



BEA eLink™ for Mainframe TCP IMS User Guide

BEA eLink for Mainframe TCP 3.2
Document Edition 3.2
September 2001

Copyright

Copyright © 2001 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, WebLogic, Tuxedo, and Jolt are registered trademarks of BEA Systems, Inc. How Business Becomes E-Business, BEA WebLogic E-Business Platform, BEA Builder, BEA Manager, BEA eLink, BEA WebLogic Commerce Server, BEA WebLogic Personalization Server, BEA WebLogic Process Integrator, BEA WebLogic Collaborate, BEA WebLogic Enterprise, and BEA WebLogic Server are trademarks of BEA Systems, Inc.

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

BEA eLink Adapter for Mainframe TCP IMS User Guide]

Document Edition	Part Number	Date	Software Version
3.2	N/A	September 2001	BEA eLink for Mainframe TCP 3.2

Contents

About This Document

What You Need to Know	x
e-docs Web Site.....	x
How to Print the Document.....	xi
Related Information.....	xi
Contact Us!.....	xii
Documentation Conventions	xii

1. Introducing BEA eLink TCP for IMS

Features/Benefits	1-1
BEA eLink TCP for IMS Architecture.....	1-2
BEA eLink TCP Functionality	1-3
Domains-based Gateway Connectivity	1-3
Security	1-3
Domain Name Server Support	1-4
Dynamic Configuration.....	1-4
Required System Components	1-4
BEA eLink Gateway Program.....	1-4
IBM LE/370	1-4
IBM TCP/IP for OS/390	1-5
IMS Server Transactions.....	1-5
IMS Client Transactions.....	1-5

2. Understanding How BEA eLink TCP for IMS Works

Inbound Processing	2-2
Outbound Processing.....	2-3
How BEA eLink TCP for IMS Is Initialized.....	2-4

Processing IMS Server Requests	2-5
Processing IMS Client Requests.....	2-5
Two Phases of an IMS Client Request	2-5
How BEA eLink TCP for IMS Processes an IMS Client Request.....	2-6
How BEA eLink TCP for IMS Shuts Down	2-7
3. Configuring BEA eLink TCP Security	
Security Checking from Tuxedo to Mainframe.....	3-1
Security Checking from Mainframe to UNIX.....	3-3
Setting Up Security for eLink TCP for IMS.....	3-4
Securing Connections from IMS to UNIX.....	3-4
Securing Connections from IMS to IMS.....	3-4
Securing Connections from IMS to CICS	3-4
Securing Services	3-5
4. Configuring BEA eLink TCP for IMS	
System Configuration Parameters	3-2
Syntax of the SYSTEM Statement	3-2
Description of SYSTEM Parameters.....	3-3
Defining the Local Host	3-6
Syntax of the HOST Statement	3-7
Description of the Parameters	3-7
Defining the Local Gateway.....	3-8
Syntax of the GATEWAY TYPE=LOCAL Statement.....	3-9
Description of GATEWAY TYPE=LOCAL Parameters	3-9
Defining Remote Gateways.....	3-12
Syntax of the GATEWAY TYPE=REMOTE Statement.....	3-13
Description of GATEWAY TYPE=REMOTE Parameters	3-14
Defining Remote Services	3-17
Syntax of the SERVICE TYPE=REMOTE Statement	3-17
Description of SERVICE TYPE=REMOTE Parameters	3-18
Defining Local Services	3-18
Syntax of the SERVICE TYPE=LOCAL Statement	3-19
Description of Parameters	3-19
Dynamically Configuring eLink TCP for IMS.....	3-21

Modifying SYSTEM Parameters	3-21
Writing the Configuration to a File	3-22
Processing a File.....	3-22
Changing Gateway Configurations	3-23
Adding a Gateway.....	3-23
Starting a Gateway	3-24
Stopping a Gateway	3-24
Modifying an Existing Gateway	3-24
Changing IMS Host Configurations.....	3-25
Adding a Host	3-25
Starting a Host.....	3-26
Modifying an Existing Host.....	3-26
Stopping a Host.....	3-27
Changing Remote Service Definitions	3-27
Adding a Remote Service	3-28
Stopping a Remote Service	3-28
Modifying an Existing Remote Service	3-29
Starting a Remote Service.....	3-29
Changing Local Service Definitions	3-30
Adding a Local Service.....	3-30
Stopping a Local Service	3-31
Modifying an Existing Local Service	3-31
Starting a Local Service	3-32
Display Status Information.....	3-32

5. Monitoring BEA eLink TCP for IMS

Testing Connectivity with Remote Systems.....	4-1
Monitoring and Troubleshooting OTMA.....	4-2
Operating BEA eLink TCP for IMS.....	4-3
Initialization	4-3
Normal Operation.....	4-4
Operator Commands	4-4
Issuing Session-related Messages	4-5
Termination	4-5
SHUTDOWN Command	4-5

A Client-Initiated Shutdown	4-6
The BEA eLink TCP for IMS Message Log	4-7
Message Format.....	4-7
OS/390 Console Messages	4-8
The Server Response Log File.....	4-9

6. Programming BEA eLink TCP for IMS

Programming Your Application	5-1
Programming Interface	5-2
IMS Server Requests	5-2
Message Format.....	5-2
Field Descriptions.....	5-3
Header Files and Copybooks.....	5-3
Returning a Response	5-4
OTMA Client Gateway	5-4
IMS Client Requests	5-4
Message Format.....	5-4
Field Descriptions.....	5-6
Header Files and Copybooks.....	5-9
Error Handling	5-9
Types of Errors	5-9
Other Errors	5-10
How BEA eLink TCP for Tuxedo Translates Data	5-10
BEA Tuxedo Buffer Types.....	5-11
Data Translation Rules	5-11
Strings and Numeric Data: A Closer Look	5-12
Including NULL Characters in String Length Calculations.....	5-13
Converting Numeric Data	5-13

A. System Services Descriptions

TCP/IP for OS/390	A-1
LE/370	A-2
OS/390	A-3

B. Error and Informational Messages

Logical Machine IDs	B-2
---------------------------	-----

Messages Returned to the Remote Gateway	B-2
Messages Written to the BEA eLink TCP for IMS Message Log	B-3

C. Example Configuration Files

D. Sample JCL and User Exits

JCL for Executing eLink TCP for IMS	D-1
Executing eLink TCP for IMS	D-2
Sample User Exits Enabling IMS Clients to Communicate to Remote Services....	D-2
OTMA Prerouting User Exit (DFSYPRX0)	D-3
Sample DSECT BEAOTM	D-8
OTMA Destination Resolution User Exit (DFSYDRU0).....	D-9

E. Sample IMS Client and Server Transactions

IMS Server Transaction.....	E-1
BEASVR01 - A Sample IMS Server Transaction	E-2
Notes	E-4
IMS Client Transactions.....	E-5
Request Phase.....	E-5
Response Phase	E-5
BEACRQ01 - A Sample IMS Client Request Transaction.....	E-6
Notes	E-8
BEACRP01 - A Sample IMS Client Response Transaction	E-10
Notes	E-12
BEACRR01 - A Sample IMS Client Request/Response Transaction.....	E-13
Notes	E-16

F. Error and Reason Codes

Glossary

Index



About This Document

BEA eLink TCP for Mainframe for IMS (hereafter referenced as eLink TCP for IMS) is a gateway connectivity feature that enables OLTP application programs on BEA Tuxedo systems to perform various non-transactional tasks with application programs that reside on IMS. This document describes the eLink TCP product and gives instructions for configuring and administering a gateway system.

The BEA eLink TCP for IMS User Guide is contains the following topics:

- [“Introducing BEA eLink TCP for IMS”](#) provides a description of concepts and processes.
- [“Understanding How BEA eLink TCP for IMS Works”](#) describes how to develop applications that exchange data with a remote BEA eLink gateway.
- [“Configuring BEA eLink TCP Security”](#) describes how to configure security for the gateway.
- [“Configuring BEA eLink TCP for IMS”](#) describes how to create the configuration file and dynamically change configurations.
- [“Monitoring BEA eLink TCP for IMS”](#) describes how to start the eLink TCP application and test connectivity with remote BEA eLink gateways. It also provides information on implementing the gateway system and handling errors.
- [“Programming BEA eLink TCP for IMS”](#) describes how to program your application in COBOL, including message formats and field descriptions.
- [“System Services Descriptions”](#) describes various OS/390, IMS, and TCP/IP services used by eLink TCP.
- [“Error and Informational Messages”](#) describes messages and resolutions for the system.
- [“Example Configuration Files”](#) provides sample configurations.

-
- “[Sample JCL and User Exits](#)” provides sample job control language (JCL) required to install and execute the eLink TCP application.
 - “[Sample IMS Client and Server Transactions](#)” provides sample IMS client and server transactions for use with the eLink TCP application.
 - “[Error and Reason Codes](#)” describes error and reason codes issued by the gateway.
 - “[Glossary](#)” identifies and defines eLink TCP terminology.

What You Need to Know

This document is intended for system administrators who configure and administer the eLink TCP system. In addition, programmers can find useful information for developing client programs and service routines that exchange data with the remote BEA eLink gateways.

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.bea.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 TCP 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 TCP 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 eLink TCP documentation consists of the following items:

- *BEA eLink for Mainframe TCP Installation Guide*
- *BEA eLink for Mainframe TCP Tuxedo User Guide*
- *BEA eLink for Mainframe TCP CICS User Guide*
- *BEA eLink for Mainframe TCP IMS User Guide*
- *BEA eLink for Mainframe TCP Release Notes*

For information about other BEA products, refer to <http://edocs.bea.com/>.

Contact Us!

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

In your e-mail message, please indicate that you are using the documentation for the BEA eLink TCP 3.2 release.

If you have any questions about this version of BEA eLink TCP, or if you have problems installing and running BEA eLink TCP, contact BEA Customer Support through BEA WebSupport at www.bea.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

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.

Convention	Item
<i>italics</i>	Indicates emphasis or book titles.
monospace text	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard. <i>Examples:</i> #include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float
monospace boldface text	Identifies significant words in code. <i>Example:</i> void commit ()
<i>monospace italic text</i>	Identifies variables in code. <i>Example:</i> String <i>expr</i>
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. <i>Examples:</i> LPT1 SIGNON OR
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.
[]	Indicates optional items in a syntax line. The brackets themselves should never be typed. <i>Example:</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...

Convention	Item
	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.
...	<p>Indicates one of the following in a command line:</p> <ul style="list-style-type: none"> ■ That an argument can be repeated several times in a command line ■ That the statement omits additional optional arguments ■ That you can enter additional parameters, values, or other information <p>The ellipsis itself should never be typed.</p> <p><i>Example:</i></p> <pre>buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...</pre>
.	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

1 Introducing BEA eLink TCP for IMS

BEA eLink products offer a set of communication gateways between BEA Tuxedo systems and other online transaction processing environments, including IBM's CICS and IMS. They allow applications to scale in size by providing interoperability between systems. The BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) product is a gateway connectivity feature that provides transparent communications between client and server transactions in an IMS system and a BEA Tuxedo domain, a CICS system, or another IMS system. (A BEA Tuxedo domain is a single computer [or network of computers] that shares a single BEA Tuxedo configuration.)

The following information introduces the eLink TCP for IMS product:

- [Features/Benefits](#)
- [BEA eLink TCP for IMS Architecture](#)
- [BEA eLink TCP Functionality](#)
- [Required System Components](#)

Features/Benefits

The features and benefits provided by BEA eLink for Mainframe TCP for IMS are:

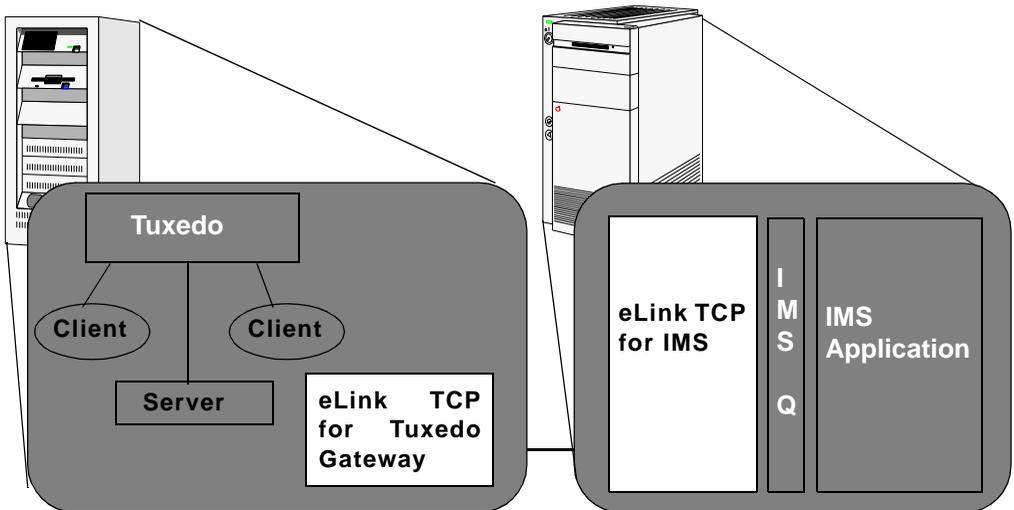
- Ensured online integration of business operations through transparent client/server communications among heterogeneous systems

- Truly open distributed applications
- High throughput through TCP/IP for online request/response messaging
- Ease of integration; modular architecture for flexible on-schedule migration
- High performance open solution

BEA eLink TCP for IMS Architecture

The eLink TCP for IMS product executes as an open transaction manager access (OTMA) client running under OS/390. It uses the OS/390 TCP/IP Sockets API to communicate with an eLink TCP for Tuxedo gateway on a remote BEA Tuxedo system, and the IMS message queue (directly or indirectly through OTMA) to communicate with IMS transactions. This product supports connectivity with remote eLink TCP for Tuxedo gateways, eLink TCP for CICS gateways, or other eLink TCP for IMS gateways. [Figure 1-1](#) illustrates connection with a remote eLink TCP for Tuxedo gateway.

Figure 1-1 The BEA eLink TCP for IMS Interoperability Solution



The configuration definition of the eLink TCP for Tuxedo gateway in the remote BEA Tuxedo system maps BEA Tuxedo local services to remote services provided by the IMS gateway. Each remote service corresponds to an IMS server transaction. Service requests issued by clients in the BEA Tuxedo system are routed through the eLink TCP for Tuxedo gateway to the eLink TCP for IMS gateway for processing by the appropriate IMS server transaction.

Similarly, the configuration definition in the eLink TCP for IMS gateway maps local service names to remote services provided by the BEA Tuxedo system. Service requests issued by IMS client transactions are routed to the eLink TCP for Tuxedo gateway for processing by the appropriate BEA Tuxedo service.

In both cases, access to a remote service is transparent to the client requesting the service.

BEA eLink TCP Functionality

The following functionality is available in the eLink TCP product.

Domains-based Gateway Connectivity

The eLink TCP product has a domains-based architecture supporting bidirectional communications, request/response support, and support for an IMS OTMA interface.

Security

The eLink TCP product grants access to Tuxedo services based on a user name that the remote gateway supplies.

The eLink TCP for IMS product has an OTMA interface that supports enhanced security. This interface allows a requester from Tuxedo to pass a user ID through the OTMA server interface for authorization through a third party security package.

For more information about configuring security, refer to the [“Monitoring BEA eLink TCP for IMS”](#) section.

Domain Name Server Support

The eLink TCP product supports domain name server (DNS) resolution of IP addresses. This support allows you to change the IP address at the domain name server to implement address changes without reconfiguring the eLink TCP gateway.

Dynamic Configuration

The eLink TCP product supports dynamic configuration updates of the eLink TCP configuration parameters. You can modify the configuration without shutting down and restarting the eLink TCP product.

Required System Components

The eLink TCP for IMS product requires the following components for transaction processing.

BEA eLink Gateway Program

The eLink TCP for IMS product executes as an OTMA client running under OS/390. It uses the OS/390 TCP/IP sockets API to communicate with remote BEA eLink gateways and the IMS message queue (directly or indirectly through OTMA in the case of the OTMA client) to communicate with IMS application transactions.

IBM LE/370

IBM LE/370 provides a common runtime environment for a number of IBM high-level languages, including C/C++. The eLink TCP for IMS gateway is written in the C programming language and requires the LE/370 runtime environment.

IBM TCP/IP for OS/390

IBM TCP/IP for OS/390 provides TCP/IP communications support for programs executing in an OS/390 environment by means of a standard, UNIX-style socket API. The eLink TCP for IMS product uses TCP/IP to communicate with remote BEA eLink gateways.

Note: This guide assumes you are using the IBM TCP/IP for OS/390 product. If you are using another supported TCP/IP product, such as Interlink TCPaccess, consult the product documentation for equivalent components.

IMS Server Transactions

IMS server transactions are user-written programs that process service requests issued by a remote system. Server transactions are ordinary IMS transactions and can be written in any language supported by IMS. For supported languages, refer to the IBM IMS manuals.

IMS Client Transactions

IMS client transactions are user-written programs that issue requests for services offered by a remote system. Client transactions are ordinary IMS message processing program (MPP) transactions and can be written in any language supported by IMS.

1 *Introducing BEA eLink TCP for IMS*

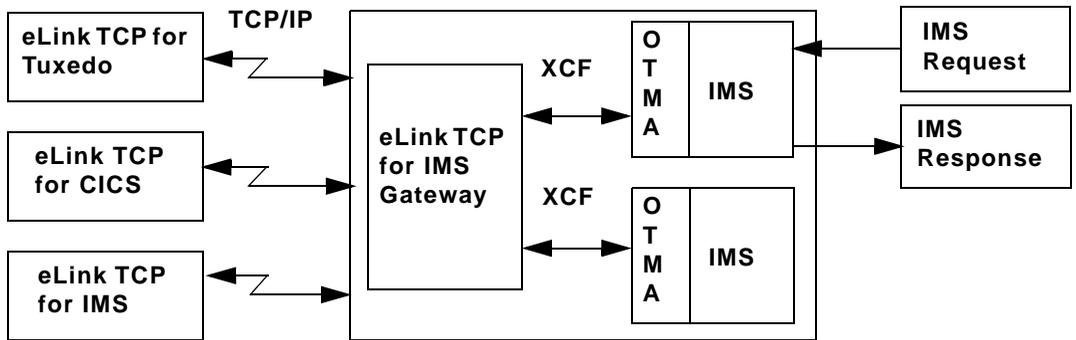
2 Understanding How BEA eLink TCP for IMS Works

The following information helps you understand how BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) works:

- [Inbound Processing](#)
- [Outbound Processing](#)
- [How BEA eLink TCP for IMS Is Initialized](#)
- [Processing IMS Server Requests](#)
- [Processing IMS Client Requests](#)
- [How BEA eLink TCP for IMS Shuts Down](#)

As shown in [Figure 2-1](#), the eLink TCP for IMS gateway serves as the interface between IMS and remote BEA eLink gateways via TCP/IP.

Figure 2-1 IMS Processing

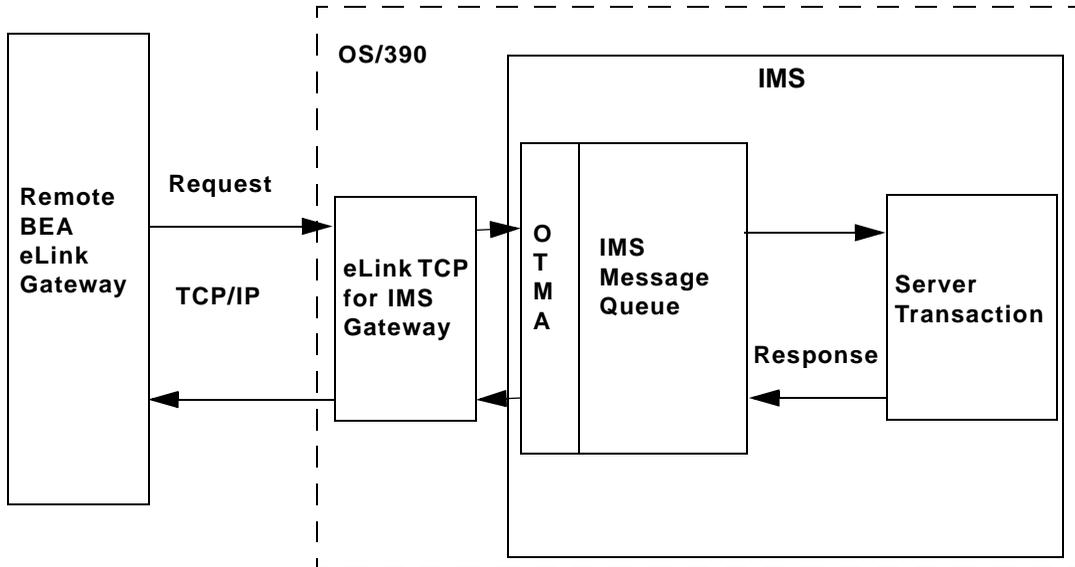


Inbound Processing

The gateway “listens” for incoming TCP/IP connection requests from remote gateways. When a request is received, an *inbound session* is established over which the remote gateway can present requests for service.

As shown in [Figure 2-2](#), when a request is received from a remote system, it is relayed to IMS which schedules the appropriate server transaction to process the request. If a response is required, the server transaction places the response in the IMS message queue. If the eLink TCP gateway is running as an OTMA client, the response is queued to the transaction pipe and delivered to the client through the cross-system coupling facility (XCF). The response is returned to the remote system over the TCP/IP connection.

Figure 2-2 IMS Inbound Processing

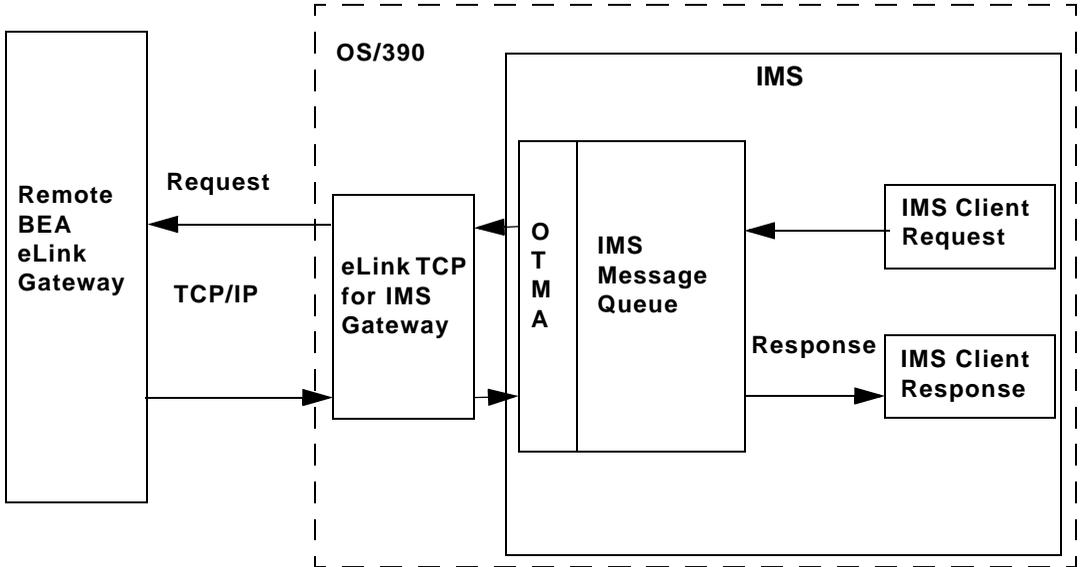


Outbound Processing

The eLink TCP for IMS product can also initiate TCP/IP connections with remote systems. These *outbound sessions* are used to send IMS client requests to remote systems for processing.

As shown in Figure 2-3, an IMS client transaction initiates a request by placing a properly formatted message into the IMS message queue. When running the gateway, two IMS user exits must be installed to route messages to the OTMA client. (For more information about request/response processing, refer to the [“Programming BEA eLink TCP for IMS”](#) section, and for sample user exits, refer to the [“Sample JCL and User Exits”](#) section.) The gateway retrieves the request and forwards it to the appropriate remote system for processing. When the response (if required) is received from the remote system, it is returned to IMS for delivery to a transaction that processes the response.

Figure 2-3 IMS Outbound Processing



How BEA eLink TCP for IMS Is Initialized

The eLink TCP for IMS gateway is started by submitting the appropriate JCL (or as a started task) for an OTMA client. The following activities then occur.

1. The gateway (load module name, BEATCPI) opens a message log (normally allocated to a disk dataset) into which all messages issued during the course of execution are written.
2. If the message log cannot be opened, the gateway immediately terminates; any messages are written to a SYSOUT dataset by default.
3. The gateway obtains configuration information from a sequential dataset containing text statements. These statements define the operating environment for eLink TCP for IMS, including the definitions of local and remote gateways. If any configuration errors are detected during initialization, the gateway immediately terminates.

Processing IMS Server Requests

An IMS server request, also referred to as an *inbound request* (relative to IMS), is a request issued by a remote client for a service provided by an IMS server transaction.

1. When eLink TCP for IMS receives a request from a remote system, a *request message* is formatted and is inserted into the IMS message queue, destined for the specified IMS server transaction (the *service name* for the request corresponds to the *transaction code* of the IMS server transaction that handles the request).
2. IMS schedules the specified server transaction to process the request message.
3. The server transaction retrieves the request message from the IMS message queue and performs the indicated processing.
4. If a response is required, the server transaction formats a *response message* and inserts it into the IMS message queue, destined for the eLink TCP for IMS gateway.
5. The eLink TCP for IMS gateway retrieves the response message from IMS and returns it to the requesting remote system via the TCP/IP connection.

Processing IMS Client Requests

An IMS client request, also referred to as an *outbound request* (relative to IMS), is a request issued by an IMS application message processing program (MPP) for a service provided by a remote system.

Two Phases of an IMS Client Request

Because of the design philosophy of IMS, processing of an IMS client request occurs in two distinct “phases.”

1. A *request phase* wherein the request is issued during the execution of one transaction (T1).
2. A corresponding *response phase* wherein the response is processed during the execution of a second transaction (T2).

Transactions T1 and T2 may in fact be the same transaction ID (with appropriate logic to perform the required request or response processing, based on execution context). T1 and T2 must be two distinct transaction executions. This distinction is necessary because T1 can only initiate a request; it cannot “wait” on the response to that request because the architecture and design philosophy of IMS does not permit this.

How BEA eLink TCP for IMS Processes an IMS Client Request

1. An IMS transaction issues a client request by inserting a properly formatted message into the IMS message queue specifying the eLink TCP for IMS OTMA client (through the use of the appropriate IMS user exit) as the destination.
Note: For information about the IMS user exits, refer to the [“Sample JCL and User Exits”](#) section.
2. Once the request has been inserted into the Message Queue, the requester is effectively “decoupled” from the request; that is, processing of the request by eLink TCP for IMS proceeds asynchronously with respect to execution of the requesting transaction. The request includes the name (transaction code) of a *response transaction* that is scheduled by eLink TCP for IMS to process the response to the request.
3. The eLink TCP for IMS gateway retrieves the request from IMS and validates it for proper format. If any errors are detected, the specified response transaction is immediately scheduled with appropriate return codes.
4. The requested service name is located in the directory of services offered by remote systems to which eLink TCP for IMS has access. If the requested service cannot be found, or the required remote gateway is currently inaccessible, an error is recognized and the response transaction is immediately scheduled with appropriate return codes.

5. If a remote service provider is successfully identified for the request, an outbound session is established with the remote system (if one is not available) and the request is sent to the remote system for processing.
6. If a response *is* required, eLink TCP for IMS schedules the response transaction (with any response data) when the response is received.
7. Eventually, if no response is received from the remote system, the request *times out*, an error is recognized, and the response transaction is scheduled with appropriate return codes.

How BEA eLink TCP for IMS Shuts Down

Once started, eLink TCP for IMS normally executes as a non-ending job, servicing inbound requests from remote systems and outbound requests originated by IMS client transactions.

Normal termination is initiated when a system operator issues the `SHUTDOWN` command. In response to a `SHUTDOWN` command, eLink TCP for IMS performs the following tasks.

1. Stops accepting new inbound TCP/IP connection requests.
2. Terminates active inbound sessions.
3. Terminates active outbound sessions.
4. Returns control to OS/390.

Note: Pending or in progress transactions are not allowed to complete prior to the system shutting down.

2 *Understanding How BEA eLink TCP for IMS Works*

3 Configuring BEA eLink TCP Security

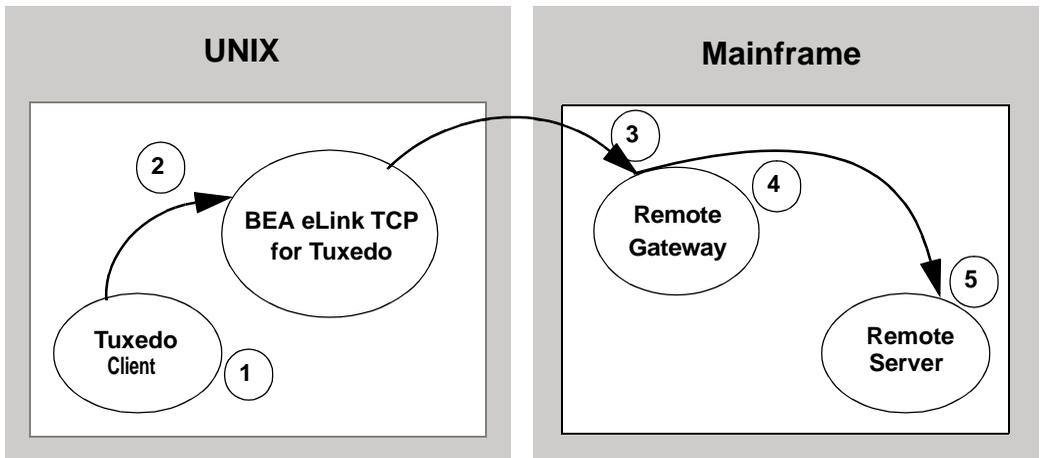
The eLink TCP product supports a security feature that allows a requester from BEA Tuxedo services to pass a user ID requirement through the OTMA or CICS server interfaces for verification through a third-party security package.

Note: The security feature on eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) runs as an OTMA client only.

Security Checking from Tuxedo to Mainframe

[Figure 3-1](#) depicts the process flow for security verifications from eLink TCP for Tuxedo on UNIX to a mainframe.

Figure 3-1 Security Checking for UNIX to Mainframe Transactions



1. Verify the user is valid on UNIX. For valid users, access is given; for invalid users, access is rejected.
2. Verify user name (reviewing the `tpusr` file), group (reviewing the `tpgrp` file), and ACL (reviewing the `tpacl` file). If all three pass, the transaction request processes. If any one of the three are rejected, the transaction request stops and a security violation occurs.

