BEA eLink Adapter for MQSeries

User Guide

BEA eLink Adapter for MQSeries Version 1.1
Document Edition 1.0
January 2000

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA eLink Adapter for MQSeries User Guide

Document Edition	Part Number	Date	Software Version
1.1		January 2000	BEA eLink Adapter for MQSeries 1.1

Contents

About This Document

This document describes the BEA eLink Adapter for MQSeries component and gives instructions for transferring data between MQSeries and the eLink Platform. This guide explains how to install and configure the eLink Adapter for MQSeries, and how to initiate data transfer requests.

The BEA eLink Adapter for MQSeries User Guide is organized as follows:

- *Understanding EAI and the Role of eLink Adapters* introduces the eLink Adapter component and explains how eLink Adapter for MQSeries fits into the BEA TUXEDO environment.
- *Installing BEA eLink Adapter for MQSeries* explains how to install the eLink Adapter component.
- *Configuring eLink Adapter for MQSeries* provides information for configuring BEA TUXEDO servers and the MQSeries Queue Manager, and for mapping the MQI and ATMI API parameters.
- *Running eLink Adapter for MQSeries* provides information about booting the BEA TUXEDO server and initiating information transfer requests between a TUXEDO environment and MQSeries.
- *Error Messages* describes error and informational messages as well as actions to resolve the errors.

What You Need to Know

This document is intended for system administrators who will install the eLink Adapter on various platforms, as well as programmers who will configure the eLink Adapter and set up TUXEDO services to execute information transfers with MQSeries. This guide assumes knowledge of BEA TUXEDO and IBM MQSeries products.

e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the "e-docs" Product Documentation page at http://e-docs.beasys.com.

How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the eLink documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the eLink documentation Home page, click the PDF files button and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at http://www.adobe.com/.

Related Information

The following BEA publications are also available for more information:

- *BEA TUXEDO Application Development Guide*
- *BEA TUXEDO Programmer's Guide*
- *BEA TUXEDO Reference Guide*

Contact Us

Your feedback on the BEA eLink documentation is important to us. Send us e-mail at **docsupport@beasys.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the eLink documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA eLink Adapter for MQSeries 1.1 release.

If you have any questions about this version of the eLink Adapter, or if you have problems installing and running the eLink Adapter, contact BEA Customer Support through BEA WebSupport at **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

Document Conventions

The following documentation conventions are used throughout this document:

Item	Examples
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 or underscores. The following are examples of variable names in text: *error_file_name* The *when-return* 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:
	`dir c:\accounting\data`
	`Volume in drive C is WIN_NT_1`
	`Volume Serial Number is 1234-5678`
	`Directory of C:\BEADIR\DATA`

Syntax	Code samples can include the following elements: ■ Variable names can include hyphens or underscores (e.g., *error_file_name)* ■ Optional items are enclosed in square brackets: [ ]. If you include an optional item, do not code the square brackets. ■ 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. ■ An ellipsis (...) indicates that the preceding element can be repeated as necessary.
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.

Item	Examples
Environment variables	Environment variables are formatted in an uppercase font. ENVFILE=${APPDIR}
Key names	Key names are presented in boldface type. Press **Enter** to continue.
Literals	Literals are formatted in a monospace font. `class extendSample`
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 **Logon window**. Select **Export** to make the service available to the client.

1 Understanding EAI and the Role of eLink Adapters

This chapter contains the following topics:

- BEA eLink Solution Overview
- Overview of the eLink Adapter for MQSeries

BEA eLink Solution Overview

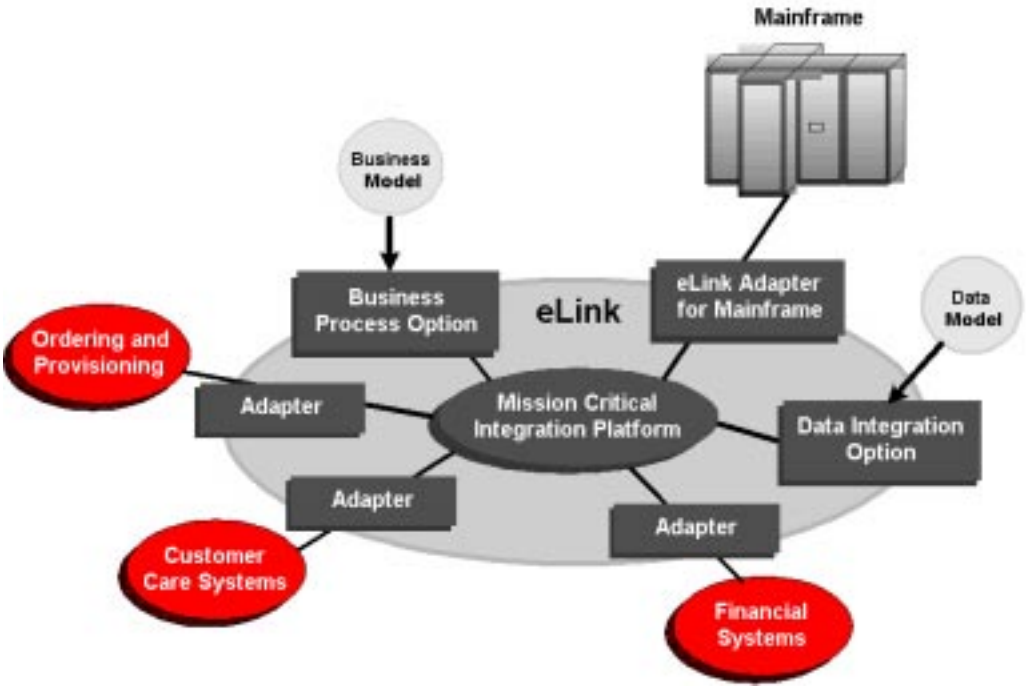
BEA Enterprise Application Integration (EAI) provides an open solution that allows applications throughout organizations to communicate seamlessly. Using EAI, you gain the long-term flexibility and investment protection you need to keep up with today's ever-changing business environment.

Typically, companies use packaged applications to automate internal operations, such as financial, manufacturing, human resources, etc. While they successfully address the needs of these specific areas, these proprietary platforms do not work together. To compete today, you need a much greater exchange of information. Systems need to communicate at both a database and a process level, within your own organization as well as with customer's and supplier's systems. BEA eLink™ Platform is the underlying basis of BEA eLink™, a family of off-the-shelf enterprise application

integration (EAI) products that leverage BEA's transaction platform to integrate existing legacy applications with customer-focused and business-to-business e-commerce initiatives.

BEA eLink Platform provides a proven, rock-solid infrastructure for integrating applications within the enterprise and across the Web. BEA eLink Platform ensures high-performance, secure transactions and transparent access to mission-critical applications and information throughout the enterprise and across the Web. Figure 1-1 illustrates the eLink logical architecture and shows where the eLink Adapters fit into the process.

Figure 1-1 BEA eLink Solution Illustration



The BEA eLink Platform

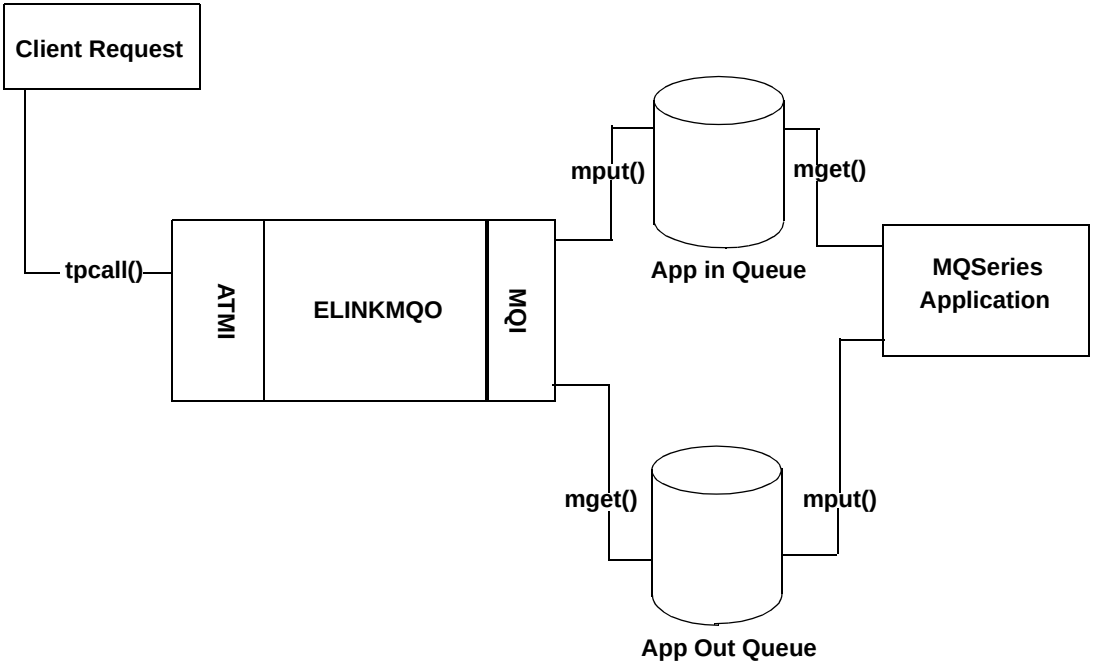
The BEA eLink Platform (in addition to all options and adapters) is highly scalable. Multiple instances of BEA eLink Platforms can collaborate so that work is divided between eLink instances and domains. BEA eLink includes SNMP integration for enterprise management. The BEA eLink Platform features compliance with the Open Group's X/Open standards including support of the XA standard for Two-phase commit processing, the X/Open ATMI API, and XPG standards for language internationalization. C, C++ and Java (via Jolt) are supported. The BEA eLink Platform connects to any RDBMS, OODBMS, file manager or queue manager. The following components operate with BEA eLink Platform:

- The Data Integration Option translates data models used by different applications into a common data format. It provides a cost-effective alternative to writing or generating programs to perform this function. It also handles complex translation with greater power and scalability than rules engines and formatters.
- The Business Process Option helps automate tasks in the business process and dynamically responds to business events and exceptions.
- The eLink Adapters provide the interface between the BEA eLink Platform and external applications, with out-of-the-box functionality (no programming required).

Overview of the eLink Adapter for MQSeries

The eLink Adapter for MQSeries provides communication between IBM MQSeries applications and BEA TUXEDO applications. The eLink Adapter consists of three TUXEDO servers: ELINKMQO, which manages eLink to MQSeries requests, ELINKMQI, which manages MQSeries to eLink requests, and TMQUEUE_MQM, which handles `tpenqueue()` and `tpdequeue()` requests. These servers are managed in the TUXEDO environment. The following diagram illustrates the flow of data when a `tpcall()` is issued from a TUXEDO client to an MQSeries application.

