BEA eLink Adapter for R/3 ALE

User Guide

.

BEA eLink Adapter for R/3 ALE, Version 1.6
Document Edition 1.6
April 2000

Copyright

Restricted Rights Legend

Trademarks or Service Marks

BEA eLink Adapter for R/3 ALE User Guide

Document Edition	Part Number	Date	Software Version
1.6	701-001216-002	April 2000	eLink Adapter for R/3 ALE v1.6

Contents

5. Configuring ALE Integration

About This Document

This document provides instructions for installing and configuring the BEA eLink Adapter for R/3 ALE product on a UNIX or Windows NT system.

The BEA eLink Adapter for R/3 ALE User Guide is organized as follows:

- Chapter 1, “Introducing BEA eLink Adapter for R/3 ALE,” introduces you to BEA eLink Adapter for R/3 ALE and provides overview information on integrating applications with R/3.
- Chapter 2, “Integrating with ALE,” provides information about ALE integration, including processing IDOCs from R/3 and IDOCs into R/3.
- Chapter 3, “Preinstallation Tasks,” describes system requirements and information that you need to review before installing BEA eLink Adapter for R/3 ALE.
- Chapter 4, “Installing BEA eLink Adapter for R/3 ALE,” provides instructions for installing BEA eLink Adapter for R/3 ALE on UNIX and Windows NT systems.
- Chapter 5, “Configuring ALE Integration,” describes how to configure BEA eLink Adapter for R/3 ALE to enable access to Application Linking and Embedding (ALE) on your SAP R/3 System.
- Chapter 6, “Configuring R/3 Connections,” describes how to configure the connections to your SAP R/3 System.
- Appendix A, “Sample Configuration Files,” describes the sample `cr3_ale.ubb` files and sample `setenv.sh` and `setenv.bat` files that accompany BEA eLink Adapter for R/3 ALE.
- Appendix B, “Configuration Quick Reference,” provides a step-by-step summary of the configuration process for UNIX and Windows NT.

What You Need to Know

This document is intended for system managers who are responsible for installing products in the UNIX or Windows NT environments and for configuring layered products. Portions of this guide are intended for experienced R/3 users with knowledge of ALE configuration.

e-docs Web Site

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

How to Print the Document

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

A PDF version of this document is available on the BEA eLink Adapter for R/3 ALE 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 BEA eLink Adapter for R/3 ALE documentation Home page, click the PDF files button and select the document you want to print.

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

Contact Us!

Your feedback on the BEA eLink Adapter for R/3 ALE documentation is important to us. Send us e-mail at **docsupport@beasys.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA eLink Adapter for R/3 ALE documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA eLink Adapter for R/3 ALE 1.6 release.

If you have any questions about this version of BEA eLink Adapter for R/3 ALE, or if you have problems installing and running BEA eLink Adapter for R/3 ALE, contact BEA Customer Support through BEA WebSupport at **www.beasys.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

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

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

Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
boldface text	Terms defined in the glossary.
Ctrl+Tab	You must press two or more keys simultaneously.

Convention	Item
italics	Emphasis or book titles.

`monospace text`	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.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

`monospace boldface text`	Significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Device names, environment variables, and logical operators. *Example*s: LPT1 SIGNON OR
`{ }`	A set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`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.

. . .	One of the following in a command line:
	■ 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.
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f file-list]...`
	`[-l file-list]...`

. . .	Omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

1 Introducing BEA eLink Adapter for R/3 ALE

This topic introduces BEA eLink Adapter for R/3 ALE. It includes the following main sections:

- BEA eLink Solution Overview
- BEA eLink Adapter for R/3 ALE
- SAP R/3 Environment Integration Issues
- BEA Enterprise Application Integration Solution
- Architectural Overview
- Integrating Applications with R/3

BEA eLink Solution Overview

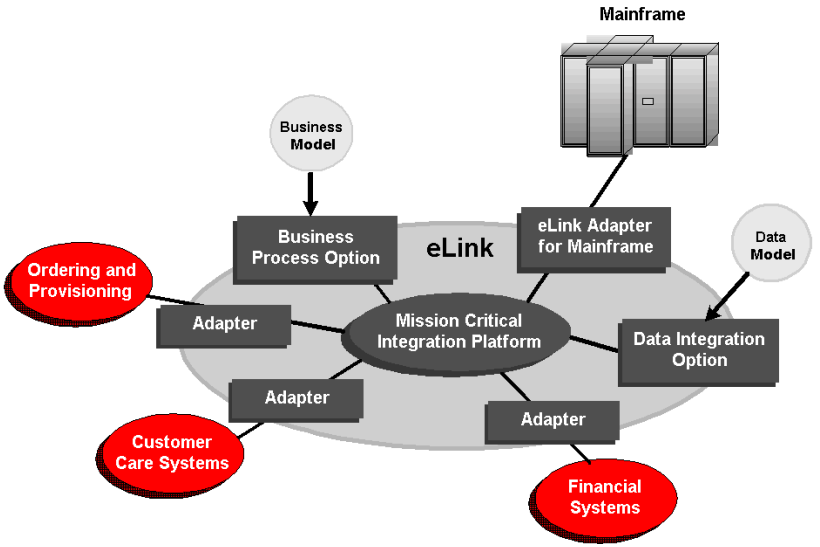
BEA eLink™ provides an open Enterprise Application Integration (EAI) solution that allows applications throughout organizations to communicate seamlessly. Using EAI, you gain the long-term flexibility and investment protection you need to keep up with today's ever-changing business environment.