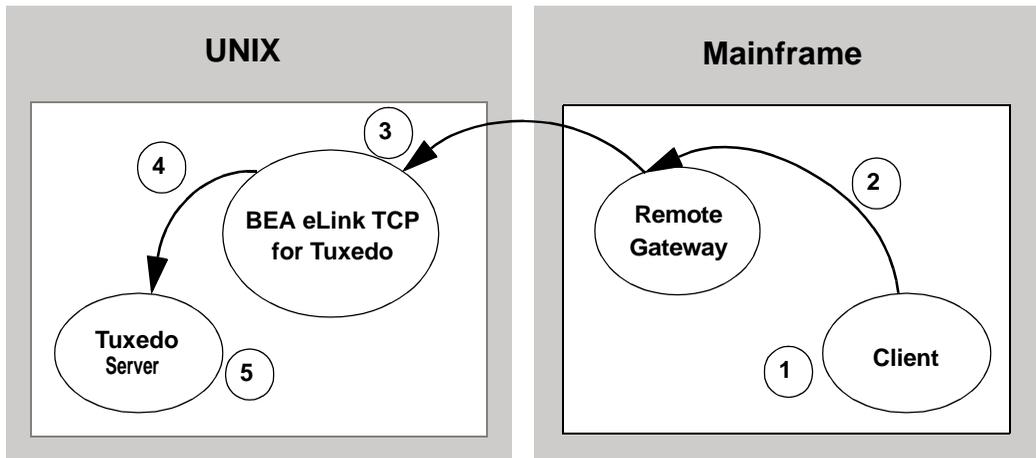
Note: The user IDs in these files must match in the Tuxedo and the mainframe environments or a security violation occurs.

3. Accept the transaction request at the mainframe gateway based on the request coming from a *trusted source*. No password is passed.
4. Verify the user name associated with the transaction against the security system (such as RACF). If the user name is not valid, the request is rejected and a security violation occurs.
5. Complete the transaction request to the server.

Security Checking from Mainframe to UNIX

Figure 3-2 depicts the process flow for security verifications from a mainframe to eLink TCP for Tuxedo on UNIX.

Figure 3-2 Security Checking for Mainframe to UNIX Transactions



1. Authorization checking is done by the mainframe security package prior to initiating the client.
2. Pass the transaction request from the client to the mainframe gateway.
3. Accept the transaction request at the UNIX gateway based on the request coming from a *trusted source*. No password is passed.
4. Decode the `appkey` to obtain the user and group numbers. Verify the user name against the security system. If the user name is valid and the user has the authority to run the transaction, the transaction request is accepted. If the user name is not valid, the request is rejected and a security violation occurs.
5. Complete the transaction request to the server if the user name is accepted.

Setting Up Security for eLink TCP for IMS

The eLink TCP for IMS product has an OTMA interface that supports enhanced security. This interface allows a requester from BEA Tuxedo services to pass a user ID through the OTMA server interface for authorization through your security package.

Securing Connections from IMS to UNIX

Complete the following tasks to enable the connection security feature.

1. Specify the `ACCOUNT` and `PASSWORD` parameters in the `GATEWAY` configuration statement for local or remote gateways.
2. Verify that the parameter values for `ACCOUNT` and `PASSWORD` in the `GATEWAY` statement match the `RMTACCT` and `PASSWORD` values in the `*FOREIGN` section of the eLink TCP for Tuxedo `GWICONFIG` file.

Securing Connections from IMS to IMS

Complete the following tasks to enable the connection security feature.

1. Specify the `ACCOUNT` and `PASSWORD` parameters in the `GATEWAY` configuration statement for local or remote gateways.
2. Verify that the parameter values for `ACCOUNT` and `PASSWORD` in the `GATEWAY` statement for the `GATEWAY TYPE=LOCAL` match the `ACCOUNT` and `PASSWORD` values in the `GATEWAY TYPE=REMOTE` statement.

Securing Connections from IMS to CICS

Complete the following tasks to enable the connection security feature.

1. Specify the `ACCOUNT` and `PASSWORD` parameters in the `GATEWAY TYPE=LOCAL` configuration statement.

2. Verify that the parameter values for `ACCOUNT` and `PASSWORD` in the `GATEWAY TYPE=LOCAL` statement match the `ACCOUNT` and `PASSWORD` values in the User Account Connection screen.

Securing Services

Complete the following tasks to enable the service security feature.

1. Set up transaction security through the mainframe with the security administrator.
2. Specify `OTMASECURITY=Y` in the `SYSTEM` statement of your eLink TCP for IMS configuration file.
3. Set the security flag for each local service using the `SECURITY` parameter in `SERVICE TYPE=LOCAL` statement. For parameter information, refer to the [“Defining Local Services”](#) section.
4. Issue the `/SEC OTMA PROFILE` command in IMS to enable security checking on a service by service basis for the OTMA interface. Issue the `/SEC OTMA FULL` command in IMS to enable security checking on all services.

Warning: If `SECURITY=N` in the `SERVICE TYPE=LOCAL` statement for any local service definition, issue `/SEC OTMA PROFILE`. A security failure results if you specify `SECURITY=N` and issue `/SEC OTMA FULL` command.

3 *Configuring BEA eLink TCP Security*

4 Configuring BEA eLink TCP for IMS

BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) makes use of a *configuration file* that is read and processed at runtime to establish the processing environment. The configuration file is a simple, plain-text file that a systems programmer can edit easily to define and alter the configuration.

The configuration is defined in terms of the local (IMS) gateways, one or more remote gateways (for example, CICS), and one or more remote services offered by various remote gateways to which eLink TCP for IMS can have access. When using the OTMA client, you must configure local services as well.

To change the configuration file without shutting down the eLink TCP for IMS system, refer to [“Dynamically Configuring eLink TCP for IMS”](#) section.

The configuration file includes the following information:

- A `SYSTEM` statement that defines the system configuration parameters (refer to the [“System Configuration Parameters”](#) section)
- Statements that define one or more IMS host systems (OTMA only) with which eLink TCP for IMS establishes communications (refer to the [“Defining the Local Host”](#) section)
- Statements that define one or more local gateways (refer to the [“Defining the Local Gateway”](#) section)
- Statements that define one or more remote gateways with which eLink TCP for IMS can establish connections and exchange requests and responses (refer to the [“Defining Remote Gateways”](#) section)

- Statements that define one or more services offered by remote gateways that can be requested by IMS clients of eLink TCP for IMS (refer to the “[Defining Remote Services](#)” section)
- Statements that define one or more services offered by local IMS Hosts (OTMA only) that can be requested by remote clients of eLink TCP for IMS (refer to the “[Defining Local Services](#)” section)
- Security information when running as an OTMA client (refer to the “[Dynamically Configuring eLink TCP for IMS](#)” section)

System Configuration Parameters

System configuration parameters are defined by the `SYSTEM` statement. Only one `SYSTEM` statement is allowed, and it should be the first statement in the configuration file. See [Listing 4-1](#) for parameter syntax.

Syntax of the SYSTEM Statement

The following listing contains the parameter syntax for the `SYSTEM` statement.

Listing 4-1 `SYSTEM` Statement Syntax

```
SYSTEM,  
    SLEEPTIME=(sleep-min,sleep-max,sleep-delta),  
    OTMACLIENT=eLink-clientname ],  
    OTMAGROUP=XCF-groupname ]  
    [ ,OTMASECURITY={Yes|No} ]  
    [ ,MSGLEVEL={msg-level|4} ]  
    [ ,CLIENTSHUTDOWN={Yes|No} ]  
    [ ,IDLETIMEOUT={nn|0} ]  
    [ ,DRUEXIT=exit-name|DFSYDRU0 ]  
    [ ,SPARESOCKETS=nn|0 ]
```

Description of SYSTEM Parameters

The `SYSTEM` statement uses the following parameters.

Parameter	Value	Description
SLEEPTIME	sleep-min	Specifies the minimum amount of time (decimal, milliseconds) that eLink TCP for IMS delays before checking for work. Default value: 150
	sleep-max	Specifies the maximum amount of time (decimal, milliseconds) that eLink TCP for IMS delays before checking for work. Default value: 500

4 Configuring BEA eLink TCP for IMS

Parameter	Value	Description
	<code>sleep-delta</code>	<p>Specifies the incremental amount of time (decimal, milliseconds) that eLink TCP for IMS uses to graduate from the minimum sleep time to the maximum sleep time.</p> <p>eLink TCP for IMS periodically “polls” for new work. The eLink TCP for IMS product uses a graduated polling interval that is set to the minimum sleep time whenever new work is found. The polling interval is then incrementally increased by the delta time until one of the following occurs:</p> <ul style="list-style-type: none">■ New work is found (in which case the polling interval is again reset to the minimum value)■ The maximum sleep time is reached (in which case the polling interval remains at the maximum value) <p>This value limits the amount of CPU time required to poll for work when traffic is heavier.</p> <p>Avoid specifying values that cause the polling interval to be very small (or zero), or values smaller than 100 milliseconds. These values may result in polling too frequently. However, also avoid specifying values that cause the polling interval to be too large (greater than one second). These values may cause eLink TCP for IMS to be unresponsive. Varying the sleep time between 100 and 500 milliseconds in 5 millisecond increments is recommended.</p> <p>Default value: 5</p>
<code>OTMACLIENT</code>	<code>clientname</code>	<p>Specifies the XCF client name for eLink TCP OTMA. This parameter is 1-16 characters.</p>

Parameter	Value	Description
OTMAGROUP	groupname	Specifies the XCF group name for IMS and eLink TCP. This parameter is 1-8 characters.
OTMASEcurity	Yes No	Specifies whether to enable or disable security for user authorization. Specify Yes to enable security or No to disable security. Default value: No
MSGLEVEL	msg-level 4	Specifies the message detail level to be logged by eLink TCP for IMS. 0 - No messages are logged 2 - Error Messages only 3 - Informational Messages and Error Messages 4 - Security Audit Messages The message level is cumulative. For example, if a message level of 4 is specified, all messages with a message level of 4 or less is logged (such as, security audit, informational, and error messages) and is also displayed on the OS/390 console. Normally, a message level of 4 should be specified, which results in eLink TCP for IMS logging all security, informational and error messages. Default value: 5

4 Configuring BEA eLink TCP for IMS

Parameter	Value	Description
CLIENTSHUTDOWN	Yes No	Specifies whether a SHUTDOWN request can be made by a remote client request or as a response to a remote client request. For additional information for shutting down the system, refer to the “Monitoring BEA eLink TCP for IMS” section. Default value: No
IDLETIMEOUT	nn 0	Specifies the number of seconds that lapses before the local gateway terminates a session. Default value: 0
DRUEXIT	exit-name	Specifies the name of the destination resolution User Exit routine in IMS that this client should use. The default is DFSYDRU0. For a sample of the DFSYDRU0 user exit, refer to the “Sample JCL and User Exits” section.
SPARESOCKETS	nn	Specifies the number of spare socket descriptors reserved for dynamic configuration changes, for example, adding a gateway, increasing the maximum number of sessions for a gateway. Default value: 0.

Defining the Local Host

IMS host systems that service IMS requests are defined with the `HOST TYPE=IMS` statement. This statement is only used when running eLink TCP for IMS as an OTMA client.

Syntax of the HOST Statement

The following listing contains the parameter syntax for the HOST statement.

Listing 4-2 HOST Statement Syntax

```
HOST,TYPE=IMS,
    IMSID=IMS-system-logical-identifier,
    OTMANAME=IMS-clientname,
    OTMAINPIPE=server-request-name,
    OTMAOUTPIPE=client-request-name]
[ ,NUMTPIPES=(nnnn | 1) ]
```

Description of the Parameters

The HOST statement uses the following parameters.

Parameter	Value	Description
IMSID	IMS-system-logical-identifier	Specifies a symbolic name that uniquely identifies the IMS system definition. This parameter is not related to the IMSID parameter specified in the IMS sysgen or on IMS region JCL statements.
OTMANAME	IMS-clientname	Specifies the XCF client name for the IMS system. This parameter is 1-16 characters.
OTMAINPIPE	server-request-name	Specifies the transaction pipe name for the IMS server requests. This parameter is 1-4 characters.

Parameter	Value	Description
OTMAOUTPIPE	client-request-name	Specifies the transaction pipe name for the IMS client requests. This parameter is 1-8 characters.
NUMTPIPES	nnnn 1	Specifies the number of transaction pipes for IMS server requests. The value can range from 1-9,999. Default value: 1

Defining the Local Gateway

The local (IMS) gateway defines the access points that remote gateways can use to connect to the IMS gateway.

The local gateway can be defined as having a single physical IP address and port number, or multiple physical IP addresses and/or port numbers. Furthermore, each physical port can be defined as accepting one or more concurrent *inbound* TCP/IP connections from remote gateways.

One or more `GATEWAY TYPE=LOCAL` statements must exist in the configuration file to define the local gateways. Each `GATEWAY TYPE=LOCAL` statement defines a specific IP address and port number through which eLink TCP for IMS accepts inbound connection requests from remote gateways. Inbound connections are used by a remote gateway to send client requests to the IMS gateway for processing by IMS server transactions. Each local gateway (as defined by a `GATEWAY TYPE=LOCAL` statement) must have a unique logical machine ID (a symbolic name used to identify a gateway).

Remote systems attempting to establish an inbound session with the local gateway can optionally be required to supply a valid account ID and/or password.

For sample configurations, refer to the [“Example Configuration Files”](#) section.

Syntax of the GATEWAY TYPE=LOCAL Statement

Listing 4-3 contains the parameter syntax for the local gateway.

Listing 4-3 Local Gateway Syntax

```
GATEWAY,TYPE=LOCAL,
    LMID=logical-machine-id,
    PORT=port-number,
    ACCOUNT={account-id|*},
    PASSWORD={password|*}
    [,HOSTADDR=(ip-address/hostname)]
    [,MAXMSGLEN={max-message-length|2048}]
    [,SESSIONS={(startup-sessions,max-sessions)|1}]
    [,IDLETIMEOUT={nn|0}]
    [,MULTIPLEX={nn|1}]
```

Description of GATEWAY TYPE=LOCAL Parameters

The GATEWAY TYPE=LOCAL statement uses the following parameters.

Parameter	Value	Description
LMID	<i>logical-machine-id</i>	Specifies a symbolic name (1-16 characters) that uniquely identifies this gateway/port. Note: Special characters are <i>not</i> supported.

4 Configuring BEA eLink TCP for IMS

Parameter	Value	Description
PORT	port-number	Specifies the port number (decimal, 1-32,767) at which inbound connections from remote gateways are accepted. Choose a port number that is not in the range of well-known port numbers reserved for use by TCP/IP applications.
ACCOUNT	account-id *	Specifies the account ID (1-8 characters) that must be specified by a remote gateway to establish a connection. If an account ID is not used, ACCOUNT=* should be specified.
PASSWORD	password *	Specifies the password (1-8 characters) that must be specified by a remote gateway to establish a connection. If a password is not used, PASSWORD=* should be specified.

Parameter	Value	Description
HOSTADDR	ip-address hostname	<p>Is the IP address or hostname for this machine over which inbound connections from remote gateways are accepted at the specified port number. The IP address is specified using standard dotted-decimal notation (for example, 199 . 203 . 45 . 99). The hostname is specified as a 1-256 character string and must match the DNS entry.</p> <p>Note: If any special characters, such as "-" are in the host address, quotes must be placed around the address name. Example: HOSTADDR= "SIA-COTS"</p> <p>Default value: If this parameter is not specified, then connections are accepted over any network interface.</p>
MAXMSGLN	max-message-length 2048	<p>Specifies the maximum TCP/IP message length that is accepted over the connection. The value range is 1-32,767.</p> <p>Default value: 2048</p>

Parameter	Value	Description
SESSIONS	(<i>startup-sessions</i> , <i>max-sessions</i>) 1	Is the maximum number of concurrent, active inbound sessions that is accepted by this gateway. The input format is (<i>startup-sessions</i> , <i>max-sessions</i>), where the <i>startup-sessions</i> represents the number of sessions to initiate at startup and <i>max-sessions</i> specifying the maximum concurrent, active, inbound sessions. The range for the <i>max-sessions</i> is 0-32,767. Default value: 1
IDLETIMEOUT	nn 0	Specifies the number of seconds that lapse before the local gateway terminates a session. Default value: 0
MULTIPLEX	nn 1	Is the maximum number of concurrent messages per session that is accepted by this gateway. Default value: 1

Defining Remote Gateways

A remote gateway is another BEA eLink gateway (TUX, CICS, or IMS) to which eLink TCP for IMS can be connected by one or more TCP/IP connections. A remote gateway definition defines the access points that eLink TCP for IMS uses to connect to the remote gateway.

The remote gateway is defined by its IP address and port number. The remote gateway may be defined as accepting one or more concurrent *outbound* TCP/IP connections initiated by the local (IMS) gateway. Outbound connections are used by the IMS gateway to send IMS client requests to a remote gateway for processing by the requested service.

One or more `GATEWAY TYPE=REMOTE` statements are used in the configuration file to define remote gateways. Each `GATEWAY TYPE=REMOTE` statement defines the IP address and port number that eLink TCP for IMS uses to establish an outbound connection with a remote gateway. The `GATEWAY TYPE=REMOTE` statement also identifies a remote gateway by a unique logical machine ID.

The configuration file may contain multiple `GATEWAY TYPE=REMOTE` statements, each defining a different remote gateway. If no `GATEWAY TYPE=REMOTE` statements are present, then eLink TCP for IMS cannot establish outbound connections with remote gateways, and therefore cannot process IMS client requests.

The eLink TCP for IMS software can optionally be configured to supply a valid account ID and valid password when establishing an outbound session with a remote gateway.

For examples of the `GATEWAY TYPE=REMOTE` statement, refer to the “[Example Configuration Files](#)” section.

Syntax of the GATEWAY TYPE=REMOTE Statement

Listing 4-4 contains the parameter syntax for the `GATEWAY TYPE=REMOTE` statement.

Listing 4-4 Remote Gateway Syntax

```
GATEWAY,TYPE=(REMOTE,{TUX|IMS|CICS,tran-id|CICSPT}),
    LMID=logical-machine-id,
    HOSTADDR=(ip-address|hostname),
    PORT=port-number,
    ACCOUNT={account-id,*},
    PASSWORD={password,*}
    [,MAXMSGLEN=max-message-length|2048]
    [,SESSIONS=(min-sessions),(max-sessions)]
    [,MULTIPLY={nn|0}]
    [,IDLETIMEOUT={nn|0}]
```

Description of GATEWAY TYPE=REMOTE Parameters

The GATEWAY TYPE=REMOTE statement uses the following parameters.

Parameter	Value	Description
TYPE	(REMOTE, type)	<p>Specifies the remote gateway type. You must specify one of the following.</p> <p>TUX - The remote gateway is an eLink TCP for Tuxedo gateway.</p> <p>IMS -The remote gateway is an eLink TCP for IMS gateway.</p> <p>CICS, <i>tran-id</i> - The remote gateway is an eLink TCP for CICS gateway. <i>tran-id</i> specifies the 1-4 character name of the eLink TCP for CICS transaction that is to be scheduled by CICS to handle incoming TCP/IP eLink requests. Refer to the <i>BEA eLink TCP for CICS User Guide</i> for more information.</p> <p>CICSCPT - The remote gateway is an eLink TCP for CICS gateway that uses the Interlink CPT product for TCP/IP access.</p>
LMID	logical-machine -id	<p>Specifies a symbolic name (1-16 characters) that uniquely identifies the remote gateway. The logical machine ID is used to associate a remote service with a particular remote gateway.</p>

Parameter	Value	Description
HOSTADDR	ip-address host name	<p>Specifies the IP address of the remote gateway. The IP address is specified using standard dotted-decimal notation (for example, 103.204.79.82). The hostname is specified as a 1-256 character string and must match the DNS entry.</p> <p>Note: If any special characters, such as "-" are in the host address, quotes must be placed around the address name. Example: HOSTADDR= "SIA-COTS"</p>
PORT	port-number	Specifies the port number (decimal, 1-32,767) to be used to establish a connection with the remote gateway.
ACCOUNT	account-id	Specifies the account ID (1-8 characters) to be used to establish a connection with the remote gateway. If an account ID is not used, ACCOUNT=* should be specified.
PASSWORD	password	Specifies the password (1-8 characters) to be used to establish a connection with the remote gateway. If a password is not used, PASSWORD=* should be specified.
MAXMSGLEN	max-message-length	<p>Specifies the maximum TCP/IP message length (decimal, 1-32,767) that is sent over the connection.</p> <p style="text-align: right;">Default value: 2048</p>

4 Configuring BEA eLink TCP for IMS

Parameter	Value	Description
SESSIONS	min-sessions	<p>Specifies the minimum number of outbound sessions (decimal, 0-32,767) that are to be automatically started with this gateway during initialization of eLink TCP for IMS. Additional sessions (up to the maximum number of sessions specified) are started on a demand basis.</p> <p>Remember that by specifying a minimum number (greater than 0) of outbound sessions for a remote gateway, you can cause outbound session to be automatically established with that gateway during initialization. This process can be helpful in verifying outbound connectivity without having to make use of test transactions.</p> <p>Default value: 0</p>
	max-sessions	<p>Specifies the maximum number of concurrent, active, outbound sessions (decimal, 0-32,767) that are established with this gateway.</p> <p>Default value: 1</p>
MULTIPLEX	nn 0	<p>Specifies the maximum number of concurrent messages per session that the gateway accepts.</p> <p>Default value: 1</p>
IDLETIMEOUT	nn 0	<p>Specifies the number of seconds that lapse before the remote gateway terminates a session. The remote gateway terminates sessions until it is equal to the minimum sessions allowed.</p> <p>Default value: 0</p>

Defining Remote Services

A remote service is a service offered by a remote gateway to which eLink TCP for IMS has access. IMS client transactions can make requests for services offered by remote gateways.

One or more `SERVICE TYPE=REMOTE` statements are used in the configuration file to define remote services. Each `SERVICE TYPE=REMOTE` statement defines the local service name (used by the IMS client transaction to request the service), the remote service name (used to invoke the service on the remote system), and the logical machine ID of a remote gateway that offers the service. There must be a corresponding `GATEWAY TYPE=REMOTE` statement in the configuration file for the specified logical machine ID.

The configuration file can contain multiple `SERVICE TYPE=REMOTE` statements for a given (local) service name, each pointing to a different logical machine ID (that is, remote gateway). In this case, eLink TCP for IMS distributes requests for the service among the available service providers, thus maximizing throughput during busy periods.

For examples of this statement, refer to the [“Example Configuration Files”](#) section.

Syntax of the SERVICE TYPE=REMOTE Statement

Listing 4-5 contains the parameter syntax for the `SERVICE TYPE=REMOTE` statement.

Listing 4-5 Remote Services Syntax

```
SERVICE,TYPE=REMOTE,  
    NAME=(local-service-name,remote-service-name),  
    LMID=logical-machine-id  
    [,TIMEOUT={timeout|30}]
```

Description of SERVICE TYPE=REMOTE Parameters

The `SERVICE TYPE=REMOTE` statement uses the following parameters.

Parameter	Value	Description
NAME	local-service-name	Specifies the name (1-16 characters) by which the service is known on <i>this</i> machine. This is the name which must be used by an IMS client to request the service.
	remote-service-name	Specifies the name (1-16 characters) by which the service is known on the <i>remote</i> machine. This name is the name used to invoke the service on the remote machine.
LMID	logical-machine-id	Specifies the logical machine ID (1-16 characters) of a remote gateway that offers this service. This name must match a logical machine ID (LMID) specified in a <code>GATEWAY TYPE=REMOTE</code> statement.
TIMEOUT	timeout 30	Specifies the time limit in seconds (decimal, 1 - 32,767) for the service. If a response is not received within the specified period of time, an error condition is recognized. Default value: 30

Defining Local Services

A service offered by a local IMS host is defined by a `SERVICE TYPE=LOCAL` statement. There can be multiple `SERVICE TYPE=LOCAL` statements and multiple statements defining the same service, provided each specifies a different local `IMSID`.

Syntax of the SERVICE TYPE=LOCAL Statement

Listing 4-6 contains the parameter syntax for the `SERVICE TYPE=LOCAL` statement.

Note: `SERVICE TYPE=LOCAL` statements are only used when running eLink TCP for IMS as an OTMA client.

Listing 4-6 Local Service Syntax

```
SERVICE,TYPE=LOCAL,
    NAME=(local-service-name,remote-service-name),
    IMSID=logica-IMS-id,
    [,V21COMPAT={Yes|No}]
    [,FASTPATH={Yes|No}]
    [,SECURITY={Yes|No}]
    [,RESPONSE=remote-service-name]
```

Description of Parameters

The `LOCAL SERVICES` statement uses the following parameters.

Parameter	Value	Description
NAME	local-service-name	Specifies the name (1-16 characters) by which the service is known on <i>this</i> machine. This name is the name passed to IMS as the transaction name.
	remote-service-name	Specifies the name (1-16 characters) by which the service is known on the <i>remote</i> machine. This name is the name passed by the remote client to invoke this service.
IMSID	logica-IMS-id	Specifies a symbolic name that uniquely identifies an IMS system definition. This name must match a logical IMS ID (IMSID) specified in a <code>HOST, TYPE=IMS</code> statement.

4 Configuring BEA eLink TCP for IMS

Parameter	Value	Description
FASTPATH	Yes No	Specifies whether this service is a FastPath transactions. Specify <code>Yes</code> to enable FastPath transactions or <code>No</code> to disable FastPath transactions. Default value: <code>No</code> .
SECURITY	Yes No	Specifies whether to enable or disable security for this service. Specify <code>Yes</code> to enable security or <code>No</code> to disable security. Default value: <code>No</code> .
RESPONSE	<code>remote-service-name</code>	Specifies the remote service name you want to send the response to as a request. Note: This parameter is only valid for local service requests with a <code>NOREPLY</code> response.
V21COMPAT	Yes No	Specifies whether the transaction specifically developed to run with version 2.1 or previous versions of eLink TCP. Specify <code>Yes</code> to run transactions that were developed with version 2.1 or previous versions of eLink TCP for IMS. Specify <code>No</code> to run transactions that were <i>not</i> developed with version 2.1 of eLink TCP for IMS or previous versions of eLink TCP for IMS. Default value: <code>No</code> .

Dynamically Configuring eLink TCP for IMS

The eLink TCP for IMS product makes use of a *configuration file* that is read and processed at runtime to establish the processing environment. The configured parameters can be modified while the gateway is running. You can make the following changes to the eLink TCP for IMS gateway configuration dynamically by entering the commands on TCP Outstanding WTOR:

- [Modifying SYSTEM Parameters](#)
- [Writing the Configuration to a File](#)
- [Processing a File](#)
- [Changing Gateway Configurations](#)
- [Changing Remote Service Definitions](#)
- [Display Status Information](#)

Modifying SYSTEM Parameters

You can modify the following SYSTEM parameters dynamically using the MOD SYSTEM command.

- SLEEPTIME
- MSGLEVEL
- CLIENTSHUTDOWN

The example in [Listing 4-7](#) uses the MOD SYSTEM command.

Listing 4-7 MOD SYSTEM Command Example

```
MOD SYSTEM, SLEEPTIME=(150,500,5), MSGLEVEL=4, CLIENTSHUTDOWN=NO
```

For parameter descriptions and syntax, refer to the “[Description of SYSTEM Parameters](#)” section.

Writing the Configuration to a File

To write the current configuration to a file, use the `CHKPT CONFIG` command with a fully qualified file name. The example in [Listing 4-8](#) saves the current configuration to a file named `BEA.ELINK.CONFIG1`.

Listing 4-8 `CHKPT CONFIG` Command Example

```
CHKPT CONFIG, FILE="BEA.ELINK.CONFIG1"
```

Processing a File

To process a file containing commands, use the `INFILE` command with a fully qualified file name. The example in [Listing 4-9](#) processes the `BEA.ELINK.TESTFILE1` file.

Listing 4-9 `INFILE` Command Example

```
INFILE BEA.ELINK.TESTFILE1
```

Changing Gateway Configurations

You can dynamically change gateway configurations by adding a new gateway, modifying an existing gateway definition, starting a gateway, and stopping a gateway. For parameter descriptions and syntax, refer to the [“Description of GATEWAY TYPE=LOCAL Parameters”](#) and [“Description of GATEWAY TYPE=REMOTE Parameters”](#) sections.

Adding a Gateway

To add a gateway to the current configuration, use the `ADD GATEWAY` command. The following `GATEWAY` statement parameters are supported with this command.

- TYPE
- LMID
- PORT
- ACCOUNT
- PASSWORD
- HOSTADDR
- MAXMSGLEN
- SESSIONS
- MULTIPLEX

The example in Listing 4-10 uses the `ADD GATEWAY` command.

Listing 4-10 `ADD GATEWAY` Command Example

```
ADD GATEWAY, TYPE=LOCAL,  
    LMID=IMS1,  
    PORT=4500,  
    ACCOUNT=ACCT101,  
    PASSWORD=PSWD101,  
    HOSTADDR=128.0.0.1,  
    MAXMSGLEN=2048,  
    SESSIONS=5,
```

4 Configuring BEA eLink TCP for IMS

```
MULTIPLEX=3
```

Starting a Gateway

To start a gateway, use the `START GATEWAY` command with the `LMID` parameter. The example in Listing 4-11 starts the OS3901 gateway.

Listing 4-11 `START GATEWAY` Command Example

```
START GATEWAY, LMID=OS3901
```

Stopping a Gateway

To stop a gateway, use the `STOP GATEWAY` command with the `LMID` parameter. Use the `IMMED=YES` parameter to stop the gateway immediately *without* waiting for requests in progress to complete. Outstanding requests are sent an error message and all gateway connections are closed. Use the `IMMED=NO` to allow the gateway to complete transaction processing and then shutdown. While waiting, the gateway does not accept any new requests. `IMMED=NO` is the default. The example in Listing 4-12 stops the OS3901 gateway after transaction processing completes.

Listing 4-12 `STOP GATEWAY` Command Example

```
STOP GATEWAY, LMID=OS3901
```

Modifying an Existing Gateway

To change an existing gateway, stop the gateway first. Then use the `MOD GATEWAY` command to change any supported parameter definitions. You must specify an existing `LMID` for the `MOD GATEWAY` command. The following `GATEWAY` statement parameters are supported with this command.

- LMID
- ACCOUNT
- PASSWORD
- MAXMSGLEN
- SESSIONS

The example in Listing 4-13 uses the `MOD GATEWAY` command.

Listing 4-13 `MOD GATEWAY` Command Example

```
MOD GATEWAY, LMID=SUN2, ACCOUNT=ACCT102, PASSWORD=PSWD102,  
MAXMSGLEN=2048
```

Changing IMS Host Configurations

You can dynamically change IMS host configurations by adding a new host, modifying an existing host definition, starting a host, and stopping a host. For parameter descriptions and syntax, refer to the [“Description of GATEWAY TYPE=LOCAL Parameters”](#) and [“Description of GATEWAY TYPE=REMOTE Parameters”](#) sections.

Adding a Host

To add a host to the current configuration, use the `ADD HOST` command. The following `HOST TYPE=IMS` statement parameters are supported with this command.

- TYPE
- IMSID
- OTMANAME
- OTMAINPIPE
- OTMAOUTPIPE

- NUMTPIPES

The example in Listing 4-14 uses the `ADD HOST` command.