Figure 1-2 Data Flow for `tpcall()` from TUXEDO to MQSeries

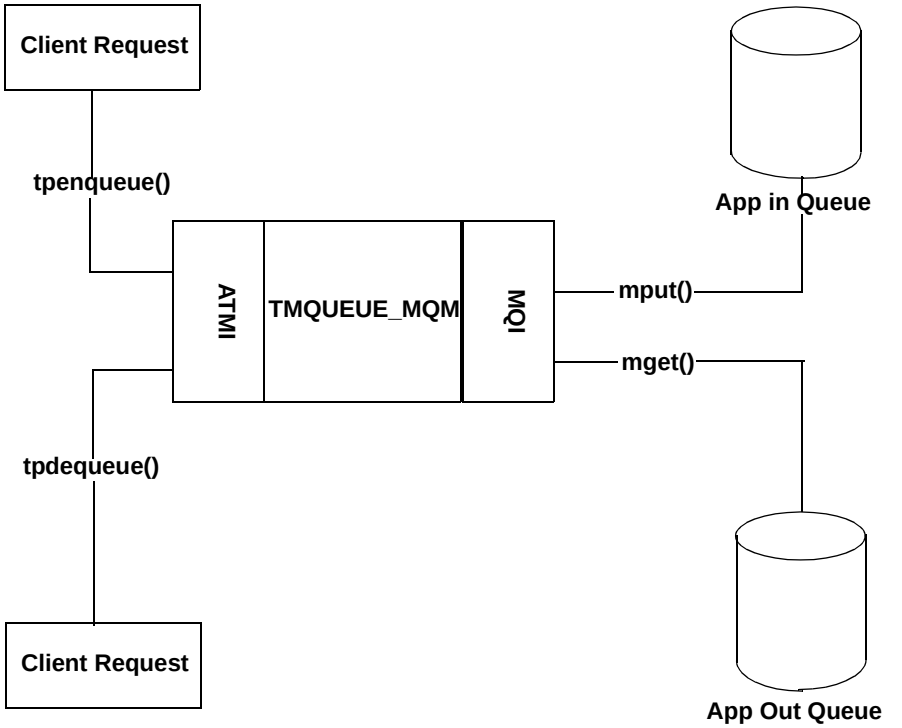


The TUXEDO client calls the service advertised by the eLink to MQSeries server (ELINKMQO). The configuration of the service dictates the queue manager name, queue name, and ReplyTo queue related to the service. The eLink Adapter then places the request on the MQSeries queue to be processed by the MQSeries application. The

eLink Adapter waits on the reply message from the output queue. When the eLink Adapter receives the reply, it returns the response data to the client's outstanding `tpcall()`.

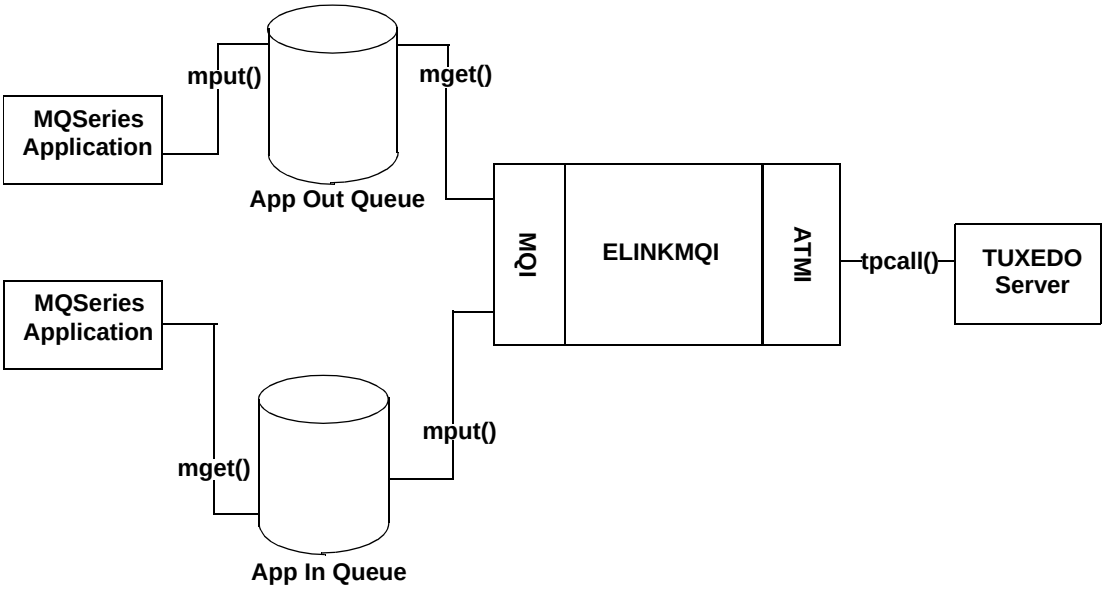
The TMQUEUE_MQM server handles `tpenqueue()` and `tpdequeue()` requests from clients who want to place messages on MQSeries queues. The following diagram shows the enqueuing and dequeuing message flows.

Figure 1-3 Message Flow for `tpenqueue()` and `tpdequeue()`



The MQSeries to eLink server (ELINKMQI), processes service requests from MQSeries applications to TUXEDO servers. The following diagram illustrates the data flow for inbound requests:

Figure 1-4 Data flow for inbound service requests



The MQSeries to eLink server (ELINKMQI) monitors specified queues for requests. When ELINKMQI receives a request, it issues a `tpcall()` request to the designated server. It then places the response data in the ReplyTo queue specified in the original request.

2 Installing BEA eLink Adapter for MQSeries

This chapter contains information for installing and uninstalling the eLink Adapter for MQSeries.

Pre-Installation Considerations

The eLink Adapter for MQSeries software runs on HP-UX, Solaris, and Windows NT. Complete the following tasks prior to installing the eLink Adapter for MQSeries:

- Read the *BEA eLink Adapter for MQSeries Release Notes.*
- Install and verify the operation of the eLink Platform product.
- Set up the required MQSeries resource manager.

The current BEA eLink Platform leverages the BEA TUXEDO infrastructure because it is based on a service-oriented architecture. Both BEA TUXEDO and BEA eLink communicate directly with each other and with other applications through the use of services. Multiple services are grouped into “application servers” or “servers”. The terms, TUXEDO services/servers and eLink services/servers can be used interchangeably. Because this document is specifically addressing the eLink family, the term “eLink service” and “eLink server” is used throughout.

Setting up the MQSeries XA Compliant Resource Manager

Before installing eLink Adapter for MQSeries, set up the MQSeries XA Compliant Resource Manager. The steps to do this are as follows.

1. Add the following line to the TUXEDO RM file: This file is located in the TUXDIR/UDATAOBJ directory (where TUXDIR is the directory where you installed TUXEDO).

   ```
   MQSeries_XA_RMI:MQRMIXASwitchDynamic: /mqmtop/lib/libmqmxa.a \
      mqmtop/lib/libmqm.s1 /opt/tuxedo/lib/libtux.s1
   ```

   Note: In the above example, the backslash ('\') character at the end of the first line is a continuation character. If you enter the entire command on one line in the RM file, do not enter the backslash ('\') character.

   The file extensions shown in the above example vary depending on the platform you are using. The example above illustrates the line in the TUXEDO RM file for the HP-UX platform. For Solaris, the line would be as follows:

   ```
   MQSeries_XA_RMI:MQRMIXASwitchDynamic: /mqmtop/lib/libmqmxa.a \
      mqmtop/lib/libmqm.so mqmtop/lib/libmqmcs.so \
      /mqmtop/lib/libmpmzse.so /opt/tuxedo/lib/libtux.so
   ```

2. Run the following command while in the TUXDIR/BIN directory to create the resource manager executable (where TUXDIR is the directory where you installed TUXEDO).

   ```
   buildtms   -o MQXA -r MQSeries_XA_RMI
   ```

3. Add the following lines to the GROUP section of the UBBCONFIG file for the group used for the eLink Adapter for MQSeries servers.

   ```
   TMSNAME=MQXA TMSCOUNT=x
   OPENINFO="MQSeries_XA_RMI:BEA.TEST.MANAGER"
   ```

   In the above example, the '*x*' indicates the number of instances to execute the MQXA.

Installing BEA eLink Adapter for MQSeries

Installing on Unix Platforms

To install the eLink Adapter for MQSeries on Unix platforms, you run the `install.sh` script. This script installs all the necessary software components.

Perform the following steps to install the eLink Adapter for MQSeries on a supported UNIX platform.

1. Log on as root.

```
$ su -
Password:
```

2. Access the CD-ROM device.

```
# ls -1 /dev/cdrom
total 0
brw-rw-rw-   1 root   sys    27,   0 January 27  10:55 c1b0t010
```

3. Mount the CD-ROM.

```
# mount -r -F cdfs /dev/cdrom/c1b0t010 /mnt
```

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

```
# cd /mnt
```

5. List the CD-ROM contents.

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

6. Execute the installation script.

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

7. The installation script runs and prompts you for responses. Listing 2-1 is an example of the installation script. The entries in bold represent user responses.

Listing 2-1 Install.sh Example

```
01) hp/hpux1020      02) hp/hpux11 03) sun5x/sol26

04) sun5x/sol7

Install which platform's files? [01-6, q to quit, l for list]: 2

** You have chosen to install from hp/hpux11 **

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     emqm       BEA eLink Adapter for MQSeries

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

BEA eLink Adapter for MQSeries
(9000) Release 1.0
Copyright (c) 2000 BEA Systems, Inc.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
BEA eLink is a trademark of BEA Systems, Inc.


Directory where MQSeries Adapter files are to be installed
(Enter your TUXEDO directory path) [?,q]: /work/cmadm/tux65

Using  /work/cmadm/tux65 as the MQSeries Adapter base directory

Determining if sufficient space is available ...
492 blocks are required
8781780 blocks are available to /work/cmadm/tux65

Unloading /cmhome/dist/emqm-1/hp/hpux1020/emqm/EMQMT65.z...
bin/ELINKMQI
bin/ELINKMQO
bin/TMQUEUE_MQM
eLink/mqseries/ELINKMQI.cfg
eLink/mqseries/ELINKMQO.cfg
eLink/mqseries/RM.emqm
eLink/mqseries/TMQUEUE_MQM.cfg
eLink/mqseries/ELINKADK.fml
```

```
eLink/mqseries/ubbconfig.emqm
lib/libadk.sl
lib/libemqcmn.sl
udataobj/binfiles.emqm
480 blocks

...finished

Changing file permissions...
...finished

Installation of BEA eLink Adapter for MQSeries was successful

Please don't forget to fill out and send in your registration card
#
```

8. Change the directory to your root directory.

   ```
   # cd /
   ```

9. Unmount the CD-ROM device.

Installing on Windows NT

Perform the following steps to install the eLink Adapter for MQSeries software on the Windows NT platform.