Typically, companies use packaged applications to automate internal operations, such as financial, manufacturing, or human resources. While they successfully address the needs of these specific areas, these proprietary platforms often do not work together. To compete today, you need a much greater exchange of information. Systems need to

communicate at a process level within your own organization, as well as with customer's and supplier's systems. BEA eLink Platform is the underlying basis of BEA eLink, a family of off-the-shelf enterprise application integration (EAI) products that leverage the BEA transaction platform to integrate existing legacy applications with customer-focused and business-to-business e-commerce initiatives.

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

Figure 1-1 BEA eLink Solution Illustration



The entire BEA eLink family (including all options and adapters) is highly scalable. Multiple instances of BEA eLink components can collaborate so that work is divided between eLink domains. BEA eLink includes SNMP integration for enterprise management.

The current BEA eLink Platform leverages the BEA Tuxedo infrastructure because it is based on a service-oriented architecture. Both BEA Tuxedo and BEA eLink communicate directly with each other and with other applications through the use of services. Multiple services are grouped into “application servers” or “servers”. The terms Tuxedo services/servers and eLink services/servers can be used interchangeably. Terms like “Tuxedo service” and “Tuxedo server” are used in this guide, and references are made to the Tuxedo product documentation. These terms and references are all applicable to the eLink Platform.

The BEA eLink Platform complies with the Open Group’s X/Open standards including support of the XA standard for two-phase commit processing, the X/Open **ATMI** API, and XPG standards for language internationalization. C, C++, COBOL, and Java are supported. The BEA eLink Platform connects to any RDBMS, OODBMS, file manager or queue manager, including a supplied XA-compliant queueing subsystem.

The following components operate with BEA eLink Platform:

- The **Data Integration Option** translates data models used by different applications into a common data format. It provides a cost-effective alternative to writing or generating programs to perform this function. It also handles complex translation with great power and scalability. The DIO leverages technology based on the TSI Mercator product, which is integrated with eLink.
- The **Business Process Option** helps automate tasks in the distributed global business process and dynamically responds to business events and exceptions. The BPO is currently implemented by integrating eLink with technology based on InConcert workflow management software.
- An **eLink Adapter** provides the interface between the BEA eLink Platform and external applications with out-of-the-box functionality.

BEA eLink Adapter for R/3 ALE

BEA eLink Adapter for R/3 ALE (application link enabling technology) is the infrastructure of choice for application integration of the SAP R/3 environment. BEA eLink Adapter for R/3 ALE works with mission-critical, high-performance middleware to enable easy application integration with and real-time access to SAP R/3 transactions, functions and data.

A production-proven solution, eLink Adapter for R/3 ALE provides an enterprise-based approach to integrating SAP R/3 with applications in all industry-leading computer environments.

SAP R/3 Environment Integration Issues

SAP R/3 is a complete information technology solution that provides an integrated suite of financial, distribution, human resources, and manufacturing applications that can be customized to meet customer needs.

Many firms implementing SAP R/3 realize that full integration between SAP R/3 and their own applications is crucial in order to maximize the effectiveness of their entire environment. However, because numerous, heterogeneous computing environments exist across the enterprise, integration has been costly to implement and difficult to maintain.

BEA Enterprise Application Integration Solution

BEA eLink Adapter for R/3 ALE assures ready, high-performance and transparent access to mission-critical applications and information across the network with a single, standard programming interface.

BEA eLink Adapter for R/3 ALE builds gateways that can be customized for specific customer needs. These gateways enable interoperability between BEA middleware applications and legacy mainframe environments. Utilization of BEA eLink Adapter for R/3 ALE provides distributed access to SAP R/3 data from industry-leading applications and eliminates the need for applications running in the SAP R/3 environment in order to access ‘bet-the-business’ data and functions.

Key Benefits

Key benefits of using the BEA eLink Adapter for R/3 ALE include:

- Incorporates third-party packages into the SAP R/3 environment
- Empowers non-SAP R/3 applications to execute SAP R/3 applications
- Enables the reduction of application development time
- Allows significant "reutilization" of existing servers
- Provides faster and smoother deployment of SAP R/3
- Accommodates links between non-SAP R/3 GUIs and SAP R/3 or between cooperating servers

