>	<b>Examples</b>
Variable names	Variable names represent information you must supply or output information that can change; they are intended to be replaced by actual names. Variable names are displayed in italics and can include hyphens but not underscores. The following are examples of variable names in text:  <i>error-file-name</i>  The <i>when-return</i> value...
User input and screen output	For screen displays and other examples of input and output, user input appears as in the first of the following lines; system output appears as in the second through fourth lines:  <b>dir c:\accounting\data</b> Volume in drive C is WIN_NT_1 Volume Serial Number is 1234-5678 Directory of C:\BEADIR\DATA

---

Item	Examples
Syntax	Code samples can include the following elements: <ul style="list-style-type: none"> <li>◆ Variable names can include hyphens but not underscores (e.g., <i>error-file-name</i>)</li> <li>◆ Optional items are enclosed in square brackets: [ ]. If you include an optional item, do not code the square brackets.</li> <li>◆ A required element for which alternatives exist is enclosed in braces { }. The alternatives are separated by the pipe (vertical bar) character:  . You must include only one of the alternatives for that element. Do not code the braces or pipe character.</li> <li>◆ An ellipsis ( ... ) indicates that the preceding element can be repeated as necessary.</li> </ul>
Omitted code	An ellipsis ( ... ) is used in examples to indicate that code that is not pertinent to the discussion is omitted. The ellipsis can be horizontal or vertical.
Environment variables	Environment variables are formatted in an uppercase font. ENVFILE=\${APPDIR}
Key names	Key names are presented in boldface type. Press <b>Enter</b> to continue.
Literals	Literals are formatted in a monospace font. <code>class extendSample</code>
Window items	Window items are presented in boldface type. Window items can be window titles, button labels, text edit box names or other parts of the window. Type your password in the <b>Logon window</b> . Select <b>Export</b> to make the service available to the client.

---

## BEA eLink Data Integration Option Documentation

The eLink Data Integration Option documentation consists of the following items:

- ◆ *BEA eLink Mercator Integrator for DIO User Guide*

- 
- ◆ *BEA eLink Data Integration Option Release Notes*

## BEA Publications

The following BEA publications are also available:

- ◆ *TUXEDO System 6 Reference Manual*
- ◆ *TUXEDO System 6 Programmer's Guide, Volumes 1 and 2*
- ◆ BEA eLink for Mainframe Online Documentation CD

## Other Publications

For more information about data mapping, refer to the Mercator documentation set. In particular, refer to the following documents when using the eLink Integrator component.

- ◆ *Mercator Type Editor Reference Guide*
- ◆ *Mercator Map Editor Reference Guide*
- ◆ *Mercator Functions & Expressions Reference Guide*

## Contact Information

The following sections provide information about how to obtain support for the documentation and software.

---

## Documentation Support

If you have questions or comments on the documentation, you can contact the BEA Information Engineering Group by e-mail at [docsupport@beasys.com](mailto:docsupport@beasys.com) (For information about how to contact Customer Support, refer to the following section.)

## Customer Support

If you have any questions about this version of BEA eLink Integrator, or if you have problems installing and running BEA eLink Integrator, contact BEA Customer Support through BEA WebSupport at [www.beasys.com](http://www.beasys.com). You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- ◆ Your name, e-mail address, phone number, and fax number
- ◆ Your company name and company address
- ◆ Your machine type and authorization codes
- ◆ The name and version of the product you are using
- ◆ A description of the problem and the content of pertinent error messages

# 1 Understanding the Data Integration Option

The BEA eLink Data Integration Option uses the eLink Integrator component with BEA TUXEDO for processing service requests for data mapping and data conversion.

This TUXEDO translation service takes an input buffer, performs a data translation on the buffer, and returns the translated data in an output buffer, exposing the Mercator wrapper for TUXEDO as a service. The data translations that can be performed corresponds to sets of Mercator maps. The service will advertise these as a set of service names that correspond to the map names. To perform a particular data translation, the service will be invoked with the appropriate service name - the service will invoke Mercator using the correspondingly named map.

For example, to call a map called TOEDI.MMC that produces EDI data in an FML format, the service name would be TOEDI.

Each map has at least one input card and one output card. The service passes the input data to input card 1 and returns data back to the service-requester via output card 1. There may be any number of additional input or output cards (to read from a file or create a file, for example). Output card 1 has a “header” with control information set in the map. This card determines the destination and format of the output buffer. *This header is required.* See the section entitled “Control Information for TUXEDO /Q” in this document for more details.

This document will describe the following topics:

- ◆ Using the BEA Solution for Data Mapping

- ◆ Using Data Mapping with a Mainframe Environment
- ◆ Using Data Mapping with SAP R/3 IDOCs
- ◆ Using Data Mapping with Other eLink Integrators
- ◆ Requirements for Data Mapping
- ◆ Overview of the eLink Integrator Data Mapping Process
  - ◆ Development Phase
  - ◆ Execution Phase

## Using the BEA Solution for Data Mapping

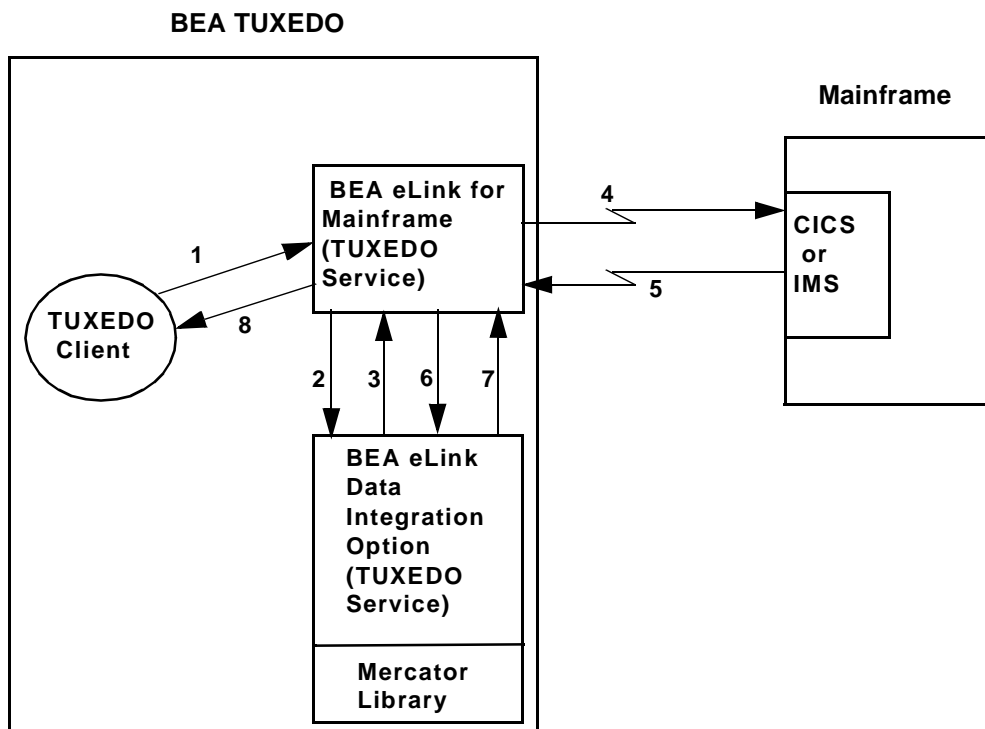
The BEA eLink Data Integration Option uses a TUXEDO environment that includes the BEA TUXEDO product as well as the eLink Mercator Integrator for DIO component for data mapping and conversion. Other products, such as the BEA eLink for Mainframe product and eLink Integrators can be integrated into your TUXEDO environment and assist in your Enterprise Application Integration (EAI) solution.

## Using Data Mapping with a Mainframe Environment

Figure 1-1 shows how the products and components work together to convert buffers between an FML and COBOL copybook format and transfer the buffers between a UNIX or Windows NT environment and a mainframe environment.

The BEA eLink for Mainframe gateway asynchronously calls external TUXEDO services, a feature which will be used to call the elinkmerc server. The elinkmerc server is a standard TUXEDO MSSQ server that receives a TUXEDO buffer as input, invokes the Mercator data mapping engine, and returns a TUXEDO buffer as output.

Figure 1-1 Data Mapping Scenario of FML to COBOL Copybook to FML



The following description explains the process flow in Figure 1-1 of a TUXEDO client making a request to send an FML buffer from a UNIX environment to a CICS region on a mainframe and return an FML buffer to the TUXEDO client.

1. A TUXEDO client issues a request to send an FML buffer to the mainframe.
2. The request passes the FML buffer to the BEA eLink for Mainframe gateway. The service name in the request causes the gateway to pass the FML buffer to the eLink Mercator Integrator for DIO component to convert the FML buffer to a COBOL copybook format.
3. The eLink Mercator Integrator for DIO component converts the FML buffer to a COBOL copybook and passes it back to the BEA eLink for Mainframe gateway.
4. The BEA eLink for Mainframe gateway passes the converted COBOL copybook to the CICS or IMS region on the mainframe.

# 1 Understanding the Data Integration Option

---

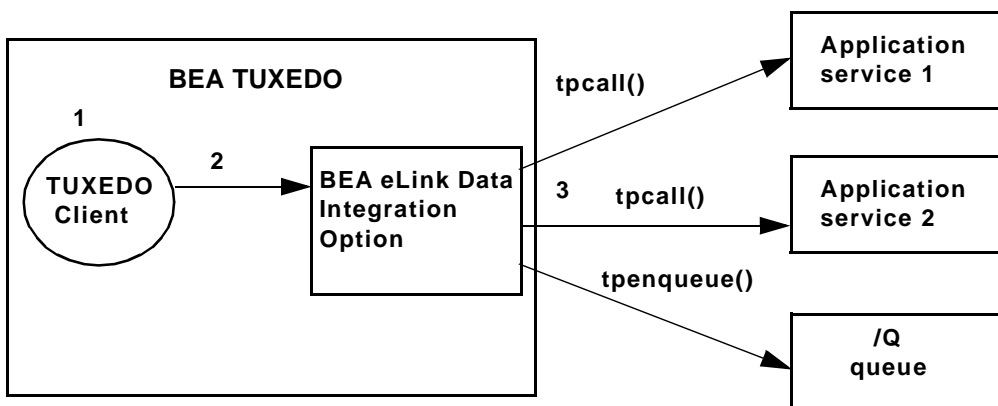
5. After the requested transaction is complete, the mainframe returns a response by sending the COBOL copybook back to the BEA eLink for Mainframe gateway.
6. The BEA eLink for Mainframe gateway passes the copybook to the eLink Integrator for conversion back to an FML buffer.
7. After the copybook is converted to an FML buffer, it is passed back to the BEA eLink for Mainframe gateway.
8. The BEA eLink for Mainframe gateway then passes the FML buffer response to the TUXEDO client.

## Using Data Mapping with SAP R/3 IDOCs

The data mapping process uses a TUXEDO environment that includes the BEA TUXEDO and the BEA eLink Mercator Integrator for DIO component. Figure 1-2 shows how the products and components work together to convert buffers between an FML and SAP Intermediate Documents (IDOCs) format.

The elinkmerc server is a standard TUXEDO MSSQ server that receives a TUXEDO buffer as input, invokes the Mercator data mapping engine, converts the buffer to another format and then passes it to multiple destinations.

**Figure 1-2 Converting FML to New Format and Passing to Multiple Destinations**





The following description explains the process flow in Figure 1-2 of a TUXEDO client making a request to send an FML buffer to multiple applications.

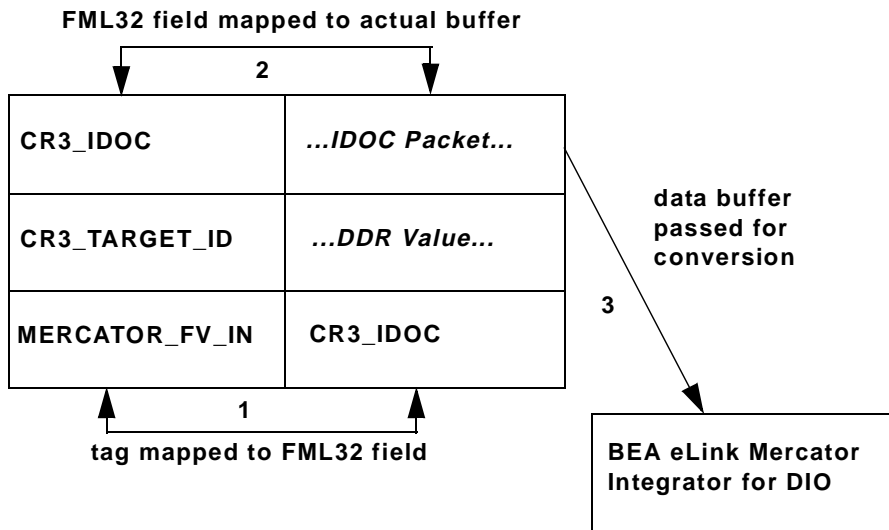
1. A TUXEDO client issues a request to send an FML buffer to multiple applications.
2. The request passes the FML buffer to the eLink Data Integration Option to convert the FML buffer to another buffer format, such as SAP R/3 IDOCs.
3. The eLink Data Integration Option converts the FML buffer to the new format and passes it to the appropriate applications.