1. Insert the product CD_ROM and click the **Run** option from the **Start** menu. The **Run** window displays. Click **Browse** to select the CD_ROM drive. Change directories to 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.

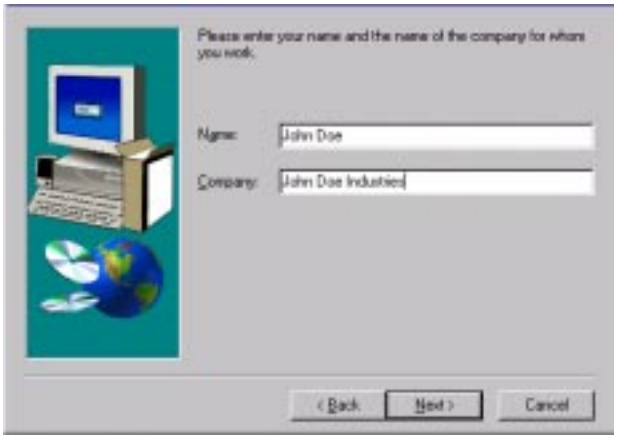
Figure 2-1 Welcome Screen

2. The BEA Software License Agreement displays. Click **Yes** to accept the terms of the agreement and continue with the product installation. Click **No** to exit the installation process.

Figure 2-2 BEA Software License Agreement Screen

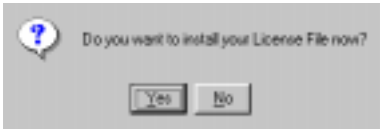
3. The **User Information** screen displays after the License Agreement. Enter the name of the eLink Platform System Administrator in the **Name** field. Enter the name of your company in the **Company** field. Click **Next** to continue with the installation.

Figure 2-3 User Information Screen



4. The **Install License File?** option screen displays next. You may select **Yes** to install your BEA Software License File, or you may select **No** to bypass this step and continue installing the eLink Adapter for Vantive software. If you select **Yes**, continue with Step 5. If you select **No**, continue with Step 7.

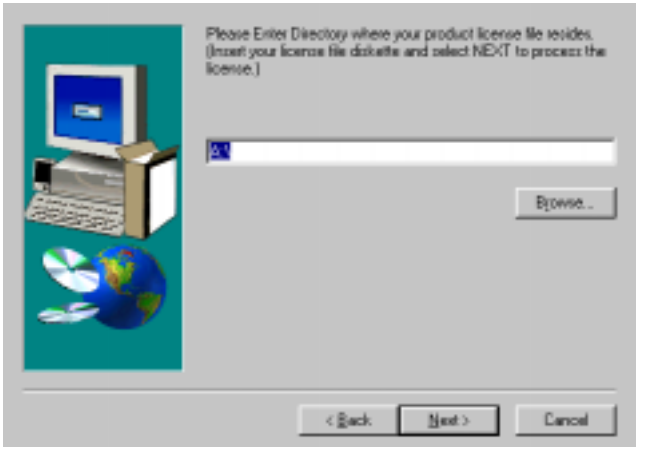
Figure 2-4 Install License File? Screen



5. The **Select License File Source Directory** screen displays. Enter the directory path where your license file resides in the field. You can browse and click directories by clicking the **Browse** button. Typically, the license file is installed in the `tuxedo/udataobj` directory.

   If you entered a valid directory path, click **Next** to continue with the installation. Go to Step 7. If you entered an invalid directory path, go to Step 6.

Figure 2-5 Select License File Source Directory Screen



6. If you do not enter a valid directory path for your license file, the installation software generates an error message dialog box. You can select **Yes** to enter a valid directory path, or you can select **No** to continue with the installation. If you select **No**, the installation software automatically searches for the TUXEDO software. If it finds TUXEDO installed, the installation software completes the process. If TUXEDO is not found, the installation software aborts the process.

   Note: If you select **No**, the installation continues but an error is generated in the ulog.*mm/dd/yy* file indicating that the product is unlicensed. Please refer to the "Using the License Key" section of the *BEA eLink Adapter for MQSeries Release Notes* for instructions on using the license file.

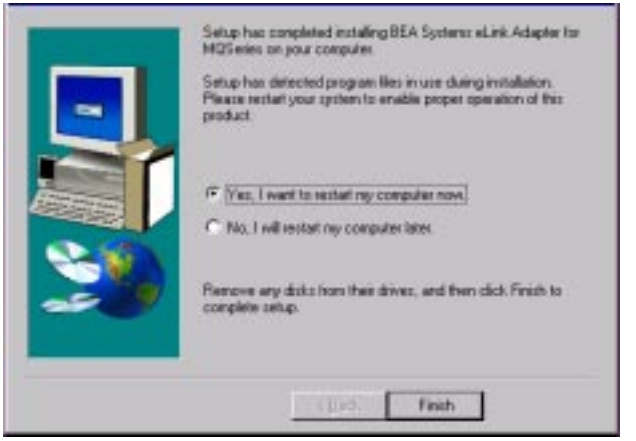
   Once you have entered a valid directory path, click **Next** to continue with the installation. Go to Step 7.

Figure 2-6 Invalid License File Directory Dialog Box

7. A progress bar displays showing the status of the installation.

8. The **Setup Complete** screen displays notifying you that the BEA eLink Adapter for MQSeries product is installed on your system. Click **Finish** to complete the Setup process.

Figure 2-7 Setup Complete Screen

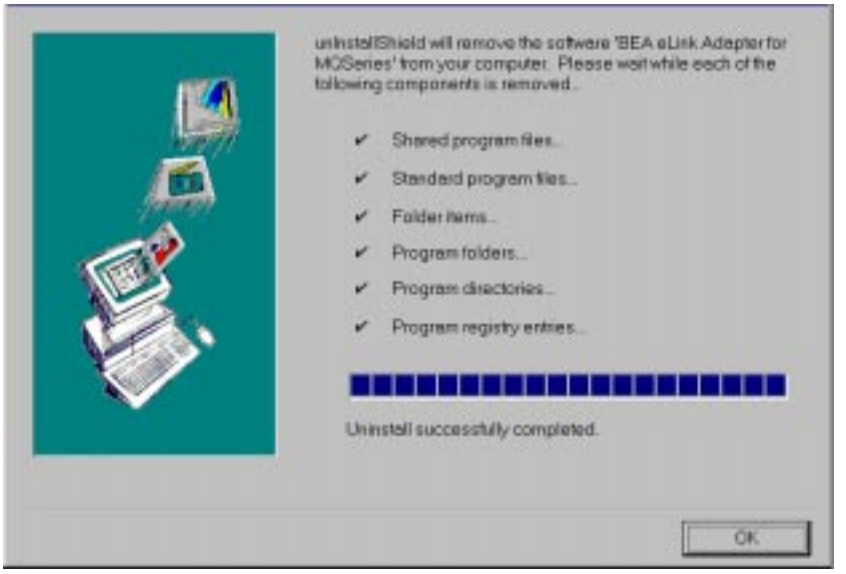


Uninstalling eLink Adapter for MQSeries on Windows NT

Perform the following steps to uninstall the eLink Adapter for MQSeries on a Windows NT system.

1. Click the Start button, and then point to **Settings**. Point to the folder that contains **Control Panel**, and then click **Control Panel**.
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 Program** properties window, select **eLink Adapter for MQSeries** from the program list and click the **Add/Remove** button.
4. The uninstall process for eLink Adapter for MQSeries begins. The **Remove Programs From Your Computer** screen displays. Click **OK** to complete the uninstall process.

Figure 2-8 Remove Programs From Your Computer Screen



Distribution Libraries and Executables

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

Verify that the following files are installed by the eLink Adapter for MQSeries software.

Table 2-1 Distribution Libraries and Executables

Directory	Files
/bin	ELINKMQI
/bin	ELINKMQO
/bin	TMQUEUE_MQM
/bin	lic.sh
/elink/mqseries	ELINKMQI.cfg
/elink/mqseries	ELINKMQO.cfg
/elink/mqseries	TMQUEUE_MQM.cfg
/elink/mqseries	ELINKADK.fml
/elink/mqseries	RM.emqm
/elink/mqseries	ELINKMQM.fml
/elink/mqseries	ubbconfig.emqm
/lib	libadk.s1
/lib	libemqcmn.s1
/udataobj	binfiles.emqm

Post-Installation Procedures

After you complete the installation procedures for the eLink Adapter for MQSeries, make sure the following has been done before continuing with the eLink Adapter configuration procedures:

- MQSeries `/lib` directory must be defined in the library path.
- MQSeries programs you intend to use must be built, must be accessible to run, and must have triggers defined, if applicable.
- All MQSeries queues you intend to use must be defined and must be accessible.

For more information on these procedures, refer to the MQSeries documentation.

3 Configuring eLink Adapter for MQSeries

Configuring the environment for eLink Adapter for MQSeries consists of the following basic tasks:

- Configuring the TUXEDO Servers
- Creating the Server Configuration Files
- Configuring the MQSeries Queue Manager

Configuring the TUXEDO Servers

The eLink Adapter for MQSeries consists of three TUXEDO servers: an eLink to MQSeries server (ELINKMQO), an MQSeries to eLink server (ELINKMQI), and a server for handling `tpenqueue()` and `tpdequeue()` requests (TMQUEUE_MQM). You must identify each of these servers in the TUXEDO UBBCONFIG file. In addition, the eLink Adapter requires that certain parameters be set for each server. You define these parameters in a server configuration file. A sample configuration file for each server is provided on the eLink Adapter for MQSeries installation CD-ROM. You can use these samples as a base and insert the specific information required for your environment.

The following sections describe how to identify each of the servers in the TUXEDO UBBCONFIG file, and how to set up the configuration file required by the eLink Adapter for each of these servers.

Configuring the eLink to MQSeries Server (ELINKMQO)

ELINKMQO routes all requests for MQSeries services from TUXEDO clients. It interacts with MQSeries via the Message Queue Interface (MQI) by enqueuing and dequeuing service requests and responses. The MQI is a common application programming interface that all MQSeries applications implement.

You define the ELINKMQO server in the SERVERS section of the TUXEDO UBBCONFIG file as follows:

Listing 3-1 Syntax for ELINKMQO Server Definition in UBBCONFIG

```
*SERVER
    ELINKMQO SRVGRP="identifier" SRVID="number"
    CLOPT="-- -C configuration_file_name"
```

For information about the SRVGRP, SRVID, and CLOPT parameter syntax and definitions, refer to the *BEA TUXEDO Reference Manual*.

CLOPT= "-- -C *eLink to MQSeries configfile*"
specifies the server's configuration file.