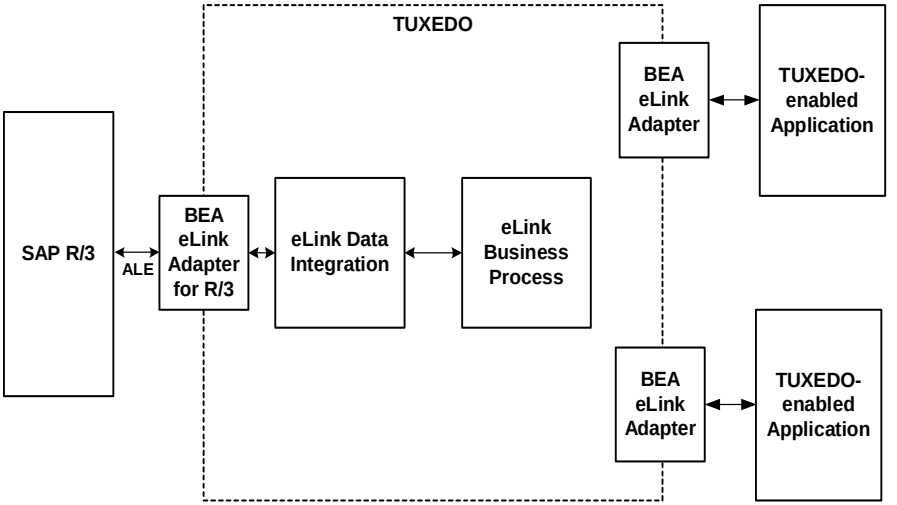
Business Applicability and Scalability

In addition to empowering the successful integration of EDO-enabled applications with SAP R/3 in heterogeneous multi-platform environments, eLink Adapter for R/3 ALE makes it possible for companies to access SAP R/3 services, maintain secure business data, and simplify forward migration to new releases of SAP R/3. BEA eLink Adapter for R/3 ALE's integration efficiency has the robustness and high-performance required of a permanent element in the overall business solution.

Architectural Overview

Figure 1-2 shows the architecture of BEA eLink Adapter for R/3 ALE:

Figure 1-2 Architecture of BEA eLink Adapter for R/3 ALE



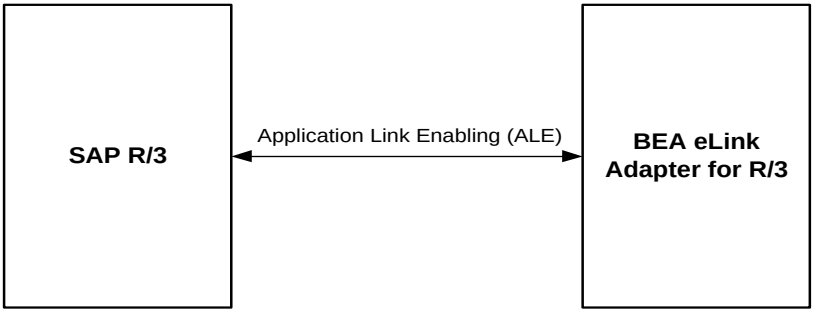
BEA eLink Adapter for R/3 ALE works with the following components:

- **eLink Data Integration** manages data transformation: data formats, data content, and rules.
- **eLink Business Process** manages process flow: state-based business processes consisting of multiple tasks.

BEA eLink Adapter for R/3 ALE communicates with R/3 via SAP's application link enabling (ALE) technology. The adapter enables bi-directional communication between R/3 and eLink.

BEA eLink Adapter for R/3 ALE provides asynchronous communication between eLink and SAP R/3, as shown in Figure 1-2. See your SAP documentation for more information about SAP ALE technologies.

Figure 1-3 Communication between BEA eLink Adapter for R/3 ALE and SAP R/3



Integrating Applications with R/3

This section introduces the key concepts involved in integrating TUXEDO-enabled applications with R/3 using BEA eLink Adapter for R/3 ALE. It includes the following topics:

- TUXEDO ATMI
- FML32 Field Definitions

TUXEDO ATMI

BEA eLink Adapter for R/3 ALE and TUXEDO-enabled applications communicate by exchanging TUXEDO FML32 buffers. Client applications that need access to the data and functionality of R/3 use the TUXEDO application-to-transaction monitor interface (ATMI) to send request messages to BEA eLink Adapter for R/3 ALE and receive the response messages. The messages exchanged are FML32 buffers. Server applications that allow R/3 access to their data and functionality will use the TUXEDO ATMI to receive request messages from BEA eLink Adapter for R/3 ALE and send response messages.

For more information on the features of TUXEDO, programming with the TUXEDO ATMI, and encoding and decoding FML buffers, see the *TUXEDO Programmer's Guide* and the *BEA TUXEDO Reference Manual* (Section 3C) in the eLink Platform documentation.

FML32 Field Definitions

BEA eLink Adapter for R/3 ALE comes with an FML field table (`cr3_ale.fml` file) that defines the fields used in FML32 buffers. This file resides in the following directory (This is also known as your applications directory or `APPDIR`):

your eLink installation path`/adapters/sapr3_ale/bin`

Note: The suggested eLink installation path for Windows NT is `C:\eLink` and the Windows NT path uses "\" instead of "/".

The `cr3_ale.fml` file contents are as follows:

Listing 1-1 FML Field Table (cr3_ale.fml file)

```
# BEA eLink Adapter for R/3 ALE - FML fields

# name                      number type          flags  comments
CR3_RFC_TID                 100    string        -      -
CR3_IDOC_CONTROL            101    string        -      -
CR3_IDOC_DATA               102    string        -      -
CR3_IDOC                    103    string        -      -
CR3_TARGET_ID               104    string        -      -
CR3_ERROR_TEXT              105    string        -      -

# Data Integration (Mercator) FML fields

MERCATOR_FV_IN              150    string        -      -

# eLink error FML fields

# name                      number type          flags  comments
ELINK_ADAPTER_ERR_CODE      200    string        -      -
ELINK_ADAPTER_ERR           201    string        -      -
ELINK_APP_ERR               202    string        -      -
```

Note: The field numbers are default values only. You can change these values if they are already in use by another application.