**Note:** The mapping to multiple destinations is done through Mercator input and output cards. For information on defining input and output cards, refer to the Mercator documentation.

## FML32 Value Buffers with SAP R/3

A particular use of the BEA eLink Mercator Integrator for DIO component with SAP R/3 is with the conversion and processing of FML32 Value buffers. Figure 1-3 shows how to map a Mercator tag to an FML32 field that then points to the actual data buffer. Once the actual data buffer is located, only it is passed to the BEA eLink Mercator Integrator for DIO application for conversion.

**Figure 1-3 Passing an FML32 Value Buffer from SAP R/3 Outbound**



# 1 Understanding the Data Integration Option

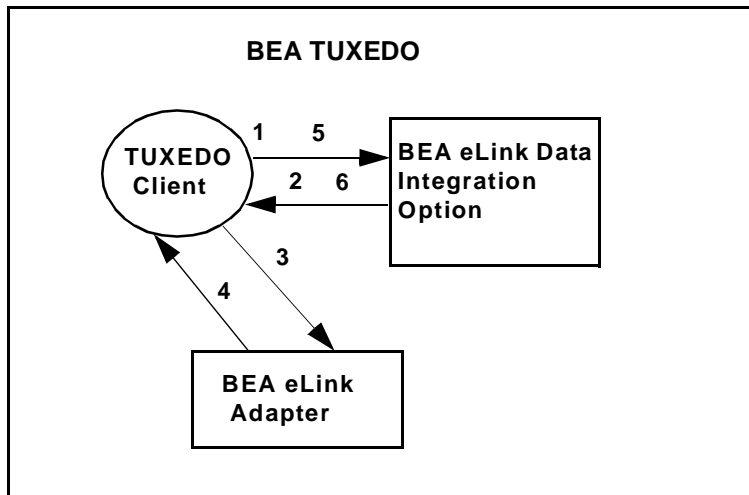
---

1. The MERCATOR\_FV\_IN field identifies the name of the FML32 field.
2. The FML32 name contains the actual data buffer to be passed to the eLink Integrator component.
3. The data is extracted from the FML32 field and sent directly to Mercator for mapping.

## Using Data Mapping with Other eLink Integrators

The data mapping process uses a TUXEDO environment that includes the BEA TUXEDO and the BEA eLink Integrator component as well as other eLink Adapters. For example, Figure 1-4 shows how the products and components work together to convert buffers between an FML and a non-FML format and then pass a request to an eLink Adapter.

**Figure 1-4 Data Mapping Scenario of FML with eLink Integrators**



The following description explains the process flow for invoking eLink DIO directly from the client to transform data prior to sending it to an eLink Adapter. The same process could be used after receiving data from an eLink Adapter to transform it back to a TUXEDO format.

1. A TUXEDO client calls eLink DIO to transform data to a format usable by an eLink Adapter.
2. The eLink DIO software sends the transformed data back to TUXEDO.
3. TUXEDO calls the eLink Adapter to process the data.
4. After the eLink Adapter processes the data, it passes the response back to TUXEDO.
5. The resulting data from the eLink Adapter is passed to the eLink DIO to be transformed into a TUXEDO-compatible format, such as FML.
6. The eLink DIO transforms the data to a TUXEDO-compatible format, such as FML, and passes it back to TUXEDO.

## Requirements for Data Mapping

The following software, components, and files must be used to issue a TUXEDO service request that maps data between an FML buffer and a non-TUXEDO data format, such as COBOL copybook or SAP R/3 IDOCs.

- ◆ BEA eLink Integrator Components
- ◆ Mercator Components

## BEA eLink Integrator Components

The following sections describe the different eLink Integrator components necessary for defining the data mapping rules in the development environment and execution of the mapping operation.

### TUXEDO elinkmerc Server

The elinkmerc server is an executable program that runs on HP-UX, SUN Solaris, AIX or Windows NT. It runs in a TUXEDO environment performing the following functions:

# 1 *Understanding the Data Integration Option*

---

- ◆ Advertises the maps specified by the -s option of the CLOPT parameter in the SERVERS section of the UBBCONFIG file. In other words, the Mercator maps are advertised as TUXEDO services.
- ◆ Maintains other command line information for execution of the Mercator Mapping Library.
- ◆ Executes TuxRunMap library which calls the Mercator Mapping Library to perform data mapping.
- ◆ Receives the FML buffer and converts it into an internal format for use by Mercator.

## **FML and VIEW Importer**

The eLink Integrator FML and VIEW Importer reads the FML Field Definition file and an FML Group Format file to create a type tree file. This component reads the FML Field Definition file and creates a Type Tree Maker Command file (.mts) which is then run through the type tree maker.

## **Mercator Components**

The following sections describe the different Mercator components necessary for defining the data mapping rules in the development environment.

### **Importers**

Use the Mercator Importers to create type tree files. For example, to convert an FML buffer to a COBOL Copybook, use the COBOL Copybook Importer to create a COBOL type tree. For information about using Importers, refer to the Mercator documentation.

### Type Editor

Use the Mercator Type Editor to make any modifications to the type tree files, such as adding the Control information to the header. The Type Editor also provides an Export feature that will produce a type tree file in a suitable format for different platforms. For information about using the Type Editor, refer to the *Mercator Type Editor Reference Guide*.

### Map Editor

The data translations are sets of Mercator maps. The TUXEDO service will advertise these as a set of service names that correspond to the map names. To perform a particular data translation, the service will be invoked with the appropriate service name - the service will invoke Mercator using the correspondingly named map.

Each map has at least one input card and one output card. The service passes the input data to input card 1 and returns data back to the service-requester via output card 1. There may be any number of additional input or output cards (for example, to read from a file or create a file). Output card 1 has a header with control information set in the map. This card determines the destination and format of the output buffer.

**Note:** The header information is *required* in the output card. For information about defining Control information with /Q, refer to the “Control Information for TUXEDO /Q.”

Use the Mercator Map Editor to define the mapping rules and compile a map file. For information about creating a Mercator map file, refer to the *Mercator Map Editor Reference Guide*.

# Overview of the eLink Integrator Data Mapping Process

The data mapping process includes two phases: the development phase and the execution phase. The development phase sets up the environment and mapping rules for data conversion. For example, this phase includes the process of taking a file type, such as a COBOL copybook, and creating a type tree file (such as `cob.mtt`) using the

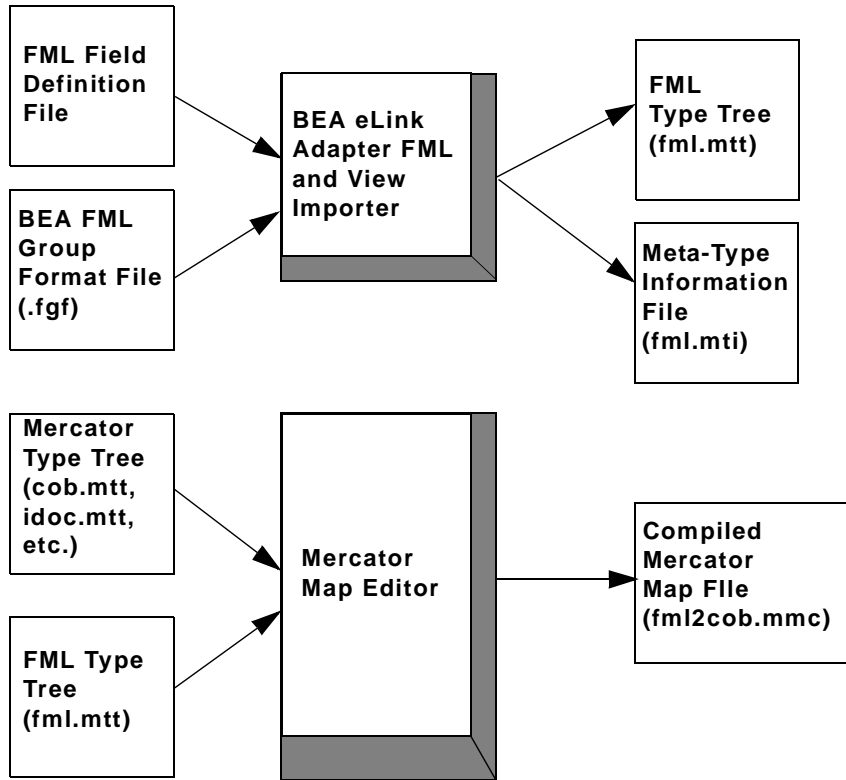
# 1 *Understanding the Data Integration Option*

---

appropriate Mercator Importer. This phase also includes taking an FML Field Definition file and an FML Group Format file and creating an associated FML type tree file (`fml.mtt`) using the eLink Integrator FML and View Importer. The FML and View Importer in the eLink Integrator component also creates a Meta-Type Information (`.mti`) file which the TUXEDO `elinkmerc` server uses during execution. After both the COBOL and the FML type trees are created for both the copybook and the FML files, a Mercator map file (`.mmc`) is created and compiled using the type tree files. The execution phase issues the TUXEDO service request for data mapping. It is during execution that the actual mapping occurs.

Figure 1-5 describes the data mapping development process flow. Simply put, the process flow begins with an input file, such as a COBOL copybook or an FML Field Definition File and an FML Group Format file (`.fgf`) and creates a compiled Mercator map file (`.mmc`).

Figure 1-5 Data Map Development



## Development Phase

The development phase is separated into two parts, the creation of Mercator type tree files and the creation of the Mercator map file. You must have a source type tree, such as a COBOL type tree, and a destination type tree, such as an FML type tree. Use the Mercator and eLink Data Integration Option tools to create the type tree files. Before creating the type tree files, you must describe your input data as well as the format of your output data.

**Note:** The development phase is *only* supported on Windows NT.

## Using Importers to Create Mercator Type Tree Files

Use a Mercator Importer to input a type of file and create a type tree file as output. For more information on how to use the Mercator Importers, refer to the Mercator documentation. Use the eLink Integrator FML and View Importer to create an FML type tree.

**Note:** For conversions to COBOL copybook, the COBOL type tree does not contain the eLink Integrator header, so you must edit the type tree using the Type Editor. You must add the eLink Integrator control header from the FML type tree to output COBOL copybooks.

You can input into the eLink Integrator FML and View Importer an FML Field Definition file and an FML Group Format (.fgf) file. To define these groups, refer to “Data Mapping Conversion.” After defining these groups in the FML Group Format file, you will use the eLink Integrator FML and View Importer to create a Meta-Type Information file (.mti) and create an output FML type tree file (.mtt).

The eLink Integrator reads the FML data to convert it to the proprietary format. The Meta-Type Information (.mti) file determines which FML fields will be retrieved and in what order.

## Compiling Type Tree Files into Mercator Map Files

Mercator map files define adapters, source and destination type trees, and transformation rules. One map must be created for each message that is to be mapped. One map is needed to convert the FML buffer to another buffer type such as COBOL for a mainframe request, and another map is needed for the conversion from the buffer type (COBOL) back to FML for the mainframe response.

## Execution Phase

Prior to a TUXEDO service request executing a Mercator map file, be sure that you have specified the mapping operation in the UBBCONFIG file and the DMCONFIG file. To request a mapping operation, the gateway will use the service name as the data mapping operation specified in the INBUFTYPE parameter of the DM\_GATEWAY\_SERVICE section of the DMCONFIG file. For more information about the UBBCONFIG and DMCONFIG file entries for running the eLink Integrator,



refer to “Configuring the Environment for eLink Integrator.” The `elinkmerc` server will then use that service name as the base name for the map definition that Mercator is to use.

The execution phase requires that certain files exist in the execution environment. After moving the files to this environment, `elinkmerc` server can process a service request for data mapping.

### Setting Up the Execution Environment

Certain files must be transferred from the development environment on Windows NT to the execution environment, which is also where APPDIR is pointing. If your execution environment is HP-UX, SUN Solaris, or AIX, use the Mercator **Map>Port** functionality to convert the Mercator map files (\*.`mmc`) to a format that is suitable for these UNIX platforms.

**Note:** If your execution environment is Windows NT, you do not need to perform the **Map>Port** step.

After performing the Port process for the Mercator map files, transfer the required files to the execution environment. When transferring these files from the development environment on Windows NT to the execution environment, such as HP-UX, AIX, or SUN Solaris, specify the appropriate mode (ASCII or binary) to use during the transfer.