A configuration file provides a list of services and their associated parameters to the server at startup. The sample configuration file included on the installation CD-ROM for ELINKMQO is called ELINKMQO.CFG. Refer to "Creating the Server Configuration Files" for an explanation of the parameters you need to define.

Defining FML32 Fields for the eLink to MQSeries Server

You must also define FML32 fields for ELINKMQO in a field definition table. BEA eLink Platform uses FML functions to manipulate fielded buffers related to errors. Refer to the *BEA TUXEDO FML Programmer's Guide* for more information on FML32.

The syntax for the field definition table for ELINKMQO is as follows:

Listing 3-2 Syntax for field definition table for ELINKMQO

```
# name                    number    type    flags   comments
ELINK_ADAPTER_ERR          'n'     string      -        -
ELINK_ADAPTER_ERR_CODE     'n'     string      -        -
ELINK_APP_ERR              'n'     string      -        -
```

The FML32 fields you need to define are as follows.

ELINK_ADAPTER_ERR
: stores the details of the errors specific to the eLink Adapter for MQSeries.

ELINK_ADAPTER_ERR_CODE
: stores the category code for the adapter-specific errors.

ELINK_APP_ERR
: stores the details of the errors specific to MQSeries.

A sample field definition table (elinkadk.fml) is included on the installation CD-ROM. You can edit this file to assign field ID numbers that are appropriate for your installation.

Note: Make sure the field ID numbers you assign do not conflict with any other field ID numbers you may have for other applications.

When you have created the field definition table, copy it to the directory defined in the environment variable FLDTBLDIR32 and update the environment variable to include this file (elinkadk.fml). Refer to the *BEA TUXEDO Reference Manual* for more information on environment variables.

Configuring the MQSeries to eLink Server (ELINKMQI)

ELINKMQI forwards messages requested from MQSeries applications to TUXEDO services. The application queues a request to a designated queue that is monitored by ELINKMQI. The requested service is specified in the message descriptor. Like ELINKMQO, ELINKMQI must perform data and semantic transformations on the data stored on a queue before delivering it to a service. It must do the same to replies.

You define the ELINKMQI server in the SERVERS section of the TUXEDO UBBCONFIG file as follows:

Listing 3-3 Syntax for ELINKMQI Definition in UBBCONFIG

```
*SERVER
    ELINKMQI="identifier" SRVID="number" REPLYQ=N
    CLOPT="--  -C configuration_file_name
```

For information about the SRVGRP, SRVID, REPLYQ, and CLOPT parameter syntax and definitions, refer to the *BEA TUXEDO Reference Manual.*

CLOPT= "-- -C *MQSeries to eLink configfile*"
specifies the server's configuration file.

A configuration file provides a list of services and their associated parameters to the server at startup. The sample configuration file included on the installation CD-ROM for ELINKMQI is called ELINKMQI.CFG. Refer to "Creating the Server Configuration Files" for an explanation of the parameters you need to define.

Configuring the TMQUEUE_MQM Server

TMQUEUE_MQM processes the `tpenqueue()` and `tpdequeue()` requests from TUXEDO applications that need to send or retrieve data to or from an MQSeries queue.

You define the TMQUEUE_MQM server in the SERVERS section of the TUXEDO UBBCONFIG file as follows:

Listing 3-4 Syntax for TMQUEUE_MQM Definition in UBBCONFIG

```
*SERVER
    TMQUEUE_MQM="identifier" SRVID="number" REPLYQ=N
    CLOPT="--  -C configuration_file_name
```

For information about the SRVGRP, SRVID, REPLYQ, and CLOPT parameter syntax and definitions, refer to the *BEA TUXEDO Reference Manual.*

CLOPT= "-- -C *enqueue/dequeue configfile*"
specifies the server's configuration file.

A configuration file provides a list of services and their associated parameters to the server at startup. The sample configuration file included on the installation CD-ROM for TMQUEUE-MQM is called TMQUEUE-MQM.CFG. Refer to "Creating the Server Configuration Files" for an explanation of the parameters you need to define.

Creating the Server Configuration Files

You must create a configuration file for each of the three eLink Adapter for MQSeries servers. The installation CD-ROM contains sample files you can use as a base for creating your own configuration files. You can substitute the parameter settings in the sample files with the settings required for your own environment.

Creating the eLink to MQSeries Server Configuration File

The ELINKMQO.CFG file controls the operation of the eLink to MQSeries server (ELINKMQO). Following are the sections of the ELINKMQO configuration file and the parameters you can define for each section. A sample configuration file follows the descriptions.

Note: ELINKMQO.CFG is a generic filename. You can name this file anything you choose, but the filename must match the `-C` *`configuration_file_name`* parameter you specify in the TUXEDO UBBCONFIG file. (See Configuring the eLink to MQSeries Server (ELINKMQO) for instructions on configuring the ELINKMQO server in the UBBCONFIG file.)

The ELINKMQO configuration file is divided into the following required sections:

- QUEUE_MANAGER

  Defines the queue manager name and logical ID.

  Note: The configuration file can only have one QUEUE_MANAGER section.

- SERVICES

  Defines various parameters for services and messages.

The ELINKMQO configuration file has one optional section:

- SERVER

  Defines minimum and maximum settings for services and messages.

  Note: The configuration file can only have one SERVER section.

These sections and the parameters within each section can be in any order in the configuration file, as long as the required sections and parameters are defined.

Defining the QUEUE_MANAGER Section (Required)

The syntax for the QUEUE_MANAGER section of the ELINKMQO configuration file is as follows:

Listing 3-5 Syntax for QUEUE_MANAGER section

```
*QUEUE_MANAGER
   LQMID=string
   NAME=string
```

Note: The configuration file can only have one QUEUE_MANAGER section.

Required Parameters

The following parameters must be included in the QUEUE_MANAGER section of the ELINKMQO configuration file.

LQMID= *string*
: Specifies the logical Queue Manager ID for mapping Queues to the QUEUE_MANAGER. Can contain up to 8 alphanumeric characters, upper and lowercase.

NAME= *string*
: Specified the name of the MQSeries Queue Manager. Can contain up to 48 characters, all uppercase. Refer to MQSeries documentation for the specific format of Queue Manager names.

Optional Parameters

There are no optional parameters for the QUEUE_MANAGER section.

Defining the SERVICE Section (Required)

The syntax for the SERVICE section of the ELINKMQO configuration file is as follows:

Listing 3-6 Syntax for SERVICE section

```
*SERVICE
   NAME=string
   MQNAME=string
   LQMID=string
   FORMAT=string
   TRAN=Y/N
   MAXMSGLEN=integer
   REPLYTOQ=string
   TIMEOUT=integer
   INFIELD=string
   OUTFIELD=string
```

Required Parameters

The following parameters must be included in the SERVICE section of the ELINKMQO configuration file.

NAME = *string*

Specifies the TUXEDO Service Name. Can contain up to a maximum of 15 characters, upper or lowercase.

MQNAME = *string*

Specifies the name of the MQSeries Queue. Can contain up to a maximum of 48 characters, all uppercase. Refer to MQSeries documentation for the specific format of Queue names.

LQMID= *string*

Specifies the logical queue manager ID for mapping the SERVICE queue defined by MQNAME to a QUEUE_MANAGER section. Can contain up to 8 alphanumeric characters, upper and lowercase.

MAXMSGLEN= *integer*

Specifies the maximum message length expected for buffers received by this service. Required parameter unless DEFMAXMSGLEN is specified in the

SERVER section. (See Defining the SERVER Section (Optional) for more information).

TIMEOUT = *integer*

Specifies the amount of time, in seconds, that is allowed for processing of the indicated service. The value must be greater than or equal to 0. A value of 0 indicates that the service will not be timed out.

Required parameter unless DEFTIMEOUT is specified in the SERVER section.

Optional Parameters

The following parameters are optional.

FORMAT = *string*

Maps to the MQSeries Message Descriptor Format. Can contain up to 8 characters. Refer to MQSeries documentation for specific format of the Message Descriptor Format field.

TRAN ={Y|N}

Specifies whether the Service is transactional. Y indicates transactional; N indicates non-transactional.
Default value is N.

REPLYTOQ = *string*

Specifies the name of the MQSeries Reply To Queue. Can contain up to 48 characters, all uppercase. Refer to MQSeries documentation for the specific format of Queue names.

INFIELD= *string*

Specifies the FML32 field name of the input data in an FML32 buffer. If you specify this parameter, all requests for this service must pass FML32 buffers for input.

OUTFIELD= *string*

Specifies the FML32 field name of the output data to be used when building an FML32 response buffer. If you specify this parameter, you must also specify the INFIELD parameter. The omission of this parameter causes the response data field name to match the name specified by the INFIELD parameter.

Defining the SERVER Section (Optional)

The syntax for the SERVER section of the ELINKMQO configuration file is as follows:

Listing 3-7 Syntax for SERVERS section

```
*SERVER
    DEFTIMEOUT=integer
    DEFMSGLEN=integer
    MINMSGLEVEL=integer
    MAXMSGLEVEL=integer
```

Note: The configuration file can only have one SERVER section.

Required Parameters

There are no required parameters in the SERVER section.

Optional Parameters

MINMSGLEVEL = *integer*
: Specifies minimum debug level desired for ADKDebug logging.

MAXMSGLEVEL = *integer*
: Specifies maximum debug level desired for ADKDebug logging.

DEFMAXMSGLEN = *integer*
: Specifies the default maximum message length expected for buffers received by eLink to MQSeries services. The DEFMAXMSGLEN is overridden by the Service MAXMSGLEN parameter.

DEFTIMEOUT = *integer*
: Specifies the amount of time, in seconds, that is allowed for processing of the indicated service. The value must be greater than or equal to 0. A value of 0 indicates that the service will not be timed out. The DEFTIMEOUT is overridden by the Service TIMEOUT parameter.

Sample ELINKMQO.CFG

Listing 3-8 Sample Configuration File for ELINKMQO Server

```
*SERVER
    DEFTIMEOUT=60
    DEFMSGLEN=4096
    MINMSGLEVEL=10
    MAXMSGLEVEL=30

*QUEUE_MANAGER
    LQMID=QM1
    NAME=BEA.TEST.MANAGER

*SERVICE
    NAME=MQTest1
    LQMID=QM1
    TIMEOUT=20
    MQNAME=TEST.SAMPLE.ECHO
    REPLYTOQ=TEST.REPLY.QUEUE

*SERVICE
    NAME=MQTest2
    LQMID=QM1
    TIMEOUT=15
    REPLYTOQ=TEST.REPLY.QUEUE
    MQNAME=TEST.SAMPLE.TOUPPER

*SERVICE
    NAME=MQTest3
    LQMID=QM1
    MQNAME=TEST.NOREPLY.QUEUE
    TRAN=YES
    MAXMSGLEN=1024
```

Creating the MQSeries to eLink Server Configuration File

The ELINKMQI.CFG file controls the operation of the MQSeries to eLink server (ELINKMQI). Following are the sections of the ELINKMQI configuration file and the parameters you can define for each section. A sample configuration file follows the descriptions.