Listing 4-14 `ADD HOST` Command Example

```
ADD HOST, TYPE=IMS,
      IMSID=IMS3,
      OTMANAME=IMS61SYS3,
      OTMAINPIPE=SYS3,
      OTMAOUTPIPE=SYS3OUT,
      NUMTPIPES=5
```

Starting a Host

To start a host, use the `START HOST` command with the `IMSID` parameter. The example in Listing 4-15 starts the `IMS3` host.

Listing 4-15 `START HOST` Command Example

```
START HOST, IMSID=IMS3
```

Modifying an Existing Host

The following tasks explain how to change an existing host.

1. Stop the host first.
2. Then use the `MOD HOST` command to change any supported parameter definitions. You must specify an `IMSID` for the `MOD HOST` command.

The following `HOST` statement parameters are modifiable with this command.

- OTMAINPIPE
- OTMAOUTPIPE

■ NUMPIPES

The example in Listing 4-16 uses the `MOD HOST` command.

Listing 4-16 MOD HOST Command Example

```
MOD HOST, IMSID=IMS3, NUMPIPES=6
```

Stopping a Host

To stop a host, use the `STOP HOST` command with the `IMSID` parameter with a value matching the `IMSID` specification on the `HOST, TYPE=IMS` statement. Refer to the [“Defining the Local Gateway”](#) section. Use the `IMMED=YES` parameter to stop the host immediately *without* waiting for requests in progress to complete. Outstanding requests are sent an error message. (For a description of error messages, refer to the [“Error and Informational Messages”](#) section.) Use the `IMMED=NO` to allow the host to complete processing outstanding requests and then shutdown. While waiting, the host does not accept any new requests. `IMMED=NO` is the default. The example in Listing 4-17 stops the IMS3 host after transaction processing completes.

Listing 4-17 STOP HOST Command Example

```
STOP HOST, IMSID=IMS3
```

Changing Remote Service Definitions

You can dynamically change remote service definitions by adding a new service, modifying an existing service definition, or starting and stopping services. Existing services must be stopped prior to modifying them. For parameter descriptions and syntax, refer to the [“Description of SERVICE TYPE=REMOTE Parameters”](#) section.

Adding a Remote Service

To add a remote service to the current configuration, use the `ADD SERVICE` command. The following `SERVICE TYPE=REMOTE` statement parameters are supported with this command. `TYPE`, `LMID`, and `NAME` are required parameters.

- `TYPE`
- `LMID`
- `NAME`
- `TIMEOUT`

The example in Listing 4-18 uses the `ADD SERVICE` command to add a remote service.

Listing 4-18 `ADD SERVICE` Command (RemoteService) Example

```
ADD SERVICE, TYPE=REMOTE,
    LMID=TUX1,
    NAME=( ECHO, ECHO ),
    TIMEOUT=60
```

Stopping a Remote Service

To stop a remote service, use the `STOP SERVICE` command with the `LMID` and `NAME` parameters. In this case, the `NAME` parameter only specifies the *local name*, rather than the fully qualified (*local name, remote name*) syntax. Use the `IMMED=YES` parameter to stop the service immediately *without* waiting for requests in progress to complete. Outstanding requests are sent an error message. Use the `IMMED=NO` to allow the requests to complete processing. While waiting, the gateway does not accept any new requests. `IMMED=NO` is the default. The example in Listing 4-19 stops `ECHO` on the `TUX1` gateway.

Listing 4-19 `STOP SERVICE` Command (Remote Service) Example

```
STOP SERVICE,
    LMID=TUX1,
```

```
NAME=ECHO
```

Modifying an Existing Remote Service

The following tasks explain how to change an existing remote service definition.

1. Stop the service first.
2. Then use the `MOD SERVICE` command to change the service information. Use the `LMID` and `NAME` parameters to specify the service you want to modify. The `TIMEOUT` parameter in the `SERVICE TYPE=REMOTE` statement is the only parameter that can be modified with this command and is required.

The example in Listing 4-20 uses the `MOD SERVICE` command. In this case, the `NAME` parameter only specifies the *local name*, rather than the fully qualified (*local name, remote name*) syntax.

Listing 4-20 `MOD SERVICE` Command (Remote Service) Example

```
MOD SERVICE,  
  LMID=TUX1,  
  NAME=ECHO,  
  TIMEOUT=60
```

Starting a Remote Service

To start a service, use the `START SERVICE` command with the `LMID` and `NAME` parameters. In this case, the `NAME` parameter only specifies the *local name* rather than the fully qualified (*local name, remote name*) syntax. The example in Listing 4-21 starts the `TUX1` service.

Listing 4-21 `START SERVICE` Command (Remote Service) Example

```
START SERVICE,  
  LMID=TUX1,
```

```
NAME=ECHO
```

Changing Local Service Definitions

You can dynamically change local service definitions by adding a new service, modifying an existing service definition, or starting and stopping services. Existing services must be stopped prior to modifying them. For parameter descriptions and syntax, refer to the “[Defining Local Services](#)” section.

Adding a Local Service

To add a local service to the current configuration, use the `ADD SERVICE` command. The `TYPE`, `IMSID`, and `NAME` are required parameters when adding a local service. The following `SERVICE TYPE=LOCAL` statement parameters are supported with this command.

- `TYPE`
- `IMSID`
- `NAME`
- `SECURITY`
- `FASTPATH`
- `V21COMPAT`

The example in Listing 4-22 uses the `ADD SERVICE` command.

Listing 4-22 `ADD SERVICE` Command (Local Service) Example

```
ADD SERVICE, TYPE=LOCAL,
    IMSID=IMS1,
    NAME=( BEASVR01 , ECHO ) ,
    SECURITY=YES,
    FASTPATH=NO,
    V21COMPAT=NO
```

Stopping a Local Service

To stop a local service, use the `STOP SERVICE` command with the `IMSID` and `NAME` parameters. In this case, the `NAME` parameter only specifies the *remote name*, rather than the fully qualified (*local name, remote name*) syntax. Use the `IMMED=YES` parameter to stop the service immediately *without* waiting for requests in progress to complete. Outstanding requests are sent an error message. Use the `IMMED=NO` to allow the requests to complete processing. While waiting, the gateway does not accept any new requests. `IMMED=NO` is the default. The example in Listing 4-23 stops `ECHO` on the `IMS1` host.

Listing 4-23 `STOP SERVICE` Command (Local Service) Example

```
STOP SERVICE, IMSID=IMS1, NAME=ECHO
```

Modifying an Existing Local Service

The following tasks explain how to change an existing local service definition.

1. Stop the service first.
2. Then use the `MOD SERVICE` command to change any supported parameters.

The following `SERVICE TYPE=LOCAL` statement parameters can be modified with this command.

- `SECURITY`
- `FASTPATH`
- `V21COMPAT`

4 Configuring BEA eLink TCP for IMS

The example in Listing 4-24 uses the `MOD SERVICE` command. In this case, the `NAME` parameter only specifies the *remote name*, rather than the fully qualified (*local name, remote name*) syntax.

Listing 4-24 MOD SERVICE Command (Local Service) Example

```
MOD SERVICE, IMSID=IMS1, NAME=ECHO, SECURITY=YES
```

Starting a Local Service

To start a local service, use the `START SERVICE` command with the `IMSID` and `NAME` parameters. In this case, the `NAME` parameter only specifies the *remote name*, rather than the fully qualified (*local name, remote name*) syntax. The example in Listing 4-25 starts the `ECHO` service for `IMS1`.

Listing 4-25 START SERVICE Command (Local Service) Example

```
START SERVICE, IMSID=IMS1, NAME=ECHO
```

Display Status Information

Use the `STATUS` command to display the status of the eLink TCP gateway. The command displays XCF status with all IMS hosts, the number of messages processed, the number of XCF messages received, message control block usage, and service counts. The following status codes for IMS hosts can result:

- ACTIVE
- ERROR
- RESTART

- BID SENT
- UNKNOWN

Listing 4-26 STATUS Command Response Example

```
BEA2128I      XCF Status with IMS Client IMS61CR1 = Active
BEA2128I      XCF Status with IMS Client IMS61CR2 = Active
BEA2184I      XCF Messages Received = 4
BEA2183I      Messages Processed = 0
BEA2185I      MCBs: Allocated = 771 Assigned = 0
BEA2186I      Services: Remote = 7 Local = 6
```

5 Monitoring BEA eLink TCP for IMS

The following topics explain operating and monitoring the BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) product:

- [Testing Connectivity with Remote Systems](#)
- [Operating BEA eLink TCP for IMS](#)
- [The BEA eLink TCP for IMS Message Log](#)
- [The Server Response Log File](#)

Testing Connectivity with Remote Systems

When you start eLink TCP for IMS for the first time, you should do so in a controlled environment, using a small, simple configuration that allows you to methodically test the system to verify your installation and the configuration.

An appropriate test configuration would include the IMS gateway (one port) and one remote gateway. The `SOURCE` distribution library includes several sample IMS client and server transactions that can be used to test connectivity with a remote BEA eLink gateway.

By specifying a minimum number (greater than 0) of outbound sessions for a remote gateway, you can cause outbound sessions to be automatically established with that gateway during initialization. This can help verify outbound connectivity without having to make use of test transactions.

Execute test client and server transactions from both sides of the configuration to verify bidirectional connectivity. If errors are encountered, use the diagnostic messages issued by *both* sides of the configuration (that is, the IMS gateway *and* the remote BEA Tuxedo gateway) to identify and correct the problem.

For a description of the eLink TCP for IMS error messages, refer to the [“Error and Informational Messages”](#) section.

Monitoring and Troubleshooting OTMA

The following commands are a few of the available IMS commands that may assist you in monitoring or troubleshooting problems with the OTMA connection. For definitions and command syntax for these IMS commands, refer to the *IMS/ESA Operator’s Reference* or the *Open Transaction Manager Access Guide*.

- `/SEC OTMA security-level`

Specify `FULL` for *security-level* to check security on all messages.

Specify `PROFILE` for *security-level* to check security only when requested.

Specify `NONE` for *security-level* to disable security checking.
- `/DIS OTMA`

Displays the OTMA status.
- `/STO OTMA`

Stops OTMA messaging.
- `/STA OTMA`

Enables OTMA messaging.
- `/DIS TMEMBER name TPIPE ALL`

Displays the status of the client.
- `/STO TMEMBER name TPIPE ALL`

Stops an individual client from accessing a service.
- `/STA TMEMBER name TPIPE ALL`

Enables an individual client and allows access to a service.

Operating BEA eLink TCP for IMS

The eLink TCP for IMS product is started by submitting JCL. For sample JCL, refer to the [“Sample JCL and User Exits”](#) section.

During operation, eLink TCP for IMS writes all messages to the message log dataset (DDNAME=MSGLOG). The message log is primarily useful for historical purposes after eLink TCP for IMS has ended. The message log remains open for output during execution and recent messages cannot be viewed (using ISPF Browse, for example) until the dataset is closed at termination.

However, informational and error messages are also written to the OS/390 console, where they can be viewed in real time by a system operator. When MSGLEVEL=4 is specified in the configuration file (the normal mode), all messages written to the message log are also displayed on the OS/390 console.

Initialization

During initialization, the following activities occur:

- The runtime datasets (configuration file, message log, and server response log) are opened
- The configuration file is read and processed to establish the configuration
- The eLink TCP for IMS product establishes a TCP/IP Listener for each defined local gateway (port)
- If requested, outbound sessions are established with remote gateways
- The eLink TCP for IMS product puts up an outstanding write-to-operator with reply (WTOR) on the OS/390 console by means of which operator commands can be entered

If any errors are detected during initialization, eLink TCP for IMS issues the appropriate error messages and initialization fails. If initialization fails, correct the indicated errors and restart eLink TCP for IMS.

Normal Operation

If initialization is successful, normal operation begins, during which the following activities occur:

- Requests for new inbound sessions, initiated by remote systems, are accepted.
- New outbound sessions are requested and established with remote systems as required.
- The eLink TCP for IMS product monitors the TCP/IP network for requests and responses from remote systems.
- Inbound requests are received from remote systems through the TCP/IP network and are relayed to the appropriate IMS server transactions.
- IMS is polled for requests and responses from IMS client and server transactions.
- Outbound requests issued by IMS client transactions are relayed to the appropriate remote systems through the TCP/IP network.
- The eLink TCP for IMS product monitors requests-in-progress to ensure that a response (if required) is received within a specified period of time.
- Operator commands (entered through a reply to the outstanding command `WTOR`) are processed.

Operator Commands

During normal operation, eLink TCP for IMS puts up an outstanding `WTOR` (message ID `BEA2113I`) that can be used to enter commands. A command is entered by simply replying to the outstanding `WTOR` in the format in [Listing 5-1](#).

Listing 5-1 Syntax for Replying to `WTOR`

R *nn,command*

In [Listing 5-1](#), *nn* is the OS/390-assigned reply ID and *command* is the text of the command.

Issuing Session-related Messages

During normal operation, only messages for session-related events are issued. These include requests to establish a TCP/IP connection, requests to establish a session between two gateways, session disconnect, and session termination. This minimizes the amount of console traffic that occurs under normal operating conditions. Messages associated with TCP/IP connections include the 4-digit TCP/IP socket ID (immediately following the message ID).

Termination

Normal operation continues until a SHUTDOWN command is received. Under normal circumstances, eLink TCP for IMS is terminated by entering an operator SHUTDOWN command (through the outstanding command WTOR).

When a SHUTDOWN command is received and accepted, the following activities occur:

1. The eLink TCP for IMS gateway stops accepting requests for new inbound connections (from remote systems).
2. Active inbound and outbound sessions are terminated without permitting in progress transactions to complete.
3. The Message Log and Server Response Log datasets are closed.
4. The eLink TCP for IMS gateway terminates and returns control to OS/390.

SHUTDOWN Command

The SHUTDOWN command allows an operator to initiate termination of eLink TCP for IMS, and is entered in the format in [Listing 5-2](#).

Listing 5-2 Syntax for Termination Initiation

```
R nn,SHUTDOWN
```

A Client-Initiated Shutdown

The eLink TCP for IMS product can be configured to allow shutdown to be initiated by a remote client request or a response to a remote client request. This may be useful if it is necessary to shut down eLink TCP for IMS from a remote system, rather than by operator command from an OS/390 console.

Shutdown is requested by embedding a modify command into the user request data of an IMS server request, or the user response data of an IMS server response in the following format.

Format the command as shown in [Listing 5-3](#). If the *jobname* specified is incorrect, eLink TCP for IMS simply ignores the command and processes the request or response in the usual way.

Note: To use this feature, the configuration file must specify the `CLIENTSHUTDOWN=YES` option on the `SYSTEM` statement (the default is `NO`). Otherwise, eLink TCP for IMS ignores a remote client request to initiate shutdown processing.

Listing 5-3 Syntax for Client-Initiated Shutdown

```
F jobname TERM=type
```

jobname

is the OS/390 jobname assigned to eLink TCP for IMS.

TERM=*type*

is the method for shutting down the system. The values for *type* are as follows.

STOP

is a normal shutdown. If TERM=STOP is specified, eLink TCP for IMS initiates normal shutdown processing, as if an operator had entered the SHUTDOWN command from an OS/390 console.

DUMP

is an abend with a dump. If `TERM=DUMP` is specified, eLink TCP for IMS issues a U3166 abend. If a `SYSUDUMP DD` statement is included in the JCL, a standard OS/390 dump is produced.

The BEA eLink TCP for IMS Message Log

The eLink TCP for IMS product uses a message log dataset to record all messages issued. Normally, the message log (`DDNAME=MSGLOG`) is allocated to a disk dataset, but it can be allocated to another destination (such as `sysout`) if desired.

The message log is primarily intended for historical use; that is, as a means to review eLink TCP for IMS activity after-the-fact. The message log remains open for output during the entire execution of eLink TCP for IMS. Consequently, messages cannot ordinarily be viewed interactively (for example, using ISPF Browse) due to the dataset disposition and/or buffering of messages by OS/390.

The `MSGLEVEL` parameter of the `SYSTEM` statement in the configuration file controls the type of messages written to the log. Specifying a `MSGLEVEL` of 4 causes all informational and error messages to be recorded. Specifying a `MSGLEVEL` of 2 records only error messages. A `MSGLEVEL` of 0 (zero) suppresses all logging. Under normal circumstances, a `MSGLEVEL` of 4 should be specified.

You may elect to have messages appended to the existing log (thus preserving messages from previous executions of eLink TCP for IMS) by coding `DISP=MOD` in the `MSGLOG DD` statement in the JCL for eLink TCP for IMS. Alternatively, coding `DISP=OLD` or `DISP=SHR` causes the log to be overwritten, discarding any messages from a previous execution of eLink TCP for IMS.

Message Format

Each message written to the message log has the general format as in [Listing 5-4](#).

Listing 5-4 Message Log Format

```
mm-dd-yyyy hh:mm:ss msgid ssss text
```

mm-dd-yyyy

mm is the month of the year (1-12) in which the message was logged.

dd is the day of the month (1-31) on which the message was logged.

yyyy is the year that the message was logged.

hh:mm:ss

hh is the hour of the day during which the message was logged.

mm is the minute of the hour during which the message was logged.

ss is the second of the minute during which the message was logged.

Figure 5-1 *msgid*

is the message ID, in the form *BEAnnnnt*, where *nnnn* is a unique message number and *t* is the message type.

ssss

is the socket number for the TCP/IP connection with which the message is associated. If the message is not associated with a TCP/IP connection, this field is blank.

text

is the text of the message.

For additional information on messages issued by the gateway, refer to the [“Error and Informational Messages”](#) section.

OS/390 Console Messages

The eLink TCP for IMS product also logs messages to the OS/390 console so that operators can monitor the operation of eLink TCP for IMS and respond to any conditions requiring their attention.

In general, eLink TCP for IMS logs the same messages to the console that are recorded in the message log (that is, informational and error messages). However, because eLink TCP for IMS issues few informational messages during normal operation, console traffic is minimized.

The Server Response Log File

When eLink TCP for IMS receives a client request from a remote system, the request is inserted into the IMS message queue for delivery to the specified IMS server transaction. The IMS server transaction processes the request and inserts the response (if required) into the IMS message queue for delivery to eLink TCP for IMS. When the response is received, it is returned to the requesting remote system.

Each IMS server request and the associated response contains a unique request/response ID, consisting of the date and time that eLink TCP for IMS was started, and a serial number. The eLink TCP for IMS gateway uses the request/response ID to correlate each response with a pending server request.

It is possible that eLink TCP for IMS may receive a response from an IMS server transaction for which no pending request exists. This can occur under any of the following conditions:

- The eLink TCP for IMS gateway was restarted after the request was sent to IMS but before the response was received. In this case, the response is usually received when eLink TCP for IMS is subsequently restarted.
- The TCP/IP connection with the remote gateway was lost after the request was sent to IMS but before the response was received.
- The request was sent to IMS, but timed out and was aborted before the response was received from IMS.
- The IMS server transaction returned a response to a request that did not require a response.

When a response cannot be correlated with a pending request (that is, a pending request with a matching request/response ID cannot be found), eLink TCP for IMS writes the response to a server response log file (DDNAME=SVRLOG). The information in the server

response log file can be useful as part of a manual recovery procedure. Message BEA2033E is also issued, indicating that a server response has been logged and specifying the reason (“Server Request not found” or “No response was expected”).

Server responses are logged as two separate records: the BEA server request/response header (containing the unique request/response ID), and the response data.

Note: The dataset attributes of the server response Log File are fixed by architecture. Refer to the *BEA eLink TCP Installation Guide* for additional information.

6 Programming BEA eLink TCP for IMS

BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) allows user-written IMS application transactions to service requests received from remote systems (server requests) and to initiate requests for services provided by remote systems (client requests).

The following topics are programming considerations:

- [Programming Your Application](#)
- [Programming Interface](#)
- [Returning a Response](#)
- [IMS Client Requests](#)
- [Error Handling](#)
- [How BEA eLink TCP for Tuxedo Translates Data](#)

Programming Your Application

IMS client and server transactions are ordinary IMS message processing programs (MPPs) and with the use of OTMA, server transactions can also be FastPath programs. These programs may be written in any programming language supported by IMS.

Because COBOL is the traditional language used for this purpose, the examples in this document are written in COBOL. However, these examples may be translated easily to other supported programming languages such as C.

Programming Interface

The eLink TCP for IMS product and user-written IMS client and server transactions communicate by exchanging standard IMS messages (using the IMS message queue directly or through OTMA).

IMS Server Requests

An IMS server transaction is one that services requests originated by a remote system. The request message is inserted directly into the IMS message queue or indirectly through OTMA as a single segment. The IMS server transaction retrieves the request message from the IMS message queue and processes the request as required.

Message Format

When running the eLink TCP gateway as an OTMA client, the request or response message is in the standard IMS message format. The following message is a sample when using OTMA.

Listing 6-1 Sample Message using OTMA

```
01 BEA-SERVER-REQUEST-RESPONSE-MESSAGE .
05 BEA-SERVER-HEADER
   10 LL                PIC 9(4) COMP VALUE 0 .
   10 ZZ                PIC 9(4) COMP VALUE 0 .
   10 TRAN-CODE        PIC X(8) VALUE SPACES .
```

REQUEST-DATA.

Field Descriptions

The request messages contain the following fields.

Field	Description
LL	The length of the complete request or response message (including this field).
ZZ	Reserved for use by IMS; normally zero.
TRAN-CODE	For a request message, the transaction code of the IMS server transaction. For a response message, the transaction code of the eLink TCP for IMS gateway.
FILL-CHAR	Message fill character (space).

Header Files and Copybooks

The `INCLUDE` distribution library contains the following header files and copybooks for use by IMS server transactions.

Member Name	Description
BEATCPI	C Header file
SERVER	COBOL copybook

Returning a Response

If a response is required, the server transaction formats a client response message.

OTMA Client Gateway

The request message presented to the server transaction contains the name of the transaction pipe (`tpipe`) in the `LTERM` field of the I/O PCB. If the security feature is in use, the `USERID` and `GROUPID` fields of I/O PCB also contain relevant data.

If a response is required, the response message is inserted into the IMS message queue and routed back to the gateway by OTMA. The response message is inserted either as a single segment, or as multiple segments. The eLink TCP for IMS gateway can handle either format.

IMS Client Requests

An IMS client transaction is one that issues a request for a service provided by a remote system. You must install the `DFSYPRX0` and `DFSYDRU0` IMS user exits prior to using an IMS client transaction with the OTMA client gateway. Refer to the IBM document, *Open Transaction Manager Access Guide* for additional information on these user exits. Basic samples of these exits are in the [“Sample JCL and User Exits”](#) section.

Message Format

An eLink TCP for IMS client request or response message consists of three contiguous “sections”:

- BEA-defined request/response header
- User-defined context data

- User-defined request or response data

An eLink TCP for IMS client request or response has the following message format.

Listing 6-2 Client Request/Response Message Format

```

01 BEA-CLIENT-REQUEST-RESPONSE-MESSAGE.
   05 BEA-CLIENT-HEADER.
       10 LL                               PIC 9(04) COMP VALUE 0.
       10 ZZ                               PIC 9(04) COMP VALUE 0.
       10 TRAN-CODE                        PIC X(08) VALUE SPACES.
       10 FILL-CHAR                        PIC X      VALUE SPACES.
       10 BEA-RESERVED-1                   PIC X(03) VALUE LOW-VALUES.
       10 HEADER-LENGTH                    PIC 9(08) COMP VALUE 0. .
       10 CONTEXT-LENGTH                   PIC 9(08) COMP VALUE 0.
       10 DATA-LENGTH                     PIC 9(08) COMP VALUE 0.
       10 MAX-RESPONSE-LENGTH              PIC 9(08) COMP VALUE 0.
       10 REQUEST-TYPE                     PIC 9(08) COMP VALUE 0.
           88 REQUEST-RESPONSE-REQUIRED      VALUE 1.
           88 REQUEST-NO-RESPONSE           VALUE 2.
       10 RESPONSE-FORMAT                  PIC 9(08) COMP VALUE 0.
           88 RESPONSE-SINGLE-SEGMENT        VALUE 0.
           88 RESPONSE-MULTI-SEGMENT        VALUE 1.
       10 ERROR-CODE                       PIC S9(08) COMP VALUE +0.
           88 ERR_NO_ERROR                   VALUE +0.
           88 ERR_RESPONSE_TRUNCATED        VALUE +4.
           88 ERR_HOST_NOT_AVAILABLE        VALUE +8.
           88 ERR_HOST_BUSY                  VALUE 12.
           88 ERR_TIMEOUT                     VALUE 16.
           88 ERR_SERVICE_NAME               VALUE 20.
           88 ERR_NO_STORAGE                 VALUE 24.
           88 ERR_REQUEST_FORMAT             VALUE 28.
           88 ERR_HOST_ERROR                 VALUE 32.
           88 ERR_TP_ERROR                   VALUE 36.
           88 ERR_INTERNAL_ERROR             VALUE 40.
       10 REASON-CODE                      PIC S9(08) COMP VALUE +0.
           88 RSN_NO_REASON                  VALUE 0.
           88 RSN_HEADER_LENGTH              VALUE 4.
           88 RSN_HEADER_LENGTHS             VALUE 8.
           88 RSN_REQUEST_TYPE               VALUE 12.
           88 RSN_NO_RESPONSE_TRAN          VALUE 16.
           88 RSN_CONTEXT_LENGTH             VALUE 20.
           88 RSN_DATA_LENGTH                VALUE 24.
           88 RSN_MAX_DATA_LENGTH            VALUE 28.
           88 RSN_NO_SERVICE_NAME            VALUE 32.
           88 RSN_SERVICE_NAME_UNDEFINED     VALUE 36.
       10 SERVICE-NAME                     PIC X(16) VALUE SPACES.

```

```

10 RESPONSE-TRAN          PIC X(08) VALUE SPACES.
10 ORIGIN-TERMINAL        PIC X(08) VALUE SPACES.
10 BEA-RESERVED-2         PIC X(16) VALUE LOW-VALUES.
05 USER-CONTEXT-DATA.
   10 (user-defined...) |
05 USER-REQUEST-DATA.
   10 (user-defined...)
05 USER-RESPONSE-DATA REDEFINES USER-REQUEST-DATA.
   10 (user-defined...)

```

Field Descriptions

The client request or response messages contain the following fields.

Field	Description
LL	The length of the complete request or response message (including this field).
ZZ	Reserved for use by IMS; normally zero.
TRAN-CODE	For a request message, the transaction code of the eLink TCP for IMS gateway. For a response message, the transaction code of the IMS client response transaction.
FILL-CHAR	Message fill character (space).
BEA-RESERVED-1	Reserved for use by eLink TCP for IMS. For a request message, this field <i>must</i> be initialized to binary zero (low values).
HEADER-LENGTH	The length of the BEA request/response header.
CONTEXT-LENGTH	The length of user context data, if any. If there is no context data, the length is set to zero.
DATA-LENGTH	The length of user request or response data, if any. If there is no request or response data, the length is set to zero.

Field	Description
MAX-RESPONSE-LENGTH	<p>For a request, the maximum length of user response data that can be accepted. If the response data exceeds this length, it is truncated as necessary and appropriate return codes are set.</p>
REQUEST-TYPE	<p>For a request, indicates whether or not a response is required.</p> <p>If no response is required, eLink TCP for IMS schedules the specified response transaction as soon as the request has been sent to the remote system.</p> <p>Otherwise, the response transaction is scheduled when the required response has been received from the remote system.</p>
RESPONSE-FORMAT	<p>For a request, specifies the type or response message format desired.</p> <p>If the single segment option is specified (the default), eLink TCP for IMS inserts the complete response message into the IMS message queue as a single message segment.</p> <p>If the multi-segment option is requested, the response message is inserted into the IMS message queue in three separate segments.</p> <ol style="list-style-type: none"> 1. response header 2. user context data 3. user response data
ERROR-CODE	<p>This field is set in the response message to indicate that an error occurred. If this field is zero, the request successfully processed. Otherwise, the error code indicates the nature of the error that occurred.</p> <p>Error codes are documented in the “Error and Reason Codes” section.</p>
REASON-CODE	<p>For certain kinds of errors, this field may be set to provide additional information about the error. If this field is zero, no additional information is available.</p> <p>Reason Codes are documented in the “Error and Reason Codes” section.</p>

Field	Description
SERVICE-NAME	Specifies the name of the service (as locally defined to eLink TCP for IMS) that processes the request. Note that the local service name may not be the same as the remote system name.
RESPONSE-TRAN	The name of the transaction (IMS transaction code) that is to be scheduled to process the response to a request. A response transaction <i>must</i> be specified for <i>all requests</i> , even if an application-level response is not required.
ORIGIN-TERMINAL	The name of the terminal (IMS LTERM) associated with this request. For example, if the request is issued by a transaction processing a terminal message, this name is the name of the associated terminal.
BEA-RESERVED-2	Reserved for use by eLink TCP for IMS. For a request message, this field <i>must</i> be initialized to binary zero (low values).
USER-CONTEXT-DATA	This user-defined data area is preserved by eLink TCP for IMS and returned in the response message. This data can be used for any purpose; for example, the response transaction can use this data to correlate the response with the original request. The context data, if present, must immediately follow the BEA request header with no intervening (slack) bytes.
USER-REQUEST-DATA	For a request message, user-defined data specific to the request. The request data, if present, must immediately follow the context data (if any), with no intervening (slack) bytes.
USER-RESPONSE-DATA	For a response message, the user-defined data specific to the response. The response data, if any, must immediately follow the context data (if any) with no intervening (slack) bytes.

Header Files and Copybooks

The `INCLUDE` distribution library contains the following header files and copybooks for use by IMS client transactions.

Member Name	Description
BEATCPI	C Header file
SERVER	COBOL copybook

Error Handling

Every client request *must* designate a response transaction (even though the request itself may not require an application-level response). The eLink TCP for IMS product schedules the response transaction if an error is encountered that prevents processing of the request (for example, the request may be improperly formatted, or the specified service may be unavailable).

The response transaction should always check the error code to determine whether the request was successfully processed.

Types of Errors

The eLink TCP for IMS product recognizes a wide variety of error conditions that may prevent successful processing of a client request. These are briefly summarized in the following sections.

Invalid Request Format

The eLink TCP for IMS gateway validates a client request prior to sending it to a remote system for processing. Types of validation errors include invalid service name, invalid length values, etc.

Remote System Not Available

The eLink TCP for IMS gateway may be unable to process a request because the required remote system is either inaccessible or currently busy processing other requests.

Request Timeout

The eLink TCP for IMS gateway monitors each request sent to a remote system for processing. If a response is required and is not received within a prescribed period of time, the request *times out*.

Remote System Errors

The request may fail due to an error that occurs on the remote system. For example, the requested service may detect an application-level error, a security authorization failure, or the remote gateway may be unable to invoke the requested service.

Other Errors

Although unlikely, it is possible that eLink TCP for IMS may be unable to process a request due to an internal or system error, such as a temporary storage shortage.

How BEA eLink TCP for Tuxedo Translates Data

When a client program on the remote BEA Tuxedo system sends data to (or receives data from) a service routine on a different model of computer, eLink TCP for Tuxedo automatically translates data as required. Translation involves changing the representation of intrinsic data types by changing attributes such as the following:

- Word length
- Byte order

BEA Tuxedo Buffer Types

The following table contains definitions for buffer types.