◆ `fm12cob.mti` (ASCII file)

**Note:** This file is a copy of the `sample.mti` renamed to match the name of the `fm12cob.mmc` map file.

◆ `cob2fm1.mti` (ASCII file)

**Note:** This file is a copy of the `sample.mti` renamed to match the name of the `cob2fm1.mmc` map file.

◆ `sample.fml` (ASCII file)

◆ `fm12cob.mmc` (binary file)

◆ `cob2fm1.mmc` (binary file)

## Initiating a Service Request for Data Mapping

After the execution environment is set up, you can issue a TUXEDO service request that will implement the data mapping rules defined in the map files. Before issuing the TUXEDO service request, be sure the following tasks are complete.

1. Edit the TUXEDO UBBCONFIG file to define the elinkmerc server and the conversion service names.
2. Edit the INBUFTYPE and OUTBUFTYPE parameters in the DM\_GATEWAY\_SERVICE section in the DMCONFIG file to specify the correct map name.
3. Transfer the required files to the execution environment.
4. Add `sample.fml` to FIELDTBLS and FIELDTBL32. Also add the path for `sample.fml` to FLDTBLDIR and FLDTBLDIR32.
5. Ensure that TUXDIR and APPDIR are appropriately set.
6. Initiate the BEA TUXEDO and BEA eLink for Mainframe software.
7. Issue the TUXEDO service request.

**Note:** For information on the BEA TUXEDO software, refer to the BEA TUXEDO documentation. For information on the BEA eLink for Mainframe software, refer to the BEA eLink for Mainframe online documentation.

# 2 Installing BEA eLink Integrator

This chapter consists of the following topics:

- ◆ Installation Prerequisites
- ◆ Installing on the Development Platform (Windows NT)
- ◆ Installing on the Execution Platform
- ◆ Distribution Libraries and Executables
- ◆ Uninstalling eLink Integrator on Windows NT

## Installation Prerequisites

Refer to the *BEA eLink Data Integration Option Release Notes* for information on prerequisite software that must be installed and operational prior to installing the BEA eLink Integrator software. Also, for planning purposes, refer to the “Data Mapping Worksheet” section.

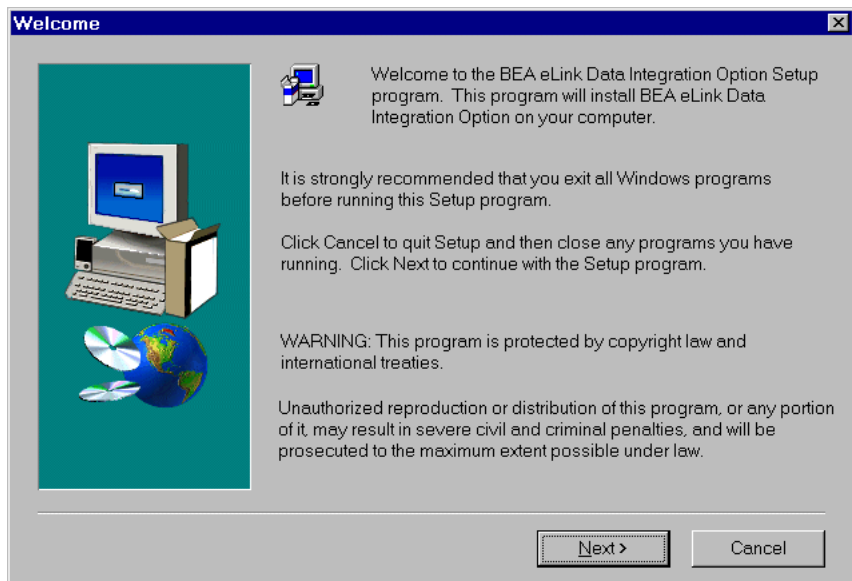
**Note:** BEA TUXEDO must be installed prior to installing the eLink Integrator component for your execution environment.

# Installing on the Development Platform (Windows NT)

Perform the following steps to install the eLink Integrator software on a Windows NT system.

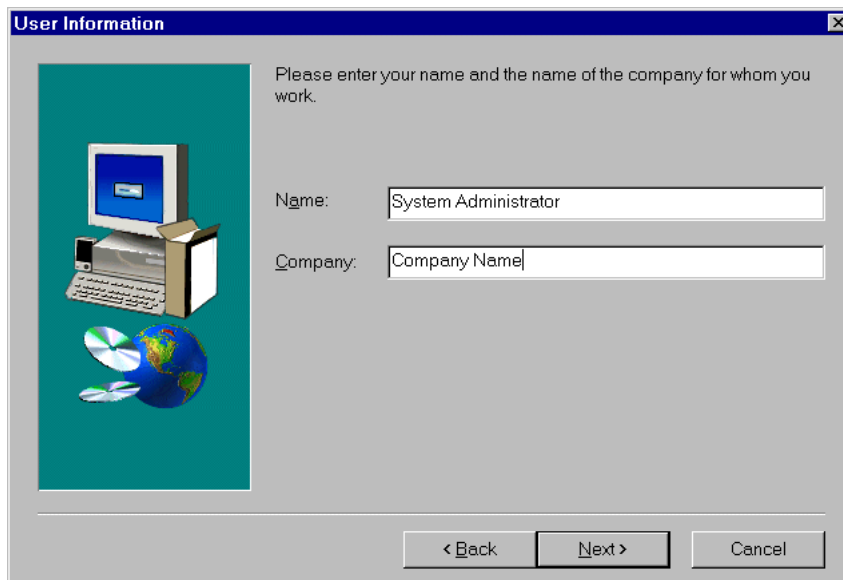
1. Insert the product CD-ROM and click on the **Run** option from the **Start menu**. The **Run** window displays. Click on the **Browse** button to select the CD-ROM drive. Select the `winnt` directory and select the `Setup.exe` program. Click **OK** to run the executable and begin the installation. The following **Welcome** screen displays. Click **Next** to continue with the installation.

Figure 2-1 Welcome



2. The **User Information** screen displays after the **Welcome** screen. Enter your name in the **Name** field. Enter the name of your company in the **Company** field. Click **Next** to continue with the installation.

**Figure 2-2 User Information**



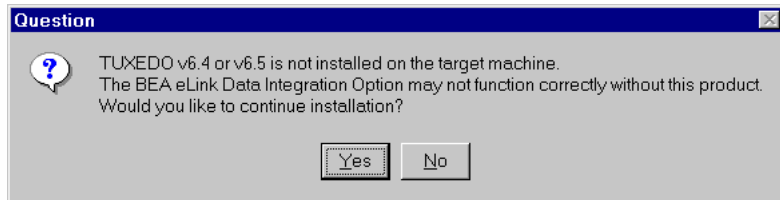
- a. If BEA TUXEDO is installed and detected on your system, the installation begins and a progress bar displays the status. The eLink Integrator components install into the TUXEDO directory. You may abort the installation process anytime prior to completion by clicking the **Cancel** button.

When the installation completes, the **Setup Complete** screen shown in Step 4 notifies you that the eLink Integrator software is installed on your system.

- b. If BEA TUXEDO is not installed on your system, the following **Question** screen displays.

**Warning:** If Windows NT is your execution environment, BEA TUXEDO should be installed first and the eLink Integrator should be installed within the same directory. If you install the eLink Integrator outside of the TUXEDO directory, you will need to copy the files into the TUXEDO directory for processing of data mapping service requests.

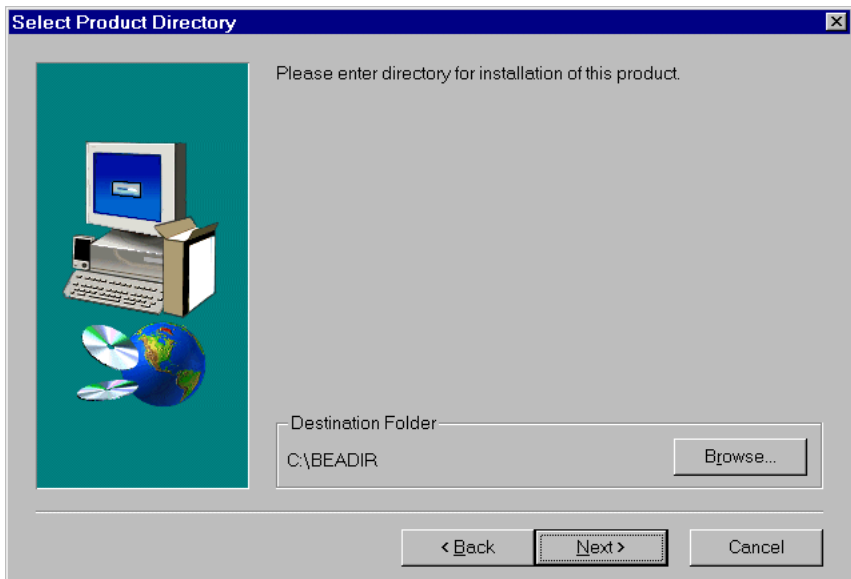
**Figure 2-3 Question**



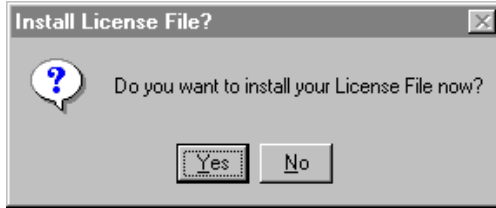
Click **Yes** to continue the installation or **No** to quit.

3. The **Select Product Directory** screen displays after the **Question** screen. Click the **Browse** button and select a new destination folder if you do not want to install to the default destination folder. Click **Next** to continue the installation.

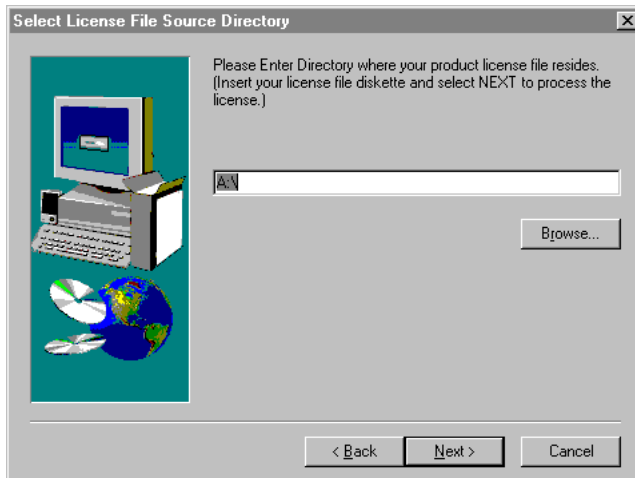
**Figure 2-4 Select Product Directory**



4. The **Install License File** screen displays after the **Select Product Directory** screen. For additional license key information, refer to the *BEA eLink Data Integration Option Release Notes*.



- a. Click **Yes** to install the license file now. The following screen displays prompting you for the location of the license file. Click **Next** to continue with the installation process.



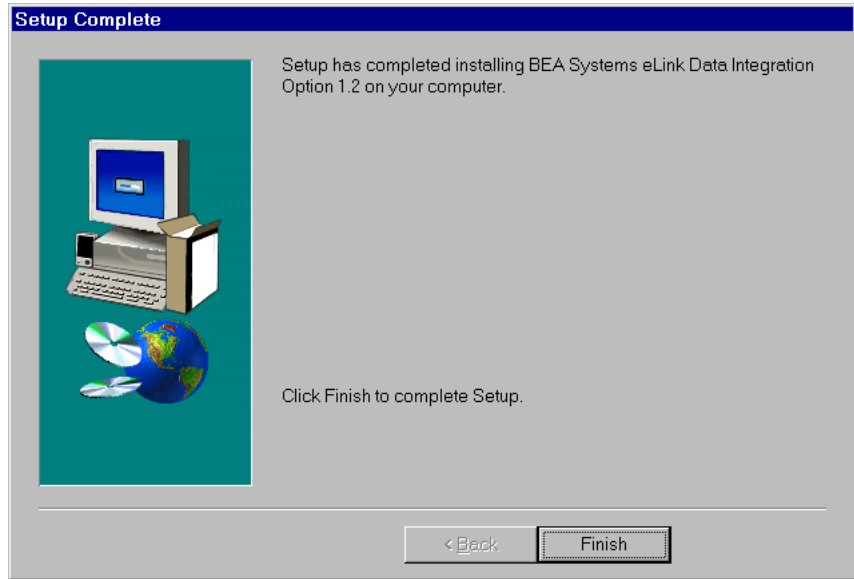
A progress bar displays the status of the installation. You may abort the installation process anytime prior to completion by clicking the **Cancel** button.

- b. Click **No** to bypass the installation of the license file now. Be sure to install the license file prior to initializing the software.

A progress bar displays the status of the installation. You may abort the installation process anytime prior to completion by clicking the **Cancel** button.

5. The **Setup Complete** screen notifies you that the eLink Integrator software is installed on your system. Click **Finish** to complete the setup process.