Table 1-1 describes these FML fields:

Table 1-1 FML Field Definitions in the cr3_ale.fml File

Field Name	Description
CR3_RFC_TID	Field that specifies the transaction ID. See "Transaction IDs (TIDs)" in Chapter 2, "Integrating with ALE."
CR3_IDOC_CONTROL	Contains one or more control records for the IDOC packet.
CR3_IDOC_DATA	Contains one or more data records for the IDOC packet.
CR3_IDOC	Field that contains IDOC packet data. See Chapter 2, "Integrating with ALE."
CR3_TARGET_ID	Data Dependent Routing value. See Chapter 2, "Integrating with ALE."
CR3_ERROR_TEXT	Error text.
ELINK_ADAPTER_ERR_CODE	eLink error category.
ELINK_ADAPTER_ERR	BEA eLink Adapter for R/3 ALE error code and text.

Note: BEA eLink Adapter for R/3 ALE imposes no restriction on the length of string values sent and received using eLink Platform (TUXEDO). The size is determined by the target application. For example, the length of an import parameter is specified by R/3. Note that the maximum size of a TUXEDO message is 2GB.

See your TUXEDO documentation for more information about FML32 buffers and field definition tables.

2 Integrating with ALE

This topic describes how to integrate SAP R/3 with SAP application linking and embedding (ALE) technology in the BEA eLink environment. It includes the following main sections:

- ALE Integration
- Processing eLink-to-R/3 IDOCs
- Processing R/3-to-eLink IDOCs

For information about setting up ALE processing, see the following:

- Chapter 5, “Configuring ALE Integration.”
- Chapter 6, “Configuring R/3 Connections.”

ALE Integration

The following topics provide important conceptual information about integrating with ALE:

- Usage Scenarios for ALE Integration
- Information Flow for ALE Integration
- Key ALE Concepts

Usage Scenarios for ALE Integration

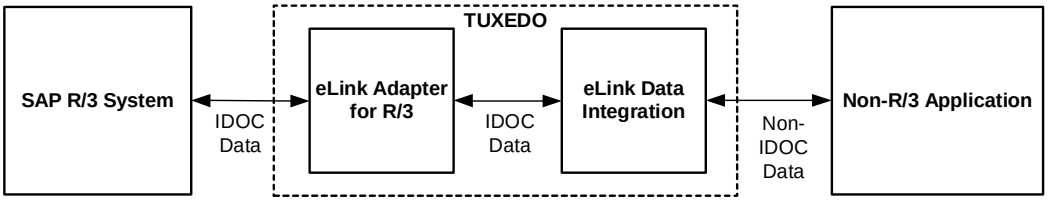
Common ALE integration implementations of eLink Adapter for R/3 ALE include:

- **Integrating R/3 and non-R/3 systems** by using eLink Platform (TUXEDO) and eLink Adapter for R/3 ALE to exchange intermediate documents (IDOCs) and non-IDOCs across application and platform boundaries. For a discussion of IDOCs, see "Key ALE Concepts" later in this section.
- **Communicating among R/3 systems** by using eLink Platform (TUXEDO) and eLink Adapter for R/3 ALE to transport IDOCs reliably and efficiently among R/3 logical systems.

Integrating R/3 and Non-R/3 Systems

Figure 2-1 shows how eLink Adapter for R/3 ALE, in conjunction with BEA eLink Data Integration, can be used to integrate R/3 with non-R/3 systems:

Figure 2-1 Integrating R/3 and Non-R/3 Systems

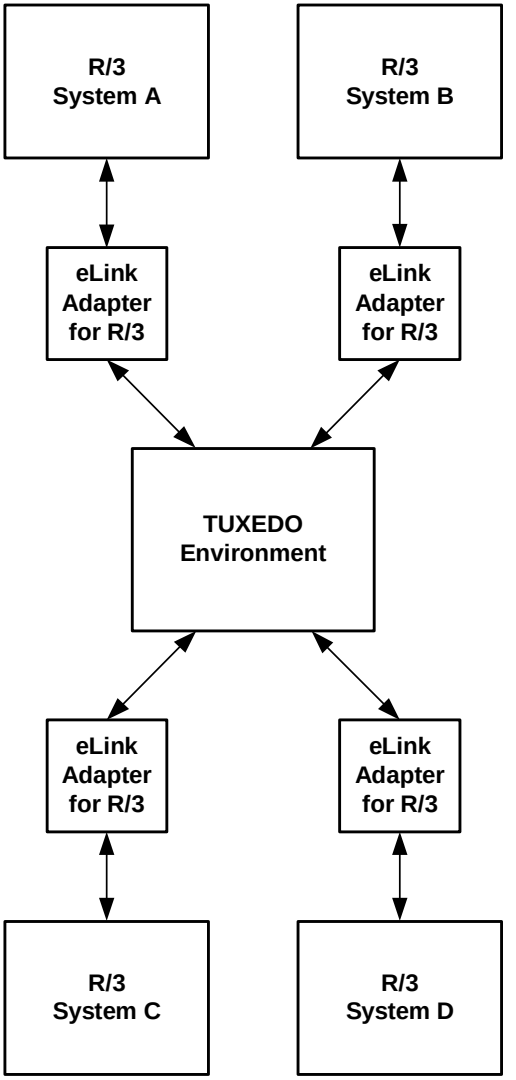


In this scenario, these BEA components provide communication and data transformation services that enable the exchange of IDOC (R/3) and non-IDOC (non-R/3) data between R/3 and non-R/3 systems.