Term	Definition
STRING	A buffer of character data that is terminated by the first null character in the buffer. Typically these character string buffers undergo translation when sent to a system that is dissimilar from the sending system.
CARRAY	A CARRAY is a buffer of raw data that contains no terminating character. The data in the array undergoes no conversion or translation; it is sent from one system to another without modification. (A CARRAY is an exemplary buffer type for a graphics file.)
VIEW	A VIEW buffer is a collection of field definitions that can be treated as a single entity. It is comparable to a record layout in COBOL or a structure in C.
FML	Fielded manipulation language (FML) buffers are variable length, dynamic, self-describing buffers. Each field in the buffer has its own descriptive header. In the BEA Tuxedo product, FML buffers can be tied closely to VIEW buffers so that conversion from one to the other is direct.

Data Translation Rules

The eLink TCP for Tuxedo gateway adheres to the following data translation rules:

- CARRAY fields are passed untranslated as sequences of bytes.
- STRING and CHAR fields undergo ASCII to EBCDIC translation (if needed).
- SHORT and LONG fields are translated to S9(4) COMP and S9(9) COMP, respectively.
- FLOAT and DOUBLE fields are translated to COMP-1 and COMP-2, respectively.

Note: The BEA Tuxedo product provides a field type named `dec_t` that supports decimal values within `VIEWS`. The eLink TCP for IMS gateway translates these fields into machine independent representations of packed decimals. For example, `dec_t(m,n)` becomes `S9(2*m-(n+1))V9(n) COMP-3`. Therefore, a decimal field with a size of 8,5 corresponds to `S9(10)V9(5) COMP-3`.

The following table summarizes the rules for translating between C and IBM 370 data types.

Remote Data Type	Description	View Field Type/Length
PIC X(n)	Alphanumeric characters	string / n
PIC X	Single alphanumeric character	char
PIC X(n)	Raw bytes	carray / n
PIC 9	Single numeric byte	carray / 1
PIC S9(4) COMP	32-bit integer	short
PIC S9(9) COMP	64-bit integer	long
COMP-1	Single-precision floating point	float
COMP-2	Double-precision floating point	double
PIC S9((m+(n+1))/2)V9(n)) COMP-3	Packed decimal	dec_t / m,n

Strings and Numeric Data: A Closer Look

This section provides the following information:

- Suggestions that help you develop `VIEW` definitions for input and output buffers and records
- Explanations of how eLink TCP for Tuxedo treats string data and numeric data

Including NULL Characters in String Length Calculations

When you create `VIEW` definitions for input and output buffers that are used by IMS applications, do not specify extra characters for terminating NULL characters that are used in string fields.

For example, when a remote IMS application program expects 10 characters in an input record, specify *10* for that field, not 10 plus 1 (for the terminating NULL character).

Note: Although eLink TCP for Tuxedo does not require strings to be NULL-terminated, it respects NULL termination. Therefore, when eLink TCP for Tuxedo detects a NULL (zero) character within a string, it does not process any subsequent characters. To pass full 8-bit data that contains embedded NULL values, use a `CARRAY` type field or buffer.

The character set translations performed by eLink TCP for Tuxedo are fully localizable, in accordance with the *X/Open XPG Portability Guides*. ASCII and EBCDIC translations are loadable from message files. The eLink TCP for Tuxedo product contains default behaviors that should meet the requirements of most English-language applications. However, you may find it necessary to customize tables. For complete instructions, see the *BEA eLink TCP for Tuxedo User Guide*.

Converting Numeric Data

You can convert numeric data easily into different data types, provided that you have enough range in the intermediate and destination types to handle the maximum value you need to represent.

For example, you can convert an FML field of double into a packed decimal field on the remote target system by specifying an appropriate `dec_t` type `VIEW` element.

In addition, you can convert numeric values into strings (and the reverse). For example, while FML buffers do not directly support the `dec_t` type, you can place decimal values in string fields and map these to `dec_t` fields within `VIEW` definitions.

A System Services Descriptions

BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) uses the following system services:

- TCP/IP for OS/390
- LE/370
- OS/390

TCP/IP for OS/390

The eLink TCP for IMS product makes use of the following TCP/IP for OS/390 services by issuing the appropriate API calls.

TCP/IP for OS/390 Service	Description
accept	Accept a TCP/IP connection request
bind	Bind a socket to an IP address and port
close	Close a socket and free associated resources
connect	Initiate a TCP/IP connection with a specified IP address and port
fcntl	Set the operational characteristics of a socket

TCP/IP for OS/390 Service	Description
gethostbyaddr	Resolve an IP address into a domain name
gethostbyname	Resolve a hostname into a domain name
htonl	Convert a long integer from host to network byte order
htons	Convert a short integer from host to network byte order
inet_network	Convert an IP address from character format to internal format
inet_ntoa	Convert an IP address from internal format to character format
listen	Initiate listening on a socket bound to an IP address and port
read	Read data on a TCP/IP connection
selectex	Wait on one or more socket FDs to be posted, a timeout to occur, or an ECB to be posted
setsockopt	Set socket-related processing options
socket	Obtain (allocate) a socket descriptor for a TCP/IP connection
shutdown	Shut down a TCP/IP connection
write	Write data to a TCP/IP connection

LE/370

The eLink TCP for IMS product makes use of the following LE/370 services by issuing the appropriate API calls.

LE/370 Service	Description
CEE3ABD	Issue an abend

OS/390

The eLink TCP for IMS product makes use of the following OS/390 services by issuing the appropriate API calls.

OS/390 Service	Description
WTO	Issue a write-to-operator (WTO) message
WTOR	Issue a write-to-operator-with-reply (WTOR) message
XCF	Uses cross-system coupling facility to communicate to OTMA
Latch Manager Services	Locks storage resources to prevent simultaneous updates of shared memory

B Error and Informational Messages

All messages issued by BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) have an IBM-standard 8-character message ID beginning with the characters, "BEA". Messages have the following general format: *BEAnnnnt ssss text*.

Message Part	Description
<i>nnnn</i>	A unique message number in the range 2000 - 2500.
<i>t</i>	The message type: I=Informational E=Error W=Warning
<i>ssss</i>	The socket number identifying the TCP/IP connection with which the message is associated. If the message is not associated with a specific TCP/IP connection, this field is blank.
<i>text</i>	The actual text of the message.

Logical Machine IDs

A logical machine ID (LMID) is a symbolic name used to identify a BEA eLink gateway. LMIDs are defined in the BEA eLink TCP for IMS configuration file, and appear in selected messages issued by BEA eLink TCP for IMS.

When used in a message to identify a gateway, the logical machine ID displays as follows:

LMID (*logical-machine-id*, *type*)

logical-machine-id

is the logical machine ID (1-16 characters)

type

is the type of gateway: L is the local gateway; R is a remote gateway

Note: The logical machine IDs are defined in the configuration file for BEA eLink TCP for IMS.

The terms *local* and *remote*, as used in this document, are relative to the BEA eLink TCP for IMS gateway. *Local* refers to the IMS gateway; *remote* refers to another BEA eLink gateway with which the IMS gateway is communicating.

The following are examples of LMIDs used in messages issued by BEA eLink TCP for IMS:

- ◆ LMID(OS3901,L) is a local gateway named “OS3901”
- ◆ LMID(TUXC9A,R) is a remote gateway named “TUXC9A”

Messages Returned to the Remote Gateway

BEA eLink TCP for IMS returns the following messages to a remote gateway in conjunction with positive and negative responses to requests presented by a remote gateway:

- ◆ BEA2030E

- ◆ BEA2034E
- ◆ BEA2038I
- ◆ BEA2039E
- ◆ BEA2042I
- ◆ BEA2045I
- ◆ BEA2047E
- ◆ BEA2049E
- ◆ BEA2078E

These messages may also be recorded in the message log and displayed on the OS/390 console. For a complete description of each of these messages, refer to the “Messages Written to the BEA eLink TCP for IMS Message Log” section that follows.

Messages Written to the BEA eLink TCP for IMS Message Log

BEA2000I *jobname* BEA Connect TCP For IMS (*vv.rr*) has ended - RC = *rc*

Description: This message is issued during termination just before BEA eLink TCP for IMS returns to OS/390.

jobname is the OS/390 job name assigned to BEA eLink TCP for IMS.

vv is the product version number.

rr is the product release number.

rc is a return code indicating the reason for termination. The most common return code is 24, indicating that termination occurred in response to a SHUTDOWN request.

Response: BEA eLink TCP for IMS normally terminates in response to a SHUTDOWN request, in which case the return code is 24 and no special response is required.

If BEA eLink TCP for IMS terminates with a return code other than 24, termination is the result of an abnormal condition. In that case, there should be other messages that provide additional information about the reason for the termination.

BEA2001I *jobname* BEA Connect TCP For IMS (*vv.rr*) starting

Description: This message is issued when BEA eLink TCP for IMS begins execution.

jobname is the OS/390 job name assigned to BEA eLink TCP for IMS.

vv is the product version number.

rr is the product release number.

Response: None required.

BEA2002I *jobname* BEA Connect TCP For IMS (*vv.rr*) initialization complete

Description: This message is issued when BEA eLink TCP for IMS has successfully completed initialization and is ready to begin normal processing.

jobname is the OS/390 job name assigned to BEA eLink TCP for IMS.

vv is the product version number.

rr is the product release number.

Response: None required.

BEA2003I BEA Connect TCP For IMS (*vv.rr*) is terminating (RSN=*rc*)

Description: This message is issued when BEA eLink TCP for IMS begins the termination process (for example, in response to an operator SHUTDOWN command).

vv is the product version number.

rr is the product release number.

rc is a reason code indicating the reason for termination.

Response: BEA eLink TCP for IMS normally terminates in response to a SHUTDOWN request, in which case the return code is 24 and no special response is required.

If BEA eLink TCP for IMS terminates with a return code other than 24, termination is the result of an abnormal condition. In that case, there should be other messages that provide additional information about the reason for the termination.

BEA2004E Invalid IMS Client Request: *description*, Error(*error-code*), Reason(*reason-code*)

Description: An IMS client transaction has issued a request that is invalid. *description* is a text description of the error detected. *error-code* is the error code. For a description of error codes issued by BEA eLink TCP for IMS, refer to the “[Error and Reason Codes](#)” section. *reason-code* is the reason code. For a description of reason codes issued by BEA eLink TCP for IMS, refer to the “[Error and Reason Codes](#)” section.

Response: Identify and correct the indicated problem.

BEA2005E IMS Client Request for Service(*service*) failed: Requested service not defined

Description: The requested service is not defined to BEA eLink TCP for IMS. *service* is the specified service name.

Response: The specified service is not defined in the BEA eLink TCP for IMS configuration file. Ensure that the correct local service name is being specified, and that the service is properly defined in the configuration file with the specified local service name.

BEA2006E IMS Client Request for Service(*service*) failed: All outbound sessions busy

Description: All available outbound sessions with remote systems offering the service are currently in use (and no additional sessions can be started due to session limits). *service* is the specified service name.

B Error and Informational Messages

	Response:	Retry the request later. Increase the maximum number of outbound sessions defined for remote systems that offer the specified service.
BEA2007E	Client Request routing error: Service(<i>service-name</i>), RC(<i>rc</i>)	
	Description:	An internal error occurred while attempting to select a remote system and outbound session to process an IMS client request. <i>service-name</i> is the specified service name. <i>rc</i> is the return code returned by the GetServiceInfo() function.
	Response:	Report the error to the systems programmer responsible for BEA eLink TCP for IMS.
BEA2008E	Operator command unrecognized - rejected	
	Description:	An operator command has been entered that is syntactically invalid.
	Response:	Correct and re-enter the command.
BEA2009I	Operator request for SHUTDOWN accepted	
	Description:	BEA eLink TCP for IMS has recognized and accepted an operator request to initiate shutdown processing.
	Action:	BEA eLink TCP for IMS begins the process of terminating.
	Response:	None required.
BEA2010I	Initializing IMS interface	
	Description:	This message is issued during initialization to indicate that the interface with IMS (i.e., via the IMS message queue) is being initialized.
	Response:	None required.

BEA2011E	Abnormal return from IMS <i>function</i> request - Status = <i>ss</i>
Description:	An IMS API call has returned an abnormal or unexpected status code. <i>function</i> is the IMS function code: GU (Get Unique) GN (Get Next) CHNG (Change) ISRT (Insert) PURG (Purge) CHKP (Checkpoint). <i>ss</i> is the status code from the program control block (PCB).
Response:	Identify the type of error and determine the reason for it. For example, the error may be the result of a temporary change in the IMS configuration.
BEA2012E	Open failed for Message Log: DDNAME(<i>ddname</i>)
Description:	BEA eLink TCP for IMS was unable to open the message log. <i>ddname</i> is the DDNAME of the message log dataset.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Verify that the correct DDNAME is coded in the JCL for BEA eLink TCP for IMS and that the message log dataset has the correct attributes.
BEA2013I	Initializing network (TCP/IP) interface
Description:	This message is issued during initialization to indicate that the TCP/IP network interface is being initialized. This includes starting a Listener session for each local gateway defined in the BEA eLink TCP for IMS configuration file.
Response:	None required.
BEA2014I	Terminating network (TCP/IP) interface
Description:	This message is issued during termination to indicate that the TCP/IP network interface is being terminated. This involves: - terminating active inbound sessions - terminating active outbound sessions - terminating Listener sessions
Response:	None required.

BEA2015I *ssss* Inbound Sessions are no longer being accepted by LMID(*lmid*,L)

Description: The Listener for the specified logical machine ID is being stopped; inbound TCP/IP connection requests are no longer accepted by this LMID.

ssss is the socket ID assigned to the Listener by TCP/IP.

lmid is the logical machine ID of the local gateway.

Response: None required.

BEA2016I Starting Listener for LMID(*lmid*,L), IP(*ip-address*), Port(*port-number*)

Description: A Listener is being started for the local gateway to accept TCP/IP connection requests from remote systems.

lmid is the logical machine ID of the local gateway.

ip-address is the IP address at which TCP/IP connection requests are accepted. If an asterisk (*) appears, connection requests are accepted through any available network interface on the machine.

port-number is the port number at which TCP/IP connection requests are accepted.

Response: None required.

BEA2017E Unable to allocate Session Control Block for Listener

Description: BEA eLink TCP for IMS was unable to allocate storage for an internal control block used to manage a Listener session.

Response: This is an abnormal condition indicating a severe storage shortage or corruption of the storage allocation control blocks. It may be necessary to terminate BEA eLink TCP for IMS (preferably with a dump) to identify and correct the problem.

BEA2018I ssss LMID(*lmid*,L) is accepting TCP/IP connection requests at IP(*ip-address*), Port(*port-number*)

Description: A Listener was successfully started for a local gateway and is ready to accept inbound TCP/IP connection requests from remote systems.

ssss is the socket ID assigned to the Listener by TCP/IP.

lmid is the logical machine ID of the local gateway.

ip-address is the IP address at which TCP/IP connection requests are accepted. If an asterisk (*) appears, connection requests are accepted through any available network interface.

port-number is the port number on which TCP/IP connection requests are accepted.

Response: None required.

BEA2019I ssss Inbound TCP/IP connection request received for LMID(*lmid*,L) from IP(*ip-address: host-name*)

Description: A TCP/IP connection request was received from a remote system for a local gateway.

ssss is the socket ID assigned to the connection by TCP/IP.

lmid is the logical machine ID of the local gateway receiving the request.

ip-address is the IP address of the remote system making the request.

host-name is the network name of the remote system making the request. TCP/IP obtains this information from a domain name server or from a local /etc/hosts file. If the name of the remote host cannot be determined, "Unknown" is displayed.

Response: None required.

BEA2020E *ssss* Inbound TCP/IP connection request rejected: Session limit (*max-sessions*) exceeded

Description: A TCP/IP connection request from a remote system was rejected because the local gateway is already operating at the specified maximum number of concurrent, inbound sessions. *ssss* is the socket ID assigned to the connection by TCP/IP. *max-sessions* is the maximum number of concurrent, active inbound sessions supported by the local gateway.

Response: If necessary, increase the maximum number of inbound sessions for the local gateway (defined by the SESSIONS parameter in the GATEWAY TYPE=LOCAL statement).

BEA2021E *ssss* Unable to allocate Session Control Block for Inbound Session

Description: BEA eLink TCP for IMS was unable to allocate storage for an internal control block used to manage an inbound session. *ssss* is the socket ID assigned to the connection by TCP/IP.

Response: This is an abnormal condition indicating a severe storage shortage or corruption of the storage allocation control blocks. It may be necessary to terminate BEA eLink TCP for IMS (preferably with a dump) to identify and correct the problem.

BEA2022I *ssss* Inbound TCP/IP connection accepted by LMID(*lmid*,L(*port*)) from R(*ip-address:port*)

Description: An inbound TCP/IP connection from a remote system has been accepted by a local gateway. *ssss* is the socket ID assigned to the connection by TCP/IP. *lmid* is the port for logical machine ID of the local gateway accepting the connection. *port* is the port for the local gateway accepting the connection. *ip-address:port* is the IP address and port for the remote system making the connection.

Response: None required.

BEA2023I	Requesting outbound TCP/IP connection with LMID(<i>lmid</i>,R) at IP(<i>ip-address</i>), Port(<i>port-number</i>)
Description:	The local gateway is attempting to establish an outbound TCP/IP connection with a remote system. <i>lmid</i> is the logical machine ID of the remote gateway. <i>ip-address</i> is the IP address of the remote gateway. <i>port-number</i> is the port number of the remote gateway.
Response:	None required.
BEA2024I	<i>ssss</i> Unable to establish TCP/IP connection with LMID(<i>lmid</i>,R)
Description:	The local gateway failed to establish an outbound TCP/IP connection with a remote system. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the remote gateway.
Response:	None required.
BEA2025E	<i>ssss</i> Received message length (<i>rcvd-length</i>) exceeds maximum allowed on session (<i>max-length</i>)
Description:	The local gateway received a gateway message whose length exceeds the maximum allowed for the session. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>rcvd-length</i> is the length of the message received. <i>max-length</i> is the maximum message length allowed on the session.
Action:	The session is terminated.

-
- Response:** This error can be caused by any of the following:
1. The configurations of the local and/or remote gateways may be incorrect or inconsistent with respect to the maximum message length allowed. The maximum message length of the local gateway is specified by the MAXMSGLEN parameter on the GATEWAY TYPE=LOCAL statement.
 2. The remote gateway may not be properly configured to connect to an IMS gateway. In particular, the remote gateway may be configured incorrectly to connect to a CICS gateway.
 3. Some other TCP/IP application (i.e., not a BEA eLink gateway) may be attempting to connect to the IP address and port number used by BEA eLink TCP for IMS. Check the configuration and verify that BEA eLink TCP for IMS is not using an IP address and port number which are also being used by another TCP/IP application.

BEA2026E *ssss* Message header length (*length*) is not valid

Description: The local gateway received a gateway message containing an invalid header length.
ssss is the socket ID assigned to the connection by TCP/IP.
length is the length of the message header received.

Action: The session is terminated.

- Response:** This error may be caused by any of the following:
1. The versions of the local and remote gateways may be incompatible.
 2. Some other TCP/IP application (i.e., not a BEA eLink gateway) may be attempting to connect to the IP address and port number used by BEA eLink TCP for IMS. Check the configuration and verify that BEA eLink TCP for IMS is not using an IP address and port number which is also being used by another TCP/IP application.
-

BEA2027E	ssss Message length (<i>length</i>) is not valid
Description:	The local gateway received a gateway message whose overall length is invalid (the total length is less than the length of the message header). <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>length</i> is the length of the gateway message received.
Action:	The session is terminated.
Response:	Contact your BEA eLink TCP for IMS system programmer.
BEA2028E	ssss IMS Client Request timed out: LMID(<i>lmid</i>,R), Service(<i>service</i>)
Description:	An IMS client request was sent to a remote system for processing, but no response was received within a specified period of time. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the remote gateway. <i>service</i> is the remote name of the service requested (i.e., the name of the service as defined on the remote system).
Action:	BEA eLink TCP for IMS aborts the request and schedules the specified IMS response transaction with the appropriate return codes.
Response:	The remote system may be experiencing problems, or may be extremely busy. An adjustment to the time out value may resolve this problem.
BEA2029E	ssss Allocation failed for buffer (size = <i>buffer-size</i>)
Description:	BEA eLink TCP for IMS is unable to allocate a buffer of the specified size for a gateway message. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP (this field is blank if there is no associated connection). <i>buffer-size</i> is the size of the requested buffer.
Action:	The session is terminated.

Response: Assuming a reasonable buffer size, this is an abnormal condition indicating a severe storage shortage or corruption of the storage allocation control blocks. It may be necessary to terminate BEA eLink TCP for IMS (preferably with a dump) to identify and correct the problem.

BEA2030E *ssss* IMS Server Request failed - Invalid IMS transaction code: Service(*service-name*)

Description: BEA eLink TCP for IMS received an abnormal return from IMS (PCB Status Code = "A1") while attempting to insert a server request message into the IMS message queue.
ssss is the socket ID assigned to the connection by TCP/IP.
service-name is the name of the requested service (i.e., the IMS transaction code for the server transaction).

Action: The request fails. This message is also returned to the remote gateway as part of a negative response to the request.

Response: The requested service name must be the (correct) IMS transaction code for the server transaction that processes the request. Check the remote configuration and verify that a valid IMS transaction name is specified as the remote service name.

BEA2031I *ssss* Shutdown requested by client transaction: Type(*type*)

Description: A client transaction requested a shutdown of the IMS gateway. Note that this request can be made either by a remote (e.g., BEA TUXEDO) client or an IMS server transaction if CLIENTSHUTDOWN is set to Yes.
ssss is the socket ID assigned to the connection by TCP/IP.
type is the type of shutdown requested: STOP - Normal shutdown DUMP - User 3166 abend with dump

Action: BEA eLink TCP for IMS initiates shutdown processing.

Response: Assuming that this is a valid request, no response is required.

BEA2032E *ssss* IMS Server Request (*size=size*) exceeds maximum supported IMS message length (*max-length*)

Description: BEA eLink TCP for IMS received a server request which is too large to be formatted into a single IMS message segment for insertion into the IMS message queue.

ssss is the socket ID assigned to the connection by TCP/IP.

size is the length of the request message received.

max-length is the maximum supported length of an IMS message segment.

Action: The request fails.

Response: Server requests are limited by the maximum supported IMS message segment length.

BEA2033E Logging IMS Server Response: *mmddyyy.hhmmss.nnnnnn* (*reason*)

Description: BEA eLink TCP for IMS is unable to correlate a response message returned by an IMS server transaction with a pending, active request. Each server request and response message is identified by a timestamp consisting of the date and time that BEA eLink TCP for IMS was started and a unique serial number:

mm month (1-12)

dd day (1-31)

yyyy year

hh hour (0-23)

mm minute (0-59)

ss second (0-59)

nnnnnn serial number

reason specifies the reason that the response is being logged.

Action: The response message is logged in the server response log dataset.

Response: If the reason is "Server Request not found", one of the following occurred:

1. BEA eLink TCP for IMS was restarted after the server request was sent to IMS but before the response was received. This message may be issued following a restart of BEA eLink TCP for IMS if server requests were in progress at the time BEA eLink TCP for IMS was terminated.
2. The TCP/IP connection with the remote gateway was lost after the server request was sent to IMS but before the response was received.
3. A server request timed out and was aborted (either by the local or the remote gateway) before the response was received from IMS.

Response: If the reason is "No response was expected", it indicates that an IMS server transaction issued a response to a request for which a response was not required. Correct the IMS server transaction. One other possible cause for this error is that BEA eLink TCP for IMS was unable to identify an IMS client request and processed it as if it were an IMS server response. This can happen if an IMS client transaction improperly initializes the BEA Client request header by failing to initialize the BEA-RESERVED1 field (BEA_Reserved1 in the C header file) to binary zero (low values).

BEA2034E *ssss Invalid or unsupported request received by LMID(*lmid*,L): Opcode(*opcode*)*

Description: BEA eLink TCP for IMS received a gateway message from a remote gateway containing an invalid or unsupported request type.

ssss is the socket ID assigned to the connection by TCP/IP.

lmid is theological machine ID of the local gateway receiving the request.

opcode is a numeric value identifying the request type.

Action: The session is terminated. This message is also sent to the remote gateway as part of a negative response to the request.

Response: The local and remote gateways may be incompatible (different versions).

BEA2035E	Unable to add service <i>service-name</i>, maximum service count reached.
Description:	The maximum number of dynamically added services has been reached.
Action:	The ADD SERVICE operator command is rejected.
Response:	The service cannot be added dynamically. The BEA eLink TCP for IMS gateway must be stopped and restarted before additional services can be added dynamically.
BEA2036I	<i>ssss</i> CONNECT request received by LMID(<i>lmid</i>,L)
Description:	BEA eLink TCP for IMS received a CONNECT request from a remote gateway. A CONNECT request is issued to establish an inbound session with the local gateway. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the local gateway receiving the request.
Response:	None required.
BEA2037I	<i>ssss</i> CONNECT request accepted by LMID(<i>lmid</i>,L)
Description:	BEA eLink TCP for IMS accepted a CONNECT request from a remote gateway. An inbound session has been established. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the local gateway receiving the request.
Response:	None required.
BEA2038I	Reserved for future use by BEA Connect TCP for IMS.
BEA2039E	<i>ssss</i> CONNECT request rejected by LMID(<i>lmid</i>,L): Not Authorized
Description:	The local gateway rejected a CONNECT request from a remote gateway because the account ID and/or password presented is not correct. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the local gateway.

B Error and Informational Messages

Action:	The session is terminated. This message is also returned to the remote gateway as part of a negative response to the CONNECT request.
Response:	Check the respective configurations of the local and remote gateways to verify that they are consistent with respect to account ID and/or password. If the local gateway definition specifies that no account ID and/or password are to be used (by coding an asterisk for the account ID and/or password), the remote gateway must present a blank (i.e., spaces) account ID and/or password.
BEA2040I <i>ssss</i> SHUTDOWN request received by LMID(<i>lmid</i> ,L)	
Description:	BEA eLink TCP for IMS received a SHUTDOWN request from a remote gateway. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the local gateway receiving the request.
Action:	BEA eLink TCP for IMS initiates termination processing.
Response:	None required.
BEA2041I <i>ssss</i> DISCONNECT request received by LMID(<i>lmid</i> ,L)	
Description:	BEA eLink TCP for IMS received a DISCONNECT request from a remote gateway. A DISCONNECT request may be sent by a remote gateway to logically terminate a session. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the local gateway receiving the request.
Action:	BEA eLink TCP for IMS terminates the inbound session with the remote gateway. Note, however, that the TCP/IP connection is not affected and remains open. The remote gateway may issue a subsequent CONNECT request to establish a new inbound session, or it may close the TCP/IP connection at its option.
Response:	None required.

BEA2042I	<i>ssss DISCONNECT request accepted by LMID(lmid,L)</i>
Description:	BEA eLink TCP for IMS accepted a DISCONNECT request received from a remote gateway. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the local gateway receiving the request.
Action:	This message is also returned to the remote gateway as part of a positive response to the DISCONNECT request.
Response:	None required.
BEA2043I	<i>Inbound (request/response) for LMID(lmid,L)</i>
Description:	A network message was received. <i>lmid</i> is the logical machine ID of the local gateway receiving the request.
Action:	None.
Response:	None required. This message is used only when MSGLEVEL is set to DEBUG(10).
BEA2044I	<i>ssss PING request received by LMID(lmid,L)</i>
Description:	BEA eLink TCP for IMS received a PING request from a remote gateway. A PING request may be sent by a remote gateway to verify that a logical session exists and that the partner gateway is operative. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the local gateway receiving the request.
Response:	None required.

BEA2045I	PING response: <i>jobname</i> BEA Connect TCP For IMS (<i>vv</i>.<i>rr</i>)
Description:	<p>This message is returned to the remote gateway by BEA eLink TCP for IMS in response to a PING request. This message is not logged.</p> <p><i>jobname</i> is the OS/390 job name assigned to BEA eLink TCP for IMS.</p> <p><i>vv</i> is the product version number.</p> <p><i>rr</i> is the product release number.</p>
Response:	None required.
BEA2046I	<i>ssss</i> COMMAND received by LMID(<i>lmid</i>,L)
Description:	<p>BEA eLink TCP for IMS has received a command from a remote gateway.</p> <p><i>ssss</i> is the socket ID assigned to the connection by TCP/IP.</p> <p><i>lmid</i> is the logical machine ID of the local gateway receiving the request.</p>
Response:	The COMMAND request is not currently supported by BEA eLink TCP for IMS and should not be sent by a remote gateway. (See ID: BEA2047E).
BEA2047E	<i>ssss</i> Command rejected by LMID(<i>lmid</i>,L): Remote commands are not supported
Description:	<p>BEA eLink TCP for IMS does not currently support commands sent by remote gateways.</p> <p><i>ssss</i> is the socket ID assigned to the connection by TCP/IP.</p> <p><i>lmid</i> is the logical machine ID of the local gateway.</p>
Action:	BEA eLink TCP for IMS ignores the request. This message is also returned to the remote gateway as part of a negative response to the COMMAND request.
Response:	The remote gateway should not be sending commands. Check the configuration and version of the remote gateway to determine why this is occurring.
BEA2048I	Allocating Buffer (size=<i>n</i>)
Description:	An internal storage buffer was allocated.

	Action:	None.
	Response:	None required. This message is used only when MSGLEVEL is set to DEBUG(10).
BEA2049E	ssss Protocol error detected by LMID(<i>imid</i>,L) on Inbound Session: Request(<i>request</i>), State(<i>state</i>)	
	Description:	<p>BEA eLink TCP for IMS received a request from a remote gateway, but the session is currently in a state which is inconsistent with the request.</p> <p><i>ssss</i> is the socket ID assigned to the connection by TCP/IP.</p> <p><i>imid</i> is the logical machine ID of the local gateway receiving the request.</p> <p><i>request</i> is the type of request received: CONNECT, DISCONNECT, etc.</p> <p><i>state</i> is a numeric value corresponding to the current state of the session, as viewed by BEA eLink TCP for IMS.</p>
	Action:	BEA eLink TCP for IMS takes no specific action. However, subsequent errors (including additional protocol errors) can occur, which likely leads to termination of the session. This message is also returned to the remote gateway as part of a negative response to the request.
	Response:	This error can occur if two gateways become "out of sync" on a session with respect to their respective states. If this error occurs frequently or persistently, notify the systems programmer responsible for BEA eLink TCP for IMS.
BEA2050I	IMS Host hostname has been added.	
	Description:	The ADD HOST operator command has completed successfully.
	Action:	None.
	Response:	None required.