**Figure 2-5 Setup Complete**



# Installing on the Execution Platform

This section explains how to install the eLink Adapter software on the following execution platforms:

- ◆ Microsoft Windows NT
- ◆ HP-UX
- ◆ SUN Solaris
- ◆ IBM AIX

**Warning:** You must install the eLink Integrator execution components within the TUXEDO directory.



## Installing for Execution on Windows NT

Verify that the following libraries and program executables are installed on your system within the TUXEDO directory:

- ◆ elinkmerc.exe
- ◆ m4elink.dll
- ◆ dbutil.dll
- ◆ dbutil32.dll
- ◆ runmer32.dll
- ◆ mercma32.dll
- ◆ mercad32.dll
- ◆ mercrm32.dll

## Installing for Execution on a UNIX Platform (HP-UX, AIX, or SUN Solaris)

To install the eLink Integrator software, you run the `install.sh` script. This script installs all the necessary software components.

**Note:** The eLink Integrator should be installed into the BEA TUXEDO directory. Prior to initiating the installation script, determine the directory location of BEA TUXEDO.

Perform the following steps to install the eLink Integrator software on a supported Unix platform:

1. Log on as root to install the eLink Integrator software.

```
$ su -  
Password:
```

2. Access the CD-ROM device.

```
# ls -l /dev/cdrom
```

## 2 Installing BEA eLink Integrator

---

```
total 0
brw-rw-rw-    1 root    sys    22,    0 May 17 10:55 clb0t010
```

3. Mount the CD-ROM.

```
# mount -r -F cdfs /dev/cdrom/clb0t010 /mnt
```

4. Change the directory to your CD-ROM device.

```
# cd /mnt
```

5. List the CD-ROM contents.

```
# ls
install.sh  hp  sun5x
```

6. Execute the installation script.

```
# sh ./install.sh
```

7. The installation script runs and prompts you for responses.

### Listing 2-1 Install.sh Example

---

```
Script started on Wed Oct  6 12:16:25 1999
HPW1: /cmhome/dist/dio-2> install.sh

01) hp/hpux1020      02) hp/hpux11      03) ibm/aix43
04) sun5x/sol26     05) sun5x/sol7

Install which platform's files? [01-5, q to quit, l for list]: 1

** You have chosen to install from hp/hpux1020 **

BEA eLink Data Integration Option Release 1.2

This directory contains the BEA eLink Data Integration Option System
for
HP-UX 10.20 on 9000/800 series.

Is this correct? [y,n,q]: y

To terminate the installation at any time press the interrupt key,
typically <del>, <break>, or <ctrl+c>.
```

The following packages are available:

```
1      dio          BEA eLink Data Integration Option
```

Select the package(s) you wish to install (or 'all' to install all packages) (default: all) [?,??,q]: 1

```
BEA eLink Data Integration Option
(9000) Release 1.2
Copyright (c) 1999 BEA Systems, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
BEA eLink is a trademark of BEA Systems, Inc.
```

Directory where Data Integration Option files are to be installed  
(Enter your Tuxedo directory path) [?,q]: **/bea/work/TUX65**

Using /bea/work/TUX65 as the Data Integration Option base directory

```
Determining if sufficient space is available ...
1848 blocks are required
8634614 blocks are available to /bea/work/TUX65
```

```
Unloading /cmhome/dist/dio-2/hp/hpux1020/dio/DIOT65.Z ...
bin/elinkmerc
bin/lic.sh
bin/mti2fgf
lib/libdbutil.sl
lib/libm4elink.sl
1770 blocks
... finished
```

```
Changing file permissions...
... finished
```

If your license file is accessible, you may install it now.  
Install license file? [y/n]: **n**

Please don't forget to use lic.sh located in your product bin directory to install the license file from the enclosed floppy. Refer to your product Installation Guide for details on how to do this.

Installation of BEA eLink Data Integration Option was successful

Please don't forget to fill out and send in your registration card  
HPW1: /cmhome/dist/dio-2> exit

```
script done on Wed Oct 6 12:18:01 1999
```

---

8. Change the directory to your root directory.

```
# cd /
```

9. Unmount the CD-ROM device.

```
# unmount /mnt
```

# Distribution Libraries and Executables

The eLink Integrator CD-ROM contains the following libraries and executable programs. After installing the eLink Integrator software, verify that these libraries and programs are installed on your system.

## HP-UX

Verify that the following files are installed by the eLink Integrator software:

**Table 2-1 HP-UX Installed Files**

Directory	Files
/bin	elinkmerc mti2fgf
/lib	libdbutil.sl libm4elink.sl

## SUN Solaris

Verify that the following files are installed by the eLink Integrator software:

**Table 2-2 SUN Solaris Installed Files**

Directory	Files
/bin	elinkmerc mti2fgf
/lib	libdbutil.so libm4elink.so

## IBM AIX

Verify that the following files are installed by the eLink Integrator software:

**Table 2-3 IBM AIX Installed Files**

Directory	Files
/bin	elinkmerc mti2fgf
/lib	libdbutil.so libm4elink.so

## Windows NT 4.0

Verify that the following files are installed by the eLink Integrator software:

**Table 2-4 Windows NT Installed Files**

<b>Directory</b>	<b>Files</b>
/bin	elinkmerc.exe elinkimprt.exe mti2fgf.exe m4elink.dll dbutil.dll dbutil32.dll runmer32.dll mercma32.dll mercad32.dll mercrm32.dll mercadpt.dll

---

# Uninstalling eLink Integrator on Windows NT

Perform the following steps to uninstall the eLink Integrator software on a Windows NT system:

1. Access the **Control Panel** window from the **Start>Settings>Control Panel** menu option.
2. Double-click on the **Add/Remove Programs** option from the Control panel listings to access the **Add/Remove Programs Properties** window.
3. In the **Add/Remove Programs Properties** window, select **BEA eLink Mercator Integrator for DIO 1.2** from the program list and click the **Add/Remove** button.
4. The uninstall process for eLink Mercator Integrator for DIO begins. The **Remove Programs From Your System** screen displays. Click **OK** to complete the uninstall process.

Figure 2-6 Removing Programs From Your Computer



## 2 *Installing BEA eLink Integrator*

---



# 3 Configuring the Environment for eLink Integrator

Before running the eLink Integrator, you must configure the BEA TUXEDO server and the eLink for Mainframe gateway to run with an alternate data mapping product, such as Mercator.

Configuring the environment for eLink Integrator consists of the following basic tasks:

- ◆ Configuring the elinkmerc Server to Run with eLink Integrator
- ◆ Configuring the eLink for Mainframe Gateway to Run with eLink Data Integration Option

## Configuring the elinkmerc Server to Run with eLink Integrator

Before running the eLink Integrator, you must edit the UBBCONFIG file to recognize the services that will be used for the data mapping. For UBBCONFIG parameter syntax and definitions, refer to the *BEA TUXEDO Reference Manual*.

## Define the elinkmerc Server

Define the elinkmerc server as the server responsible for the data mapping operation. To define this server, add the elinkmerc information in the SERVERS section of the UBBCONFIG file. The following parameters are required for defining the elinkmerc server.

**Note:** You must specify `elinkmerc` as the file to execute for the data mapping process. Within this server definition, you must also specify the `-s` option in the CLOPT parameter.

### Listing 3-1 Syntax for elinkmerc Server Definition in the UBBCONFIG File

---

```
*SERVERS

elinkmerc
  SRVGRP=groupname      SRVID=n
  CLOPT="-s servicename1:XLATE_SERVICE
        -s servicename2:XLATE_SERVICE -- -WUD -TIO -AE
        -NOMTI"
```

---

For information about the SRVGRP, SRVID, and CLOPT parameter syntax and definitions, refer to the *BEA TUXEDO Reference Manual*. The definition of the `-s` option in the CLOPT parameter follows and is required for the data mapping process.

CLOPT="-s *servicename*"

specifies the services to advertise. In this case, the *servicename* is the advertised service to use for the data conversion. It is the name of the Mercator map files, such as `fm12cob.mmc` and `cob2fm1.mmc`.

**Note:** The *servicename* specified in the CLOPT parameter must also be defined in the SERVICE section of the UBBCONFIG file.

`-NOMTI` specifies the FML Default Format which groups FML according to the order of fields defined in the FML Definition File. Specify `-NOMTI` in upper case only. This option must be specified, for example, when using the eLink Mercator Integrator for DIO component with SAP applications and conversions from FML to IDOCs.

If you do *not* specify `-NOMTI`, then the default is used, which is the FML Field Definition file and the FGF file.

The options specified to the right of the double dashes (`-- -WUD -TIO - AE`) in Listing 3-1 are optional and write trace files and audit logs. These Mercator options may slow performance; however, may assist in troubleshooting. For an explanation of these options, refer to the *Execution Command Reference Guide* in the Mercator online documentation.

## Advertise the Services for Data Mapping

Advertise the service for data mapping by defining it in the SERVICES section of the UBBCONFIG file.

### Listing 3-2 Syntax for Advertising the Mapping Service

---

```
*SERVICES
```

```
servicename1  
servicename2
```

---

```
servicename1
```

specifies a 1-15 character name of the service for data mapping. An example would be `fm12cob` and `cob2fml`.

**Note:** The name of each of the services must match the map names in the CLOPT parameter in the elinkmerc server definition. For more information about defining services, refer to the BEA TUXEDO documentation about the UBBCONFIG.

## Sample UBBCONFIG File for Data Mapping

Listing 3-3 is a sample UBBCONFIG file for Windows NT. In this sample, the elinkmerc server is defined in the SERVERS section with the required CLOPT -s option specified.

**Listing 3-3 Sample UBBCONFIG File for Data Mapping**

---

```
*RESOURCES

IPCKEY          123791
DOMAINID        simpapp
MASTER          simple

*MACHINES

DALNT6
    LMID          = simple
    TUXDIR        = "\tuxedo"
    TUXCONFIG     = "\myappdir\tuxconfig"
    APPDIR        = "\myappdir"
    FIELDTBL     = "sample.fml"
    FIELDTBL32   = "sample.fml"
    FLDTBLDIR    = "\myappdir"
    FLDTBLDIR32  = "\myappdir"
    ULOGPFX      = "\myappdir\ULOG"
                # LD_LIBRARY_PATH=\mercator
                # SHLIB_PATH=\mercator
                PATH=\mercator

*GROUPS

eLINK
    LMID=simple   GRPNO=1

*SERVERS

DEFAULT:
    CLOPT="-A"

elinkmerc
    SRVGRP=eLINK      SRVID=10
    REPLYQ=N
    CLOPT="-sFML2COB:XLATE_SERVICE -sCOB2FML:XLATE_SERVICE --"
```

```
-WUD -TIO -AE"

GWIDOMAIN
    SRVGRP=eLINK    SRVID=21
    CLOPT="-A -- -eMERC"

*SERVICES

FML2COB
COB2FML

*ROUTING
```

---

## **Configuring the eLink for Mainframe Gateway to Run with eLink Data Integration Option**

To configure the products and components to work together to convert buffers between an FML and COBOL copybook format and transfer the buffers between a UNIX or Windows NT environment and a mainframe environment, you must set up the eLink for Mainframe product. The eLink for Mainframe product will invoke an alternate data mapping product to perform the actual data conversion.

Before running the eLink for Mainframe gateway, you must edit the UBBCONFIG file and the DMCONFIG file to recognize the alternate data mapping product to use for data mapping. For configuration information for the eLink for Mainframe gateway, refer to the eLink for Mainframe online product documentation.

### **3** *Configuring the Environment for eLink Integrator*

---

# 4 Data Mapping Conversion

Data mapping allows data in different formats to be passed between a UNIX or Windows NT environment and a mainframe environment. Several applications work together to accomplish this task. The following information will outline these tasks.

- ◆ Preparing for Data Mapping Conversion
- ◆ Developing Rules for Conversions Between FML Buffers and Other Applications
- ◆ Required Files for Conversion Process
- ◆ Executing a Service Request with Data Mapping

## Preparing for Data Mapping Conversion

Before starting the development of a map file that will define the rules for data conversion, the development environment must be set up. Preparing the development environment includes the following steps.

- ◆ Creating Files for Conversion
- ◆ Transferring Files to Development Environment

# Creating Files for Conversion

You must create source and destination files prior to the conversion process.

- ◆ For converting COBOL copybooks to FML buffers or FML buffers to COBOL copybooks, you must have the following files:
  - ◆ COBOL copybook that meets the Mercator conversion standards  
For information about the copybook requirements, refer to the Mercator Help file.
  - ◆ FML Field Definition file  
For information about creating an FML Field Definition file, refer to the *BEA TUXEDO Programmer's Guide* and the *BEA TUXEDO Reference Manual*.
  - ◆ FML Group file  
For information on the format and definition of the FML Group file, refer to the "Creating an FML Group Format File" section.
- ◆ For converting SAP R/3 IDOCs to FML buffers or FML buffers to SAP R/3 IDOCs, you must have the following files:
  - ◆ SAP IDOC file that meets the Mercator conversion standards  
For information about the IDOC requirements, refer to the Mercator Help file.
  - ◆ FML Field Definition file  
For information about creating an FML Field Definition file, refer to the *BEA TUXEDO Programmer's Guide* and the *BEA TUXEDO Reference Manual*.