Communicating Among R/3 Logical Systems

Figure 2-2 shows how eLink Platform (called TUXEDO in this and most other diagrams) and eLink Adapter for R/3 ALE can transport IDOCs among R/3 logical systems:

Figure 2-2 Communication Among R/3 Logical Systems

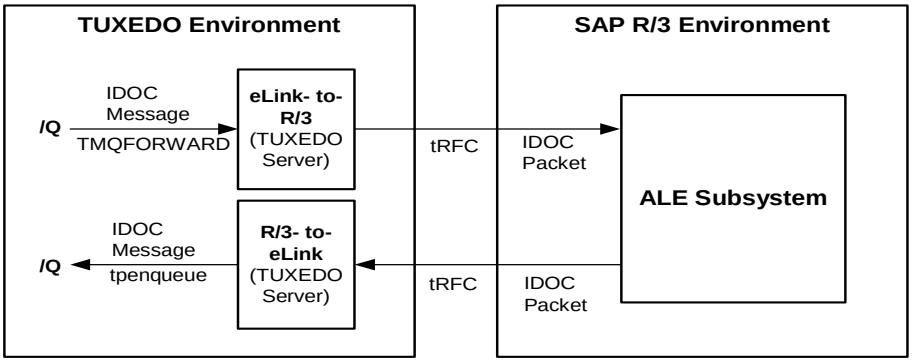


In this scenario, eLink Adapter for R/3 ALE and TUXEDO provide reliable and efficient communication services that enable the transport of IDOC packets between R/3 logical systems, thereby reducing the load on SAP communication services.

Information Flow for ALE Integration

Figure 2-3 shows the information flow for the two main ALE processes: R/3-to-eLink (outbound process) and eLink-to-R/3 (inbound process):

Figure 2-3 Overview of Information Flow for ALE Processing



- **eLink-to-R/3** is a TUXEDO server that submits IDOC packets to R/3. eLink-to-R/3 receives each IDOC packet as an FML32 message buffer (forwarded from a TUXEDO queue), adds the IDOC data for R/3, and submits the IDOC packet to R/3 via transactional RFC (tRFC). See Information Flow for eLink-to-R/3 IDOCs later in this section for more information.
- **R/3-to-eLink** is a TUXEDO server that receives IDOC packets from R/3 via transactional RFC (tRFC). R/3-to-eLink adds each IDOC packet data into an FML32 message buffer and queues it into one or more TUXEDO queues. See "Information Flow for R/3-to-eLink IDOCs" later in this section for more information.

Key ALE Concepts

Two key concepts are used in ALE processing:

- Intermediate Documents (IDOCs)
- Transaction IDs (TIDs)

Intermediate Documents (IDOCs)

In the SAP R/3 environment, an intermediate document (**IDOC**) is a container for distributing R/3 application data among R/3 logical systems and for exchanging R/3 application data with non-R/3 systems.

In ALE processing, an IDOC consists of two types of records:

- The control record uniquely identifies the IDOC, specifying such information as the identity of the sender, the target (logical system ID), message type, IDOC-type, and status. The length of a control record is 464 bytes for version 3.x IDOCs and 524 bytes for version 4.x IDOCs.
- The data records uniquely identify a segment that contains application data. The length of a data record is 1055 bytes for version 3.x IDOCs and 1063 bytes for version 4.x IDOCs, which consists of a 55-byte header and a 1000-byte segment. The header identifies the segment type and hence the segment's structure.

Each IDOC is a sequential buffer that contains one control record and one or more data records, as shown in Figure 2-4:

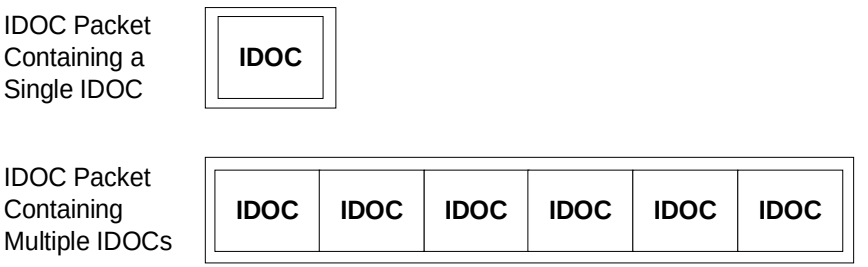
Figure 2-4 Structure of an ALE IDOC

Control Record	
Sender	
Receiver	
Message Type	
IDOC Type	
Status	
Data Records	
Header	Segment
Header	Segment
Header	Segment
Header	Segment ...

For eLink-to-R/3 IDOCs, eLink-to-R/3 validates the size and structure of control records and data records. For each IDOC, eLink-to-R/3 also verifies that the DOCNUM data in the control record matches the DOCNUM data in associated data records.

An **IDOC packet** is a message that contains one or more individual IDOCs, as shown in Figure 2-5:

Figure 2-5 Types of IDOC Packets



The R/3 System separately maintains status information about the creation, receipt, and processing of IDOCs. See your SAP R/3 documentation for more information about IDOCs.

In the TUXEDO environment, IDOC packets are transmitted in FML32 messages. These field definitions are specified in the `cr3_ale.fml` file, as described in "FML32 Field Definitions" in Chapter 1, "Introducing BEA eLink Adapter for R/3 ALE."