BEA2051E	<i>ssss</i> Invalid or unsupported response received from LMID(<i>lmid</i>,R): Opcode(<i>opcode</i>)
Description:	BEA eLink TCP for IMS received a response message from a remote gateway on an outbound session that cannot be identified (that is, the Opcode in the BEA eLink message header is not valid). <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the remote gateway. <i>opcode</i> is the numeric value of the Opcode in the Message Header.
Action:	The session is terminated.
Response:	If this error occurs, notify the systems programmer responsible for BEA eLink TCP for IMS.
BEA2052I	<i>ssss</i> Issuing CONNECT request for LMID(<i>lmid</i>,R)
Description:	BEA eLink TCP for IMS is requesting an outbound session with a remote gateway. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the remote gateway.
Response:	None required.
BEA2053E	<i>ssss</i> CONNECT request rejected by LMID(<i>lmid</i>,R): GWError(<i>gwwerror</i>), GWMsg(<i>gwmsg</i>)
Description:	BEA eLink TCP for IMS received a negative response to a CONNECT request to establish an outbound session with a remote gateway. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the remote gateway. <i>gwwerror</i> is the gateway error code returned by the remote gateway. <i>gwmsg</i> is the gateway error message returned by the remote gateway.
Action:	The CONNECT request fails and the TCP/IP connection with the remote gateway is terminated.

Response: Refer to the documentation for the remote gateway for information about the reason for the error. Also, consult the remote gateway's log file for additional information.

BEA2054I *ssss* **CONNECT request accepted by LMID(*lmid*,R)**

Description: BEA eLink TCP for IMS has received a positive response to a CONNECT request to establish an outbound session with a remote gateway. The session is established and is ready to process requests.

ssss is the socket ID assigned to the connection by TCP/IP.
lmid is the logical machine ID of the remote gateway.

Response: None required.

BEA2055I *ssss* **Issuing DISCONNECT for LMID(*lmid*,R)**

Description: BEA eLink TCP for IMS is ending an outbound session with a remote gateway.

ssss is the socket ID assigned to the connection by TCP/IP.
lmid is the logical machine ID of the remote gateway.

Response: None required.

BEA2056E *ssss* **DISCONNECT rejected by LMID(*lmid*,R): GWError(*gwwerror*), GWMsg(*gwwmsg*)**

Description: BEA eLink TCP for IMS received a negative response to a DISCONNECT request to end an outbound session with a remote gateway.

ssss is the socket ID assigned to the connection by TCP/IP.
lmid is the logical machine ID of the remote gateway.
gwwerror is the gateway error code returned by the remote gateway.
gwwmsg is the gateway error message returned by the remote gateway.

Action: The session is terminated.

Response: Refer to the documentation for the remote gateway for information about the reason for the error. Also, consult the remote gateway's log file for additional information.

BEA2057I *ssss DISCONNECT accepted by LMID(lmid,R)*

Description: BEA eLink TCP for IMS received a positive response to a DISCONNECT request to end an outbound session with a remote gateway.
ssss is the socket ID assigned to the connection by TCP/IP.
lmid is the logical machine ID of the remote gateway.

Response: None required.

BEA2058I *ssss Sending PING request to LMID(lmid,R)*

Description: BEA eLink TCP for IMS is sending a PING request to a remote gateway.
ssss is the socket ID assigned to the connection by TCP/IP.
lmid is the logical machine ID of the remote gateway.

Response: None required.

BEA2059E *ssss PING request rejected by LMID(lmid,R): GWError(gwerror), GWMsg(gwmsg)*

Description: BEA eLink TCP for IMS received a negative response to a PING request sent to a remote gateway.
ssss is the socket ID assigned to the connection by TCP/IP.
lmid is the logical machine ID of the remote gateway.
gwerror is the gateway error code returned by the remote gateway.
gwmsg is the gateway error message returned by the remote gateway.

Action: The session is terminated.

Response: Refer to the documentation for the remote gateway for information about the reason for the error. Also, consult the remote gateway's log file for additional information.

BEA2060I	ssss PING response from LMID(<i>lmid</i>,R): response
Description:	BEA eLink TCP for IMS received a positive response to a PING request sent to a remote gateway. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the remote gateway. <i>response</i> is the text of the response message returned by the remote gateway.
Response:	None required.
BEA2061I	ssss Sending command to LMID(<i>lmid</i>,R)
Description:	BEA eLink TCP for IMS is sending a command to a remote gateway. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the remote gateway.
Response:	None required.
BEA2062E	ssss COMMAND rejected by LMID(<i>lmid</i>,R): GWError(<i>gwwerror</i>), GWMsg(<i>gwwmsg</i>)
Description:	BEA eLink TCP for IMS received a negative response to a command sent to a remote gateway. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the remote gateway. <i>gwwerror</i> is the gateway error code returned by the remote gateway. <i>gwwmsg</i> is the gateway error message returned by the remote gateway.
Action:	The session is terminated.
Response:	Refer to the documentation for the remote gateway for information about the reason for the error. Also, consult the remote gateway's log file for additional information.

BEA2063E *ssss* Unable to allocate buffer for IMS Client Request (*size=buffer-size*)

Description: BEA eLink TCP for IMS was unable to allocate storage for an internal buffer required to process an IMS client request. *ssss* is the socket ID assigned to the connection by TCP/IP. *buffer-size* is the size of the required buffer.

Response: This is an abnormal condition indicating a severe storage shortage or corruption of the storage allocation control blocks. It may be necessary to terminate BEA eLink TCP for IMS (preferably with a dump) to identify and correct the problem.

BEA2064E *ssss* IMS Client Request sent to LMID(*lmid*,R) failed: GWError(*gwwerror*), GWMsg(*gwmsg*)

Description: BEA eLink TCP for IMS received a negative response to an IMS client request sent to a remote gateway. *ssss* is the socket ID assigned to the connection by TCP/IP. *lmid* is the logical machine ID of the remote gateway. *gwwerror* is the gateway error code returned by the remote gateway. *gwmsg* is the gateway error message returned by the remote gateway.

Action: The specified IMS client response transaction is scheduled with the appropriate return codes.

Response: Refer to the documentation for the remote gateway for information about the reason for the error. Also, consult the remote gateway's log file for additional information.

BEA2065E	<i>ssss</i> State error on <i>inout</i> Session: LMID(<i>lmid,type</i>), Request(<i>request</i>), State(<i>state</i>)
Description:	<p>BEA eLink TCP for IMS attempted to execute an operation against a session which is not currently in a state consistent with the requested operation.</p> <p><i>ssss</i> is the socket ID assigned to the connection by TCP/IP.</p> <p><i>inout</i> session type: inbound or outbound.</p> <p><i>lmid</i> is the logical machine ID of the gateway.</p> <p><i>type</i> is the gateway type: L (local) or R (remote).</p> <p><i>request</i> is the request type: CONNECT, DISCONNECT, etc.</p> <p><i>state</i> describes the current state of the session.</p>
Action:	The session is terminated.
Response:	Refer this error to the systems programmer responsible for BEA eLink TCP for IMS.
BEA2066E	Unable to allocate Session Control Block for Outbound Session
Description:	BEA eLink TCP for IMS was unable to allocate storage for an internal control block required to manage an outbound session.
Response:	This is an abnormal condition indicating a severe storage shortage or corruption of the storage allocation control blocks. It may be necessary to terminate BEA eLink TCP for IMS (preferably with a dump) to identify and correct the problem.
BEA2067I	<i>ssss</i> TCP/IP connection closed by session partner
Description:	<p>BEA eLink TCP for IMS was notified that a TCP/IP connection with a remote system was closed by the session partner.</p> <p><i>ssss</i> is the socket ID assigned to the connection by TCP/IP.</p>
Action:	The session is terminated.
Response:	None required.

BEA2068I	<i>ssss</i> TCP/IP connection reset
Description:	BEA eLink TCP for IMS was notified that a connection with a remote system was reset. A reset occurs when the connection is hard-closed (i.e., closed without first issuing a TCP/IP shutdown request). <i>ssss</i> is the socket ID assigned to the connection by TCP/IP.
Action:	The session is terminated.
Response:	None required.
BEA2069E	<i>ssss</i> Length error on socket read
Description:	TCP/IP reported an illogical length read from a connection; for example, the length reported was greater than the actual length requested. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP.
Action:	The session is terminated.
Response:	This error may indicate a problem with TCP/IP for OS/390.
BEA2070E	<i>ssss</i> Abnormal return from TCP/IP API request: <i>fcn(function)</i>, <i>rc(rc)</i>, <i>errno(errno)</i>
Description:	BEA eLink TCP for IMS received an abnormal return from a TCP/IP socket operation. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>function</i> is the TCP/IP API function called: read, write, etc. <i>rc</i> is the TCP/IP return code (usually -1). <i>errno</i> is the TCP/IP errno value returned.
Action:	The session is terminated.
Response:	Consult the IBM TCP/IP for OS/390 documentation for a description of possible errno values and their meanings.

BEA2071I *ssss* Session termination in progress (RSN=*reason*)

Description: BEA eLink TCP for IMS has initiated termination of a session with a remote gateway.
ssss is the socket ID assigned to the connection by TCP/IP.
reason is a numeric value indicating the reason the session is being terminated. Some typical reason codes are: 12 A socket error occurred.
16 The connection was closed by the session partner.
20 An internal gateway request (such as CONNECT) timed out.
24 BEA eLink TCP for IMS is terminating in response to a SHUTDOWN request.
28 Session limit exceeded (a new session cannot be started).
60 A session protocol error occurred.

Response: This action typically occurs as the result of a session-related error condition (identified by the reason code), or in conjunction with a SHUTDOWN request. If a reason code other than one of those previously listed is given, the error should be reported to the systems programmer responsible for BEA eLink TCP for IMS.

BEA2072I *ssss* TCP/IP connection closed

Description: As part of the session termination process, BEA eLink TCP for IMS successfully closed the TCP/IP connection. *ssss* is the socket ID assigned to the connection by TCP/IP.

Response: None required.

BEA2073I *ssss* Session termination complete

Description: A session was successfully terminated by BEA eLink TCP for IMS.
ssss is the socket ID assigned to the connection by TCP/IP.

Response: None required.

BEA2074E Error writing to Server Response Log File

Description: BEA eLink TCP for IMS received an abnormal return attempting to write a record to the server response log file.

B Error and Informational Messages

	Response:	Check the server response log file to ensure that it is allocated with the correct dataset attributes, allocated using the correct DDNAME, and is not full or experiencing I/O errors.
BEA2075E	Open failed for server response log file: DDNAME(<i>ddname</i>)	
	Description:	BEA eLink TCP for IMS is not able to open the Server Response Log File. <i>ddname</i> is the DDNAME used by BEA eLink TCP for IMS to open the dataset.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Check the server response log file to ensure that it is allocated with the correct dataset attributes and is allocated using the correct DDNAME as shown.
BEA2076E	ssss IMS Server Request received with previous request pending (timeout assumed)	
	Description:	BEA eLink TCP for IMS received an IMS server request from a remote gateway, but a prior request is still pending (i.e., currently being serviced by IMS). <i>ssss</i> is the socket ID assigned to the connection by TCP/IP.
	Action:	BEA eLink TCP assumes that the remote gateway timed out while waiting for a response to the previous request. Therefore, the previous request (currently being serviced by IMS) is aborted (discarded) and BEA eLink TCP for IMS begins processing the new request.
	Response:	This error can occur when a remote gateway cancels a previously sent request (i.e., due to timeout) and immediately sends another request.
BEA2077E	ssss Pending IMS Server Request for Service(<i>service</i>) aborted	
	Description:	BEA eLink TCP for IMS aborted processing of the current server request. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>service</i> is the name of the IMS server transaction scheduled to process the current request.

	Action:	The current request is discarded and the session is ready to accept a new request.
	Response:	None required.
BEA2078E	ssss IMS Server Request timed out: Service(service)	
	Description:	An IMS server request requiring a response was being processed by IMS, but no response was received within a specified period of time. ssss is the socket ID assigned to the connection by TCP/IP. service is the name of the IMS server transaction scheduled to process the request.
	Action:	The request is aborted. This message is also returned to the remote gateway as part of a negative response to the request.
	Response:	Check the IMS server transaction (service) that processes the request to make sure that it is not stopped or has an unusually high backlog.
BEA2079E	Client Response failed: Sense Code=sc Reason=reason	
	Description:	A response to a client request was rejected by OTMA. sc is the sense code. reason is a numeric value indicating the reason for the response failure.
	Action:	The response is logged.
	Response:	Consult the IMS Open Transaction Manager Access Guide to determine the cause of the error.
BEA2080E	Malloc failed for parse work area	
	Description:	BEA eLink TCP for IMS was unable to allocate storage for an internal work area used to parse the configuration file.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Report this error to the systems programmer responsible for BEA eLink TCP for IMS.

BEA2081I	IOPCB after CHKP Call:
Description:	Displays the contents of the IOPCB after a CHKP DLI call.
Action:	None.
Response:	None required. This message is used only when MSGLEVEL is set to DEBUG(10).
BEA2082E	Missing LMID option for REMOTE GATEWAY
Description:	A required keyword parameter (LMID) was omitted from a GATEWAY TYPE=REMOTE statement in the configuration File.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2083E	Missing HOSTADDR option for REMOTE GATEWAY
Description:	A required keyword parameter (HOSTADDR) was omitted from a GATEWAY TYPE=REMOTE statement in the configuration file.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2084E	Missing PORT option for REMOTE GATEWAY
Description:	A required keyword parameter (PORT) was omitted from a GATEWAY TYPE=REMOTE statement in the configuration file.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2085E	NAME option is invalid for REMOTE GATEWAY
Description:	An invalid keyword parameter (NAME) was specified on a GATEWAY TYPE=REMOTE statement in the configuration file.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.

Response: Correct the problem and restart BEA eLink TCP for IMS.

BEA2086E TIMEOUT option is invalid for REMOTE GATEWAY

Description: An invalid keyword parameter (TIMEOUT) was specified on a GATEWAY TYPE=REMOTE statement in the configuration file.

Action: Initialization fails and BEA eLink TCP for IMS terminates.

Response: Correct the problem and restart BEA eLink TCP for IMS.

BEA2087E Cannot malloc for Remote Host

Description: BEA eLink TCP for IMS was unable to allocate storage for an internal control block used to manage a remote gateway.

Action: Initialization fails and BEA eLink TCP for IMS terminates.

Response: Report this error to the systems programmer responsible for BEA eLink TCP for IMS.

BEA2088E Missing LMID option for LOCAL GATEWAY

Description: A required keyword parameter (LMID) was omitted from a GATEWAY TYPE=LOCAL statement in the configuration file.

Action: Initialization fails and BEA eLink TCP for IMS terminates.

Response: Correct the problem and restart BEA eLink TCP for IMS.

BEA2089E Missing HOSTADDR option for LOCAL GATEWAY

Description: A required keyword parameter (HOSTADDR) was omitted from a GATEWAY TYPE=LOCAL statement in the configuration file.

Action: Initialization fails and BEA eLink TCP for IMS terminates.

Response: Correct the problem and restart BEA eLink TCP for IMS.

BEA2090E Missing PORT option for LOCAL GATEWAY

Description: A required keyword parameter (PORT) was omitted from a GATEWAY TYPE=LOCAL statement in the configuration file.

B Error and Informational Messages

	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2091E	Missing ACCOUNT option for LOCAL GATEWAY	
	Description:	A required keyword parameter (ACCOUNT) was omitted from a GATEWAY TYPE=LOCAL statement in the configuration file.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2092E	Missing PASSWORD option for LOCAL GATEWAY	
	Description:	A required keyword parameter (PASSWORD) was omitted from a GATEWAY TYPE=LOCAL statement in the configuration file.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2093E	NAME option is invalid for LOCAL GATEWAY	
	Description:	An invalid keyword parameter (NAME) was specified on a GATEWAY TYPE=LOCAL statement in the configuration file.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2094E	TIMEOUT option is invalid for LOCAL GATEWAY	
	Description:	An invalid keyword parameter (TIMEOUT) was specified on a GATEWAY TYPE=LOCAL statement in the configuration file.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Correct the problem and restart BEA eLink TCP for IMS.

BEA2095E	Cannot malloc for Local Host
Description:	BEA eLink TCP for IMS was unable to allocate storage for an internal control block used to manage the local gateway.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Report this error to the systems programmer responsible for BEA eLink TCP for IMS.
BEA2096E	Missing LMID option for REMOTE SERVICE
Description:	A required keyword parameter (LMID) was omitted from a SERVICE TYPE=REMOTE statement in the configuration file.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2097E	HOSTADDR option is invalid for REMOTE SERVICE
Description:	An invalid keyword parameter (HOSTADDR) was specified on a SERVICE TYPE=REMOTE statement in the configuration file.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2098E	PORT option is invalid for REMOTE SERVICE
Description:	An invalid keyword parameter (PORT) was specified on a SERVICE TYPE=REMOTE statement in the configuration file.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2099E	ACCOUNT option is invalid for REMOTE SERVICE
Description:	An invalid keyword parameter (ACCOUNT) was specified on a SERVICE TYPE=REMOTE statement in the configuration file.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.

B Error and Informational Messages

	Response: Correct the problem and restart BEA eLink TCP for IMS.
BEA2100E	PASSWORD option is invalid for REMOTE SERVICE
	Description: An invalid keyword parameter (PASSWORD) was specified on a SERVICE TYPE=REMOTE statement in the configuration file.
	Action: Initialization fails and BEA eLink TCP for IMS terminates.
	Response: Correct the problem and restart BEA eLink TCP for IMS.
BEA2101E	MAXMSGLEN option is invalid for REMOTE SERVICE
	Description: An invalid keyword parameter (MAXMSGLEN) was specified on a SERVICE TYPE=REMOTE statement in the configuration file.
	Action: Initialization fails and BEA eLink TCP for IMS terminates.
	Response: Correct the problem and restart BEA eLink TCP for IMS.
BEA2102E	SESSIONS option is invalid for REMOTE SERVICE
	Description: An invalid keyword parameter (SESSIONS) was specified on a SERVICE TYPE=REMOTE statement in the configuration file.
	Action: Initialization fails and BEA eLink TCP for IMS terminates.
	Response: Correct the problem and restart BEA eLink TCP for IMS.
BEA2103E	Missing NAME option for REMOTE SERVICE
	Description: A required keyword parameter (NAME) was omitted from a SERVICE TYPE=REMOTE statement in the configuration file.
	Action: Initialization fails and BEA eLink TCP for IMS terminates.
	Response: Correct the problem and restart BEA eLink TCP for IMS.
BEA2104E	Missing TIMEOUT option for REMOTE SERVICE
	Description: A required keyword parameter (TIMEOUT) was omitted from a SERVICE TYPE=REMOTE statement in the configuration file.

	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Correct the problem and restart BEA eLink TCP for IMS.
BEA2105E	Cannot malloc for Work Service	
	Description:	BEA eLink TCP for IMS was unable to allocate storage used to process remote service definitions.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Report this error to the systems programmer responsible for BEA eLink TCP for IMS.
BEA2106E	Cannot malloc for CurrWorkSer Info	
	Description:	BEA eLink TCP for IMS was unable to allocate storage used to process remote service definitions.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Report this error to the systems programmer responsible for BEA eLink TCP for IMS.
BEA2107E	The command lines which are in error are: <i>text</i>	
	Description:	When a syntax error is detected during processing of the configuration file, BEA eLink TCP for IMS displays the lines containing the error. <i>text</i> is the line containing the text in error.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Correct the error and restart BEA eLink TCP for IMS.
BEA2108E	Cannot malloc for ServiceKeys	
	Description:	BEA eLink TCP for IMS was unable to allocate storage used to process remote service definitions.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Report this error to the systems programmer responsible for BEA eLink TCP for IMS.

BEA2109E	Could not find LMID [<i>lmid</i>] for Service [<i>service</i>]
Description:	<p>The logical machine ID (LMID) specified on a SERVICE TYPE=REMOTE statement could not be matched with a GATEWAY TYPE=REMOTE statement specifying the same (matching) LMID in the configuration file.</p> <p><i>lmid</i> is the logical machine ID specified in the SERVICE TYPE=REMOTE statement.</p> <p><i>service</i> is the local service name specified in the SERVICE TYPE=REMOTE statement.</p>
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Each Remote Service must be associated with a particular remote gateway (identified by a unique logical machine ID) that provides the service. Check the GATEWAY TYPE=REMOTE and SERVICE TYPE=REMOTE statements to verify that they specify the same logical machine ID. Correct any errors and restart BEA eLink TCP for IMS.
BEA2110E	Syntax error in configuration file, line <i>line-no</i>
Description:	<p>A syntax error was detected during processing of the configuration file.</p> <p><i>line-no</i> is the line number of the line containing the syntax error.</p>
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Correct the error and restart BEA eLink TCP for IMS.
BEA2111E	Line in error: <i>text</i>
Description:	<p>This message is displayed in conjunction with message BEA2110E.</p> <p><i>text</i> is the line containing the text in error.</p>
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Correct the error and restart BEA eLink TCP for IMS.

BEA2112E	Open failed for configuration file: DDNAME(<i>ddname</i>)
Description:	BEA eLink TCP for IMS was unable to open the configuration file. <i>ddname</i> is the logical dataset name of the configuration file dataset.
Action:	Initialization fails and BEA eLink TCP for IMS terminates.
Response:	Verify that the correct DDNAME is coded in the JCL for BEA eLink TCP for IMS and that the configuration file dataset has the correct attributes.
BEA2113I	<i>jobname</i> BEA Connect TCP For IMS (<i>vv.rr</i>) ready
Description:	This message is issued by BEA eLink TCP for IMS in conjunction with an outstanding WTOR which can be used to enter operator commands. <i>jobname</i> is the OS/390 job name assigned to BEA eLink TCP for IMS. <i>vv</i> is the product version number. <i>rr</i> is the product release number.
Response:	None required. You can reply to the associated WTOR to enter an operator command for BEA eLink TCP for IMS.
BEA2114I	<i>ssss</i> Session TIMEOUT detected on LMID(<i>lmid,type</i>)
Description:	This message is issued by BEA eLink TCP for IMS when an IDLE TIMEOUT has occurred. <i>ssss</i> is the socket ID assigned to the connection by TCP/IP. <i>lmid</i> is the logical machine ID of the gateway. <i>type</i> is the gateway type: L (local) or R (remote).
Response:	Refer this error to the systems programmer responsible for BEA eLink TCP for IMS.

BEA2115I	Service <i>service-name</i> for IMSID <i>logical-IMS-identifier</i> has been added.
Description:	The ADD SERVICE operator command for a local service has completed successfully. <i>service-name</i> is the local service name which is passed to IMS as the transaction name. <i>logical-IMSID-identifier</i> is a symbolic name that uniquely identifies an IMS system definition. This name must match a logical IMS ID (IMSID) specified in a HOST, TYPE=IMS statement.
Action:	None.
Response:	None required.
BEA2116	Reserved for future use.
BEA2117I	IMS Host IMSID has been modified.
Description:	The MOD HOST operator command has completed successfully. IMSID is a symbolic name that uniquely identifies the IMS system definition.
Action:	None.
Response:	None required.
BEA2117I	IMS Host <i>IMSID</i> has been modified.
Description:	The MOD HOST operator command has completed successfully. <i>IMSID</i> is a symbolic name that uniquely identifies the IMS system definition.
Action:	None.
Response:	None required.
BEA2118E	IMS Host must be stopped prior to modifying.
Description:	The STOP HOST command must be issued before entering the MOD HOST command.
Action:	The MOD HOST command is aborted.

Response: Enter the STOP HOST command and then retry the MOD HOST command.

BEA2119E Missing IMSID option for MOD HOST.

Description: The IMSID parameter was omitted and is required.

Action: The MOD HOST command is aborted.

Response: Enter the IMSID parameter and then retry the MOD HOST command.

BEA2120E Allocation failed for OTMA Work Area

Description: Upon initialization of the OTMA interface, BEA eLink TCP for IMS was unable to allocate storage used to interface with OTMA.

Action: Initialization fails and BEA eLink TCP for IMS terminates.

Response: Refer this error to the systems programmer responsible for BEA eLink TCP for IMS.

BEA2121I Terminating OTMA interface

Description: During shutdown, BEA eLink TCP for IMS prepares to close its connection with OTMA.

Action: The OTMA interface is closed and its associated resources are returned to the system.

Response: None required.

BEA2122E Cannot resolve host name

Description: During configuration file processing, a hostname for a local or remote gateway could not be resolved by the domain name server or the local host file.

Action: If the error occurs during initialization, BEA eLink TCP for IMS terminates. If the error occurs during the dynamic addition of a gateway, the gateway is not added.

Response: Check the hostname to verify spelling. Contact your network administrator to verify the hostname or change the hostname if necessary.

B Error and Informational Messages

BEA2123I	Host address for <i>hostname</i> is <i>IP-address</i>
Description:	A hostname in the configuration file has been resolved to the following IP address.
Action:	None.
Response:	None required.
BEA2124E	Client Bid Failed
Description:	The initial communication between BEA eLink TCP for IMS and the IMS OTMA interface failed.
Action:	Initialization of the OTMA interface fails.
Response:	Verify that the OTMA interface for the IMS system you want to access is enabled and that the OTMAGROUP parameter defined for BEA eLink TCP for IMS matches the XCF group defined to IMS.
BEA2125I	OTMA Msg Received
Description:	Informational message only displayed when message level is set to DEBUG(10).
Action:	None
Response:	None required.
BEA2126I	OTMA Msg Sent
Description:	Informational message only displayed when message level is set to DEBUG(10).
Action:	None
Response:	None required.
BEA2127I	Client-Bid Message Sent
Description:	Informational message only displayed when message level is set to DEBUG(10).
Action:	None
Response:	None required.

BEA2128I	XCF Status with IMS Client OTMANAME=<i>status</i>
Description:	Response message to the STATUS operator command. <i>status</i> is one of the following: ACTIVE, UNKNOWN, ERROR, RESET, or BID SENT
Action:	None
Response:	None required.
BEA2129I	Processing input from file <i>filename</i>
Description:	Response message to the INFILE operator command.
Action:	None
Response:	None required.
BEA2130I	Writing Configuration to file <i>filename</i>
Description:	Response message to the CHKPT CONFIG operator command.
Action:	None
Response:	None required.
BEA2131E	Open failed for Checkpoint File: <i>filename</i>
Description:	The filename given in the CHKPT CONFIG operator command was not a valid file name or the file could not be created.
Action:	The configuration checkpoint is aborted.
Response:	Check the file name and re-enter the command. Contact the system programmer responsible for BEA eLink TCP for IMS to check for availability of DASD resources.
BEA2132I	Configuration checkpoint complete
Description:	Normal response to a CHKPT CONFIG operator command.
Action:	None
Response:	None required.

B Error and Informational Messages

BEA2133E	Error opening file <i>filename</i>
Description:	The file name entered with the INFILE operator command could not be opened.
Action:	The INFILE command processing is aborted.
Response:	Check the file name and retry if it is correct. Resolve any contention issues with other processes needing exclusive access to the file.
BEA2134E	Host LMID not Unique
Description:	The LMID for a gateway definition is not a unique name. No two gateways can share the same LMID.
Action:	If the error occurs during BEA eLink TCP for IMS startup, initialization aborts. If the error occurs while dynamically adding a gateway, the gateway is not added.
Response:	Change the LMID to a unique name.
BEA2135I	Gateway <i>lmid</i> has been added
Description:	This is the normal response to an ADD GATEWAY operator command. <i>lmid</i> is the logical machine ID specified for the ADD GATEWAY command.
Action:	None
Response:	None required.
BEA2136E	Missing LMID option for MOD GATEWAY
Description:	The LMID option is required for a MOD GATEWAY operator command.
Action:	The command action is aborted.
Response:	Add the LMID parameter to the command and retry.

BEA2137E	Gateway <i>lmid</i> is invalid
Description:	The LMID parameter entered in a MOD GATEWAY or MOD SERVICE command is not a currently defined gateway. <i>lmid</i> is the logical machine ID specified for the MOD command.
Action:	The MOD command is aborted.
Response:	Change the LMID to match the name of the defined gateway to be modified.
BEA2138E	Gateway must be stopped prior to modifying
Description:	A MOD GATEWAY operator command was issued prior to stopping the gateway with a STOP GATEWAY command.
Action:	The MOD command processing is aborted.
Response:	Issue a STOP GATEWAY command and retry the MOD GATEWAY command.
BEA2139I	Gateway <i>lmid</i> has been modified
Description:	This is the normal response to a MOD GATEWAY operator command. <i>lmid</i> is the logical machine ID specified for the MOD GATEWAY command.
Action:	None
Response:	None required.
BEA2140I	Gateway <i>lmid</i> has been started
Description:	This is the normal response to a START GATEWAY command. <i>lmid</i> is the logical machine ID specified for the START GATEWAY command.
Action:	None
Response:	None required.

BEA2141I	Gateway <i>lmid</i> is being stopped
Description:	This is the initial response to a STOP GATEWAY operator command. <i>lmid</i> is the logical machine ID specified for the STOP GATEWAY command.
Action:	All outstanding messages are allowed to continue processing unless the IMMED=YES option was specified. All sessions for the gateway are then closed.
Response:	None required.
BEA2142I	Gateway <i>lmid</i> has been stopped
Description:	The STOP GATEWAY command has completed stopping the gateway. <i>lmid</i> is the logical machine ID specified for the STOP GATEWAY command.
Action:	None
Response:	None required.
BEA2143E	Local Service Name not Unique
Description:	A service of the same name exists for the LMID associated with this service.
Action:	If the error occurs during BEA eLink TCP for IMS startup, initialization aborts and the program terminates. If the error occurs during processing of an ADD SERVICE operator command, command processing aborts.
Response:	Choose a unique name for the service and retry the command.
BEA2144I	Service <i>service-name</i> for LMID <i>lmid</i> has been added
Description:	This is the normal response to an ADD SERVICE operator command. <i>service</i> is the local service name specified in the SERVICE TYPE=REMOTE statement. <i>lmid</i> is the logical machine ID specified for the ADD SERVICE command.

	Action: None
	Response: None required.
BEA2145I	Service <i>service-name</i> for LMID <i>lmid</i> is invalid
	Description: The service name specified in the MOD SERVICE operator command does not exist, or the LMID is incorrect. <i>service</i> is the local service name specified in the SERVICE TYPE=REMOTE statement. <i>lmid</i> is the logical machine ID specified for the MOD SERVICE command.
	Action: The MOD SERVICE command processing aborts.
	Response: Correct the error and retry the command.
BEA2146E	Service must be stopped prior to modifying
	Description: A MOD SERVICE operator command was entered prior to stopping the service with a STOP SERVICE command.
	Action: The MOD SERVICE command processing aborts.
	Response: Issue a STOP SERVICE command and retry the MOD SERVICE command.
BEA2147I	Service <i>service-name</i> for LMID <i>lmid</i> has been modified
	Description: This is the normal response for a MOD SERVICE operator command. <i>service</i> is the local service name specified in the SERVICE TYPE=REMOTE statement. <i>lmid</i> is the logical machine ID specified for the MOD SERVICE command.
	Action: None
	Response: None required.
BEA2148I	System parameters have been modified
	Description: This is the normal response to a MOD SYSTEM operator command.