Note: ELINKMQI.CFG is a generic filename. You can name this file anything you choose, but the filename must match the `-C` *`configuration_file_name`* parameter you specify in the TUXEDO UBBCONFIG file. (See Configuring the MQSeries to eLink Server (ELINKMQI) for instructions on configuring the ELINKMQI server in the UBBCONFIG file.)

The ELINKMQI configuration file is divided into the following required sections:

- QUEUE_MANAGER

  Defines the queue manager name and logical ID.

  Note: The configuration file can only have one QUEUE_MANAGER section.

- QUEUE

  Defines various parameters for the incoming message queue.

- SERVICE

  Defines various parameters for the TUXEDO service.

The ELINKMQI configuration file has one optional section:

- SERVER

  Defines the minimum and maximum debug level.

  Note: The configuration file can only have one SERVER section.

These sections and the parameters within each section can be in any order in the configuration file, as long as the required sections and parameters are defined.

Defining the QUEUE_MANAGER Section (Required)

The syntax for the QUEUE_MANAGER section of the ELINKMQI configuration file is as follows:

Listing 3-9 Syntax for QUEUE_MANAGER section

```
*QUEUE_MANAGER
   LQMID=string
   NAME=string
```

Note: The configuration file can only have one QUEUE_MANAGER section.

Required Parameters

The following parameters must be included in the QUEUE_MANAGER section of the ELINKMQO configuration file.

LQMID= *string*
: Specifies the logical Queue Manager ID for mapping Queue Managers to queues. Can contain up to 8 alphanumeric characters, upper and lowercase.

NAME= *string*
: Specified the name of the MQSeries Queue Manager assigned to the queue to be processed by the server. Can contain up to 48 characters, all uppercase. Refer to MQSeries documentation for the specific format of Queue Manager names.

Optional Parameters

There are no optional parameters for the QUEUE_MANAGER section.

Defining the QUEUE Section (Required)

The syntax for the QUEUE section of the ELINKMQI configuration file is as follows:

Listing 3-10 Syntax for QUEUE section

```
*QUEUE_MANAGER
   LQMID=string
   MQNAME=string
   MAXMSGLEN=integer
```

Required Parameters

The following parameters must be included in the QUEUE section of the ELINKMQI configuration file.

LQMID= *string*

Specifies the logical Queue Manager ID for mapping Queue Managers to Queues. Can contain up to 8 alphanumeric characters, upper and lowercase.

MQNAME= *string*

Specifies the name of an MQSeries queue to be processed by the server. Can contain up to 48 characters, all uppercase. Refer to MQSeries documentation for specific format of queue names.

MAXMSGLEN = *integer*

Specifies the maximum message length expected for queue buffers received by this server.

Optional Parameters

There are no optional parameters for the QUEUE section.

Defining the SERVICE Section (Required)

The syntax for the SERVICE section of the ELINKMQI configuration file is as follows:

Listing 3-11 Syntax for SERVICE section

```
*SERVICE
   NAME=string
   FORMAT=string
   TRAN=Y/N
```

Required Parameters

The following parameters must be included in the SERVICE section of the ELINKMQI configuration file.

NAME = *string*

Specifies the TUXEDO Service Name. Can contain up to a maximum of 15 characters, upper or lowercase.

FORMAT = *string*

Maps the MQSeries Message Descriptor Format field to the service being called. Can contain up to 8 characters. Refer to MQSeries documentation for specific format of the Message Descriptor Format field.

Optional Parameters

The following parameters are optional.

TRAN ={Y|N}

Specifies whether the Service is transactional. Y indicates transactional; N indicates non-transactional.
Default value is N.

Defining the SERVER Section (Optional)

The syntax for the SERVER section of the ELINKMQI configuration file is as follows:

Listing 3-12 Syntax for SERVER section

```
*SERVER
    MINMSGLEVEL=integer
    MAXMSGLEVEL=integer
```

Note: The configuration file can only have one SERVER section.

Required Parameters

There are no required parameters in the SERVER section.

Optional Parameters

MINMSGLEVEL = *integer*
Specifies minimum debug level desired for ADKDebug logging.

MAXMSGLEVEL = *integer*
Specifies maximum debug level desired for ADKDebug logging.

Sample ELINKMQI.CFG

Listing 3-13 Sample Configuration File for ELINKMQI Server

```
*SERVER
    MINMSGLEVEL=1
    MAXMSGLEVEL=100
*QUEUE_MANAGER
    LQMID=QM1
    NAME=BEA.TEST.MANAGER
*SERVICE
    NAME=SvcToupper
    FORMAT=UPPER
*SERVICE
    NAME=SvcEcho
    FORMAT=ECHO
    TRAN=YES
*QUEUE
    MQNAME=TEST.SAMPLE.QUEUE1
    LQMID=QM1
    MAXMSGLEN=200
*QUEUE
    MQNAME=TEST.SAMPLE.QUEUE2
    LQMID=QM1
```

Creating the enqueue/dequeue Server Configuration File

The TMQUEUE_MQM.CFG file controls the operation of the server that handles `tpenqueue` and `tpdequeue` requests (TMQUEUE_MQM). Following are the sections of the TMQUEUE_MQM configuration file and the parameters you can define for each section. A sample configuration file follows the descriptions.

Note: TMQUEUE_MQM.CFG is a generic filename. You can name this file anything you choose, but the filename must match the `-C` *`configuration_file_name`* parameter you specify in the TUXEDO UBBCONFIG file. (See Configuring the TMQUEUE_MQM Server for instructions on configuring the TMQUEUE_MQM server in the UBBCONFIG file.)

The TMQUEUE_MQM configuration file is divided into the following required sections:

- QUEUE_MANAGER

  Defines the queue manager name and logical ID.

  Note: The configuration file can only have one QUEUE_MANAGER section.

- QUEUE

  Defines various parameters for the incoming message queue.

The TQMQUEUE_MQM configuration file has one optional section:

- SERVER

  Defines the minimum and maximum debug level and the default message length.

  Note: The configuration file can only have one SERVER section.

These sections and the parameters within each section can be in any order in the configuration file, as long as the required sections and parameters are defined.

Defining the QUEUE_MANAGER Section (Required)

The syntax for the QUEUE_MANAGER section of the TQMQUEUE_MQM configuration file is as follows:

Listing 3-14 Syntax for QUEUE_MANAGER section

```
*QUEUE_MANAGER
   LQMID=string
   NAME=string
```

Note: The configuration file can only have one QUEUE_MANAGER section.

Required Parameters

The following parameters must be included in the QUEUE_MANAGER section of the TQMQUEUE_MQM configuration file.

LQMID= *string*

Specifies the logical Queue Manager ID for mapping Queue Managers to queues. Can contain up to 8 alphanumeric characters, upper and lowercase.

NAME= *string*

Specifies the name of the MQSeries Queue Manager assigned to the queue to be processed by the server. Can contain up to 48 characters, all uppercase. Refer to MQSeries documentation for the specific format of Queue Manager names.

Optional Parameters

There are no optional parameters for the QUEUE_MANAGER section.

Defining the QUEUE Section (Required)

The syntax for the QUEUE section of the TQMQUEUE_MQM configuration file is as follows:

Listing 3-15 Syntax for QUEUE section

```
*QUEUE_MANAGER
   LQMID=string
   MQNAME=string
   TUXNAME=string
   MAXMSGLEN=integer
```

Required Parameters

The following parameters must be included in the QUEUE section of the TQMQUEUE_MQM configuration file.

LQMID= *string*

Specifies the logical Queue Manager ID for mapping Queue Managers to Services. Can contain up to 8 alphanumeric characters, upper and lowercase.

MQNAME= *string*

Specifies the name of an MQSeries queue to be processed by the server. Can contain up to 48 characters, all uppercase. Refer to MQSeries documentation for specific format of queue names.

TUXNAME = *string*
Specifies the name of the queue used for ATMI enqueue/dequeue calls. Can contain up to 15 charaters.

Optional Parameters

The following parameter is optional for the QUEUE_MANAGER section of the

MAXMSGLEN = *integer*
Specifies the maximum message length expected for this queue. Value must be less than or equal to the MaxMsgLength of the MQSeries Queue. This parameter overrides the SERVER DEFMAXMSGLEN parameter.

Defining the SERVER Section (Optional)

The syntax for the SERVER section of the TQMQUEUE_MQM configuration file is as follows:

Listing 3-16 Syntax for SERVER section

```
*SERVER
    MINMSGLEVEL=integer
    MAXMSGLEVEL=integer
    DEFMAXMSGLEN=integer
```

Note: The configuration file can only have one SERVER section.

Required Parameters

There are no required parameters in the SERVER section.

Optional Parameters

MINMSGLEVEL = *integer*
Specifies minimum debug level desired for ADKDebug logging.

MAXMSGLEVEL = *integer*
Specifies maximum debug level desired for ADKDebug logging.

DEFMAXMSGLEN=*integer*

Specifies the default maximum message length expected for buffers received by the Queues. The DEFMAXMSGLEN is overridden by the QUEUE MAXMSGLEN parameter. Value must be less than or equal to the MaxMsgLength parameter specified for the MQSeries Queue.

Sample TMQUEUE_MQM.CFG

Listing 3-17 Sample Configuration File for the TMQUEUE_MQM Server

```
*SERVER
    DEFMAXMSGLEN=4096
*QUEUE_MANAGER
    LQMID=QM1
    NAME=BEA.TEST.MANAGER
*QUEUE
    TUXNAME=MQTest1
    LQMID=QM1
    MQNAME=TEST.NOREPLY.QUEUE
*QUEUE
    TUXNAME=MQEcho
    LQMID=QM1
    MQNAME=TEST.SAMPLE.ECHO
    MAXMSGLEN=2048
*QUEUE
    TUXNAME=MQReply
    LQMID=QM1
    MQNAME=TEST.REPLY.QUEUE
```

Configuring the MQSeries Queue Manager

You must configure the MQSeries queue manager in order to run the eLink Adapter for MQSeries. Refer to your MQSeries documentation for specific instructions on configuring queue managers.

4 Running eLink Adapter for MQSeries

Running the eLink Adapter for MQSeries consists of the following basic tasks:

- Booting the Servers
- Initiating a TUXEDO-to-MQSeries Request
- Initiating an MQSeries-to-TUXEDO Request
- Sending and Receiving Messages to and from MQSeries

Booting the Servers

The eLink Adapter for MQSeries servers boot as part of the TUXEDO application using standard TUXEDO utilities, such as `tmboot`. The eLink Adapter reads the server configuration files and attempts to connect to the specified queue manager. Once the eLink Adapter establishes a connection with the queue manager, the eLink to MQSeries server (ELINKMQO) advertises the services associated with that queue manager.