Transaction IDs (TIDs)

R/3 assigns a unique transaction ID (TID) to each IDOC packet it processes. R/3 uses TIDs to manage transactional integrity:

- **eLink-to-R/3**. For eLink-to-R/3 IDOC packets, R/3 uses TIDs to guarantee receipt once and only once.
- **R/3-to-eLink**. For R/3-to-eLink IDOC packets, R/3 uses TIDs to guarantee delivery once and only once.

In the TUXEDO environment, R/3-to-eLink and eLink-to-R/3 monitor TIDs through the use of TID log files. See "Managing Transactional Integrity for eLink-to-R/3 IDOCs" and "Managing Transactional Integrity for R/3-to-eLink IDOCs" for more information.

Processing eLink-to-R/3 IDOCs

The following topics describe how to process eLink-to-R/3 IDOCs using the eLink-to-R/3 server. It includes the following topics:

- eLink-to-R/3 Server
- Information Flow for eLink-to-R/3 IDOCs
- Splitting eLink-to-R/3 IDOC Packets
- Managing Transactional Integrity for eLink-to-R/3 IDOCs
- Managing Transactional Integrity for eLink-to-R/3 IDOCs
- Handling Problems with eLink-to-R/3 IDOCs

eLink-to-R/3 must be properly configured before it can process IDOCs. See "Configuring the eLink-to-R/3 Server" in Chapter 5, "Configuring ALE Integration," for more information.

eLink-to-R/3 Server

eLink-to-R/3 is a TUXEDO server that submits IDOC packets to R/3. eLink-to-R/3 receives each IDOC packet as an FML32 message buffer (forwarded from a TUXEDO queue) and it submits the IDOC packet to R/3 via tRFC. eLink-to-R/3 uses a TID log to track the TIDs associated with IDOC packets to guarantee delivery to R/3 once and only once. The name of the executable for eLink-to-R/3 is `cr3alein`.

eLink-to-R/3 Services (CR3_SUBMIT and CR3_IDOC_IN)

eLink-to-R/3 provides two services that process eLink-to-R/3 IDOC packets:

Service Name	Description
`CR3_SUBMIT`	Receives an incoming FML32 buffer containing an IDOC packet from a TUXEDO queue; validates the IDOC packet data; obtains a TID from R/3 for the IDOC packet; binds the TID into the IDOC packet; and queues the IDOC message into the CR3_IDOC_IN queue.
`CR3_IDOC_IN`	Receives the IDOC packet from the CR3_IDOC_IN queue; encodes the IDOC data for R/3; and submits the IDOC packet to R/3 for processing.

FML32 Field Definitions for eLink-to-R/3 IDOCs

The eLink-to-R/3 uses the following FML32 field definitions in IDOC messages:

Table 2-1 FML32 Fields for eLink-to-R/3 Messages

Field	Data Type	Description
`CR3_IDOC`	string	Contains an IDOC packet consisting of one or more IDOCs.
`CR3_TARGET_ID`	string	Contains a data-dependent routing value. Required even if it contains only a dummy value.
`CR3_RFC_TID`	string	Contains the transaction ID (TID) for the IDOC packet.

Table 2-1 FML32 Fields for eLink-to-R/3 Messages

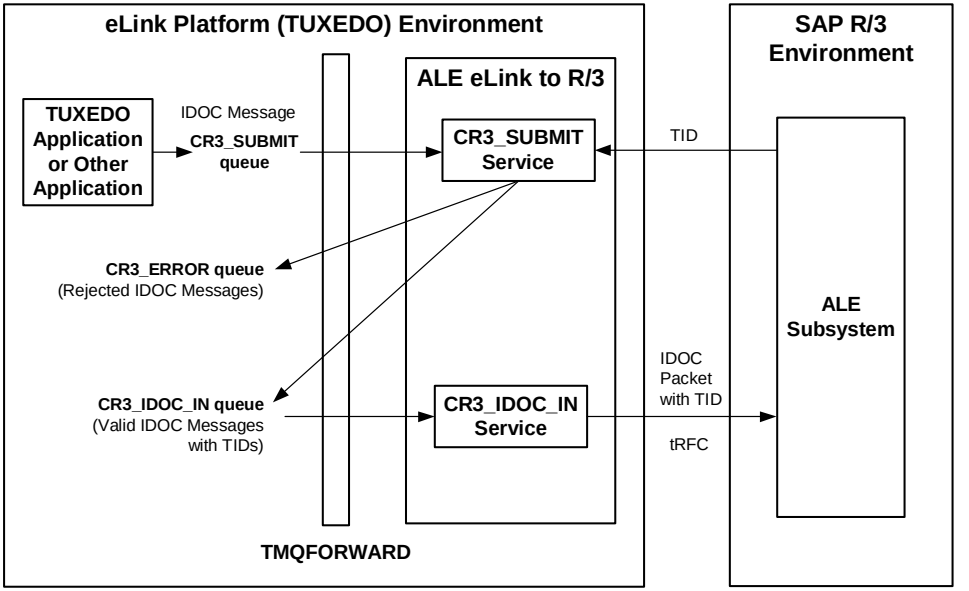
Field	Data Type	Description
CR3_IDOC_CONTROL	string	Contains one or more control records for the IDOC packet.
CR3_IDOC_DATA	string	Contains one or more data records for the IDOC packet.

These fields are defined in the `cr3_ale.fml` file, as described in "FML32 Field Definitions" in Chapter 1, "Introducing BEA eLink Adapter for R/3 ALE."