B Error and Informational Messages

	Action:	None
	Response:	None required.
BEA2149E	Inbound TCP/IP connection refused: MCB Error	
	Description:	A connection attempt from a remote gateway was rejected because of a shortage of message control blocks probably due to an allocation of system resource failure.
	Action:	The connection is refused.
	Response:	Refer this error to the system programmer responsible for BEA eLink TCP for IMS.
BEA2150S	Security Violation: Tran=<i>tran-id</i>, UserID=<i>userid</i>, GroupID=<i>groupid</i>, TermID=<i>terminalid</i>	
	Description:	A request to execute a transaction on behalf of the given <i>userid</i> failed authorization.
	Action:	A security error response is returned to the requestor.
	Response:	None required. The message is given for auditing purposes only.
BEA2151I	XCF Send Failed Ret=<i>return-code</i> Rsn=<i>reason-code</i>	
	Description:	An attempt to send a message to OTMA failed.
	Action:	A system error response is sent back to the requestor.
	Response:	Verify the OTMA system parameters, correct any errors, and retry. If the error persists, refer this error to the systems programmer responsible for BEA eLink TCP for IMS.
BEA2152I	Message rejected by LMID(<i>lmid</i>,L) on Inbound Session - Gateway Stopped	
	Description:	A new request was received by a gateway in a stopping condition. <i>lmid</i> is the logical machine ID specified for the MOD SERVICE command.
	Action:	A system error response is sent to the requestor.
	Response:	None required.

BEA2153I	Service <i>service-name</i> for LMID <i>lmid</i> has been started
Description:	This is the normal response for a START SERVICE operator command. <i>service-name</i> is the name of the requested service (i.e., the IMS transaction code for the server transaction). <i>lmid</i> is the logical machine ID specified for the gateway.
Action:	None
Response:	None required.
BEA2154I	Service <i>service-name</i> for LMID <i>lmid</i> is being stopped
Description:	This is the initial response to a STOP SERVICE operator command. <i>service-name</i> is the name of the requested service (i.e., the IMS transaction code for the server transaction). <i>lmid</i> is the logical machine ID specified for the gateway.
Action:	All outstanding requests are allowed to continue processing unless the IMMED=YES option was specified.
Response:	None required.
BEA2155I	Service <i>service-name</i> for LMID <i>lmid</i> has been stopped
Description:	This is the normal response to a STOP SERVICE operator command. <i>service-name</i> is the name of the requested service (i.e., the IMS transaction code for the server transaction). <i>lmid</i> is the logical machine ID specified for the gateway.
Action:	None
Response:	None required.
BEA2156I	Response received for previously timed out request
Description:	A response was received for a request for which timeout processing had already initiated.
Action:	None
Response:	None required.

B Error and Informational Messages

BEA2157I	Hostname maximum length (256) exceeded.
Description:	The hostname string used in the HOSTADDR gateway parameter is too long.
Action:	If the error occurs during BEA eLink TCP for IMS startup, initialization aborts. If the error occurs while dynamically adding a gateway, the gateway is not added.
Response:	Contact your network administrator to provide a new hostname less than 256 characters, or use the IP address in dotted decimal format.
BEA2158I	Maximum number of hostnames (256) exceeded.
Description:	More than 256 gateways were defined in the configuration using the hostname option of the HOSTADDR parameter.
Action:	If the error occurs during BEA eLink TCP for IMS startup, initialization aborts. If the error occurs while dynamically adding a gateway, the gateway is not added.
Response:	Refer this error to the systems programmer responsible for BEA eLink TCP for IMS.
BEA2159I	Inbound TCP/IP connection request rejected: Gateway Stopped
Description:	A remote gateway connection attempt to a stopped gateway was rejected.
Action:	A connection error response was returned to the remote gateway.
Response:	None required.
BEA2160E	IMS Error Message received
Description:	IMS rejected a service request for a variety of reasons.
Action:	An IMS DFS Error message was returned to the gateway on behalf of a client request.
Response:	Examine the text of the message to determine the nature of the error and take corrective action.

BEA2161E	IMS Server Request failed - IMS Unavailable
Description:	The IMS host system required to service this request is not available.
Action:	An error message is returned to the client with the appropriate error and reason codes set.
Response:	Determine why the IMS Host is unavailable and change the configuration to reroute the request to another available Host, if possible.
BEA2162I	OTMA Status change detected with status: Old=<i>oldstatus</i> New=<i>newstatus</i>
Description:	The eLink gateway detected a status change with an IMS Host.
Action:	The message is logged and displayed to the console.
Response:	Determine why the status of the IMS Host has changed.
BEA2163E	OTMAINPIPE maximum length n exceeded.
Description:	An IMS Host was configured with an OTMAINPIPE parameter.
Action:	The message is logged and displayed to the console.
Response:	Correct the configuration information and retry.
BEA2164I	Multi-Segment Message Reconstructed
Description:	A multi-segment client response was received.
Action:	None.
Response:	None required. This message is used only when MSGLEVEL is set to DEBUG(10).
BEA2165E	Not enough spare sockets available.
Description:	A dynamic configuration change was attempted that required the use of additional TCP/IP socket resources and not enough were available. Message BEA2166E should follow with additional details.
Action:	The configuration change was rejected.

	Response:	The eLink TCP for IMS gateway must be shut down to accomplish the change. Consider increasing the value of the SPARESOCKETS parameter for future dynamic updates.
BEA2166E	Requested: <i>n</i>, Available: <i>n</i>	
	Description:	A follow on message to BEA2165E. This message details the TCP/IP socket resources requested and available.
	Action:	The configuration change was rejected.
	Response:	The eLink TCP for IMS gateway must be shut down to accomplish the change. Consider increasing the value of the SPARESOCKETS parameter for future dynamic updates.
BEA2167E	Invalid Command Syntax	
	Description:	The command was entered incorrectly.
	Action:	The command was rejected.
	Response:	For correct command syntax, refer to the “Configuring BEA eLink TCP for IMS” section.
BEA2168E	IMS Server Request for Service(<i>service-name</i>) failed: Requested Service not defined	
	Description:	The service request from a remote client was for an undefined service.
	Action:	The client request is rejected with an appropriate error and reason code.
	Response:	Update the remote client to request the correct service name or define a new local service with the name requested.
BEA2169E	IMS Server Request for Service (<i>service-name</i>) failed: Requested Service not available	
	Description:	The service request from a remote client was for a service that is temporarily unavailable.
	Action:	The client request is rejected with an appropriate error and reason code.
	Response:	Determine why the service is currently unavailable and take corrective action, if necessary.

BEA2170E	IMS Server Request for Service (<i>service-name</i>) failed: Buffer Size Error
Description:	The service request from a remote client failed due to an internal buffer size error. Typically, this is caused by exceeding the maximum message size of 64K, including internal header data.
Action:	The client request is rejected with an appropriate error and reason code.
Response:	Contact BEA Support to report the error.
BEA2171E	Multiplex Count must be greater than zero (0)
Description:	A GATEWAY configuration statement contained a MULTIPLEX parameter set to zero.
Action:	The configuration is rejected. If the error occurred on initial startup, the eLink TCP for IMS gateway exits.
Response:	Change the MULTIPLEX parameter to a number greater than zero or remove the parameter and use the default (1).
BEA2172E	Missing IMSID option for IMS HOST
Description:	A HOST configuration statement was missing an IMSID parameter which is required.
Action:	The configuration is rejected. If the error occurred on initial startup, the eLink TCP for IMS gateway exits.
Response:	Correct the configuration information and retry.
BEA2173E	Missing OTMANAME option for IMS HOST
Description:	A HOST configuration statement was missing an OTMANAME parameter which is required.
Action:	The configuration is rejected. If the error occurred on initial startup, the eLink TCP for IMS gateway exits.
Response:	Correct the configuration information and retry.
BEA2174E	Missing OTMAINPIPE option for IMS HOST
Description:	A HOST configuration statement was missing an OTMAINPIPE parameter which is required.

B Error and Informational Messages

	Action:	The configuration is rejected. If the error occurred on initial startup, the eLink TCP for IMS gateway exits.
	Response:	Correct the configuration information and retry.
BEA2175E	Missing OTMAOUTPIPE option for IMS HOST	
	Description:	A HOST configuration statement was missing an OTMAOUTPIPE parameter which is required.
	Action:	The configuration is rejected. If the error occurred on initial startup, the eLink TCP for IMS gateway exits.
	Response:	Correct the configuration information and retry.
BEA2176E	Cannot malloc for IMS HOST	
	Description:	BEA eLink TCP for IMS was unable to allocate storage used to process IMS HOST definitions.
	Action:	Initialization fails and BEA eLink TCP for IMS terminates.
	Response:	Report this error to the systems programmer responsible for BEA eLink TCP for IMS.
BEA2177E	Host IMSID not Unique	
	Description:	A HOST configuration with the same IMSID already exists.
	Action:	If the error occurs during BEA eLink TCP for IMS startup, initialization is aborted and the program terminates. If the error occurs during processing of an ADD HOST operator command, command processing is aborted.
	Response:	Choose a unique name for the HOST and retry the command.
BEA2178E	Missing IMSID option for Service	
	Description:	A local service definition is missing an IMSID parameter which is required.
	Action:	If the error occurs during BEA eLink TCP for IMS startup, initialization is aborted and the program terminates. If the error occurs during processing of an ADD SERVICE operator command, command processing is aborted.

	Response: Change the SERVICE, TYPE=LOCAL statement to include an IMSID parameter.
BEA2179E	Local Service <i>service-name</i> for IMSID <i>IMS-logical-identifier</i> is invalid
	Description: A dynamic configuration operator command (MOD, START, or STOP SERVICE) was entered for a local service that does not exist.
	Action: Command processing is aborted.
	Response: Enter the correct remote name of the local service in the NAME parameter of the dynamic configuration command.
BEA2180I	Service <i>service-name</i> for IMSID <i>IMS-logical-identifier</i> is being stopped
	Description: A STOP SERVICE operator command for a local service is being processed.
	Action: The STOP processing continues for the local service.
	Response: None Required. Informational message only.
BEA2181I	Service <i>service-name</i> for IMSID <i>IMS-logical-identifier</i> has been stopped
	Description: A STOP SERVICE operator command for a local service has completed processing.
	Action: The STOP processing has completed.
	Response: None Required. Informational message only.
BEA2182E	Could not find IMSID <i>IMS-logical-identifier</i> for service <i>service-name</i>
	Description: A dynamic configuration operator command was entered for a local service which did not contain a valid IMSID parameter.
	Action: Command processing is aborted.
	Response: Enter the correct IMSID of the local service in the IMSID parameter of the dynamic configuration command.
BEA2183I	Messages Processed = <i>n</i>
	Description: Part of the STATUS display showing the number of service requests processed.

B Error and Informational Messages

	Action: None.
	Response: None Required. Informational message only.
BEA2184I	XCF Messages Received = <i>n</i>
	Description: Part of the STATUS display showing the number of XCF messages received by the gateway.
	Action: None.
	Response: None Required. Informational message only.
BEA2185I	MCBs: Allocated = <i>number of MCBs</i> Assigned = <i>number of MCBs</i>
	Description: Part of the STATUS display showing the number of message control blocks defined and currently in use.
	Action: None.
	Response: None Required. Informational message only.
BEA2186I	Services: Remote = <i>number of remote services</i> Local = <i>number of local services</i>
	Description: Part of the STATUS display showing the number of services currently defined.
	Action: None.
	Response: None Required. Informational message only.
BEA2187E	IMS HOST <i>IMS-logical-identifier</i> is invalid
	Description: A dynamic configuration operator command (MOD, START, or STOP was entered for an IMSID that does not exist.
	Action: Operator command processing is aborted.
	Response: Retry the command with a valid IMSID.
BEA2188I	IMS HOST <i>IMS-logical-identifier</i> has been started
	Description: A START HOST operator command for an IMS host has completed processing.

	Action:	None
	Response:	None Required. Informational message only.
BEA2189I	IMS HOST <i>IMS-logical-identifier</i> is being stopped	
	Description:	A STOP HOST operator command for an IMS host is being processed.
	Action:	The STOP HOST processing continues for the host.
	Response:	None Required. Informational message only.
BEA2190I	IMS HOST <i>IMS-logical-identifier</i> has been stopped	
	Description:	A STOP SERVICE operator command for an IMS host has completed processing.
	Action:	The STOP SERVICE processing has completed.
	Response:	None Required. Informational message only.
BEA2191E	Service <i>service-name</i> for IMSID <i>IMS-logical-identifier</i> already exists.	
	Description:	An ADD SERVICE operator command for a local service has been attempted but the service has been previously defined.
	Action:	The ADD SERVICE processing has been aborted.
	Response:	Change the service name and retry, if necessary,
BEA2192E	Missing LMID option for START GATEWAY	
	Description:	A START GATEWAY operator command has been attempted but the LMID parameter, which is required, has been omitted.
	Action:	The START GATEWAY processing has been aborted.
	Response:	Add the LMID parameter to the command and retry.
BEA2193E	Missing LMID option for STOP GATEWAY	
	Description:	A STOP GATEWAY operator command has been attempted but the LMID parameter, which is required, has been omitted.
	Action:	The STOP GATEWAY processing has been aborted.

	Response: Add the LMID parameter to the command and retry.
BEA2194E	Missing IMSID option for START HOST
	Description: A START HOST operator command has been attempted but the IMSID parameter, which is required, has been omitted.
	Action: The START HOST processing has been aborted.
	Response: Add the IMSID parameter to the command and retry.
BEA2195E	Missing IMSID option for STOP HOST
	Description: A STOP HOST operator command has been attempted but the IMSID parameter, which is required has, been omitted.
	Action: The STOP HOST processing has been aborted.
	Response: Add the IMSID parameter to the command and retry.
BEA2196E	OTMANAME for IMSID <i>IMS-logical-identifier</i> not unique
	Description: A HOST statement contains an OTMANAME parameter value that already is used in another HOST statement.
	Action: If the error occurs during BEA eLink TCP for IMS startup, initialization is aborted and the program terminates. If the error occurs during processing of an ADD HOST operator command, command processing is aborted.
	Response: Change the OTMANAME parameter to match the unique OTMA client name of the IMS system and retry.
BEA2197 - BEA2213	Reserved for internal trace messages used only when debugging is enabled.
BEA2214E	Response Service <i>remote-service-name</i> for Service <i>local-service-name</i> is invalid.
	Description: The specified RESPONSE service on a SERVICE, TYPE=LOCAL statement was not a valid remote service.
	Action: If the error occurs during BEA eLink TCP for IMS startup, initialization is aborted and the program terminates. If the error occurs during processing of an ADD SERVICE operator command, command processing is aborted.

	Response: Change the service name and retry.
BEA2215E	Response Transaction <i>remote-service-name</i> from IMS Client is invalid.
	Description: The response transaction specified in the BEA client request/response header is not defined as a local service.
	Action: The client request is aborted.
	Response: Define the response transaction as a local service.
BEA2216E	Client request for Service <i>remote-service-name</i> timed out on queue.
	Description: A service request from an IMS client timed out while waiting for an available session on a remote host.
	Action: A timeout error is returned to the response transaction.
	Response: Reconfigure the SESSIONS or MULTIPLEX count for the desired remote gateway, if necessary.
BEA2217I	Service <i>local-service-name</i> for IMSID <i>IMS-logical-identifier</i> has been started.
	Description: This is the normal response for a START SERVICE operator command for a local service.
	Action: None.
	Response: None Required.
BEA2218 - BEA2500	Reserved for future use by BEA Connect TCP for IMS.
BEA9999E	Unable to locate message number <number>.
	Description: An undefined error code has been generated.
	Action: The gateway is unable to describe an error which has occurred.
	Response: Contact BEA Support to report the error. Be sure to note the error number listed.

C Example Configuration Files

This document illustrates several BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) configurations, from a simple, server-only configuration to a fully-functional, bidirectional configuration. Member `CONFIGOT` in the `CONTROL` distribution library contain a sample configuration file, which you can use as a starting point for developing a configuration file tailored to your environment.

[Listing C-1](#) is a sample configuration for running eLink TCP for IMS. Every configuration file must contain one `SYSTEM` statement that is the first statement in the file. The `SYSTEM` statement contains the OTMA-specific parameters including the definition for security. In this example, `FASTPATH` is not enabled.

Listing C-1 Sample OTMA Configuration

```
#-----#
#
#       BEA CONNECT TCP FOR IMS CONFIGURATION FILE
#
#-----#

SYSTEM,                # SYSTEM PARAMETERS
  SLEEPTIME=(150,500,5), # IMS POLLING INTERVAL (MIN,MAX,DELTA) MS
  CLIENTSHUTDOWN=NO,    # DO NOT ALLOW CLIENT-REQUESTED SHUTDOWN
  MSGLEVEL=4,          # MESSAGE LOGGING LEVEL
                        # 2 = ERROR MESSAGES (ONLY)
                        # 4 = ERROR & INFORMATIONAL MESSAGES
  SPARESOCKETS=50,     # SPARE SOCKETS USED TO DYNAMICALLY ADD
                        # OR MODIFY GATEWAYS
  OTMACLIENT=BEAOTMA01, # XCF GROUP MEMBER NAME THAT CONNECT FOR
                        # IMS WILL USE
```

C Example Configuration Files

```
OTMAGROUP=BEAGROUP,          # XCF GROUP NAME
DRUEXIT=DFSYBEA0,           # DRU EXIT NAME USED BY CONNECT FOR IMS
OTMASECURITY=YES            # ENABLE SECURITY

#-----#
#           INBOUND SESSIONS (FROM REMOTE GATEWAYS TO IMS)           #
#-----#

GATEWAY,TYPE=LOCAL,         # LOCAL IMS GATEWAY DEFINITION
LMID=OS3901,                # LOGICAL MACHINE ID
HOSTADDR=111.111.111.111,   # IP ADDRESS (STANDARD DOTTED-DECIMAL)
PORT=4500,                  # PORT NUMBER (DECIMAL)
ACCOUNT=ACCT101,           # ACCOUNT ID
PASSWORD=PSWD101,          # PASSWORD
MAXMSGLEN=2048,            # MAXIMUM GATEWAY MESSAGE SIZE
SESSIONS=1                  # MAXIMUM NUMBER INBOUND SESSIONS
                           # VIA THIS IP/PORT

GATEWAY,TYPE=LOCAL,         # LOCAL IMS GATEWAY DEFINITION
LMID=OS3902,                # LOGICAL MACHINE ID
HOSTADDR=111.111.111.111,   # IP ADDRESS (STANDARD DOTTED-DECIMAL)
PORT=4501,                  # PORT NUMBER (DECIMAL)
ACCOUNT=*,                  # CONNECTION SECURITY NOT USED
PASSWORD=*,                 # BY THIS LISTENER
MAXMSGLEN=32512,           # MAXIMUM GATEWAY MESSAGE SIZE
SESSIONS=1,                 # MAXIMUM NUMBER OF INBOUND SESSIONS
MULTIPLEX=5                 # MAXIMUM NUMBER OF SIMULTANEOUS
                           # REQUESTS PER SESSION

GATEWAY,TYPE=LOCAL,         # LOCAL IMS GATEWAY DEFINITION
LMID=OS3903,                # LOGICAL MACHINE ID
HOSTADDR=111.111.111.111,   # IP ADDRESS (STANDARD DOTTED-DECIMAL)
PORT=4502,                  # PORT NUMBER (DECIMAL)
ACCOUNT=ACCT103,           # ACCOUNT ID
PASSWORD=PSWD103,          # PASSWORD
MAXMSGLEN=2048,            # MAXIMUM GATEWAY MESSAGE SIZE
SESSIONS=5                  # MAXIMUM NUMBER INBOUND SESSIONS

#-----#
#           OUTBOUND SESSIONS (FROM IMS TO REMOTE GATEWAYS)           #
#-----#

GATEWAY,TYPE=(REMOTE,TUX)   # REMOTE TPS GATEWAY DEFINITION
LMID=TUX1,                  # LOGICAL MACHINE ID
HOSTADDR=111.111.111.111,   # IP ADDRESS (STANDARD DOTTED-DECIMAL)
PORT=3333,                  # PORT NUMBER (DECIMAL)
ACCOUNT=*,                  # ACCOUNT ID (* MEANS "NOT USED")
PASSWORD=*,                 # PASSWORD (* MEANS "NOT USED")
MAXMSGLEN=2048,            # MAXIMUM GATEWAY MESSAGE SIZE
```

```

SESSIONS=(0,5), # (MIN,MAX) NUMBER OUTBOUND SESSIONS
MULTIPLEX=5 # MAXIMUM NUMBER OF SIMULTANEOUS
# REQUESTS PER SESSION

GATEWAY,TYPE=(REMOTE,IMS) # REMOTE IMS GATEWAY DEFINITION
LMID=IMSR, # LOGICAL MACHINE ID
HOSTADDR=111.111.111.111, # IP ADDRESS (STANDARD DOTTED-DECIMAL)
PORT=3333, # PORT NUMBER (DECIMAL)
ACCOUNT=*, # ACCOUNT ID (* MEANS "NOT USED")
PASSWORD=*, # PASSWORD (* MEANS "NOT USED")
MAXMSGLEN=2048, # MAXIMUM GATEWAY MESSAGE SIZE
SESSIONS=(0,5) # (MIN,MAX) NUMBER OUTBOUND SESSIONS

GATEWAY,TYPE=(REMOTE,CICS,BEAH), # REMOTE CICS GATEWAY DEFINITION
LMID=CICS1, # LOGICAL MACHINE ID
HOSTADDR=111.111.111.111, # IP ADDRESS (STANDARD DOTTED-DECIMAL)
PORT=4444, # PORT NUMBER (DECIMAL)
ACCOUNT=*, # ACCOUNT ID (* MEANS "NOT USED")
PASSWORD=*, # PASSWORD (* MEANS "NOT USED")
MAXMSGLEN=2048, # MAXIMUM GATEWAY MESSAGE SIZE
SESSIONS=(0,5) # (MIN,MAX) NUMBER OUTBOUND SESSIONS

#-----#
# IMS HOST SYSTEMS TO BE USED BY CONNECT FOR IMS #
#-----#

HOST,TYPE=IMS, # IMS HOST DEFINITION
IMSID=IMS1, # LOGICAL IMS IDENTIFIER
OTMANAME=IMS61CR1, # XCF GROUP MEMBER NAME OF TARGET IMS
OTMAINPIPE=BEAI, # TPIPE BASE NAME FOR REQUESTS OF IMS
# AS A SERVER
OTMAOUTPIPE=BEAPIPE, # TPIPE NAME FOR REQUESTS OF IMS AS A
# CLIENT
NUMTPIPES=5 # NUMBER OF INBOUND TPIPES

#-----#
# REMOTE SERVICES (CAN BE REQUESTED BY IMS CLIENTS) #
#-----#

SERVICE,TYPE=REMOTE, # REMOTE SERVICE DEFINITION
LMID=TUX1, # LOGICAL MACHINE ID
NAME=(ECHO,ECHO), # LOCAL SERVICE NAME, REMOTE SERVICE NAME
TIMEOUT=60 # PROCESSING TIME LIMIT (SECONDS)

SERVICE,TYPE=REMOTE, # REMOTE SERVICE DEFINITION
LMID=TUX1, # LOGICAL MACHINE ID
NAME=(TOUPPER,TOUPPER), # LOCAL SERVICE NAME, REMOTE SERVICE NAME
TIMEOUT=60 # PROCESSING TIME LIMIT (SECONDS)

```

C Example Configuration Files

```
SERVICE,TYPE=REMOTE,           # REMOTE SERVICE DEFINITION
  LMID=TUX1,                    # LOGICAL MACHINE ID
  NAME=(ANSWERBACK,DATALOG)     # LOCAL SERVICE NAME, REMOTE SERVICE NAME
  TIMEOUT=10                    # PROCESSING TIME LIMIT (SECONDS)

SERVICE,TYPE=REMOTE,           # REMOTE SERVICE DEFINITION
  LMID=IMSR,                    # LOGICAL MACHINE ID
  NAME=(TOUPPER,TOUPPER),       # LOCAL SERVICE NAME, REMOTE SERVICE NAME
  TIMEOUT=60                    # PROCESSING TIME LIMIT (SECONDS)

SERVICE,TYPE=REMOTE,           # REMOTE SERVICE DEFINITION
  LMID=CICS1,                   # LOGICAL MACHINE ID
  NAME=(TOUPPER,TOUPPER),       # LOCAL SERVICE NAME, REMOTE SERVICE NAME
  TIMEOUT=60                    # PROCESSING TIME LIMIT (SECONDS)

#-----#
#          LOCAL SERVICES (CAN BE REQUESTED BY REMOTE CLIENTS)          #
#-----#

SERVICE,TYPE=LOCAL,           # LOCAL SERVICE DEFINITION
  IMSID=IMS1,                   # LOGICAL IMS ID
  NAME=(BEASVR01,ECHO),         # LOCAL SERVICE NAME, REMOTE SERVICE NAME
  SECURITY=NO,                  # SERVICE LEVEL SECURITY SETTING
  FASTPATH=NO,                 # NOT A FASTPATH TRANSACTION
  V21COMPAT=YES                # MESSAGE FORMAT IS VERSION 2.1 AND
                                # EARLIER COMPATIBLE

SERVICE,TYPE=LOCAL,           # LOCAL SERVICE DEFINITION
  IMSID=IMS1,                   # LOGICAL IMS ID
  NAME=(BEASVR02,REDIRECT)      # LOCAL SERVICE NAME, REMOTE SERVICE NAME
  SECURITY=YES,                 # SERVICE LEVEL SECURITY SETTING
  FASTPATH=NO,                 # NOT A FASTPATH TRANSACTION
  V21COMPAT=NO,                # MESSAGE FORMAT IS VERSION 2.1 AND
                                # EARLIER COMPATIBLE
  RESPONSE=ANSWERBACK           # REROUTE RESPONSE TO THIS REMOTE
                                # IF THE REQUEST WAS REQUEST NO REPLY
```

D Sample JCL and User Exits

Use the following sample JCL and user exits to run the BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) product:

- [JCL for Executing eLink TCP for IMS](#)
- [Sample User Exits Enabling IMS Clients to Communicate to Remote Services](#)
 - [OTMA Prerouting User Exit \(DFSYPRX0\)](#)
 - [OTMA Destination Resolution User Exit \(DFSYDRU0\)](#)

JCL for Executing eLink TCP for IMS

The JCL distribution library contains sample JCL for executing the eLink TCP for IMS product. You must modify the sample JCL to conform to the conventions and standards for your installation. Modifications include the following:

- Adding an appropriate job card
- Changing the names of IBM product datasets (IMS, TCP/IP for OS/390, and LE/370) to match those used by your installation
- Changing the names of eLink TCP for IMS datasets to match those used by your installation
- Changing the names of certain program modules (for example, the linkage editor), if necessary, to match those used by your installation

Executing eLink TCP for IMS

[Listing D-1](#) is sample JCL for executing as an OTMA client. Use the following JCL only as a reference for modifying the standard OTMA JCL for use with the eLink TCP for IMS product.

Listing D-1 Sample Run JCL

```
//JOBNAME JOB      ETC.
//*****
//* BRING UP BEA CONNECT TCP FOR IMS/OTMA *
//*****
//STEP1      EXEC PGM=BEATCPI,REGION=0M,TIME=1440
//STEPLIB DD  DSN=hlq1.hlq2,DISP=SHR
//* Place the name of the authorized library in the STEPLIB DD above
//SYSTCPD DD  DISP=SHR,DSN=TCPIP.DATA
//MSGLOG DD   DISP=SHR,DSN=BEATCPI.MSGLOG
//MSGCAT DD   DISP=SHR,DSN=BEATCPI.MSGCAT
//SVRLOG DD   DISP=SHR,DSN=BEATCPI.SVRLOG
//CONFIG DD   DISP=SHR,DSN=hlq1.hlq2.JCL(CONFIG)
//* Place the name of the Config file in the CONFIG DD above
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
```

Sample User Exits Enabling IMS Clients to Communicate to Remote Services

To enable IMS clients to communicate to remote services through OTMA, you must install both the `DFSYPRX0` and `DFSYDRU0` IBM user exits available in IMS. Customize the exits to recognize specific transactions as well as the eLink OTMA client name as defined in the configuration file. For more information about setting up the configuration file, refer to the [“Configuring BEA eLink TCP for IMS”](#) section. Refer to the IBM document *Open Transaction Manager Access Guide* for additional information on the IBM user exits.

OTMA Prerouting User Exit (DFSYPRX0)

Listing D-2 is the sample IBM OTMA prerouting user exit which can be found in the dataset BEATCPI source. The IBM DFSYPRX0 user exit enables IMS to search for a destination for an application request. In the following sample, the lines in bold are modifications that you must make to the IBM DFSYPRX0 user exit.