Initiating a TUXEDO-to-MQSeries Request

TUXEDO clients can call services advertised by the eLink Adapter for MQSeries eLink to MQSeries server (ELINKMQO). A TUXEDO-to-MQSeries request consists of the following actions.

1. The TUXEDO client initiates a request for a service advertised by ELINKMQO.
2. The eLink Adapter uses the MQSeries Message Queue Interface (MQI) to forward these requests to the appropriate MQSeries queue.
3. The eLink Adapter retrieves response data (if any) from the designated reply queue and returns this data to the TUXEDO client.

Initiating an MQSeries-to-TUXEDO Request

MQSeries applications can request TUXEDO services via the MQSeries to eLink server ELINKMQI. An MQSeries-to-TUXEDO request consists of the following actions.

1. The MQSeries application queues a message requesting the service to a designated queue.
2. The eLink Adapter retrieves the message from the incoming queue.
3. The eLink Adapter forwards the message data to the appropriate service.
4. The eLink Adapter places response data (if any) on the specified reply queue.

Sending and Receiving Messages to and from MQSeries

The TMQUEUE_MQM server processes `tpenqueue` and `tpdequeue` requests, as well as `tpcall` requests, from a TUXEDO server.

Processing a tpenqueue Request

The syntax for a `tpenqueue` request is as follows:

Listing 4-1 Syntax for `tpenqueue` requests

```
tpenqueue (qspace, qname, qctl, data, len, flags)
```

A `tpenqueue` request requires all parameters shown previously. Following are brief descriptions of these parameters. For more information, refer to the *BEA TUXEDO Programmer's Guide*.

QSPACE

The name of the service advertised by TMQUEUE_MQM. This value can be overwritten in the TUXEDO UBBCONFIG file using the `-s` option. For example:

```
-s myname:MQMQUEUE
```

Default is MQMQUEUE.

QNAME

The name of the queue where you want the adapter to place or retrieve messages. Can contain up to 15 characters. Corresponds to an MQSeries queue defined in the TMQUEUE_MQM.CFG file (MQNAME).

QCTL

Provides additional information about the message. The supported options you can define for `tpenqueue` requests are as follows:

TPNOFLAGS
: No options apply to this message

TPQPRIORITY
: Message priority. TUXEDO values for this field can be from 1-100. The eLink Adapter values can be 0-9 and are mapped to TUXEDO values as follows: 0=TUXEDO 1-10, 1=TUXEDO 11-20, etc.

TPQCORRID
: Correlation ID. Identifies the response to a request. TUXEDO supports up to 32 bytes. The eLink Adapter supports up to 24 bytes and truncates TUXEDO values on `tpenqueue` requests.

TPQREPLYQ
: Name of the queue where you want to place reply messages. Can contain up to 15 characters.

TPQMSGID
: Return the message ID generated when the message is placed on the queue. TUXEDO supports up to 32 bytes. The eLink Adapter supports up to 24 bytes and pads the ID with nulls.

DATA
: Data to be placed on the queue

LEN
: Length of the data to be placed on the queue

FLAGS
: Defines the flag settings for the message. The supported flags for `tpenqueue` requests are as follows:

  TPNOTRAN
  : Messages from callers that are in transaction mode are not queued within the same transaction as the caller.

Processing a tpdequeue Request

The syntax for a `tpdequeue` request is as follows:

Listing 4-2 Syntax for `tpdequeue` requests

```
tpdequeue (qspace, qname, qctl, data, len, flags)
```

A `tpdequeue` request requires all parameters shown previously. Following are brief descriptions of these parameters. For more information, refer to the *BEA TUXEDO Programmer's Guide*.

QSPACE

The name of the service advertised by TMQUEUE_MQM. This value can be overwritten in the TUXEDO UBBCONFIG file using the `-s` option. For example:

```
-s myname:MQMQUEUE
```

Default is MQMQUEUE

QNAME

The name of the queue where you want the adapter to place or retrieve messages. Can contain up to 15 characters. Corresponds to an MQSeries queue defined in the TMQUEUE_MQM.CFG file (MQNAME).

QCTL

Provides additional information about the message. The supported options you can define for `tpdequeue` requests are as follows:

TPNOFLAGS

No options apply to this message

TPQGETBYMSGID

Dequeues the specified message ID.

TPQGETBYCORRID

Dequeues the message with the specified correlation ID.

TPQPRIORITY

Message priority. TUXEDO values for this field can be from 1-100. The eLink Adapter values can be 0-9 and are mapped to TUXEDO values as follows: 0=TUXEDO 1-10, 1=TUXEDO 11-20, etc.

TPQCORRID

Correlation ID. Identifies the response to a request. TUXEDO supports up to 32 bytes. The eLink Adapter supports up to 24 bytes and truncates TUXEDO values on `tpenqueue` requests.

TPQREPLYQ

Name of the queue where you want to place reply messages. Can contain up to 15 characters.

TPQMSGID

Return the message ID generated when the message is placed on the queue. TUXEDO supports up to 32 bytes. The eLink Adapter supports up to 24 bytes and pads the ID with nulls.

DATA

Data to be placed on the queue

LEN

Length of the data to be placed on the queue

FLAGS

Defines the flag settings for the message. The supported flags for `tpdequeue` requests are as follows:

TPNOTRAN

Messages from callers that are in transaction mode are not queued within the same transaction as the caller.

Processing a tpcall Request

Processing a `tpcall` request with the eLink Adapter for MQSeries requires no special parameters. The only requirement is that, if you request a response, the REPLYTOQ must be specified in the server configuration file. Refer to Configuring eLink Adapter for MQSeries for more information on the server configuration files. Refer to the *BEA TUXEDO Programmer's Guide* for more information on `tpcall`.

A Error Messages

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

1000: ELINK_EATMI	**tpopen failed error = *error_text***	
	DESCRIPTION	The Resource Manager failed to open correctly. The exact reason for the failure is indicated by the error. More information concerning the reason a resource manager failed to open can be obtained by interrogating a resource manager in its own specific manner.
	ACTION	Check to determine if the Resource Manager is running. If the Resource Manager is running check the specific error to determine why the open failed.
1001 ELINK_EAPP_API	**Connect to LQMID *queue_manager_LQMID* failed, Reason = *reason_code***	
	DESCRIPTION	An error occurred while attempting to connect to the Queue Manager specified by the LQMID. The specific Reason is shown.
	ACTION	Check the MQSeries Configuration to determine if the specified Queue Manager is defined correctly and is currently running. Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code.

1002: ELINK_EAPP_API	**Inquire on LQMID *queue_manager_LQMID* failed, Reason = *reason_code***	
	DESCRIPTION	The error occurred while attempting to inquire about the Queue Manager specified by the LQMID. The specific Reason is shown.
	ACTION	Check the MQSeries Configuration to determine if the specified Queue Manager is defined correctly and is currently running. Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code.
1003: ELINK_EAPP_API	**Inquire on LQMID *LQMID_value* Queue *queue_name* failed, Reason = *reason_code***	
	DESCRIPTION	An error occurred while attempting to inquire about the Queue specified by the LQMID. The specific Reason is shown.
	ACTION	Check the MQSeries Configuration to determine if the specified Queue Manager and Queue are defined correctly and are currently running. Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code.
1004: ELINK_ECONFIG	**Get First Element failed for LQMID *queue_manager_LQMID***	
	DESCRIPTION	The ELINKMQ configuration file did not contain a Queue Manager definition.
	ACTION	Add the QUEUE_MANAGER section to the configuration file and restart the server.
1005: ELINK_ECONFIG	**Get First Queue Element failed LQMID *queue_LQMID***	
	DESCRIPTION	The ELINKMQ configuration file did not contain a QUEUE definition with an LQMID for the QUEUE_MANAGER associated with the specified LQMID.
	ACTION	Add a QUEUE section to the configuration file and restart the server.

1006: ELINK_WARNING	**Received unexpected tmadmin message**	
	DESCRIPTION	The server received an unexpected message from the Administration service.
	ACTION	No Action is needed.
1007: ELINK_ENOENT	**Unable to locate QUEUE_MANAGER for requested SERVICE *service_name***	
	DESCRIPTION	The Service specified in the call is not associated with a QUEUE_MANAGER in the current configuration.
	ACTION	Check the ELINKMQ configuration file for the specified service and be sure the SERVICE LQMID maps to a valid QUEUE_MANAGER section.
1008: ELINK_EAPP_API	**MQOPEN failed on queue *queue_name* failed Reason = *reason_code*, marking queue down**	
	DESCRIPTION	An error occurred while attempting to open the specified queue. The specific reason is shown.
	ACTION	This queue will be marked inactive and processing of the queue will no longer be performed. Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code. Correct the problem if necessary and restart the server to restart polling.
1009: ELINK_EAPP_API	**MQOPEN failed Reason = *reason_code* - shutting down**	
	DESCRIPTION	A fatal error occurred while attempting to open the specified queue. The specific reason is shown.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code. Correct the problem if necessary and restart the server.

1010: ELINK_WARNING	**MQOPEN failed Reason = *reason_code***	
	DESCRIPTION	An warning occurred while attempting to open the specified queue. The specific reason is shown.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code. Take the appropriate action to resolve the problem with the MQSeries queue.
1011: ELINK_ETRAN	**Unable to start transaction error = *error_text*, queue *queue_name***	
	DESCRIPTION	While processing the specified queue, the request to begin a transaction failed with the indicated error.
	ACTION	Refer to the BEA TUXEDO documentation on `tpbegin` for the specific error. Correct the error if necessary.
1012: ELINK_EATMI	**Could not allocate buffer *error_text***	
	DESCRIPTION	A buffer could not be allocated, probably due to a shortage of system resources.
	ACTION	Verify that the system resources (memory) are available and restart the server.
1013: ELINK_EATMI	**Receive buffer length *length_value* exceeds Queue max message length *length_value***	
	DESCRIPTION	The buffer received from an `MQGET` exceeded the Queue max message size. The Queue max message size is defined to be the lesser of the QUEUE configuration section MAXMSGLEN or the MQSeries default MAXMSGLEN defined for the Queue.
	ACTION	Either shorten the message being sent or increase the MAXMSGLEN of the Queue.