Information Flow for eLink-to-R/3 IDOCs

Figure 2-6 shows the information flow for eLink-to-R/3 IDOCs:

Figure 2-6 Information Flow for eLink-to-R/3 IDOCs



The information flow for eLink-to-R/3 proceeds in the following sequence:

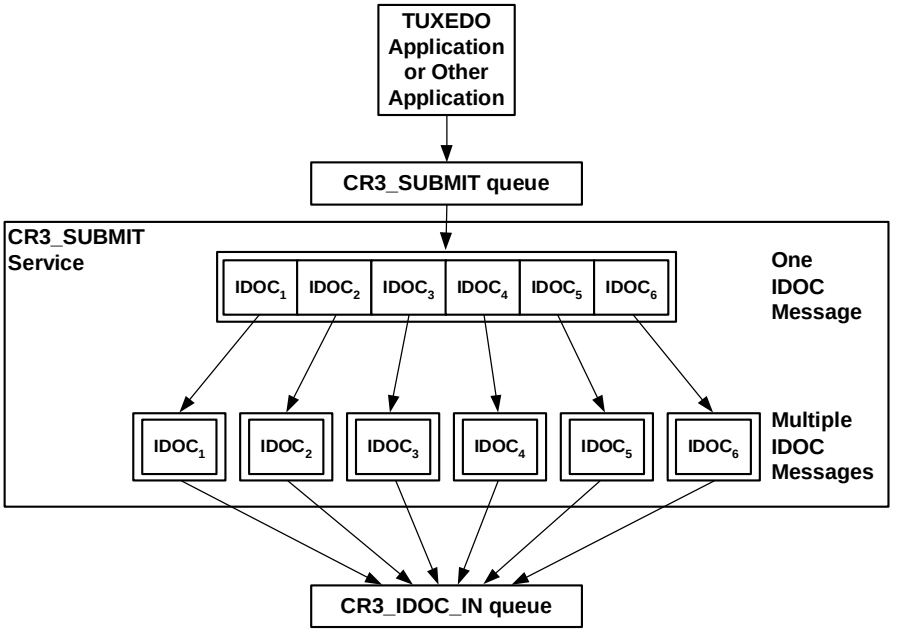
1. One or more instances of eLink-to-R/3 (a TUXEDO server) start up.
2. An application (a TUXEDO application, eLink Data Integration, or some other tool) constructs an FML32 buffer containing IDOC data and queues it into the `CR3_SUBMIT` queue.
3. The `TMQFORWARD` TUXEDO service dequeues the IDOC message from the `CR3_SUBMIT` queue and submits it to the `CR3_SUBMIT` service of the eLink-to-R/3 server.
4. The `CR3_SUBMIT` service receives the IDOC message and validates its contents:
   - It checks for the existence of the `CR3_TARGET_ID` field in the message buffer.
   - It checks the structure and size of the control record and associated data records.
   - Within the IDOC, it compares the `DOCNUM` fields in the control record and associated data records to verify that they match.
5. After validation, the `CR3_SUBMIT` service takes one of the following actions:
   - If the IDOC message fails validation, `CR3_SUBMIT` sends it to the error queue (`CR3_ERROR` queue).
   - If the IDOC message passes validation, `CR3_SUBMIT` obtains a TID from R/3, binds the TID to the IDOC packet in the FML32 buffer by encoding the TID in the message buffer (the `CR3_RFC_TID` field), and queues the IDOC message into the `CR3_IDOC_IN` queue.
6. `TMQFORWARD` dequeues the IDOC message from the `CR3_IDOC_IN` queue and submits it to the `CR3_IDOC_IN` service of the eLink-to-R/3 server.
7. The `CR3_IDOC_IN` service submits the IDOC packet and TID to R/3.

eLink-to-R/3 uses a TID log file to manage transactional integrity. See “Managing Transactional Integrity for R/3-to-eLink IDOCs” later in this section for more information.

Splitting eLink-to-R/3 IDOC Packets

By default, the eLink-to-R/3 server passes an IDOC message containing multiple IDOCs to R/3 in a single packet. You can configure eLink-to-R/3 to split IDOC messages containing multiple IDOCs into individual IDOC messages, each with its own TID. For example, if an IDOC message contains six IDOCs, eLink-to-R/3 can create six separate IDOC packets, each containing a single IDOC and associated with a unique TID. Figure 2-7 shows splitting IDOC packets and queueing them into the `CR3_IDOC_IN` queue:

Figure 2-7 Splitting eLink-to-R/3 IDOC Packets



Splitting IDOC packets provides additional flexibility for processing eLink-to-R/3 IDOCs. However, this configuration can also increase load on the R/3 System and reduce throughput performance. For example, an IDOC packet containing six IDOCs requires two RFC calls: one to request the TID and another to submit the IDOC packet to R/3. Six IDOC packets containing a single IDOC each, however, requires twelve separate RFC calls: six to request TIDs and six to submit each IDOC packet to R/3.

See "Splitting eLink-to-R/3 IDOC Packets Containing Multiple IDOCs" in Chapter 5, "Configuring ALE Integration," for instructions.

Managing Transactional Integrity for eLink-to-R/3 IDOCs

The eLink-to-R/3 server manages transactional integrity for eLink-to-R/3 IDOCs to guarantee that it delivers an IDOC packet to R/3 once and only once. R/3 uses the TID to guarantee that it processes the IDOC packet exactly once. If an attempt to submit an IDOC packet to R/3 fails, eLink-to-R/3 retries using the same TID. eLink-to-R/3 uses a TID log file to track the transaction IDs (TIDs) that R/3 assigns to each eLink-to-R/3 IDOC packet. See "Transaction IDs (TIDs)" earlier in this document for an introduction to TIDs.