## Creating an FML Group Format File

The FML Group Format file (\*.fgf) allows you to group information on top of the FML Field Definition file.

**Note:** The FML Group Format file is not used when you have specified the `-NOMTI` option in the `CLOPT` parameter.

Listing 4-1 is the syntax for the FML Group Format file.



**Listing 4-1 Syntax for FML Group Format File**

---

```
GROUP <groupname> [OCCURS n]
      field-type FML-field-name
END
```

---

**GROUP**

represents all of the fields or groups which will be used in the FML buffer. FML Groups are started using the GROUP keyword and ended using the END keyword. GROUPS can be nested.

**Note:** The top GROUP (or root) cannot have an OCCURS clause.

<groupname>

specifies the name for the group. You can use the name of the FML Field Definition file as the *groupname* for the top GROUP.

<field-type> <FML-field-name>

specifies fields within an FML Group Format. Fields are identified using the field type and the FML field name. Valid field types are string, long, short, char, double, float, or carry. The FML field name must match the name specified in the FML Field Definition file or an error will occur.

**Note:** Group names and field names *must* be unique.

OCCURS *n*

tells FML how many occurrences exist for this group or field. The OCCURS clause is optional. A field or group without an OCCURS clause defaults to a single occurrence.

## 4 Data Mapping Conversion

---

Listing 4-2 is a sample FML Group Format file with nested groups and occurrences.

### Listing 4-2 Sample FML Group Format File

---

```
GROUP fml_bufname
  GROUP payroll OCCURS 10
    GROUP name OCCURS 3
      string FML_FNAME
      string FML_LNAME
      string FML_MIDDLE_INIT
    END
    long    HOURS WORKED OCCURS 10
  END
END
```

---

## Using the mti2fgf Utility

The BEA eLink Integrator component installs a utility that uses the Meta-Type Information file and transposes it to the original FML Group file format. In other words, you can take `sample.mti` and, using this utility, create `sample.fgf`. This utility is helpful if you have the Meta-Type Information file, but no longer have the source FML Group Format file. After using the utility, the resulting `*.fgf` file is sent to stdout and can be used in the eLink Integrator FML and View Importer.

To use the `mti2fgf` utility, issue the following command.

### Listing 4-3 mti2fgf Utility

---

```
$ mti2fgf filename.mti
```

---

## Transferring Files to Development Environment

Using a file transfer tool, such as FTP, transfer the following files to the Windows NT development environment.

**Note:** Be sure to specify ASCII mode.

- ◆ FML files (.fml) from an existing TUXEDO configuration
- ◆ COBOL copybooks from the mainframe

## Developing Rules for Conversions Between FML Buffers and Other Applications

After you transfer the necessary files to the development environment, you can begin the process for building a Mercator map file. The map file will contain the specific mapping rules for the data being passed between a UNIX or Windows NT environment and a mainframe. The following tasks must be done in the development environment prior to the execution of a service request.

1. Set System Properties for the Mercator Open Edition Client
2. Create an FML Type Tree File Using the FML and View Importer
3. Create a Mercator Map File

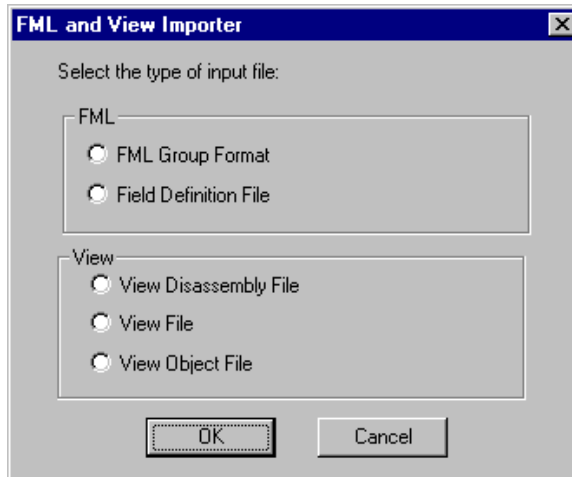
### Set System Properties for the Mercator Open Edition Client

Before developing your data maps, you must put the Mercator Open Edition Client in your system path. Select **Program>Settings>Control Panel>System Properties** to access properties page. Select the **Environment** tab and enter the path for Mercator Open Edition Client under **User Variables**. The default installation directory is `C:\mercator`.

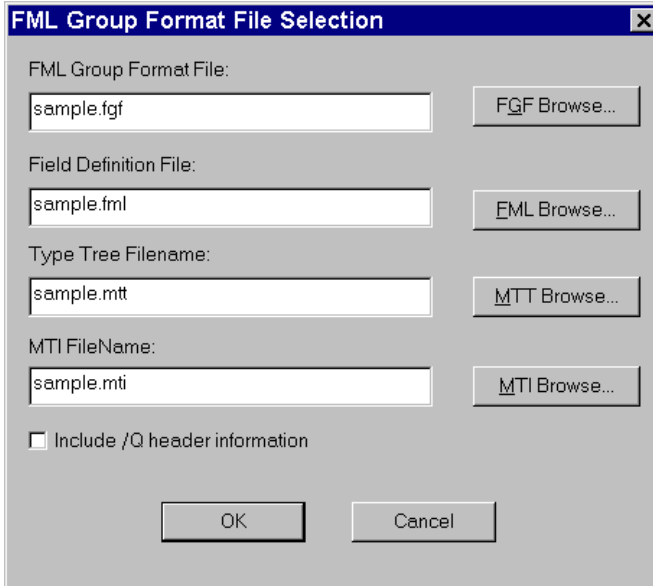
## Create an FML Type Tree File Using the FML and View Importer

To create an FML type tree file, complete the following tasks.

1. Create the necessary FML Group file (.fgf) and FML Field Definition file (.fml). For information on the FML Group Format file, refer to the “Creating an FML Group Format File” section. For information on creating FML Field Definitions file, refer to the *BEA TUXEDO Programmer’s Guide* and the *BEA TUXEDO Reference Manual*.
2. Access the eLink Integrator **FML and View Importer** dialog box by invoking `elinkimprt.exe`.
3. Select the type of input file by clicking one of the radio buttons under FML.



- a. If you select the FML Group Format radio box and click **OK**, the FML Group Format File Selection dialog box displays.



Browse to specify the following files on the FML Group Format File Selection.

FML Group Format File (.fgf) specifies the FML Group Format file to use in the data mapping conversion. This file is used in conjunction with an FML Field Definition file for the conversion.

FML Field Definition File (.fml) specifies the FML buffer to use in the data mapping conversion.

Type Tree File Name is the directory location and name of the FML type tree file to use for creating the Mercator Map file. If you reference only a file name and no directory path, the file will be created in the current directory.

MTI File Name is the directory path and file name for the metadata associated with the FML type tree file. If you reference only a file name and no directory path, the file will be created in the current directory.

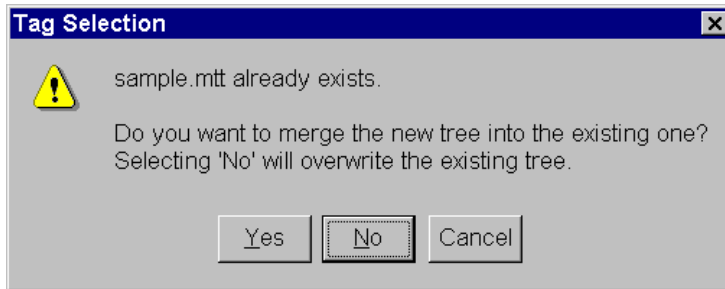
**Note:** Check the Include /Q header information box when using eLink Data Integration Option with the TUXEDO /Q feature. Do *not* check the Include /Q header information box when using the BEA eLink Mercator Integrator for DIO component with eLink for Mainframe.

## 4 Data Mapping Conversion

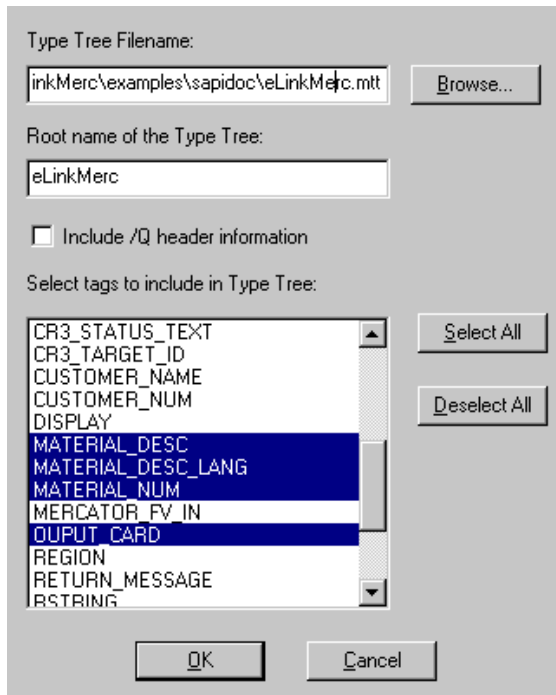
Click **OK** to validate the .fgf and .fml files and create the .mtt and .mti files.

If the following screen displays, click No to overwrite the file. No is the default.

**Note:** Do not click Yes. It will merge the existing tree with the new tree.



b. If you click the Field Definition File radio button under the FML section and click **OK**, the Type Tree Filename screen displays.



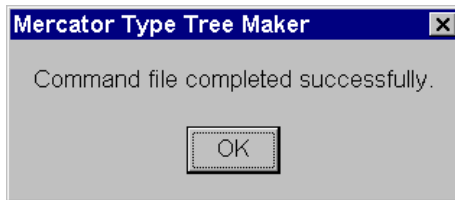
In the **Type Tree Filename**, browse to the directory where the output file resides.

In the **Root name of the Type Tree** field, enter the base value of the type tree.

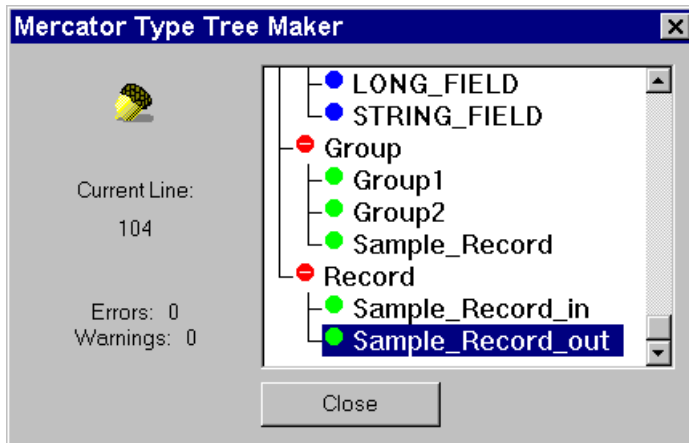
Select the **Include /Q header information** check box, if you intend to use TUXEDO queues.

In the display box, select the FML fields in the type tree.

4. If the FML type tree processes without error, the following pop up will display. Click **OK** to continue with the process.



5. The Mercator Type Tree Maker displays the newly created FML type tree file. Click **Close** to save the type tree file.



### FML Type Tree Fields

The FML type tree specifies the type definitions of the input and output files. There are four branches for the type tree file; Control, Field, Message, and SET. The SET line is not used.

- ◆ Control contains general information about the field, which is defined in the Header file.
- ◆ Field contains individual field definitions
- ◆ Message
  - ◆ Input contains one or more sets
  - ◆ Output contains header information plus one or more sets
- ◆ SETS contain multiple occurrences of zero or more specific FML fields. There may be multiple sets.

### Create a non-FML Type Tree File Using Importers

From your source file, create a type tree file (\*.mtt).

1. Using the Mercator product, load your source file such as a COBOL copybook into the appropriate Importer and create a type tree file. For specific information about how to create a type tree using the Importer, refer to the Mercator Help file.
2. Using a text editor or the Mercator Type Editor, open the type tree file (\*.mtt). If the type tree file does not contain the Control information (from the FML type tree file), add it into the header of the type tree file (\*.mtt). This can be done by dragging and dropping the Control information from the FML type tree file into the other type tree file. Listing 4-4 is a sample of the header information that must be added to the COBOL type tree file before creating the Mercator map file (.mnc).