Listing D-2 OTMA Pre-routing Sample User Exit (DFSYPRX0)

```
TITLE 'DFSYPRX0: OTMA PRE-ROUTING USER EXIT'
DFSYPRX0 CSECT
DFSYPRX0 AMODE 31
DFSYPRX0 RMODE ANY
*****
*
* DESCRIPTION: OTMA PRE-ROUTING USER EXIT
*
*****@SCPVRT**
*
*     Licensed Materials - Property of IBM
*
*     "Restricted Materials of IBM"
*
*     5655-158 (C) Copyright IBM Corp. 1991,1996
*
*****@ECPVRT**
*
* MODULE TYPE:  DC
*
* PROCESSOR:    ASSEMBLER H
*
* MODULE SIZE:  SEE EXTERNAL SYMBOL DICTIONARY PRODUCED BY
*               ASSEMBLER
*
* ATTRIBUTE:    REENTRANT
*
* ENTRY POINT: DFSYPRX0
*
* ENTRY INTERFACES :
*
* STATUS:       RELEASE 5.1          CHANGE LEVEL - 0
*
* FUNCTION:     THIS MODULE IS CALLED BY IMS WHEN AN APPLICATION
*               CALL (IE CHNG OR ISRT) RESULTS IN A SEARCH FOR
```


D Sample JCL and User Exits

```
*          +25      = SYNCHRONIZATION LEVEL IF THE INPUT WAS          *
*                  FROM AN OTMA CLIENT.                               *
*                                                                    *
*          +26      = 2-BYTE RESERVED                                *
*                                                                    *
*          +28      = 8-BYTE USERID                                  *
*                  (LEFT-JUSTIFIED, PADDED WITH BLANKS)             *
*                  THIS IS AN ENTRY PARAMETER ONLY.                 *
*                                                                    *
*          +36      = 8-BYTE GROUP ID                                *
*                  (LEFT-JUSTIFIED, PADDED WITH BLANKS)             *
*                  THIS IS AN ENTRY PARAMETER ONLY.                 *
*                                                                    *
*          +44      = 4-BYTE RESERVED                                *
*                                                                    *
*          +48      = 16-BYTE NAME OF INPUT CLIENT MEMBER NAME      *
*                  IF INPUT WAS FROM A CLIENT, BLANKS OTHERWISE.    *
*                  (LEFT-JUSTIFIED, PADDED WITH BLANKS)             *
*                  THIS IS AN ENTRY PARAMETER ONLY.                 *
*                                                                    *
*          +64      = ADDR(MESSAGE CONTROL INFORMATION),            *
*                  AVAILABLE FROM INPUT OTMA MESSAGE PREFIX.        *
*                  THIS IS AN ENTRY PARAMETER ONLY.                 *
*                                                                    *
*          +68      = ADDR(STATE DATA),                             *
*                  AVAILABLE FROM INPUT OTMA MESSAGE PREFIX.        *
*                  THIS IS AN ENTRY PARAMETER ONLY.                 *
*                                                                    *
*          +72      = ADDR(USER DATA),                              *
*                  AVAILABLE FROM INPUT OTMA MESSAGE PREFIX.        *
*                  THIS IS AN ENTRY PARAMETER ONLY.                 *
*                                                                    *
*          +76      = 4-BYTE RESERVED                                *
*                                                                    *
*                                                                    *
*          +80      = ADDR(16-BYTE MEMBER OVERRIDE)                 *
*                  THIS FIELD IS USED WHEN THE TRANSACTION IS      *
*                  INVOKED FROM A NON-OTMA LTERM, TO INDICATE      *
*                  THE CLIENT NAME WHEN AN OTMA DESTINATION         *
*                  IS DESIRED.                                       *
*                  (LEFT JUSTIFIED, PADDED WITH BLANKS)             *
*                  THIS IS AN EXIT PARAMETER ONLY.                 *
*                                                                    *
*                                                                    *
*          CONTENTS OF REGISTERS AT EXIT:                             *
*                                                                    *
*          R15      = RETURN CODE                                    *
*                                                                    *
*                                                                    *
*                  RETURN CODE CAN BE ONE OF THE FOLLOWINGS:      *
*                                                                    *
```

Sample User Exits Enabling IMS Clients to Communicate to Remote Services

```

*           0 : CONTINUE DEFAULT PROCESSING:           *
*           IF THE TRANSACTION WAS INVOKED FROM AN    *
*           OTMA TPIPE, THEN OTMA FIND-DESTINATION    *
*           PROCESSING SHOULD OCCUR.                 *
*           IF THE TRANSACTION WAS INVOKED FROM A NON- *
*           OTMA LTERM, THEN LEGACY FIND-DESTINATION  *
*           PROCESSING SHOULD OCCUR.                 *
*           4 : OTMA FIND-DESTINATION PROCESSING SHOULD *
*           OCCUR. IF THE TRANSACTION WAS ENTERED    *
*           FROM A NON-OTMA LTERM, THEN A CLIENT     *
*           MEMBER NAME MUST BE PROVIDED IN THE      *
*           PRE-ROUTING PARAMETER LIST.             *
*           8 : LEGACY FIND-DESTINATION PROCESSING    *
*           SHOULD OCCUR.                            *
*
*
* REGISTER 13 IS THE ADDRESS OF SAVE AREA.           *
*
* NORMAL LINKAGE CONVENTIONS FOR REGISTER 13, 14 AND 15
*
*
*****
*
*
*           SAVE (14,12),,DFSYPRX0&SYSDATE&SYSTIME
*           SPACE 2
*           LR R12,R15 MODULE ADDRESSABILITY
*           USING DFSYPRX0,R12
*           USING OTMAPARM,R10
*           USING SXPL,R9
*
*           LR R9,R1 LOAD PARAMETER BASE
*           L R10,SXPLFSPL OTMA SPECIFIC PARAMETERS
*           CLC OTMDEST,MYPIPE SPECIAL DESTINATION?
*           BNE PRX01500 B NO
*           ICM R8,15,OTMCOVRD GET ADDRESS OF OVERRIDING NAME
*           BZ PRX01550 B IF NO ADDRESS
*           MVC 0(16,R8),MYCLIENT SET OVERRIDING CLIENT NAME
*           LA R15,4 ROUTE TO OTMA
*           B PRX01600 RETURN
*
*
* PRX01500 DS 0H
*           CLC OTMTRANS,MYTRAN SPECIAL TRANSACTION?
*           BNE PRX01550 B NO
*           LA R15,8 INVOKE LEGACY PROCESSING
*           B PRX01600 RETURN
*

```

D Sample JCL and User Exits

```
PRX01550 DS    0H
          LA    R15,0          RC = 0
PRX01600 DS    0H
*
          RETURN (14,12),,RC=(15)  RETURN WITH RETURN CODE IN R15
          SPACE 2
          REQUATE
          SPACE 2
MYPIPE   DC    CL8'BEAOPIPE'      TEST TPIPE
MYTRAN   DC    CL8'BEACRP01'     TEST TRANSACTION
MYCLIENT DC    CL16'BEAOTMA01'   TEST CLIENT
*
* DATA MACROS
*
          DFSSXPL
          BEAOTM
*
          END
```

Sample DSECT BEAOTM

[Listing D-3](#) is the DSECT BEAOTM macro referenced previously. A DSECT for the OTMA parameter area passed to the DFSYPRX0 exit is not provided by IBM.

Listing D-3 DSECT BEAOTM

```
MACRO
          BEAOTM
OTMAPARM DSECT
OTMLTERM DS    CL8          OTMA SPECIFIC PARAMETER LIST
OTMDEST  DS    CL8          ORIGINATING LTERM/TPIPE
OTMTRANS DS    CL8          DESTINATION NAME
OTMFLAGS DS    X            TRANSACTION NAME
#OTMINP  EQU    X'80'      OTMA FLAGS
OTMSYNCH DS    X            INPUT OTMA MSG PREFIX EXISTS
OTMRESVD DS    XL2         OTMA SYNCHRONIZATION LEVEL
OTMUSER  DS    CL8          RESERVED
OTMGROUP DS    CL8          USER ID
OTMRESV2 DS    XL4         GROUP NAME
OTMCNAME DS    CL16        RESERVED
OTMMSGCL DS    A           ORIGINATING CLIENT NAME
OTMSTATE DS    A           ADDRESS OF INPUT MESSAGE CTL
OTMUSERD DS    A           ADDRESS OF INPUT STATE DATA
OTMRSVD3 DS    XL4        ADDRESS OF INPUT USER DATA
          RESERVED
```

```
OTMCOVRD DS      A                      ADDRESS OF OVERRIDING CLIENT NAME
*
                                MEND
```

OTMA Destination Resolution User Exit (DFSYDRU0)

[Listing D-4](#) is the sample IBM OTMA destination resolution user exit. The IBM DFSYDRU0 user exit enables IMS to determine whether the destination is an OTMA client, a non-OTMA LTERM, a different OTMA client, or an invalid destination. In the following sample, the lines in bold are modifications that you must make to the IBM DFSYDRU0 user exit.

Listing D-4 OTMA Destination Resolution User Exit (DFSYDRU0)

```
TITLE 'DFSYDRU0: OTMA DESTINATION RESOLUTION USER EXIT'
DFSYDRU0 CSECT
DFSYDRU0 AMODE 31
DFSYDRU0 RMODE ANY
*****
*                                                                 *
* DESCRIPTION: OTMA DESTINATION RESOLUTION USER EXIT           *
*                                                                 *
*****@SCPVRT**
*                                                                 *
*       Licensed Materials - Property of IBM                   *
*                                                                 *
*       "Restricted Materials of IBM"                            *
*                                                                 *
*       5655-158 (C) Copyright IBM Corp. 1991,1996             *
*                                                                 *
*****@ECPVRT**
*                                                                 *
* MODULE TYPE:  DC                                             *
*                                                                 *
* PROCESSOR:    ASSEMBLER H                                    *
*                                                                 *
* MODULE SIZE:  SEE EXTERNAL SYMBOL DICTIONARY PRODUCED BY    *
* ASSEMBLER                                           *
*                                                                 *
* ATTRIBUTE:    REENTRANT                                       *
*                                                                 *
*
```


D Sample JCL and User Exits

```
*           THIS IS AN ENTRY PARAMETER ONLY.           *
*
*
*           +16   = 8-BYTE TRANSACTION NAME             *
*                (LEFT-JUSTIFIED, PADDED WITH BLANKS)   *
*                THIS IS AN ENTRY PARAMETER ONLY.     *
*
*
*           +24   = 1-BYTE FLAG                         *
*                X'80' - IF SET, INDICATES THAT AN INPUT *
*                   OTMA PREFIX EXISTS.  THE MCI,      *
*                   STATE DATA, AND USER DATA ARE   *
*                   ACCESSIBLE IN THIS PARMLIST.      *
*                X'40' - IF SET, INDICATES THAT A DIFFERENT *
*                   OTMA CLIENT MAY BE SPECIFIED USING *
*                   RETURN CODE OF R15=8. (I.E. MEMBER *
*                   OVERRIDE IS ALLOWED FOR THIS CALL). *
*                THIS IS AN ENTRY PARAMETER ONLY.     *
*
*
*           +25   = SYNCHRONIZATION LEVEL IF THE INPUT WAS *
*                FROM AN OTMA CLIENT.                  *
*                THIS IS AN ENTRY PARAMETER ONLY.     *
*
*
*           +26   = 1-BYTE FLAG                         *
*                X'80' - IF SET, TPIPE EXISTS FOR THE CLIENT *
*                X'40' - IF SET, LTERM EXISTS IN IMS (LEGACY) *
*                X'20' - IF SET, LU62 DESCRIPTOR EXISTS *
*                X'10' - IF SET, ETO IS AVAILABLE      *
*                X'08' - IF SET, CLIENT IS ACTIVE      *
*                THIS IS AN ENTRY PARAMETER ONLY.     *
*
*
*           +27   = 1-BYTE RESERVED                     *
*
*
*           +28   = 8-BYTE USERID                      *
*                (LEFT-JUSTIFIED, PADDED WITH BLANKS)   *
*                THIS IS AN ENTRY PARAMETER ONLY.     *
*
*
*           +36   = 8-BYTE GROUP ID                    *
*                (LEFT-JUSTIFIED, PADDED WITH BLANKS)   *
*                THIS IS AN ENTRY PARAMETER ONLY.     *
*
*
*           +44   = 16-BYTE DESTINATION CLIENT NAME     *
*                (LEFT-JUSTIFIED, PADDED WITH BLANKS)   *
*                THIS IS AN ENTRY PARAMETER ONLY.     *
*
*
*           +60   = 4-BYTE RESERVED                    *
*
*
*           +64   = 16-BYTE NAME OF INPUT CLIENT MEMBER NAME *
*                IF INPUT WAS FROM A CLIENT, BLANKS OTHERWISE. *
*                (LEFT-JUSTIFIED, PADDED WITH BLANKS)   *
*                THIS IS AN ENTRY PARAMETER ONLY.     *
```

Sample User Exits Enabling IMS Clients to Communicate to Remote Services

```
*
*
*      +80      = ADDR(MESSAGE CONTROL INFORMATION),
*                AVAILABLE FROM INPUT OTMA MESSAGE PREFIX.
*                THIS IS AN ENTRY PARAMETER ONLY.
*
*
*      +84      = ADDR(STATE DATA),
*                AVAILABLE FROM INPUT OTMA MESSAGE PREFIX.
*                THIS IS AN ENTRY PARAMETER ONLY.
*
*
*      +88      = ADDR(USER DATA),
*                AVAILABLE FROM INPUT OTMA MESSAGE PREFIX.
*
*                THIS AREA IS ALSO USED TO RETURN NEW OR
*                MODIFIED USER DATA, UP TO A MAXIMUM OF
*                1024 BYTES.
*
*                @PQ07683
*
*                THE FORMAT OF THE USER DATA IS:
*                0-1 LENGTH OF FOLLOWING USER DATA, WHICH
*                INCLUDES THIS LENGTH FIELD
*                2- USER DATA
*
*
*      +92      = 4-BYTE RESERVED
*
*
*      +96      = ADDR(OUTPUT PARAMETER LIST)
*                THE OUTPUT PARAMETER LIST IS USED TO
*                RETURN INFORMATION TO IMS.  SEE UNDER
*                CONTENTS OF REGISTERS AT EXIT.
*
*
*
*
*      CONTENTS OF REGISTERS AT EXIT:
*
*
*      R15      = RETURN CODE
*
*
*                RETURN CODE CAN BE ONE OF THE FOLLOWINGS:
*                0 : DESTINATION IS OTMA CLIENT TPIPE
*                4 : DESTINATION IS NON-OTMA LTERM (LEGACY)
*                8 : DESTINATION IS IN A DIFFERENT CLIENT.
*                   THE CLIENT NAME MUST BE RETURNED IN
*                   THE DFSYDRU0 PARMLIST.
*                12: DESTINATION IS INVALID
*
*
*      CONTENTS OF OTMA PRE-ROUTING USER EXIT PARMLIST:
*
*
*      ALL FIELDS IN THE PARMLIST SHOULD REMAIN UNCHANGED.
*      OUTPUT INFORMATION IS RETURNED AS FOLLOWS:
*
*
*      +96      = ADDR(OUTPUT PARAMETER LIST)
*                +00 = (16-BYTE MEMBER OVERRIDE)
*                THIS FIELD IS USED WHEN THE DESTINATION *
```


Sample User Exits Enabling IMS Clients to Communicate to Remote Services

```
MYCLIENT DC      CL16 'BEAOTMA01'      TEST CLIENT
*
      END
```

E Sample IMS Client and Server Transactions

BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) client and server transactions are ordinary IMS MPP transactions, and when running the BEA eLink TCP for IMS gateway as an OTMA client, FastPath transactions as well. Transactions can be written in any application programming language supported by IMS. Source code for the sample transactions in this document can be found in the BEA eLink TCP for IMS `SOURCE` distribution library. The following sample transactions are provided in this document:

- [IMS Server Transaction](#)
- [IMS Client Transactions](#)

IMS Server Transaction

An IMS server transaction offers a specified service and processes client requests for that service. A server request takes place according to the following sequence of events.

1. A client transaction running on a remote system makes a request for a service offered by the IMS gateway. The remote gateway forwards the request to the IMS gateway, which formats and inserts a *server request message* indirectly through OTMA. The server request message is destined for the IMS server transaction that offers the service.

2. IMS schedules the specified server transaction, which retrieves the server request message from the IMS message queue and performs whatever processing is appropriate. If a response is required, the server transaction formats a *server response message* and inserts it into the IMS message queue destined for the BEA eLink TCP for IMS server response transaction.
3. BEA eLink TCP for IMS retrieves the server response message from the IMS message queue or the OTMA interface and returns the response to the remote system making the request.

BEASVR01 - A Sample IMS Server Transaction

Listing E-1 is a simple *echo transaction*; that is, it simply returns a response consisting of whatever request data it received.

Listing E-1 Sample IMS Server Transaction

```
CBL APOST
*****
FILE:          BEASVR01
PURPOSE:       SAMPLE IMS SERVER TRANSACTION FOR USE WITH
                BEA CONNECT TCP FOR IMS
COMMENTS:      THIS IS A SIMPLE ECHO SERVER TRANSACTION THAT ISSUES A
                RESPONSE CONSISTING OF THE SAME DATA AS RECEIVED.
LANGUAGE:      COBOL
COPYRIGHT      (C) 1997 BEA SYSTEMS, INC. ALL RIGHTS RESERVED
*****

IDENTIFICATION DIVISION.
PROGRAM-ID.    BEASVR01.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  IBM-370.
OBJECT-COMPUTER.  IBM-370.
DATA DIVISION.
WORKING-STORAGE SECTION.

77 GU          PIC X(04) VALUE 'GU '.
77 CHNG        PIC X(04) VALUE 'CHNG'.
77 ISRT        PIC X(04) VALUE 'ISRT'.
77 PURG        PIC X(04) VALUE 'PURG'.
```

```

01 BEAGW PIC X(08) VALUE 'BEATCPI '.

*****
*                REQUEST MESSAGE                *
*****
01 REQUEST-MESSAGE.
(1) COPY SERVER.
    05 USER-DATA PIC X(512).
*****
*                RESPONSE MESSAGE                *
*****
01 REQUEST-MESSAGE.
(1) COPY SERVER.
    05 USER-DATA PIC X(512).
LINKAGE SECTION.
*****
*                I/O PCB                        *
*****
01 01 IOPCB.
    05 LTERM PIC X(08).
    05 FILLER PIC X(02).
    05 IOPCB-STATUS PIC X(02).
    05 FILLER PIC X(28).
*****

PROCEDURE DIVISION.
    ENTRY 'DLITCBL' USING IOPCB, ALTPCB.
    PERFORM IOPCB-GET THRU IOPCB-GET-EXIT.
    PERFORM ECHO-MSG THRU ECHO-MSG-EXIT.
    PERFORM ISRT-MSG THRU ISRT-MSG-EXIT.
    GO TO 9999-RETURN.

*****
*                RETRIEVE THE REQUEST MESSAGE FROM THE IMS QUEUE                *
*****
IOPCB-GET.
(2) CALL 'CBLTDLI' USING GU, IOPCB, REQUEST-MESSAGE.
    IF IOPCB-STATUS NOT = SPACES
        GO TO 9999-RETURN.

IOPCB-GET-EXIT.
EXIT.
*****
*                COMPOSE THE RESPONSE MESSAGE                *
*****
ECHO-MSG.
* THE RESPONSE MESSAGE HAS THE SAME BASIC FORMAT AS

```

E Sample IMS Client and Server Transactions

```
* THE REQUEST MESSAGE. WE CHANGE THE TRAN CODE TO THAT
* OF THE DESTINATION. NOTE THAT THE FIELD
* BEA-RESERVED MUST BE COPIED INTACT FROM THE
* REQUEST MESSAGE TO THE RESPONSE MESSAGE. FOR RE-
* SPONSE DATA, WE SIMPLY RETURN THE REQUEST DATA.
* HENCE THIS IS AN ECHO TRANSACTION.

(3)      MOVE REQUEST-MESSAGE TO RESPONSE-MESSAGE.
(4,6)    MOVE BEAGW           TO TRAN-CODE IN RESPONSE-MESSAGE.

        ECHO-MSG-EXIT.
        EXIT.
*****
*          INSERT RESPONSE MESSAGE TO IMS QUEUE          *
*****
        ISRT-MSG.
* (5)     CALL 'CBLTDLI' USING ISRT, IOPCB, RESPONSE-MESSAGE.
        IF IOPCB-STATUS NOT = SPACES
            GO TO 9999-RETURN.
ISRT-MSG-EXIT.
        EXIT.
*****
*          INSERT RESPONSE MESSAGE TO ALTERNATE PCB      *
*****
        PCB-ISRT.
        9999-RETURN.
        GOBACK.
```

Notes

1. The format of the BEA eLink TCP for IMS server request/response message header is supplied by a COBOL copybook, `SERVER`. The equivalent C language header is provided by the `BEATCP1.H` header file. Both are available in the BEA eLink TCP for IMS `INCLUDE` distribution library.
2. A `GU` (Get Unique) call is issued to retrieve the request message from the IMS message queue. A server request message is always formatted as a single message segment, so a `GN` (Get Next) call is not strictly required.
3. The `USER-DATA` (user-defined request data) is simply copied from the request message to the response message.

4. The `TRAN-CODE` in the response message is set to that of the BEA eLink TCP for IMS (in this example, `BEAGW= "BEATCPT"`; the name may be different in your installation).
5. An `ISRT` (Insert) call is issued to insert the message into the IMS message queue, returning the response to BEA eLink TCP for IMS.
6. In cases where no response is required, the server transaction simply performs any processing required by the request and returns to IMS (without sending a response message to BEA eLink TCP for IMS).

IMS Client Transactions

An IMS client transaction is one that issues a request for a specified remote service made available through BEA eLink TCP for IMS. IMS client requests (unlike server requests) are processed in two distinct “phases”: a *request* phase, followed by a *response* phase. The request and response must be processed by two separate transaction executions (for example, `T1` and `T2`).

Request Phase

A client request takes place according to the following scenario.

1. The request transaction, `T1`, issues a request for a service provided by a remote gateway by inserting a *client request message* into the IMS message queue. The client request message is destined for eLink TCP for IMS, and must specify the name (transaction code) of a response transaction (`T2`).
2. The eLink TCP for IMS gateway retrieves the client request message and forwards the request to a remote system which offers the requested service.

Response Phase

A client response takes place according to the following scenario.

1. When the response (if required) is received, a *client response message* is formatted and inserted directly into the IMS message queue. If running the OTMA client, the response is placed into the IMS message queue indirectly through the OTMA interface. The client response message is destined for the specified response transaction (T2).
2. The response transaction is scheduled by IMS and retrieves the client response message from the IMS message queue. The client response message contains sufficient information to enable the response transaction to correlate the response with the original request.

BEACRQ01 - A Sample IMS Client Request Transaction

The following sample IMS client request transaction makes a request for a service called `TOUPPER`, provided by a remote system. The `TOUPPER` service returns a response consisting of the request data translated to uppercase.

Listing E-2 Sample IMS Client Request Transaction

```
CBL APOST
      IDENTIFICATION DIVISION.
      PROGRAM-ID. BEACRQ01.
      ENVIRONMENT DIVISION.
      CONFIGURATION SECTION.
      SOURCE-COMPUTER. IBM-370.
      OBJECT-COMPUTER. IBM-370.
      DATA DIVISION.
      WORKING-STORAGE SECTION.

      77 GU          PIC X(04) VALUE 'GU '.
      77 CHNG       PIC X(04) VALUE 'CHNG'.
      77 ISRT       PIC X(04) VALUE 'ISRT'.
      77 PURG       PIC X(04) VALUE 'PURG'.

      01 MSG-IN.
         05 MI-LL          PIC S9(04) COMP VALUE +0.
         05 MI-ZZ          PIC S9(04) COMP VALUE +0.
         05 MI-TRANCODE    PIC X(08) VALUE SPACES.
         05 FILLER         PIC X(01) VALUE SPACES.
         05 MI-AREA.      PIC X(91) VALUE SPACES.
      10 MI-DATA
```

```

(1) 01 CLIENT-REQUEST.
      05 CRR-LL PIC 9(04) COMP VALUE 0.
      05 CRR-ZZ PIC 9(04) COMP VALUE 0.
      05 CRR-TRANCODE PIC X(08) VALUE SPACES.
      05 FILLER PIC X(01) VALUE SPACES.
      05 BEA-RESERVED-1 PIC X(03) VALUE LOW-VALUES.
      05 CRR-HEADER-LEN PIC 9(08) COMP.
      05 CRR-CONTEXT-DATA-LEN PIC 9(08) COMP.
      05 CRR-REQUEST-DATA-LEN PIC 9(08) COMP.
      05 CRR-MAX-RESPONSE-LEN PIC 9(08) COMP.
      05 CRR-REQUEST-TYPE PIC 9(08) COMP.
      05 CRR-RESPONSE-FORMAT PIC 9(08) COMP.
      05 CRR-ERROR-CODE PIC 9(08) COMP.
      05 CRR-REASON-CODE PIC 9(08) COMP.
      05 CRR-SERVICE-NAME PIC X(16) VALUE LOW-VALUES.
      05 CRR-RESPONSE-TRAN PIC X(08) VALUE LOW-VALUES.
      05 CRR-ORIGIN-TERMINAL PIC X(08) VALUE LOW-VALUES.
      05 BEA-RESERVED-2 PIC X(16) VALUE LOW-VALUES.
      05 CRR-CONTEXT-AREA.
          10 CRR-CONTEXT-DATA PIC X(31) VALUE SPACES.
      05 CRR-REQUEST-AREA.
          10 CRR-REQUEST-DATA PIC X(100) VALUE SPACES.
  
```

LINKAGE SECTION.

```

01 IOPCB.
      05 LTERM PIC X(08).
      05 FILLER PIC X(02).
      05 IOPCB-STATUS PIC X(02).
      05 FILLER PIC X(28).
01 ALTPCB.
      05 ALTPCB-DEST PIC X(08).
      05 FILLER PIC X(02).
      05 ALTPCB-STATUS PIC X(02).
      05 FILLER PIC X(10).
  
```

PROCEDURE DIVISION.

ENTRY 'DLITCBL' USING IOPCB, ALTPCB.

```

(2) CALL 'CBLTDLI' USING GU, IOPCB, MSG-IN.
      IF IOPCB-STATUS NOT = SPACES
      GO TO 9999-RETURN.

(3) MOVE 'BEATCPI' TO CRR-TRANCODE.
      MOVE 96 TO CRR-HEADER-LEN.
      MOVE 1 TO CRR-REQUEST-TYPE.
      MOVE 0 TO CRR-RESPONSE-FORMAT.
      MOVE LTERM TO CRR-ORIGIN-TERMINAL.
      MOVE 'TOUPPER' TO CRR-SERVICE-NAME.
      MOVE 'BEACRP01' TO CRR-RESPONSE-TRAN.
  
```

E Sample IMS Client and Server Transactions

```
MOVE ALL 'A' TO CRR-REQUEST-DATA.  
MOVE LENGTH OF CRR-REQUEST-AREA TO  
CRR-REQUEST-DATA-LEN.  
MOVE CRR-REQUEST-DATA-LEN TO CRR-MAX-RESPONSE-LEN.  
  
(4) MOVE ALL 'B' TO CRR-CONTEXT-DATA.  
MOVE LENGTH OF CRR-CONTEXT-AREA TO CRR-CONTEXT-DATA-LEN.  
  
(5) COMPUTE CRR-LL = CRR-REQUEST-DATA-LEN +  
CRR-CONTEXT-DATA-LEN +  
CRR-HEADER-LEN.  
  
(6) CALL 'CBLTDLI' USING CHNG, ALTPCB, CRR-TRANCODE.  
IF ALTPCB-STATUS NOT = SPACES  
GO TO 9999-RETURN.  
  
(7) CALL 'CBLTDLI' USING ISRT, ALTPCB, CLIENT-REQUEST.  
IF ALTPCB-STATUS NOT = SPACES  
GO TO 9999-RETURN.  
  
(8) CALL 'CBLTDLI' USING PURG, ALTPCB.  
IF ALTPCB-STATUS NOT = SPACES  
GO TO 9999-RETURN.  
  
(9) 9999-RETURN.  
  
GOBACK.
```

Notes

1. Although not used in the previous example, the format of the eLink TCP for IMS client request/response message header is supplied by a COBOL copybook, CLIENT. The equivalent C language header is provided by the BEATCPI.H header file. Both are available in the eLink TCP for IMS INCLUDE distribution library.
2. A GU (Get Unique) call is issued to retrieve the terminal input message from the IMS message queue.
3. The client request header is initialized as follows:
 - CRR-TRANCODE is set to the transaction code of the eLink TCP for IMS gateway (in this example, "BEATCPI"; the name may be different in your installation).
 - BEA-RESERVED-1 *must be initialized to binary zero.*

- CRR-HEADER-LEN is set to the length of the client request header (96 bytes).
 - CRR-REQUEST-TYPE is set to 1, specifying that a response is required.
 - CRR-RESPONSE-FORMAT is set to 0, specifying that the response is to be returned as a single message segment.
 - CRR-ORIGIN-TERMINAL is set to the name of the requesting terminal (LTERM), obtained from the IOPCB.
 - CRR-SERVICE-NAME is set to the name of the service being requested ("TOUPPER"). *This name must be the name of the service as locally defined to eLink TCP for IMS. This name may (or may not) be the same as the name of the service as defined on the remote system which offers the service.*
 - CRR-RESPONSE-TRAN is set to the name of the designated response transaction ("BEACRP01").
 - CRR-REQUEST-DATA-LEN is set to the length of user-defined request data being sent.
 - CRR-MAX-RESPONSE-LEN is set to the maximum amount of response data that can be accommodated.
 - BEA-RESERVED-2 *must be initialized to binary zero.*
4. Context data is optional and can be any data desired. Typically, context data is used by the response transaction to correlate the response with the original request.
 - CRR-CONTEXT-DATA-LEN is set to the length of any user-defined context data included.
 5. The length of the message segment (LL) is set to the overall length of the client request message (header + context data + request data).
 6. A CHNG (Change) call is issued to set the destination to the transaction code of the eLink TCP for IMS gateway (in this example, "BEATCPI"; the name may be different in your installation).
 7. An ISRT (Insert) call is issued to insert the client request message into the IMS message queue. Although the client request message is formatted as a single message segment in this example, the client request message may be formatted as multiple message segments, if desired.
 8. A PURG call is issued to signal that the message is complete; for example, there are no additional message segments.

9. The requesting transaction does not wait on the response. Rather, the specified response transaction is scheduled by eLink TCP for IMS when processing of the request completes.

BEACRP01 - A Sample IMS Client Response Transaction

The following sample IMS client response transaction processes the response from a request for a service called `TOUPPER`, provided by a remote system. The `TOUPPER` service returns a response consisting of the request data translated to uppercase.

Listing E-3 Sample IMS Response Transaction

```
CBL APOST
    IDENTIFICATION DIVISION.
    PROGRAM-ID. BEACRP01.
    ENVIRONMENT DIVISION.
    CONFIGURATION SECTION.
    SOURCE-COMPUTER. IBM-370.
    OBJECT-COMPUTER. IBM-370.
    DATA DIVISION.

WORKING-STORAGE SECTION.
    77 GU      PIC X(04) VALUE 'GU'.
    77 CHNG    PIC X(04) VALUE 'CHNG'.
    77 ISRT    PIC X(04) VALUE 'ISRT'.
    77 PURG    PIC X(04) VALUE 'PURG'.

1 MSG-OUT.
    05 MO-LL          PIC S9(04) COMP VALUE +0.
    05 MO-ZZ          PIC S9(04) COMP VALUE +0.
    05 MO-TRANCODE    PIC X(08) VALUE SPACES.
    05 FILLER         PIC X(01) VALUE SPACES.
    05 MO-AREA
        10 MO-DATA    PIC X(91) VALUE SPACES.
(1) 01 CLIENT-RESPONSE.
    05 CRR-LL        PIC 9(04) COMP VALUE 0.
    05 CRR-ZZ        PIC 9(04) COMP VALUE 0.
    05 CRR-TRANCODE  PIC X(08) VALUE SPACES.
    05 FILLER        PIC X(01) VALUE SPACES.
    05 BEA-RESERVED-1 PIC X(03) VALUE LOW-VALUES.
    05 CRR-HEADER-LEN PIC 9(08) COMP.
    05 CRR-CONTEXT-DATA-LEN PIC 9(08) COMP.
```

```

(7) 05 CRR-RESPONSE-DATA-LEN PIC 9(08) COMP.
(7) 05 CRR-MAX-RESPONSE-LEN PIC 9(08) COMP.
    05 CRR-REQUEST-TYPE      PIC 9(08) COMP.
(3) 05 CRR-RESPONSE-FORMAT  PIC 9(08) COMP.
    05 CRR-ERROR-CODE       PIC S9(08) COMP.
    05 CRR-REASON-CODE      PIC S9(08) COMP.
    05 CRR-SERVICE-NAME     PIC X(16) VALUE LOW-VALUES.
    05 CRR-RESPONSE-TRAN    PIC X(08) VALUE LOW-VALUES.
    05 CRR-ORIGIN-TERMINAL  PIC X(08) VALUE LOW-VALUES.
    05 BEA-RESERVED-2       PIC X(16) VALUE LOW-VALUES.
(6) 05 CRR-CONTEXT-AREA.
    10 CRR-CONTEXT-DATA     PIC X(31) VALUE SPACES.
    05 CRR-RESPONSE-AREA.
    10 CRR-RESPONSE-DATA   PIC X(100) VALUE SPACES.

```

LINKAGE SECTION.

```

01 IOPCB.
05 LTERM          PIC X(08).
05 FILLER         PIC X(02).
05 IOPCB-STATUS  PIC X(02).
05 FILLER         PIC X(28).
01 ALTPCB
05 ALTPCB-DEST   PIC X(08).
05 FILLER         PIC X(02).
05 ALTPCB-STATUS PIC X(02).
05 FILLER         PIC X(10).
01 IOPCB.