1014: ELINK_EAPP_API	**MQGET on queue *queue_name* failed Reason = *reason_code*, marking queue down**	
	DESCRIPTION	An attempt to get a message from the specified queue failed for the reason indicated. This queue will be marked inactive and processing of the queue will no longer be performed.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code. Correct the problem if necessary and restart the server.
1015: ELINK_EAPP_API	**MQGET failed Reason = *reason_code* - shutting down**	
	DESCRIPTION	A fatal error occurred while attempting to get a message from the specified queue as indicated by the reason code.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code. Correct the problem if necessary and restart the server.
1016: ELINK_ETRAN	**Transaction abort failed error = *error_text,* queue *queue_name***	
	DESCRIPTION	The abort of a transaction failed while processing the specified queue.
	ACTION	No action necessary.
1017: ELINK_INFO	**Message received not a request or datagram and discarded**	
	DESCRIPTION	The MQSeries message received must be of type MQMT_REQUEST or MQMT_DATAGRAM.
	ACTION	No action necessary, however, the message will be dropped.
1018: ELINK_EPERM	**User Identifier *user_identifier_value* not found, unable to Authorize**	
	DESCRIPTION	The user is not authorized to use the requested service.
	ACTION	Check user authorization and add user authorization list or assure the request is from a valid user.

1019: ELINK_ENOENT	**Unable to locate QUEUE_MANAGER for requested QUEUE *queue_name***	
	DESCRIPTION	The requested queue is not associated with a QUEUE_MANAGER in the ELINKMQ configuration file.
	ACTION	Correct the problem in the ELINKMQ configuration file and restart the server.
1020: ELINK_ECONFIG	**Duplicate QUEUE_MANAGER *queue_manager_LQMID* found in configuration**	
	DESCRIPTION	A duplicate QUEUE_MANAGER section was found in the configuration file. A duplicate QUEUE_MANAGER is one whose LQMID parameter matches that of another QUEUE_MANAGER.
	ACTION	Correct the problem and restart the server.
1021: ELINK_ENOENT	**Could not find SERVICE mapping to *service_name***	
	DESCRIPTION	The service requested in the MQSeries Message Descriptor Format field does not map to a SERVICE defined in the ELINKMQ configuration file.
	ACTION	Correct the problem and restart the server. A SERVICE, defined in the configuration file, must be contain a FORMAT field equal to the one in the Message Descriptor Format field.
1022: ELINK_EATMI	**Service *service_name* request failed, error = *error_text***	
	DESCRIPTION	The service request failed with the indicated error.
	ACTION	Refer to the TUXEDO `tpcall` documentation to determine the what action to take for the specified error.

1023: ELINK_EAPP_API	**MQPUT1 failed, Reason =** ***reason_code***	
	DESCRIPTION	An attempt to put a message on the specified queue failed for the reason indicated.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code and correct the problem if necessary.
1024: ELINK_EPROTO	**FML32 buffer received but INFIELD not defined for SERVICE** ***service_name***	
	DESCRIPTION	An FML32 field name to receive data from was not defined for the SERVICE.
	ACTION	Either add the field name to the service configuration or modify the client application to use a different buffer type for the service request.
1025: ELINK_ETRAN	**Commit failed, error =** ***error_text,*** **queue** ***queue_name***	
	DESCRIPTION	An error occurred when attempting to commit a transaction while processing the specified queue.
	ACTION	Refer to the TUXEDO `tpcommit` documentation to determine the what action to take for the specified error.
1026: ELINK_EAPP_API	**MQCLOSE failed, Reason =** ***reason_code*** **LQMID** ***queue_manager_LQMID*** **queue** ***queue_name***	
	DESCRIPTION	An attempt to close the specified Queue Manager queue failed for the reason indicated.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code and correct the problem if necessary.
1027: ELINK_ELIMIT	**INFIELD** ***FML32_field_name*** **parameter for SERVICE exceeds maximum length** ***field_length***	
	DESCRIPTION	The FML32 field name for input exceeds the maximum length of 30 characters.
	ACTION	Change the field name and restart the server.

1028: ELINK_EATMI	**Error *reason_code* sending buffer to Dead Letter queue *queue_manager_LQMID***	
	DESCRIPTION	An attempt to put a message on the specified dead letter queue failed for the reason indicated. The message has been dropped.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code and correct the problem if necessary.
1029: ELINK_ECONFIG	**Dead Letter Queue not defined for LQMID *queue_manager_LQMID***	
	DESCRIPTION	A Dead Letter Queue was not defined for the MQSeries Queue Manager indicated by the LQMID. The message has been dropped.
	ACTION	In order for messages which can not be processed or sent to a specified Reply Queue to be saved a Dead Letter Queue must be defined for the Queue Manager. Define the Dead Letter Queue and restart the server.
1030: ELINK_ECONFIG	**Unable to open configuration file *filename***	
	DESCRIPTION	The configuration file specified in the UBBCONFIG file could not be opened.
	ACTION	Check the ubbconfig file to make sure the configuration file name matches a valid configuration file.
1031: ELINK_ECONFIG	**Invalid SERVER parameter *tag_name***	
	DESCRIPTION	The specified parameter *tag_name* is not a valid SERVER parameter for this server.
	ACTION	Change the SERVER section to reflect a valid parameter and restart the server. Refer to Configuring the TUXEDO Servers for valid SERVER parameters.

1032: ELINK_ECONFIG	**No QUEUE_MANAGER defined**	
	DESCRIPTION	A QUEUE_MANAGER section was not defined in the configuration file. A QUEUE_MANAGER must be defined for this server.
	ACTION	Add the QUEUE_MANAGER section to the configuration file and restart the server.
1033: ELINK_ECONFIG	**Invalid QUEUE_MANAGER parameter *tag_name***	
	DESCRIPTION	The specified parameter *tag_name* is not a valid QUEUE_MANAGER parameter for this server.
	ACTION	Change the QUEUE_MANAGER section to reflect a valid parameter and restart the server. Refer to Configuring the TUXEDO Servers for valid QUEUE_MANAGER parameters.
1034: ELINK_ECONFIG	**QUEUE_MANAGER *queue_manager_LQMID* failed validation**	
	DESCRIPTION	The QUEUE_MANAGER section specified failed validation.
	ACTION	A required parameter was either invalid or missing from the QUEUE_MANAGER definition. Add the required parameters to the QUEUE_MANAGER section and restart the server.
1035: ELINK_ECONFIG	**Unable to add QUEUE_MANAGER to collection**	
	DESCRIPTION	An internal processing error occurred during the load of configuration data, probably due to a shortage of system resources.
	ACTION	Verify that system resources (memory) are available and restart the server.
1036: ELINK_ECONFIG	**No SERVICE defined**	
	DESCRIPTION	A SERVICE section was not defined in the configuration file.
	ACTION	At least one SERVICE section is a required for this server. Add the necessary SERVER section to the configuration file and restart the server.

1037: ELINK_ECONFIG	**Invalid SERVICE parameter *tag_name***	
	DESCRIPTION	The specified parameter *tag_name* is not a valid SERVICE parameter for this server.
	ACTION	Change the SERVICE section to reflect a valid parameter and restart the server. Refer to Configuring the TUXEDO Servers for valid SERVICE parameters.
1038: ELINK_ECONFIG	**Duplicate SERVICE *service_name* found in configuration**	
	DESCRIPTION	A duplicate SERVICE section was found in the configuration file. A duplicate SERVICE is one whose NAME parameter matches that of another SERVICE.
	ACTION	Correct the problem and restart the server.
1039: ELINK_ECONFIG	**SERVICE *service_name* failed validation**	
	DESCRIPTION	The SERVICE section specified failed validation.
	ACTION	A required parameter was either invalid or missing from the SERVICE definition. Add the required parameters to the SERVICE section and restart the server.
1040: ELINK_ECONFIG	**Unable to add SERVICE to collection**	
	DESCRIPTION	An internal processing error occurred during the load of configuration data, probably due to a shortage of system resources.
	ACTION	Verify that system resources (memory) are available and restart the server.
1041: ELINK_ECONFIG	**No QUEUE defined**	
	DESCRIPTION	A QUEUE section was not defined in the configuration file. At least one QUEUE section is a required for this server.
	ACTION	Add the necessary QUEUE section to the configuration file and restart the server.

1042: ELINK_ECONFIG	**Invalid QUEUE parameter *tag_name***	
	DESCRIPTION	The specified parameter is not a valid QUEUE parameter for this server.
	ACTION	Change the QUEUE section to reflect a valid parameter and restart the server. Refer to Configuring the TUXEDO Servers for valid QUEUE parameters.
1043: ELINK_ECONFIG	**QUEUE *queue_name* failed validation**	
	DESCRIPTION	The QUEUE section specified failed validation. A required parameter was either invalid or missing from the QUEUE definition.
	ACTION	Add the required parameters to the QUEUE section and restart the server.
1044: ELINK_ECONFIG	**Invalid LQMID *queue_LQMID* assigned to QUEUE *queue_name***	
	DESCRIPTION	The LQMID for the specified QUEUE did not map to a QUEUE_MANAGER LQMID.
	ACTION	Correct the LQMID parameter in the configuration file and restart the server.
1045: ELINK_ECONFIG	**Duplicate QUEUE *queue_name* found in QUEUE_MANAGER *queue_manager_LQMID* configuration**	
	DESCRIPTION	A duplicate QUEUE section was found in the configuration file. A duplicate QUEUE is one whose NAME parameter matches that of another QUEUE.
	ACTION	Correct the problem and restart the server.
1046: ELINK_ECONFIG	**Unable to add QUEUE to collection**	
	DESCRIPTION	An internal processing error occurred during the load of configuration data, probably due to a shortage of system resources.
	ACTION	Verify that system resources (memory) are available and restart the server.

1047: ELINK_ECONFIG	**Required parameter NAME missing from QUEUE_MANAGER definition**	
	DESCRIPTION	A required parameter is missing from the QUEUE_MANAGER definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.
1048: ELINK_ECONFIG	**Required parameter LQMID missing from QUEUE_MANAGER *queue_manager_name* definition**	
	DESCRIPTION	A required parameter is missing from the QUEUE_MANAGER definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.
1049: ELINK_ECONFIG	**Required parameter NAME missing from SERVICE definition**	
	DESCRIPTION	A required parameter is missing from a SERVICE definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.
1050: ELINK_ECONFIG	**Required parameter FORMAT missing from SERVICE *service_name* definition**	
	DESCRIPTION	A required parameter is missing from the indicated SERVICE definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.
1051: ELINK_ECONFIG	**Required parameter NAME missing from QUEUE definition**	
	DESCRIPTION	A required parameter is missing from a QUEUE definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.