**Note:** If you are using the TUXEDO /Q feature, refer to the “Control Information for TUXEDO /Q” section.



**Listing 4-4 Sample COBOL Type Tree Header Information**

---

```
CONTROL
  FIELD
    fieldName
    Format
    Service
    View
```

---

3. Within the non-FML type tree file, create a new category for the input and output cards. For example, create a new category named TuxMessage.
4. Select the new category that was created, for example TuxMessage. Under this category, create two new groups. One group for input messages and another group for output messages.
5. After creating the two new groups, double click on the input message group to open the component screen. Drag and drop the appropriate group under the record category into the component screen of the input message.
6. Double click on the output message group to open the component screen. Drag and drop the Header group under the Control category into the component screen of the output message. Then drag and drop the appropriate group under the Record category into the component screen of the output message.
7. Analyze the tree by selecting *Analyze* from the **Tree menu**. Select the **Results** button to view the analysis results. If there are no errors, save the file. If there are errors, refer to the *Mercator Type Editor Reference Guide* for information on resolving these errors. Be sure to save the file when finished.

## Control Information for TUXEDO /Q

Control information is used to return /Q message attributes to specify data formats of /Q message attributes in output cards. Specifically,

- ◆ In the first output card of maps used with the generic translation server, specify the data format and optional service name for the purposes of forwarding. The header information must be present in maps called by the generic translation server because it provides the only means of specifying the format of the output data.

## 4 Data Mapping Conversion

- ◆ In output cards calling TUXEDO services and /Q, use the control information to specify the data format, service name and /Q message attributes.
- ◆ In input cards calling /Q, the control information is included in the data if the -HDR option is specified. This data contains attributes of the received message.

The control information consists of the following elements.

**Table 4-1 Control Information**

	Usage		
Element	Service / /Q	When Present	Description
Format	Both	Always	Specifies the desired format of the output buffer. Possible values are "CARRAY", "STRING", "FML", "FMLV" (FML values) and "VIEW". Abbreviations of these may also be used. These are respectively: "C", "S", "F", "FV" and "V".
View	Both	Always	Specifies the name of the view if the Format type is VIEW
FieldName	Both	Always	Specifies the name of the FML field to be specified for the output data. This is used if the Format type is FMLV (FML values).
Priority	/Q	/Q only	On output, specifies the priority of the message to be sent. This overrides the -P option.
CorrID	/Q	/Q only	On output, specifies the correlation ID of the message to be sent. This overrides the -CID option.
ReplyQ	/Q	/Q only	On output, specifies the reply queue of the message to be sent. This overrides the -RQ option.
FailureQ	/Q	/Q only	On output, specifies the failure queue of the message to be sent. This overrides the -FQ option.
URCode	/Q	/Q only	On output, specifies the user return code of the message to be sent. This overrides the -UR option.

## Create a Mercator Map File

An input card and an output card must be created prior to compiling a Mercator map file (\*.mmc). Input cards contain a type tree and an adapter. Use both type tree files for the buffer types (for example FML and COBOL copybook) that are being mapped. For information on how to build a Mercator map file, refer to the *Mercator Map Editor Reference Guide* and the *Mercator Functions & Expressions Reference Guide*.

1. Access the Mercator Map Editor window.
2. Set the map file name by selecting the **Rename** option from the **Map menu**. Enter a map name that is 8 characters or less, for example `FML2COB`.
3. After naming the map file, save the file by selecting **File>Save As**. Browse to select the directory and file name, for example `FML2COB`.
4. Activate the **From** side of the **Mercator Map Editor** window. To add an input card, select **Add** from the **Card menu** to display the **Add Input Card** dialog box.

Enter a unique name for the new input card in the **Card Name** field.

In the **Type** section, click **File** to browse and select the input card type tree file (\*.mtt) for the **File** field.

In the **Name** field of the **Type** section, click the **Browse** button to display the contents of the type tree file (.mtt). From the display, select the input record.

Leave the default value, **File**, in the **Data Source** field.

In the **File** section, you can use the **Name** field as a comment box. Enter a description of the input card as your comment. Text must be entered in this field.

**Note:** The value in this field is ignored and has no effect on the data mapping process; therefore, you can use this field as a comment field.

Click **OK** to create the input card.

5. Activate the **To** side of the **Mercator Map Editor** window. To add an output card, select **Add** from the **Card menu** to display the **Add Output Card** dialog box.

Enter a unique name for the new output card in the **Card Name** field.

In the **Type** section, click **File** to browse and select the output card type tree file (\*.mtt) for the **File** field.

In the **Name** field of the **Type** section, click the **Browse** button to display the contents of the type tree file (.mtt). From the display, select the output record.

Leave the default value, **File**, in the **Data Source** field.

In the **File** section, you can use the **Name** field as a comment box. Enter a description of the output card as your comment. Text must be entered in this field.

**Note:** The value in this field is ignored and has no effect on the data mapping process; therefore, you can use this field as a comment field.

Click **OK** to create the output card.

6. On the **To** side, formulate your mapping rules. You must define the **Format** field in the Header Control Group with a mapping rule. If the output record is a COBOL copybook, enter = "C" as the rule in the **Format** field. If the output record is an FML buffer, enter = "FML" as the rule in the **Format** field.

Map the remaining fields. For information on mapping rules and defining functional maps, refer to the *Mercator Functions & Expressions Reference Guide*.

7. Compile your map file (\*.mmc) by selecting the **Build** option from the **Map menu**. If errors occur during the build process, access the building results for an explanation of the error. Resolve the errors in the mappings and execute the build process again until no errors occur.
8. Save the compiled map file by selecting **File>Save**.
9. If your execution platform is HP-UX, AIX, or SUN Solaris, you must port the maps to these platforms. To port the map, select **Map>Port** and then select the platform from a list. When the port is complete, the files will be renamed with a platform-specific extension, such as \*.hp.

## Required Files for Conversion Process

After completing the development process, the following files must exist and be transferred to the execution environment.

- ◆ FML buffer (\*.fml - ASCII file)

- ◆ Copy of type tree Meta-Type Information (.mti) file saved as the Mercator map name (FML2COB.mti and COB2FML.mti - ASCII files)

To obtain this file, copy the FML Meta-Type Information (.mti) file and rename it using the file name of both map files. For example, the FML type tree file is sample.mti and the Mercator map file is FML2COB.mmc. Copy sample.mti and rename it FML2COB.mti. The other map file is COB2FML.mmc. Then copy the sample.mti file and rename it to the COB2FML.mti.

**Note:** This step must be done for all maps using this FML type tree.

**Note:** No .mti file will exist if you are using the -NOMTI option in the CLOPT parameter when defining the server in the UBBCONFIG file.

- ◆ Type tree files (fml.mtt - Binary file)
- ◆ Mercator map files (FML2COB.mmc - Binary file)

**Note:** If you are executing on HP-UX, AIX, or SUN Solaris, the map files will have a platform-specific extension after the porting function. For example, after porting the FML2COB.mmc file to Sun Solaris, the file is named FML2COB.sun.

- ◆ Client programs that create FML (ASCII file)

## Executing a Service Request with Data Mapping

To set up the execution environment, complete the following tasks:

1. Transfer Files to APPDIR in the Execution Environment
2. Set Environment Variables
3. Start BEA TUXEDO and Other Remote Applications
4. Write a Client Program to Create FML Buffers
5. Issue a Service Request for Data Mapping

The client program requests the mainframe service advertised by the eLink for Mainframe gateway.

**Note:** Leave logging on during execution of your map files to assist in diagnostics if errors occur. Logging will produce a large amount of output. The log file (.log) can be accessed directly in the directory where the compiled map file (\*.mmc) is located.

## Transfer Files to APPDIR in the Execution Environment

Transfer the necessary files from the Windows NT development environment to the TUXEDO APPDIR directory in the execution environment. For information on required files for the execution environment, refer to the “Required Files for Conversion Process” section.

**Note:** If you ported the map files for HP-UX, AIX, or SUN Solaris, after you transfer the binary map files to the execution environment, rename the file so that the extension is .mmc. For example, rename `FML2COB.sun` to `FML2COB.mmc`.

## Set Environment Variables

Be sure to set the following environment variables.

- ◆ TUXDIR
- ◆ APPDIR
- ◆ FLDTBLDIR
- ◆ FLDTBLDIR32
- ◆ FIELDTBL
- ◆ FIELDTBL32

The following environment variables are platform-specific. Be sure to set the appropriate one for you execution environment.

- ◆ LD\_LIBRARY\_PATH (for SUN Solaris)

- ◆ SHLIB\_PATH (for HP-UX)
- ◆ LIBPATH (for AIX)
- ◆ PATH (for Windows NT)

## Start BEA TUXEDO and Other Remote Applications

After transferring the necessary files to the execution environment, be sure that the BEA TUXEDO and other products such as BEA eLink for Mainframe are running. For information on how to start these products, refer to the related BEA TUXEDO documentation and the BEA eLink for Mainframe online documentation CD.

## Write a Client Program to Create FML Buffers

The client program should be a standard TUXEDO program that uses `FML32`. All data items and occurrences specified in the FML Group Format file *must* be added to the FML buffer or an error will occur. If the FML Group Format file specifies an array of 10, then 10 occurrences must be added using the `Fadd32` function or another appropriate function.

## Issue a Service Request for Data Mapping

After starting these products, you can issue a service request for data mapping. For information on setting up TUXEDO service requests, refer to the BEA TUXEDO documentation. The service name to call is the mainframe service being advertised.

## **4** *Data Mapping Conversion*

---



---

# A Transaction Support

The TUXEDO adapter will support TUXEDO transactions. This will be integrated into the existing Mercator rollback functionality, which does not require an option to be passed to the adapter explicitly, as it is controlled via the interface structure used between Mercator and all adapters. This transactional functionality allows the following:

- ◆ A single put or get operation being a transaction
- ◆ 2 or more put or get operations being covered by 1 transaction
- ◆ A map that produces 3 operations, 2 in 1 transaction and the 3<sup>rd</sup> not in a transaction
- ◆ All TUXEDO operations define a map to be implicitly in a transaction

For any TUXEDO adapter card which has the rollback option set, it will call `tpgetlev` to see if it is already part of a TUXEDO transaction. If a transaction is active it could have either been started by a client or a previous call to the TUXEDO adapter. Only if a transaction is not active will the TUXEDO adapter call `tpbegin()` to start a transaction. At this time it will also set the 'cleanup' flag to instruct Mercator to call it back on map completion to either abort the transaction (`tpabort()`) or commit it (`tpcommit()`).

If the rollback option is not set on a particular card, and the adapter started the transaction earlier then the TUXEDO service will be called with the 'TPNOTRAN' option, to indicate that the service invocation is non-transactional. However if the process is part of a transaction but it was not started by the adapter then the TPNOTRAN flag is not set on the call, i.e. allow implicit transaction propagation.

# Transactional Message Broker

This section outlines some cases of using the powerful combination of TUXEDO global transactions and the Mercator engine. The Mercator product already supports a Rollback flag setting on an individual card basis in the map. If this flag is set it basically means that the call should support a rollback and a number of these can be grouped into a transaction.

**Note:** Mercator does not support any form of a two phase commit.

<b>TuxRun Merc() in Tx Mode</b>	<b>Output Card 1 Type</b>	<b>Output Card 1 Rollback</b>	<b>Output Card 2 Type</b>	<b>Output Card 2 Rollback</b>	<b>Output Card 3 Type</b>	<b>Output Card 3 Rollback</b>	<b>Output Card 4 Type</b>	<b>Output Card 4 Rollback</b>
Y	Tux Svc	N	Tux Svc	N	Tux Svc	N	MQSeries	N
N	Tux Svc	Y	Tux Svc	Y	Tux Svc	N	MQSeries	N
Y	Tux Svc	Y	Tux Svc	Y	Tux Svc	N	MQSeries	Y

# B Error and Informational Messages

This document contains the following descriptions of error, informational, and warning messages that can be encountered while using the BEA eLink Integrator component.

201	<b>ERROR: Cannot open '&lt;filename&gt;' (errno=&lt;os-errno-value&gt; - '&lt;os-errno-text&gt;')</b>
	<b>Description:</b> The specified Meta-Type Information file cannot be opened.
	<b>Action:</b> A Meta-Type Information file must exist and be located in the application directory identified by the APPDIR environment variable defined in the ubbconfig file (or the ENV file referenced in the ubbconfig file). Verify its presence and that 'USER' has read permissions.
	<b>See Also:</b> None.
202	<b>ERROR: Cannot restore type description from '&lt;buffer-type&gt;' (rc=&lt;response-code&gt;')</b>
	<b>Description:</b> The buffer's data type cannot be determined because its definition is not available. Refer to the supplemental code for details.
	<b>Action:</b> Examine the supplemental code and correct the problem.