```

PROCEDURE DIVISION.

ENTRY 'DLITCBL' USING IOPCB, ALTPCB.

```

(2) CALL 'CBLTDLI' USING GU, IOPCB, CLIENT-RESPONSE.
    IF IOPCB-STATUS NOT = SPACES
    GO TO 9999-RETURN.

```

```

(4) IF CRR-ERROR-CODE EQUAL 0 AND

```

```

(5) CRR-REASON-CODE EQUAL 0
    THEN MOVE 'PASSED' TO MO-DATA
    ELSE MOVE 'FAILED' TO MO-DATA.
    MOVE CRR-TRANCODE TO MO-TRANCODE.
    MOVE 84 TO MO-LL.

```

```

CALL 'CBLTDLI' USING CHNG, ALTPCB, CRR-ORIGIN-TERMINAL.
IF ALTPCB-STATUS NOT = SPACES
GO TO 9999-RETURN.

```

```

(8) CALL 'CBLTDLI' USING ISRT, ALTPCB, MSG-OUT.

```

E *Sample IMS Client and Server Transactions*

```
IF ALTPCB-STATUS NOT = SPACES
GO TO 9999-RETURN.

CALL 'CBLTDLI' USING PURG, ALTPCB.
IF ALTPCB-STATUS NOT = SPACES
GO TO 9999-RETURN.

9999-RETURN.

GOBACK.
```

Notes

1. Although not used in the previous example, the format of the eLink TCP for IMS client request/response message header is supplied by a COBOL copybook, `CLIENT`. The equivalent C language header is provided by the `BEATCPI.H` header file. Both are available in the eLink TCP for IMS `INCLUDE` distribution library.
2. A `GU` (Get Unique) call is issued to retrieve the client response message from the IMS message queue.
3. `CRR-RESPONSE-FORMAT` indicates the response format: single segment or multiple-segment, as specified in the original request. If multi-segment response format is specified, the response message consists of three segments: response header, context data, and response data.
4. `CRR-ERROR-CODE` indicates whether or not an error occurred. If this field is zero, it indicates that the request was successfully processed. Otherwise, an error occurred and the error code identifies the type of error. The error code should always be checked before processing a response.
5. `CRR-REASON-CODE` may provide additional information when an error is indicated. A value of zero indicates that no additional information is available.
6. Context data, if present, is preserved from the original request message. Typically, the context data is used by the response transaction to correlate the response with the original request.
7. `CRR-RESPONSE-DATA-LEN` contains the length of any response data returned. However, if a truncation error occurred, this field contains the actual (pretruncated) length of the response data. The amount of response data returned is truncated to the length specified by `CRR-MAX-RESPONSE-LEN`.

8. A response indicating success or failure (based on the error and reason codes) is written to the original (requesting) terminal, using the LTERM preserved in CRR-ORIGIN-TERMINAL. Note that a modifiable, alternate PCB must be used for this purpose.

BEACRR01 - A Sample IMS Client Request/Response Transaction

If desired, the functionality of the IMS client request and response transactions can be combined into a single application transaction. However, logic must be added to perform the appropriate (request or response) processing based upon the dynamic execution context.

Listing E-4 Sample IMS Request/Response Transaction

```

CBL APOST
  IDENTIFICATION DIVISION.
  PROGRAM-ID. BEACRR01.
  ENVIRONMENT DIVISION.
  CONFIGURATION SECTION.
  SOURCE-COMPUTER. IBM-370.
  OBJECT-COMPUTER. IBM-370.
  DATA DIVISION.
  WORKING-STORAGE SECTION.

      77 GU           PIC X(04) VALUE 'GU '.
      77 CHNG        PIC X(04) VALUE 'CHNG'.
      77 ISRT        PIC X(04) VALUE 'ISRT'.
      77 PURG        PIC X(04) VALUE 'PURG'.

01 MSG-IN.
      05 MI-LL       PIC S9(04) COMP VALUE +0.
      05 MI-ZZ       PIC S9(04) COMP VALUE +0.
      05 MI-TRANCODE PIC X(08) VALUE SPACES.
      05 FILLER      PIC X(01) VALUE SPACES.
      05 MI-AREA.
          10 MI-DATA PIC X(91) VALUE SPACES.
01 MSG-OUT.
      05 MO-LL       PIC S9(04) COMP VALUE +0.
      05 MO-ZZ       PIC S9(04) COMP VALUE +0.
      05 MO-TRANCODE PIC X(08) VALUE SPACES.

```

E Sample IMS Client and Server Transactions

```
05 FILLER          PIC X(01) VALUE SPACES.
05 MO-AREA.
    10 MO-DATA     PIC X(91) VALUE SPACES.

(1) 01 CLIENT-REQUEST-RESPONSE.
    05 CRR-LL          PIC 9(04) COMP VALUE 0.
    05 CRR-ZZ          PIC 9(04) COMP VALUE 0.
    05 CRR-TRANCODE    PIC X(08) VALUE SPACES.
    05 FILLER          PIC X(01) VALUE SPACES.
    05 BEA-RESERVED-1  PIC X(03) VALUE LOW-VALUES.
    05 CRR-HEADER-LEN  PIC 9(08) COMP.
    05 CRR-CONTEXT-DATA-LEN PIC 9(08) COMP.
    05 CRR-REQUEST-DATA-LEN PIC 9(08) COMP.
    05 CRR-MAX-RESPONSE-LEN PIC 9(08) COMP.
    05 CRR-REQUEST-TYPE PIC 9(08) COMP.
    05 CRR-RESPONSE-FORMAT PIC 9(08) COMP.
    05 CRR-ERROR-CODE  PIC 9(08) COMP.
    05 CRR-REASON-CODE PIC 9(08) COMP.
    05 CRR-SERVICE-NAME PIC X(16) VALUE LOW-VALUES.
    05 CRR-RESPONSE-TRAN PIC X(08) VALUE LOW-VALUES.
    05 CRR-ORIGIN-TERMINAL PIC X(08) VALUE LOW-VALUES.
    05 BEA-RESERVED-2  PIC X(16) VALUE LOW-VALUES.
    05 CRR-CONTEXT-AREA.
        10 CRR-CONTEXT-DATA PIC X(31) VALUE SPACES.
    05 CRR-REQUEST-AREA.
        10 CRR-REQUEST-DATA PIC X(100) VALUE SPACES.

LINKAGE SECTION.
01 IOPCB.
    05 LTERM          PIC X(08).
    05 FILLER          PIC X(02).
    05 IOPCB-STATUS    PIC X(02).
    05 FILLER          PIC X(28).
01 ALTPCB.
    05 ALTPCB-DEST     PIC X(08).
    05 FILLER          PIC X(02).
    05 ALTPCB-STATUS    PIC X(02).
    05 FILLER          PIC X(10).

PROCEDURE DIVISION.
ENTRY 'DLITCBL' USING IOPCB, ALTPCB.
(3) IF LTERM = SPACES OR LOW-VALUES
    THEN PERFORM RESPONSE-MESSAGE THRU RESPONSE-MESSAGE-EXIT
    ELSE PERFORM REQUEST-MESSAGE THRU REQUEST-MESSAGE-EXIT.

9999--RETURN.
    GOBACK.

REQUEST-MESSAGE.
```

```

CALL 'CBLTDLI' USING GU, IOPCB, MSG-IN.
IF IOPCB-STATUS NOT = SPACES
GO TO REQUEST-MESSAGE-EXIT.

MOVE 'BEATCPI' TO CRR-TRANCODE.
MOVE 96 TO CRR-HEADER-LEN.
MOVE 1 TO CRR-REQUEST-TYPE.
MOVE 0 TO CRR-RESPONSE-FORMAT.
MOVE LTERM TO CRR-ORIGIN-TERMINAL.
MOVE 'TOUPPER' TO CRR-SERVICE-NAME.
MOVE MI-TRANCODE TO CRR-RESPONSE-TRAN.
MOVE ALL 'A' TO CRR-REQUEST-DATA.
MOVE LENGTH OF CRR-REQUEST-AREA TO CRR-REQUEST-DATA-LEN.
MOVE CRR-REQUEST-DATA-LEN TO CRR-MAX-RESPONSE-LEN.
MOVE ALL 'B' TO CRR-CONTEXT-DATA.
MOVE LENGTH OF CRR-CONTEXT-AREA TO CRR-CONTEXT-DATA-LEN.
COMPUTE CRR-LL = CRR-REQUEST-DATA-LEN +
                 CRR-CONTEXT-DATA-LEN +
                 CRR-HEADER-LEN.

CALL 'CBLTDLI' USING CHNG, ALTPCB, CRR-TRANCODE.
IF ALTPCB-STATUS NOT = SPACES
GO TO REQUEST-MESSAGE-EXIT.

CALL 'CBLTDLI' USING ISRT, ALTPCB, CLIENT-REQUEST-RESPONSE.
IF ALTPCB-STATUS NOT = SPACES
GO TO REQUEST-MESSAGE-EXIT.

CALL 'CBLTDLI' USING PURG, ALTPCB.
IF ALTPCB-STATUS NOT = SPACES
GO TO REQUEST-MESSAGE-EXIT.

REQUEST-MESSAGE-EXIT.
EXIT.

RESPONSE-MESSAGE.

CALL 'CBLTDLI' USING GU, IOPCB, CLIENT-REQUEST-RESPONSE.
IF IOPCB-STATUS NOT = SPACES
GO TO RESPONSE-MESSAGE-EXIT.
IF CRR-ERROR-CODE EQUAL 0 AND
CRR-REASON-CODE EQUAL 0
THEN MOVE 'PASSED' TO MO-DATA
ELSE MOVE 'FAILED' TO MO-DATA.
MOVE CRR-TRANCODE TO MO-TRANCODE.
MOVE 84 TO MO-LL.

CALL 'CBLTDLI' USING CHNG, ALTPCB, CRR-ORIGIN-TERMINAL.
IF ALTPCB-STATUS NOT = SPACES

```

E *Sample IMS Client and Server Transactions*

```
GO TO RESPONSE-MESSAGE-EXIT .

CALL 'CBLTDLI' USING ISRT, ALTPCB, MSG-OUT .
IF ALTPCB-STATUS NOT = SPACES
GO TO RESPONSE-MESSAGE-EXIT .

CALL 'CBLTDLI' USING PURG, ALTPCB .
IF ALTPCB-STATUS NOT = SPACES
GO TO RESPONSE-MESSAGE-EXIT .

RESPONSE-MESSAGE-EXIT .
EXIT .
```

Notes

1. Although not used in the previous example, the format of the eLink TCP for IMS client request/response message header is supplied by a COBOL copybook, `CLIENT`. The equivalent C language header is provided by the `BEATCPI.H` header file. Both are available in the eLink TCP for IMS `INCLUDE` distribution library.
2. The actual request processing is otherwise identical to that performed by `BEACRQ01`.
3. The actual response processing is otherwise identical to that performed by `BEACRP01`.

F Error and Reason Codes

The BEA eLink for Mainframe TCP for IMS (hereafter referenced as eLink TCP for IMS) client response message header contains error and reason codes used to return error information to an IMS client. When an error occurs while processing an IMS client request, the error code is set to a unique, nonzero value that indicates the type of error that occurred. Depending on the particular error, the reason code may also contain a value that provides additional information about the error.

Error Code (Decimal Values)	Description
0	Normal Return The request completed successfully (no errors occurred).
4	Response Data Truncated The user response data was truncated because it exceeded the maximum response length specified in the request. When this error occurs, the user response data is truncated to the maximum response length specified, but the actual length of the user response data (before truncation) is returned in the data length field.
8	Remote System Unavailable The eLink TCP for IMS gateway cannot establish a session with the remote gateway required to service the request.
12	Remote System Busy All existing sessions with the required remote gateway are busy and eLink TCP for IMS cannot establish a new session (due to session limits).

Error Code (Decimal Values)	Description
16	Request Timed Out A required response was not received from the remote host within the specified time limit. This may indicate a connectivity problem or a problem with the remote system.
20	Service Not Defined The requested service name is not defined to eLink TCP for IMS. This error can also occur if the requested service name is currently out of service.
24	Internal Error An internal error occurred and eLink TCP for IMS could not allocate storage required to process the request.
28	Invalid Request An error was detected while validating the request. When this error occurs, the reason code indicates the specific reason for the error. 0=No additional information is available for this error. 4=The request header length is not valid. 8=The combined header length, context length, and data length exceed the length of the request message (LL). 12=The specified request type is not valid. 16=A response transaction is not specified (that is, the response transaction field is spaces or binary zero). 20=The context length is not valid. The length of user context data cannot be greater than 32,767 bytes. 24=The data length is not valid. The length of user request data cannot be greater than 32,767 bytes. 28=The maximum response length is not valid. The maximum length of user response data cannot be greater than 32,767 bytes. 32=A service name is not specified (that is, the service name field is spaces or binary zero). 36=The requested service name is not defined to eLink TCP for IMS.
32	Remote System Error An error was detected by the remote gateway. When this error occurs, the reason code contains the gateway error code returned by the remote system.

Error Code (Decimal Values)	Description
36	Remote Service Error An error indication (TPERROR) was returned by the remote service. When this error occurs, the reason code contains the TPERRNO value returned by the remote service.
40	Internal Logic Error An internal logic error was detected by eLink TCP for IMS. If this error occurs, notify the systems programmer responsible for eLink TCP for IMS.
44	Service Not Available The remote service is currently unavailable.
48	Service Failed The remote service returned a TPFAIL return code.
52	Security Error The user is not authorized to execute the remote service.

Glossary

A

account ID

A 1-8 character name that is exchanged between BEA eLink gateways as part of an authentication mechanism during `CONNECT` processing.

alternate PCB

Program Control Block that allows an IMS transaction to write output messages to a destination other than that from which input was received.

API

See *application programming interface*.

application programming interface (API)

A set of well-defined programming interfaces (entry points, calling parameters, and return values) by which one software program utilizes the services of another.

C

CHKP (Checkpoint)

An IMS API call that commits the changes a program has made to a database and establishes restart places within a program.

CHNG (Change)

An IMS API call that changes (sets) the destination associated with a program control block (PCB). Subsequent messages inserted into the IMS message queue using the PCB are directed to the specified destination.

CICS

See *Customer Information Control System*.

client

A software program that makes a request for a service in a client/server architecture.

client request

A request by an IMS client request transaction to BEA eLink TCP for IMS for a service provided by remote system.

client request transaction

A user-written IMS application transaction that requests a service via BEA eLink TCP for IMS that is offered by a remote system.

client response

The response to a previously issued IMS client request, processed by an IMS client response transaction.

client response transaction

A user-written IMS application transaction that processes the response to an IMS client request for a service offered by a remote system.

client/server model

A systems or application programming model in which *servers* provide architected services and *clients* request those services. Some components in the client/server model can function only as servers; others may function only as clients; still others can function both as clients and servers.

CLIENTSHUTDOWN

An option that can be specified in the BEA eLink TCP for IMS configuration file that allows shutdown processing to be initiated by a remote client request, or as a response to a remote client request.

COBOL

A standard programming language widely used for business applications, especially on IBM mainframe systems.

configuration file

A text file used by BEA eLink TCP for IMS to define the network configuration of BEA eLink gateways and services.

cross-system coupling facility (XCF)

A component of the OS/390 operating system that provides functions to support cooperation between authorized programs running within a sysplex.

Customer Information Control System (CICS)

An IBM transaction processing monitor that runs on IBM mainframe systems.

D**DASD**

See *direct access storage device*.

direct access storage device (DASD)

Any storage device, such as a hard disk, that offers random access or direct access to the stored data.

DNS

See *domain name server*.

domain name server (DNS)

A program that runs on an Internet-connected computer system (called a DNS server) and provides an automatic translation between domain names and IP addresses.

dotted decimal notation

A convention for expressing IP addresses, consisting of a series of 4 decimal numbers (0-255), separated by periods. Example: 123.205.23.99.

G**gateway**

A software program that allows dissimilar systems to communicate and exchange information. A gateway normally handles communication between systems and performs all necessary protocol translation such that the end applications communicate transparently.

gateway message

A BEA eLink architected data structure used to exchange requests and responses over a TCP/IP link connecting two BEA eLink gateways.

GN (Get Next)

An IMS API call that retrieves the next segment of a multisegment message from the IMS message queue.

GU (Get Unique)

An IMS API call that retrieves the first (or only segment) of a message from the IMS message queue.

I**IMS**

See *Information Management System*.

IMS message

An IMS-architected data structure used to implement communications between components in an IMS system.

inbound session

A logical communications path between BEA eLink TCP for IMS and another (remote) BEA eLink gateway. An inbound session is used by the remote gateway to send client requests to the local (IMS) gateway for processing.

Information Management System (IMS)

An IBM high-performance, high-volume transaction processing monitor that runs on IBM mainframe systems.

input/output program control block

An API control block used by an IMS transaction to retrieve its input and write output back to the input source (e.g., a terminal).

Interactive System Productivity Facility (ISPF)

An IBM program product used by system and application programmers to manipulate data (edit, browse, copy, etc.) on an IBM mainframe system.

Internet protocol address (IP address)

A numeric value that uniquely identifies a node in a TCP/IP network. IP addresses are usually expressed in “dotted decimal notation”, a series of 4 decimal numbers (0-255), separated by periods. Example: 123.205.23.99.

I/O PCB

See *input/output program control block*.

ISPF

See *Interactive System Productivity Facility*.

IP address

See *Internet protocol address*.

ISRT (Insert)

An IMS API call used to insert a message segment into the IMS message queue.

J**JCL**

See *job control language*.

job control language (JCL)

A language used to define and submit batch jobs on an IBM mainframe system.

L**Language Environment /370 (LE/370)**

An IBM program product which provides common runtime support for a variety of high-level IBM mainframe programming languages, including COBOL, C/C++, and PL/I.

LE/370

See *Language Environment / 370*.

listener

A special TCP/IP socket that is dedicated to “listening” for incoming TCP/IP connection requests at a specific port number.

LMID

See *logical machine ID*.

local gateway

The BEA eLink TCP for IMS gateway.

local service name

A 1-16 character name that identifies a service that can be requested by an IMS client transaction.

logical machine ID (LMID)

A symbolic name for a BEA eLink gateway (corresponding to a specific IP address and port combination).

logical terminal (LTERM)

A terminal attached to an IMS system, used by end users to enter transactions and receive responses.

LTERM

See *logical terminal*.

M**message level**

An attribute associated with messages issued by BEA eLink TCP for IMS that classifies each message as informational, error-related, etc.

message log

A dataset (usually allocated on disk) to which BEA eLink TCP for IMS logs all messages issued.

message processing program (MPP)

A user-written application program that processes transactions in an IMS system. MPPs typically confine their I/O operations to the IMS message queue (e.g., retrieving terminal input and writing terminal output) and a database.

message queue

The message storage and retrieval mechanism used by IMS components to exchange messages. For example, end-user input at a terminal causes a message to be placed into the message queue. A transaction processing program retrieves the message (containing the user input) from the message queue.

message segment

IMS messages are made up of “pieces” called message *segments*. A message can consist of a single segment, or multiple segments. A message segment is the basic unit of I/O when inserting messages into and retrieving messages from the IMS message queue.

MPP

See *message processing program*.

Multiple Virtual Storage

An IBM operating system used by IBM mainframe computers.

MVS

See *Multiple Virtual Storage*.

MVS Console

A terminal device used by the MVS operating system and other software components to display informational and error messages for the system operator. MVS consoles can also be used by the system operator to enter commands or to inquire on and modify system status.

O**online transaction processing**

A form of data processing in which users at terminals or workstations send messages to application programs that update databases in real time.

OS/390

An IBM operating system used by IBM mainframe computers.

OTMA

Open Transaction Manager Access is a transaction-based, connectionless client/server protocol.

outbound session

A logical communications path between BEA eLink TCP for IMS and another (remote) BEA eLink gateway. An outbound session is used by the local (IMS) gateway to send client requests to the remote gateway for processing.

P**password**

A 1-8 character name that is exchanged between BEA eLink gateways as part of an authentication mechanism during `CONNECT` processing.

PCB

See program control block.

port number

The entity on a TCP/IP host that identifies a logical communications channel and distinguishes one connection from another. A TCP/IP server application “listens” for incoming connection requests at a designated port. A TCP/IP client application initiates a connection with the server by specifying the host’s IP address and the server’s designated port number.

program control block (PCB)

An IMS API control block used in conjunction with application program I/O (e.g., inserting messages into and retrieving messages from the IMS message queue).

program specification block (PSB)

A control block that defines a transaction processing program to IMS.

protocol

The set of “rules” followed by two systems to communicate and exchange information.

PSB

See program specification block.

PURG (Purge)

An IMS API call that indicates that insertion of a message into the IMS message queue is complete (that is, there are no additional message segments).

R

remote gateway

A BEA eLink gateway (TUX, CICS, or IMS) to which a local BEA eLink TCP for IMS gateway is connected.

remote service name

A 1-16 character name that identifies a service offered by a remote system and which can be accessed through a remote gateway.

request phase

IMS client requests are processed in two “phases” – a request phase (in which transaction T1 issues an IMS client request), and a corresponding response phase (in which transaction T2 processes the response to the request).

response phase

IMS client requests are processed in two “phases” – a request phase (in which transaction T1 issues an IMS client request), and a corresponding response phase (in which transaction T2 processes the response to the request).

S

server

A software program that provides a service in a client/server architecture.

server request

A request (issued by a client located on a remote system) for a service provided by an IMS server transaction.

server response

The response to a request (issued by a client located on a remote system) for a service provided by an IMS server transaction.

server response log

A dataset to which BEA eLink TCP for IMS logs all IMS server responses not correlated with a pending IMS server request.

server transaction

A user-written IMS application transaction that provides a service that can be requested by a remote client through BEA eLink.

shutdown

The process of terminating (or “shutting down”) a BEA eLink gateway.

sleep time

The amount of time that BEA eLink TCP for IMS “waits” before polling the IMS message Queue for new work.

socket

The logical end point of a TCP/IP connection. An application accesses a TCP/IP connection through a socket.

socket descriptor

A TCP/IP-assigned number that uniquely identifies a socket and TCP/IP connection. An application must specify the socket descriptor in TCP/IP API calls to identify the socket/connection.

socket ID

See *socket descriptor*.

socket number

See *socket descriptor*.

T**TCP/IP**

See *Transmission Control Protocol Internet Protocol*.

timeout

When a request is sent to a remote system or to an IMS server transaction and no response is received within a specified period of time, the request “times out” and an error is recognized.

TRANCODE

See *transaction code*.

transaction code (TRANCODE)

A name that uniquely identifies a transaction in an IMS environment.

Transmission Control Protocol Internet Protocol (TCP/IP)

An open communications protocol supported by most systems, including UNIX, personal computers, and IBM mainframes. TCP/IP can be used to exchange information between dissimilar computing platforms.

W**Write-To-Operator-with-Reply (WTOR)**

A persistent message displayed on an OS/390 console which solicits operator input. A system operator replies to the message by entering the associated *reply ID* and the text of the reply.

WTOR

See *Write-To-Operator-with-Reply*.

X**XCF**

See *cross-system coupling facility*.



Index

A

- abend
 - with dump 4-7
- ADD GATEWAY command 3-23
- ADD HOST command 3-25
- ADD SERVICE command
 - local services 3-30
 - remote services 3-28
- administering eLink TCP for IMS 4-1
- API, OS/390 TCP/IP Sockets 1-2
- architecture of eLink TCP for IMS 1-2

B

- BEA eLink TCP for IMS
 - See eLink TCP for IMS
- buffers
 - CARRAY 5-11
 - FML 5-11
 - STRING 5-11
 - VIEW 5-11

C

- CARRAY buffer 5-11
- CHKPT CONFIG command 3-22
- client
 - request transaction (IMS) 2-5, 5-4, 5-9,
E-6, E-8
 - request/response transaction (IMS) E-13
 - response transaction (IMS)

- sample E-10
- scenario E-5
- transactions, IMS E-5
- codes
 - error F-1
 - reason F-1
- commands
 - dynamic configuration
 - ADD GATEWAY 3-23
 - ADD HOST 3-25
 - ADD SERVICE
 - local services 3-30
 - remote services 3-28
 - CHKPT CONFIG 3-22
 - INFILE 3-22
 - MOD GATEWAY 3-24
 - MOD HOST 3-26
 - MOD SERVICE
 - local services 3-31
 - remote services 3-29
 - MOD SYSTEM 3-21
 - START GATEWAY 3-24
 - START HOST 3-26
 - START SERVICE
 - local services 3-32
 - remote services 3-29
 - STATUS 3-32
 - STOP GATEWAY 3-24
 - STOP HOST 3-27
 - STOP SERVICE

- local services 3-31
- remote services 3-28
- IMS 4-2
- configuration
 - defining local gateways 3-8
 - defining local hosts 3-6
 - defining local services 3-18
 - defining remote gateways 3-12
 - defining remote services 3-17
- dynamic
 - description 3-21
 - modifying a gateway 3-23
 - modifying IMS hosts 3-25
 - modifying local services 3-30
 - modifying remote services 3-27
 - modifying SYSTEM statement 3-21
 - processing a file 3-22
 - writing current configuration 3-22
- example C-1
- file description 3-1
- security 3-4
- SYSTEM statement 3-2
- connections
 - securing IMS to CICS 3-4
 - securing IMS to IMS 3-4
 - securing IMS to UNIX 3-4
- connectivity
 - testing with remote systems 4-1
 - troubleshooting 4-2
- converting numeric data 5-13
- customer support contact information xii

D

- documentation, where to find it x
- dump
 - during client-initiated shutdown 4-7
- dynamic configuration
 - description 3-21
 - modifying a gateway 3-23
 - modifying IMS hosts 3-25

- modifying local services 3-30
- modifying remote services 3-27
- modifying SYSTEM statement 3-21
- processing a file 3-22
- writing current configuration 3-22

E

- eLink TCP for IMS
 - administering 4-1
 - architecture 1-2
 - configuring 3-1
 - configuring security 3-1
 - definition 1-2
 - features 1-1
 - IMS services used A-1
 - initializing 2-4, 4-3
 - message log 4-7, B-3
 - normal operation 4-4
 - programming 5-1
 - shutting down 2-7, 4-5
 - starting 4-3
 - TCP/IP for OS/390 services used A-1
 - termination consequences 4-5
- errors
 - codes F-1
 - handling 5-9
 - messages B-1
 - request timeout 5-10
 - remote system 5-10
- example
 - configuration file C-1
 - IMS client and server transactions E-1
 - JCL D-1

F

- features of eLink TCP for IMS 1-1
- FML buffer 5-11

G

gateway

- defining local (IMS) gateways 3-8
- defining remotes 3-12
- status 3-32
- supported connectivity 1-2

GATEWAY TYPE=LOCAL statement 3-8

GATEWAY TYPE=REMOTE statement 3-12

GATEWAY TYPE=REMOTE statements 3-17

H

HOST TYPE=IMS statement 3-6

hosts

- defining local (IMS) hosts 3-6

I

IBM

- LE/370 language 1-4
- TCP/IP for OS/390 1-5

IMS

- client and server transactions, sample E-1
- client requests 2-5, 5-4, 5-9
- client transaction 2-3
- commands 4-2
- message queue 1-2, 1-4, 2-3, 2-5, 5-2
- server requests 2-5, 5-2, 5-3
- services used by eLink TCP for IMS A-1

inbound

- processing 2-2

INCLUDE distribution library 5-3

INFILE command 3-22

informational messages B-1

initialization 4-3

initializing eLink TCP for IMS 2-4

initiating requests 2-3

J

JCL distribution library D-1

L

LE/370

- language 1-4

LMID

- See logical machine ID

local (IMS) gateway, defining 3-8

local (IMS) hosts, defining 3-6

local services

- configuring 3-18
- defining 3-18

log file, server response 4-9

logical machine ID B-2

M

message log 4-3, 4-7

messages

- error B-1
- format
 - description B-1
 - informational B-1
 - OS/390 console 4-8
 - returned to remote gateway B-2
 - written to eLink TCP for IMS log B-3

MOD GATEWAY command 3-24

MOD HOST command 3-26

MOD SERVICE command

- local services 3-31
- remote services 3-29

MOD SYSTEM command 3-21

modify

- gateway configuration 3-23
- IMS host configuration 3-25
- local services configuration 3-30
- remote services configuration 3-27
- SYSTEM statement 3-21

N

NULL characters in string fields 5-13
numeric data, converting 5-13

O

operating eLink TCP for IMS 4-3
OS/390

- console messages 4-8
- TCP/IP Sockets API 1-2, 1-4

OTMA

- helpful IMS commands 4-2

outbound

- processing 2-3
- requests
 - See client requests

P

printing product documentation xi
processing

- a file 3-22
- IMS client requests 2-5
- IMS server requests 2-5
- inbound 2-2
- outbound 2-3

programming for eLink TCP for IMS 5-1
programming interface 5-2

R

reason codes F-1
receiving requests 2-2
related information xi
remote gateway

- defining 3-12
- messages returned to B-2

remote services 1-3

- configuring 3-17

remote system

- errors 5-10

shutdown 4-6

- testing connectivity with 4-1

request transaction, IMS client E-6, E-8
request/response ID 4-9
request/response transaction, IMS client E-13
requests

- client E-5
- initiating 2-3
- receiving 2-2

response transaction, IMS client E-5, E-10

S

sample

- configuration file C-1
- IMS client
 - request transaction E-6, E-8
 - request/response E-13
 - response transaction E-10
- IMS client and server transactions E-1
- IMS server transaction E-2, E-4
- JCL D-1

saving configuration file 3-22
security

- connections from IMS to CICS 3-4
- connections from IMS to IMS 3-4
- connections from IMS to UNIX 3-4
- services 3-5
- setting up 3-4

server

- requests
 - processing IMS 2-5
 - response log file 4-9
 - transaction, sample IMS E-2, E-4
- server requests, IMS 5-2, 5-3
- server response log file 4-9
- SERVICE TYPE=LOCAL statements 3-19
- SERVICE TYPE=REMOTE statements 3-17

services

- configuring security 3-5
- defining local services 3-18

- defining remote services 3-17
- shut down
 - client-initiated 4-6
- SHUTDOWN command 2-7, 4-5
- shutting down eLink TCP for IMS 2-7, 4-5
- Sockets API, OS/390 TCP/IP 1-4
- START GATEWAY command 3-24
- START HOST command 3-26
- START SERVICE command
 - local services 3-32
 - remote services 3-29
- STATUS command 3-32
- status of gateway 3-32
- STOP GATEWAY command 3-24
- STOP HOST command 3-27
- STOP SERVICE command
 - local services 3-31
 - remote services 3-28
- STRING buffer 5-11
- string data 5-12
- support
 - technical xii
- system configuration parameters 3-2
- SYSTEM statement 3-2

T

- TCP/IP
 - for OS/390
 - required component 1-5
 - services used by eLink TCP for IMS
 - A-1
 - Sockets API, OS/390 1-2, 1-4
- termination
 - client-initiated shutdown 4-6
- testing
 - connectivity 4-2
 - connectivity with remote systems 4-1
- troubleshooting
 - OTMA connections 4-2

V

VIEW

- buffer 5-11
- definitions 5-13