1052: ELINK_ECONFIG	**Required parameter LQMID missing from QUEUE *queue_name* definition**	
	DESCRIPTION	A required parameter is missing from the indicated QUEUE definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.
1053: ELINK_ELIMIT	**OUTFIELD *FML32_field_name* parameter for SERVICE exceeds maximum length *length_value***	
	DESCRIPTION	The FML32 field name for output exceeds the maximum length of 30 characters.
	ACTION	Change the field name and restart the server.
1054: ELINK_ECONFIG	**Required parameter -C missing**	
	DESCRIPTION	The -C option of the CLOPTS parameter for the server is required to indicate the name of the configuration file for the server.
	ACTION	Edit the UBBCONFIG and add the -C config filename parameter to the CLOPTS for the server.
1055: ELINK_INFO	***Server_name* Started**	
	DESCRIPTION	The indicated server has started.
	ACTION	None. Informational only.
1056: ELINK_INFO	***Server_name* Shutdown**	
	DESCRIPTION	The indicated server has shutdown.
	ACTION	None. Informational only.
1057: ELINK_ELIMIT	**Configuration file name exceeds maximum length**	
	DESCRIPTION	The fully qualified configuration file name exceeds the maximum length of 256 characters.
	ACTION	Change the name/location of the configuration file to less than 256 characters, edit the UBBCONFIG to reflect the change, and restart the server.

1058: ELINK_ECONFIG	**LQMID *service_LQMID* for SERVICE *service_name* not defined**	
	DESCRIPTION	The QUEUE_MANAGER associated with the indicated SERVICE definition does not exist.
	ACTION	Change the SERVICE definition to reflect a valid QUEUE_MANAGER.
1059: ELINK_ECONFIG	**SERVICE *service_name* is defined as transactional but LQMID *queue_manager_LQMID* is non-transactional**	
	DESCRIPTION	A SERVICE was defined as transactional, however the associated QUEUE_MANAGER does support syncpoint processing.
	ACTION	Change the SERVICE definition or enable syncpoint processing for the QUEUE_MANAGER.
1060: ELINK_EATMI	**Error advertising SERVICE *service_name*, error = *error_text***	
	DESCRIPTION	An error occurred while trying to advertise a SERVICE name.
	ACTION	Verify that SERVICE name has not already been advertised by another server. Refer to the TUXEDO `tpadvertise` documentation for specific error code information.
1061: ELINK_EOS	**Memory allocation error**	
	DESCRIPTION	An internal processing error occurred during the allocation of memory from system resources.
	ACTION	Verify that system resources (memory) are available and restart the server.
1062: ELINK_EAPP_API	**Unable to disconnect from QUEUE_MANAGER, Reason = *reason_code***	
	DESCRIPTION	An error occurred during shutdown processing while disconnecting from the QUEUE_MANAGER.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code.

1063: ELINK_ENOENT	**Unable to locate requested SERVICE *service_name***	
	DESCRIPTION	The configuration information for the indicated service could not be located.
	ACTION	Verify the service name is valid and is defined in the configuration.
1064: ELINK_EAPP_API	**MQGET on QUEUE *queue* failed, Reason = *reason_code***	
	DESCRIPTION	An error occurred during the retrieval of a message from the indicated QUEUE.
	ACTION	Refer to the IBM MQSeries documentation for the specific MQSeries Reason Code.
1065: ELINK_ELIMIT	**NAME *NAME_value* parameter for QUEUE_MANAGER exceeds maximum length *maximum_length***	
	DESCRIPTION	The NAME parameter for the indicated QUEUE_MANAGER exceeds the maximum number of characters allowed.
	ACTION	Correct the NAME in the configuration and restart the server.
1066: ELINK_ELIMIT	**LQMID *LQMID_value* parameter for QUEUE_MANAGER exceeds maximum length *maximum_length***	
	DESCRIPTION	The LQMID parameter for the indicated QUEUE_MANAGER exceeds the maximum number of characters allowed.
	ACTION	Correct the LQMID in the configuration and restart the server.
1067: ELINK_ELIMIT	**NAME *NAME_value* parameter for SERVICE exceeds maximum length *maximum_length***	
	DESCRIPTION	The NAME parameter for the indicated SERVICE exceeds the maximum number of characters allowed.
	ACTION	Correct the NAME in the configuration and restart the server.

1068: ELINK_ELIMIT	**MQNAME *MQNAME_value* parameter for SERVICE exceeds maximum length *maximum_length***	
	DESCRIPTION	The MQNAME parameter for the indicated SERVICE exceeds the maximum number of characters allowed.
	ACTION	Correct the MQNAME in the configuration and restart the server.
1069: ELINK_ELIMIT	**REPLYTOQ *REPLYTOQ_value* parameter for SERVICE exceeds maximum length *maximum_length***	
	DESCRIPTION	The REPLYTOQ parameter for the indicated SERVER exceeds the maximum number of characters allowed.
	ACTION	Correct the REPLYTOQ in the configuration and restart the server.
1070: ELINK_ELIMIT	**LQMID *LQMID_value* parameter for SERVICE exceeds maximum length *maximum_length***	
	DESCRIPTION	The LQMID parameter for the indicated LQMID exceeds the maximum number of characters allowed.
	ACTION	Correct the LQMID in the configuration and restart the server.
1071: ELINK_ELIMIT	**FORMAT *FORMAT_value* parameter for SERVICE exceeds maximum length *maximum_length***	
	DESCRIPTION	The FORMAT parameter for the indicated SERVICE exceeds the maximum number of characters allowed.
	ACTION	Correct the FORMAT in the configuration and restart the server.

1072: ELINK_ECONFIG	**Required parameter MQNAME missing from SERVICE *service_name* definition**	
	DESCRIPTION	A required parameter is missing from the indicated SERVICE definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.
1073: ELINK_ECONFIG	**Required parameter LQMID missing from SERVICE *service_name* definition**	
	DESCRIPTION	A required parameter is missing from the indicated SERVICE definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.
1074: ELINK_ECONFIG	**Conflicting parameters specified (REPLYTOQ and TRAN=YES) for SERVICE *service_name***	
	DESCRIPTION	A SERVICE can not be defined as having a reply queue and transactional support.
	ACTION	Correct the SERVICE configuration and restart the server.
1075: ELINK_ECONFIG	**SERVICE parameter TIMEOUT not specified and no default available**	
	DESCRIPTION	A TIMEOUT parameter was not defined for the SERVICE and a DEFTIMEOUT parameter was not defined in the SERVER section of the configuration.
	ACTION	Define a default timeout value or add a TIMEOUT parameter to the SERVICE definition.

1076: ELINK_ECONFIG	**SERVICE parameter MAXMSGLEN not specified and no default available**	
	DESCRIPTION	A MAXMSGLEN parameter was not defined for the SERVICE and a DEFMAXMSGLEN parameter was not defined in the SERVER section of the configuration.
	ACTION	Define a default maximum message length value or add a MAXMSGLEN parameter to the SERVICE definition.
1077: ELINK_ECONFIG	**QUEUE *queue_name* parameter MAXMSGLEN not specified and no default available**	
	DESCRIPTION	A MAXMSGLEN parameter was not defined for the QUEUE and a DEFMAXMSGLEN parameter was not defined in the SERVER section of the configuration.
	ACTION	Define a default maximum message length value or add a MAXMSGLEN parameter to the QUEUE definition.
1078: ELINK_ECONFIG	**Required parameter MQNAME missing from QUEUE *queue_name* definition**	
	DESCRIPTION	A required parameter is missing from the indicated QUEUE definition in the configuration file.
	ACTION	Add the required parameter to the configuration and restart the server.
1079: ELINK_ELIMIT	**LQMID *LQMID_value* parameter for QUEUE exceeds maximum length *maximum_length***	
	DESCRIPTION	The LQMID parameter for the indicated QUEUE exceeds the maximum number of characters allowed.
	ACTION	Correct the LQMID in the configuration and restart the server.

1080: ELINK_ELIMIT	**MQNAME *MQNAME_value* parameter for QUEUE exceeds maximum length *maximum_length***	
	DESCRIPTION	The MQNAME parameter for the indicated QUEUE exceeds the maximum number of characters allowed.
	ACTION	Correct the MQNAME in the configuration and restart the server.
1081: ELINK_ELIMIT	**TUXNAME *TUXNAME_value* parameter for QUEUE exceeds maximum length *maximum_length***	
	DESCRIPTION	The TUXNAME parameter for the indicated QUEUE exceeds the maximum number of characters allowed.
	ACTION	Correct the TUXNAME in the configuration and restart the server.
1083: ELINK_EINVAL	**tpdequeue parameters TPQGETBYMSGID and TPQGETBYCORRID are mutually exclusive**	
	DESCRIPTION	The specified dequeue options cannot be specified at the same time.
	ACTION	Change the client code for the `tpdequeue` request to specify only one of these options.
1084: ELINK_EINVAL	**tpdequeue option TPQWAIT not supported**	
	DESCRIPTION	The indicated option is not supported for MQSeries queues.
	ACTION	Change the client code to not specify this option.
1085: ELINK_EINVAL	**tpenqueue parameter TPQTIME_ABS not supported**	
	DESCRIPTION	The indicated option is not supported for MQSeries queues.
	ACTION	Change the client code to not specify this option.

1086: ELINK_EINVAL	**tpenqueue parameter TPQTIME_REL not supported**	
	DESCRIPTION	The indicated option is not supported for MQSeries queues.
	ACTION	Change the client code to not specify this option.
1087: ELINK_EINVAL	**tpenqueue parameter TPQTOP not supported**	
	DESCRIPTION	The indicated option is not supported for MQSeries queues.
	ACTION	Change the client code to not specify this option.
1088: ELINK_EINVAL	**tpenqueue parameter TPQBEFOREMSGID not supported**	
	DESCRIPTION	The indicated option is not supported for MQSeries queues.
	ACTION	Change the client code to not specify this option.
1089: ELINK_EINVAL	**tpenqueue parameter TPQTPFAILUREQ not supported**	
	DESCRIPTION	The indicated option is not supported for MQSeries queues.
	ACTION	Change the client code to not specify this option.
1090: ELINK_ECONFIG	**LQMID *queue_LQMID* for QUEUE *queue_name* not defined**	
	DESCRIPTION	The QUEUE_MANAGER associated with the indicated QUEUE definition does not exist.
	ACTION	Change the QUEUE definition to reflect a valid QUEUE_MANAGER.
1091: ELINK_ECONFIG	**OUTFIELD parameter specified for SERVICE *service_name* but INFIELD parameter missing**	
	DESCRIPTION	A SERVICE definition can not contain an OUTFIELD parameter without an INFIELD parameter.
	ACTION	Either add an INFIELD parameter to the SERVICE definition or remove the OUTFIELD parameter.

1092: ELINK_EPROTO	**SERVICE *service_name* defined as transactional but called outside of a transaction**	
	DESCRIPTION	A tpbegin ATMI call to establish a transaction was not made prior to calling the indicated SERVICE which was defined as transactional.
	ACTION	Alter the client to make a tpbegin call prior to calling the service or modify the SERVICE configuration.
1095: ELINK_ECONFIG	**Only one QUEUE_MANAGER allowed**	
	DESCRIPTION	Only one QUEUE_MANAGER is allowed in the configuration
	ACTION	Remove all but one QUEUE_MANAGER section from the configuration file and restart the server.