## B Error and Informational Messages

---

	<b>See Also:</b>	None.
<b>203</b>	<b>ERROR: Cannot translate input FML data ‘(rc=&lt;response-code&gt;)’</b>	
	<b>Description:</b>	FML data could not be translated into an internal format and made available to the Mercator mapping engine. Refer to the supplemental code for details.
	<b>Action:</b>	Examine the supplemental code and correct the problem.
	<b>See Also:</b>	None.
<b>204</b>	<b>ERROR: Cannot open ‘&lt;filename&gt;’ (APPDIR not set)</b>	
	<b>Description:</b>	The specified Meta-Type Information file <filename> cannot be opened because the APPDIR environment variable is either not available or incorrectly initialized.
	<b>Action:</b>	A Meta-Type Information file must exist and be located in the application directory identified by the APPDIR environment variable defined in the ubbconfig file (or the ENV file referenced in the ubbconfig file). Verify the definition of the APPDIR environment variable.
	<b>See Also:</b>	None.
<b>205</b>	<b>ERROR: Illegal buffer type ‘&lt;buffer-type&gt;’ (tperno=&lt;tuxedo-errno-value&gt; - ‘&lt;tuxedo-errno-text&gt;’)</b>	
	<b>Description:</b>	The buffer received by the eLink Mercator Integrator for DIO service handler is not a TUXEDO buffer type.
	<b>Action:</b>	Examine the TUXEDO error response code (tperno) along with its brief explanation text to determine the nature of the programming error. More detailed information may be available in the TUXEDO documentation.
	<b>See Also:</b>	None.
<b>206</b>	<b>ERROR: INTERNAL ERROR: Invalid parameters</b>	
	<b>Description:</b>	The eLink Mercator Integrator for DIO has reached an unexpected state and may possibly become unstable.
	<b>Action:</b>	Contact Technical Support.
	<b>See Also:</b>	None.

<b>207</b>	<b>ERROR: Failure to map input buffer type ‘&lt;buffer-type&gt;’ (rc=&lt;response-code&gt;)</b>
	<p><b>Description:</b> The eLink Mercator Integrator for DIO is unable to convert the input buffer specified as &lt;buffer-type&gt; into an internal byte stream. Refer to the supplemental code for details.</p>
	<p><b>Action:</b> Examine the supplemental code and correct the problem.</p>
	<p><b>See Also:</b> None</p>
<b>208</b>	<b>ERROR: Failure to map named (&lt;field-name&gt;) input buffer type ‘&lt;buffer-type&gt;’ (rc=&lt;response-code&gt;)</b>
	<p><b>Description:</b> The eLink Mercator Integrator for DIO is unable to convert the input buffer whose &lt;buffer-type&gt; is specified by the named FML field &lt;field-name&gt; into an internal byte stream. Refer to the supplemental code for details.</p>
	<p><b>Action:</b> Examine the supplemental code and correct the problem. &lt;buffer-type&gt; is displayed “&lt;invalid-buffer-type&gt;” if &lt;field-name&gt; is not found or out of range.</p>
	<p><b>See Also:</b> None.</p>
<b>209</b>	<b>ERROR: Data mapping failure (rc=&lt;merc-code&gt; - ‘&lt;merc-message-text&gt;’)</b>
	<p><b>Description:</b> This is a general data mapping response diagnostic. &lt;merc-code&gt; and &lt;merc-message-text&gt; are detailed in the Mercator documentation.</p>
	<p><b>Action:</b> Examine &lt;merc-code&gt; and &lt;merc-message-text&gt; for the cause of the failure and correct the problem.</p>
	<p><b>See Also:</b> None.</p>
<b>210</b>	<b>ERROR: Invalid or missing control information (rc=&lt;response-code&gt;)</b>
	<p><b>Description:</b> The control information in output record data is missing or contains one or more invalid values. It cannot be processed by the eLink Mercator Integrator for DIO. Refer to the supplemental code for the cause.</p>

## B Error and Informational Messages

---

	<b>Action:</b>	The common cause for this failure is the value assigned to the FORMAT field of the Control information. Refer to this diagnostic code for details for correct assignment values.
	<b>See Also:</b>	None.
<b>211</b>	<b>ERROR: Output buffer VIEW name missing</b>	
	<b>Description:</b>	The FORMAT field in the output record's Control information specifies a TUXEDO VIEW to be used as the output buffer type, but the specific VIEW name has not been provided.
	<b>Action:</b>	Specify a VIEW name in the VIEW control information of the output record type tree.
	<b>See Also:</b>	None.
<b>212</b>	<b>ERROR: Failure to map output buffer, type not specified</b>	
	<b>Description:</b>	The eLink Mercator Integrator for DIO could not process the result received from the Mercator mapping engine because no output buffer is specified.
	<b>Action:</b>	Specify an output buffer, either by using <code>tpalloc(3)</code> and sending this together with the request in the <code>tpcall(3)</code> , or by defining it in the control information header in the output record type tree.
	<b>See Also:</b>	None.
<b>213</b>	<b>ERROR: Failure to map output byte stream to FML (rc=&lt;response-code&gt;)</b>	
	<b>Description:</b>	The eLink Mercator Integrator for DIO is unable to convert the output buffer from the internal byte stream received from the Mercator mapping engine.
	<b>Action:</b>	Examine the supplemental code and correct the problem.
	<b>See Also:</b>	None.

<b>214</b>	<b>ERROR: Failure to map output byte stream to ‘&lt;buffer-type&gt;’ (rc=&lt;response-code&gt;)</b>
	<b>Description:</b> The eLink Mercator Integrator for DIO is unable to convert the output buffer from the internal byte stream received from the Mercator mapping engine to the specified <buffer-type>.
	<b>Action:</b> Examine the supplemental code and correct the problem.
	<b>See Also:</b> None.
<b>215</b>	<b>ERROR: Illegal buffer type ‘&lt;buffer-type&gt;’</b>
	<b>Description:</b> The output buffer type is not one of the supported types and cannot be used to complete the mapping operation.
	<b>Action:</b> Refer to diagnostic 225 for a list of supported buffer types.
	<b>See Also:</b> None.
<b>216</b>	<b>ERROR: Memory allocation error</b>
	<b>Description:</b> A general memory allocation failure has occurred.
	<b>Action:</b> Call Technical Support.
	<b>See Also:</b> None.
<b>217</b>	<b>WARN: Cannot load shared library ‘&lt;filename&gt;’</b>
	<b>Description:</b> The eLink Mercator Integrator for DIO cannot load its shared library. The eLink Mercator Integrator for DIO uses a method in this library to obtain map execution status from the Mercator engine.
	<b>Action:</b> If this condition is not corrected, the eLink Mercator Integrator for DIO reports successful engine status regardless of the actual operation status. This library is expected to be installed in the TUXEDO lib directory (UNIX installations) or bin directory (Windows NT installations).
	<b>See Also:</b> None.

<b>218</b>	<b>WARN: Failed to find function in TUXEDO adapter</b>
<b>Description:</b>	The eLink Mercator Integrator for DIO cannot obtain its status reporting method from its shared library. The eLink Mercator Integrator for DIO uses this method to obtain map execution status from the Mercator engine.
<b>Action:</b>	Contact Technical Support.
<b>See Also:</b>	None.
<b>219</b>	<b>INFO: Initializing Mercator API</b>
<b>Description:</b>	This diagnostic is reported on UNIX platforms only.
<b>Action:</b>	None.
<b>See Also:</b>	None.
<b>220</b>	<b>INFO: Translation server started</b>
<b>Description:</b>	This diagnostic reports the eLinkmerc TUXEDO server is ready to service requests.
<b>Action:</b>	None.
<b>See Also:</b>	None.
<b>221</b>	<b>INFO: Exiting Mercator API</b>
<b>Description:</b>	This diagnostic reports the eLinkmerc TUXEDO server is no longer ready to service requests. It is reported on UNIX platforms only.
<b>Action:</b>	None.
<b>See Also:</b>	None.
<b>222</b>	<b>ERROR: Missing 'filename'</b>
<b>Description:</b>	The Meta-Type Information disassembler requires the name of an existing Meta-Type Information file.
<b>Action:</b>	Repeat the command using the name of an existing Meta-Type Information file.
<b>See Also:</b>	None.



<b>223</b>	<b>ERROR: Cannot restore type from '&lt;filename&gt;' (rc=&lt;response-code&gt;)</b>
	<p><b>Description:</b> The Meta-Type Information disassembler is unable to disassemble &lt;filename&gt;. &lt;response-code&gt; indicates the nature of the failure.</p>
	<p><b>Action:</b> Refer to the diagnostic &lt;response-code&gt; for a detailed reason.</p>
	<p><b>See Also:</b> None.</p>
<b>224</b>	<b>INTERNAL ERROR: TUXEDO API failure (tperrno=&lt;tuxedo-code&gt; - '&lt;tuxedo-message&gt;')</b>
	<p><b>Description:</b> The eLink Mercator Integrator for DIO encountered a failure response from TUXEDO when performing the requested operation. The &lt;tuxedo-code&gt; is the numeric value of the diagnostic code. &lt;tuxedo-message&gt; is a brief description of the diagnostic response.</p>
	<p><b>Action:</b> None.</p>
	<p><b>See Also:</b> None.</p>
<b>225</b>	<b>ERROR: Illegal FORMAT value '&lt;value&gt;' specified in control header</b>
	<p><b>Description:</b> The value of the FORMAT field of the control information in the output buffer is missing or out of range.</p>
	<p><b>Action:</b> The possible values are "CARRAY" (or "C"), "STRING" (or "S"), "VIEW" (or "V"), "FML" (or "F") and "FMLV" (or "FV"). Correctly assign this field in the control information of the output buffer.</p>
	<p><b>See Also:</b> None.</p>
<b>226</b>	<b>WARN: Cannot locate '&lt;filename&gt;' (TUXDIR not set)</b>
	<p><b>Description:</b> The Adapter cannot load its shared library. The eLink Mercator Integrator for DIO uses a method in this library to obtain map execution status from the Mercator engine.</p>

## B Error and Informational Messages

---

	<b>Action:</b>	If this condition is not corrected, the eLink Mercator Integrator for DIO reports successful engine status regardless of the actual operation status. This library is expected to be installed in the TUXEDO lib directory (UNIX Installation) or bin directory (Windows NT installations).
	<b>See Also:</b>	None.
<b>401</b>	<b>ERROR: Fieldname &lt;name&gt; not found in FML file</b>	
	<b>Description:</b>	An FML Field name referenced in this FML Group Format file is not defined in the FML Field Definition File.
	<b>Action:</b>	Provide field names in the FML Field Definition file.
	<b>See Also:</b>	None.
<b>402</b>	<b>ERROR: FieldName &lt;name&gt; has a different type from the FML file</b>	
	<b>Description:</b>	The data type specified for the FML field is not the same as the data type in the FML Field Definition file.
	<b>Action:</b>	Modify the information so that the field name and data type in the FML Field Definition file match.
	<b>See Also:</b>	None.
<b>403</b>	<b>ERROR: Fieldname &lt;name&gt; occurs multiple times in the FGF file</b>	
	<b>Description:</b>	This field name occurs multiple times in the FGF file, it should occur only once.
	<b>Action:</b>	Edit the FGF file so that the field name occurs only once.
	<b>See Also:</b>	None.
<b>404</b>	<b>ERROR: Cannot create MTI file (rc=&lt;response-code&gt;)</b>	
	<b>Description:</b>	This error occurs when the Meta-Type Information file cannot be saved.
	<b>Action:</b>	Check directory write permissions or disk space. If the error persists, contact Technical Support.
	<b>See Also:</b>	None.

<b>410</b>	<b>ERROR: Outermost GROUP in the FGF file must have exactly one occurrence</b>
	<b>Description:</b> The outermost GROUP in the FGF file cannot have an OCCURS clause with any number other than 1.
	<b>Action:</b> Remove the OCCURS clause in the outermost group or set OCCURS=1.
	<b>See Also:</b> None.
<b>411</b>	<b>ERROR: Found duplicate Group name &lt;name&gt; in FGF file</b>
	<b>Description:</b> Group Names within FGF file must be unique.
	<b>Action:</b> Modify the group Names so that they are unique.
	<b>See Also:</b> None.
<b>420</b>	<b>ERROR: Syntax Error at line &lt;line number&gt;</b>
	<b>Description:</b> There is a syntax error in the FML group format file.
	<b>Action:</b> Correct the syntax and recompile. Some guidelines for correcting the syntax follow: <ul style="list-style-type: none"> <li>◆ Verify that each GROUP statement has a corresponding END statement.</li> <li>◆ Verify that each GROUP statement has a name.</li> <li>◆ Verify that each GROUP has at least one statement.</li> <li>◆ Verify that your FML type is a string, char, array, short, long, float, or double.</li> <li>◆ Verify that each OCCURS clause is supplied with a number.</li> </ul>
	<b>See Also:</b> Section “Creating an FML Group Format File” on page 4-2
<b>3001</b>	<b>INTERNAL ERROR: Program argument value null</b>
	<b>Description:</b> An eLink Mercator Integrator for DIO contract has been violated; the eLink Mercator Integrator for DIO may be in an unreliable state.
	<b>Action:</b> Contact Technical Support.
	<b>See Also:</b> None.

## B Error and Informational Messages

---

<b>3002</b>	<b>INTERNAL ERROR: Type descriptor unreliable</b>
<b>Description:</b>	Type descriptor reference unreliable.
<b>Action:</b>	Contact Technical Support.
<b>See Also:</b>	None.
<b>3003</b>	<b>INTERNAL ERROR: Filename argument out of range</b>
<b>Description:</b>	An eLink Mercator Integrator for DIO contract has been violated; the eLink Mercator Integrator for DIO may be in an unreliable state.
<b>Action:</b>	Contact Technical Support.
<b>See Also:</b>	None.
<b>3004</b>	<b>ERROR: Cannot open file for read</b>
<b>Description:</b>	The eLink Mercator Integrator for DIO is unable to open a Meta-Type Information file for read access.
<b>Action:</b>	Verify this filename prefix matches the prefix of the corresponding Mercator map file. For example, if the name of the Mercator map file is <code>T0COBOL.mmc</code> , then the filename for the corresponding Meta-Type Information file must be <code>T0COBOL.mti</code> . This file must be located in the application's directory (APPDIR environment variable) and have appropriate read permissions.
<b>See Also:</b>	None.
<b>3005</b>	<b>ERROR: Cannot open file for writing</b>
<b>Description:</b>	The eLink Mercator Integrator for DIO is unable to open a Meta-Type Information file for write access.
<b>Action:</b>	Verify the file access permissions of the application directory (APPDIR environment variable). Appropriate write permissions must be enabled.
<b>See Also:</b>	None.
<b>3006</b>	<b>INTERNAL ERROR: Null persistence file pointer</b>

	<b>Description:</b> An eLink Mercator Integrator for DIO contract has been violated; the eLink Mercator Integrator for DIO may be in an unreliable state.
	<b>Action:</b> Contact Technical Support.
	<b>See Also:</b> None.
<b>3007</b>	<b>INTERNAL ERROR: Null member name argument</b>
	<b>Description:</b> An eLink Mercator Integrator for DIO contract has been violated; the eLink Mercator Integrator for DIO may be in an unreliable state.
	<b>Action:</b> Contact Technical Support.
	<b>See Also:</b> None.
<b>3008</b>	<b>ERROR: Field '&lt;field-name&gt;' not found</b>
	<b>Description:</b> The FML buffer does not contain the field specified by <field-name>.
	<b>Action:</b> Verify the buffer data, Meta-Type Information and Mercator map files are correct for the requested operation. This will likely occur if either one of these files or the buffer are incorrect.
	<b>See Also:</b> None.
<b>3009</b>	<b>ERROR: Field '&lt;field-name&gt;' missing matching value</b>
	<b>Description:</b> The FML buffer does not contain the value for the field specified by <field-name>.
	<b>Action:</b> Correct the buffer initialization step and proceed.
	<b>See Also:</b> None.
<b>3010</b>	<b>ERROR: Field '&lt;field-name&gt;' value not found</b>
	<b>Description:</b> The FML buffer does not contain the value for the field specified by <field-name>.
	<b>Action:</b> Correct the buffer initialization step and proceed.
	<b>See Also:</b> None.

<b>3011</b>	<b>INTERNAL ERROR: Unsafe copy</b>
<b>Description:</b>	An eLink Mercator Integrator for DIO contract has been violated; the eLink Mercator Integrator for DIO may be in an unreliable state.
<b>Action:</b>	Contact Technical Support.
<b>See Also:</b>	None.
<b>3012</b>	<b>ERROR: Failure reading Meta-Type Information</b>
<b>Description:</b>	The Meta-Type Information file could not be read into process memory.
<b>Action:</b>	The Meta-Type Information file may have become corrupted, or possibly it is not a Meta-Type Information file. Verify and regenerate this file.
<b>See Also:</b>	None.
<b>3013</b>	<b>ERROR: Cannot grow persistence file</b>
<b>Description:</b>	The Meta-Type Information cannot be loaded into memory due to resource exhaustion.
<b>Action:</b>	Contact Technical Support.
<b>See Also:</b>	None.
<b>3014</b>	<b>ERROR: Cannot allocate heap space for object persistence</b>
<b>Description:</b>	This is a general memory allocation failure due to resource exhaustion.
<b>Action:</b>	Contact Technical Support.
<b>See Also:</b>	None.
<b>3015</b>	<b>ERROR: Not a Meta-Type Information file</b>
<b>Description:</b>	The eLink Mercator Integrator for DIO does not recognize this file as a Meta-Type Information file.
<b>Action:</b>	The Meta-Type Information file may have become corrupted, or possibly it is not a Meta-Type Information file. Verify or regenerate this file.

---

	<b>See Also:</b> None.
<b>3016</b>	<b>ERROR: Meta-Type Information file contents unreliable</b>
	<b>Description:</b> The Meta-Type Information file contents could be read but not processed.
	<b>Action:</b> The Meta-Type Information file may have become corrupted, or possibly it is not a Meta-Type Information file at all. Verify or regenerate this file.
	<b>See Also:</b> None.
<b>3017</b>	<b>INTERNAL ERROR: Program argument out of range</b>
	<b>Description:</b> An eLink Mercator Integrator for DIO contract has been violated; the eLink Mercator Integrator for DIO may be in an unreliable state.
	<b>Action:</b> Contact Technical Support.
	<b>See Also:</b> None.

---

## **B** *Error and Informational Messages*

---



# C Data Mapping Worksheet

This worksheet will assist you in setting up your data mapping environment properly.

## Step 1: Install Products

The following products must be installed and operational during the execution of a data mapping service request. Install these products in the following order.

1. BEA TUXEDO 6.4 or eLink Platform 1.1

Directory location: \_\_\_\_\_

2. TSI Mercator Open Edition Client 1.4.2

**Note:** Use the SERIAL number from the eLink DIO license key when prompted for a Serial number during the Mercator Open Edition Client installation.

Directory location: \_\_\_\_\_

3. BEA eLink Data Integration Option 1.2

Directory location: \_\_\_\_\_

4. (Optional) To connect to a mainframe environment, you can use one of the following BEA products:

- ◆ BEA eLink for Mainframe TCP 3.0 with rolling patch 1
- ◆ BEA eLink for Mainframe TCP 3.1

- ◆ BEA eLink for Mainframe SNA 3.2

Directory location: \_\_\_\_\_

5. (Optional) To connect to SAP R/3 applications:

- ◆ SAP R/3 Version 3.1H

Directory location: \_\_\_\_\_

## Step 2: Set TUXEDO Environment Variables

Set the following environment variables:

- ◆ TUXDIR
- ◆ APPDIR
- ◆ FLDTBLDIR
- ◆ FLDTBLDIR32
- ◆ FIELDTBL
- ◆ FIELDTBL32

The following environment variables are platform-specific. Be sure to set the appropriate one for your execution environment.

- ◆ LD\_LIBRARY\_PATH (for SUN Solaris)
- ◆ SHLIB\_PATH (for HP-UX)
- ◆ LIBPATH (for IBM AIX)
- ◆ PATH (for Windows NT)

## Step 3: Put Mercator in the System Path

Before developing your data maps on Windows NT, you must put the Mercator Open Edition Client in your system path. Select **Program>Settings>Control Panel>System Properties** to access properties page. Select the **Environment** tab and enter the path for Mercator Open Edition Client under **User Variables**. The default installation directory is `C:\mercator`.

## Step 4: Transfer Source Files to Windows NT

1. Transfer the FML Field Definition files and non-FML source files to the development environment on the Windows NT system. Use the following table to track the location and names of these files.
2. Create the FML Group Format file to use with the FML Field Definition file in the development environment on the Windows NT system.

**Note:** This file is optional and not required when specifying the `-NOMTI` option in the `CLOPT` parameter of the server configuration.

Use the following table to track the location and names of these files.

FML Field Definition Files	FML Group Format Files	non-FML Source Files

## Step 5: UBBCONFIG File Definitions

The following configurations must be modified for data mapping service requests to process correctly.

1. Define a GROUP for data mapping. Use the following table to plan for the group information that needs to be specified.

GROUP Parameters	Arguments
<i>groupname</i>	<i>groupname</i> =
LMID= <i>logical-machine-ID</i>	<i>logical-machine-ID</i> =
GRPNO= <i>n</i>	<i>n</i> =

2. Define the elinkmerc server in the TUXEDO UBBCONFIG file.

Define the elinkmerc server in the SERVER section of the UBBCONFIG file. The elinkmerc server is responsible for the data mapping operation. Use the following table to plan for the server information that needs to be specified.

SERVER Parameters	Arguments
<i>server identifier</i>	elinkmerc
SRVGRP= <i>groupname</i>	<i>groupname</i> =
SRVID= <i>ID</i>	<i>ID</i> =
CLOPT=" -s <i>servicename1</i> <i>servicename2</i> -NOMTI "	<i>servicename1</i> = <i>servicename2</i> =

3. Define the gateway server in the TUXEDO UBBCONFIG file.

Define the gateway server in the SERVER section of the UBBCONFIG file. Use the following table to plan for the server information that needs to be specified.

<b>SERVER Parameters</b>	<b>Arguments</b>
<i>server identifier</i>	GWIDOMAIN   GWSNAX
SRVGRP= <i>groupname</i>	<i>groupname</i> =
SRVID= <i>ID</i>	<i>ID</i> =
CLOPT="-A -eMERC"	

4. Define the advertised data mapping services in the TUXEDO UBBCONFIG file.

Define the advertised data mapping services in the SERVICES section of the UBBCONFIG file. Use the following table to plan for the server information that needs to be specified.

<b>SERVICES Parameters</b>	<b>Arguments</b>
<i>servicename1</i>	<i>servicename1</i> =
<i>servicename2</i>	<i>servicename2</i> =

## Step 6: DMCONFIG File Definitions (Optional)

The following configurations are optional and must be modified for data mapping service requests to process correctly when using the BEA eLink for Mainframe product.

1. Define a DM\_LOCAL\_SERVICES for data mapping. Use the following table to plan for the service information that needs to be specified.

<b>DM_LOCAL_SERVICES Parameters</b>	<b>Arguments</b>
<i>service</i>	<i>service =</i>
INBUFTYPE=FML: <i>servicename</i>	<i>servicename =</i>
OUTBUFTYPE=FML: <i>servicename</i>	<i>servicename =</i>

2. Define a DM\_REMOTE\_SERVICES for data mapping. Use the following table to plan for the service information that needs to be specified.

<b>DM_REMOTE_SERVICES Parameters</b>	<b>Arguments</b>
<i>service</i>	<i>service =</i>
INBUFTYPE=FML: <i>servicename</i>	<i>servicename =</i>
OUTBUFTYPE=FML: <i>servicename</i>	<i>servicename =</i>

## Step 7: Create an FML Type Tree File

For information about how to create an FML type tree file, refer to “Create an FML Type Tree File Using the FML and View Importer” on page 4-6.

## Step 8: Create a non-FML Type Tree File

For information about how to create a non-FML type tree file, refer to “Create a non-FML Type Tree File Using Importers” on page 4-10.

## Step 9: Create a Mercator Map File

For information about how to create a Mercator map file, refer to “Create a Mercator Map File” on page 4-13.

## Step 10: Port Map Files for Execution on UNIX Platforms

If your execution platform is HP-UX, AIX, or SUN Solaris, you must port the Mercator map files to these platforms. To port the map, access the Mercator Map Editor and select **Map>Port** and then select the platform from a list. When the port is complete, the files will be renamed with a platform-specific extension, such as \*.hp.

**Note:** After transferring these ported files to the execution environment you must rename these files with the \*.mmc extension again. The execution environment will not recognize the ported files with the extensions such as \*.hp.

## Step 11: Renaming Meta-Type Information File (Optional)

**Note:** This step is optional and not required when specifying the -NOMTI option in the CLOPT parameter of the server configuration.

You must have a copy of the Meta-Type Information (sample.mti) file saved as the Mercator map name (FML2COB.mti and COB2FML.mti).

To obtain this file, copy the FML Meta-Type Information (.mti) file and rename it using the file name of both map files. For example, the FML type tree file is `sample.mti` and the Mercator map file is `FML2COB.mmc`. Copy `sample.mti` and rename it `FML2COB.mti`. The other map file is `COB2FML.mmc`. Then copy the `sample.mti` file and rename it to the `COB2FML.mti`.

## Step 12: Transfer Files

For information about transferring files, refer to “Transfer Files to APPDIR in the Execution Environment” on page 4-16.

## Step 13: Start BEA TUXEDO and Remote Applications

For information about how to start BEA TUXEDO and BEA eLink for Mainframe products, refer to the specific product online documentation.

## Step 14: Issue a Service Request for Data Mapping

For information about setting up service requests, refer to the BEA TUXEDO online documentation.