About the TID Log File Used for eLink-to-R/3 IDOCs

The TID log file used with eLink-to-R/3 IDOCs contains information about TIDs that eLink-to-R/3 has received and processed. Each row in the TID file represents the TID for a separate IDOC packet and contains three fixed-position columns of information:

Table 2-2 Columns in the TID Log File for eLink-to-R/3 IDOCs

Column	Description
Date-Time Stamp	Date and time at which the state of this TID was last updated in the TID log file.
TID	TID that R/3 assigned to the IDOC packet.
Status	One of the following strings: ■ **CREATED** indicates that eLink-to-R/3 has successfully associated a TID with the IDOC packet and queued it into the `CR3_IDOC_IN` queue. ■ **CONFIRMED** indicates that eLink-to-R/3 has successfully passed the IDOC packet onto R/3.

The following example shows a sample TID file for eLink-to-R/3:

```
Tue Apr 27 14:27:40 1999  0A0201FD03F937262C600004  CONFIRMED
Tue Apr 27 14:29:39 1999  0A0201FD03F937262CD90007  CONFIRMED
Tue Apr 27 14:46:58 1999  0A0201FD03F9372630E8000A  CONFIRMED
Tue Apr 27 15:52:30 1999  0A0201FD041637263FC60013  CONFIRMED
```

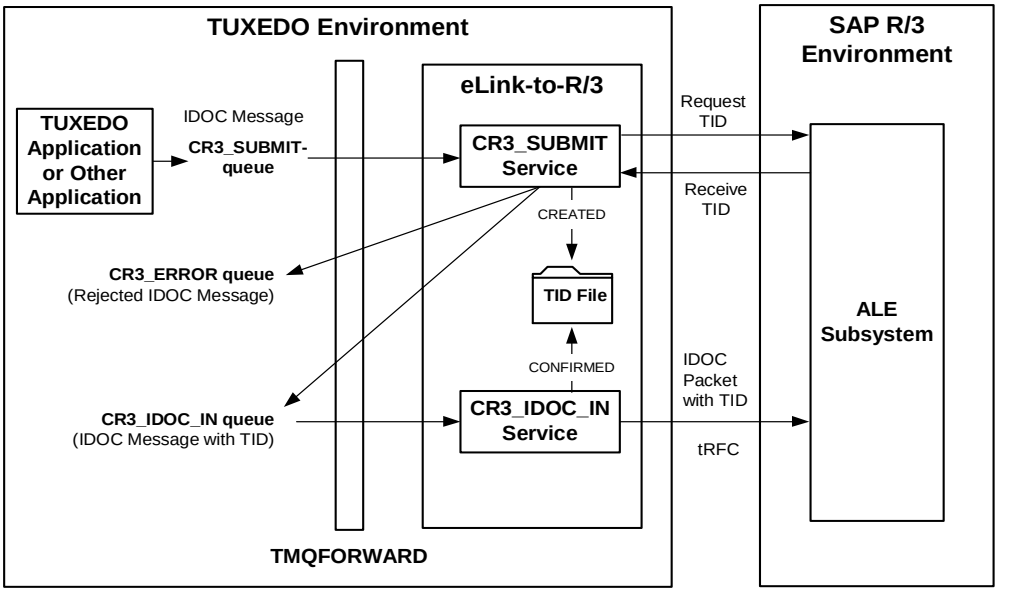
The CR3_ALEIN_TID_FILE environment variable specifies the location of the TID log file for eLink-to-R/3. See "Setting Environment Variables for eLink-to-R/3" in Chapter 5, "Configuring ALE Integration," for more information.

Use the cr3tidmanager program to manage the size and number of entries kept in TID files. See "Configuring the TID File Manager" in Chapter 5, "Configuring ALE Integration," of this guide.

Processing TIDs with eLink-to-R/3 IDOCs

Figure 2-8 shows how eLink-to-R/3 uses the TID log file to manage transactional integrity for eLink-to-R/3 IDOCs:

Figure 2-8 TID Processing for eLink-to-R/3



The eLink-to-R/3 IDOC process involves two transaction boundaries:

- The first transaction boundary ensures that the CR3_SUBMIT service has bound the IDOC packet and TID together and has successfully queued the FML32 message buffer into the CR3_IDOC_IN queue. This transaction is the entry point for any external application that submits an IDOC packet to the eLink-to-R/3 server.

- The second transaction boundary ensures that the CR3_IDOC_IN service has successfully submitted the IDOC packet and associated TID to R/3. This transaction is an internal process that submits the IDOC packet and TID to R/3 until it succeeds.

First Transaction Boundary

For the first transaction boundary, the information flow proceeds in the following sequence:

1. TMQFORWARD starts a new TUXEDO transaction, unqueues an IDOC message from the CR3_SUBMIT queue, and submits the IDOC message to the CR3_SUBMIT service of the eLink-to-R/3 server.
2. After validating the IDOC data, the CR3_SUBMIT service requests a TID from R/3.
3. R/3 generates a unique TID and returns it to the CR3_SUBMIT service.
4. The CR3_SUBMIT service opens the TID log file.
5. The CR3_SUBMIT service searches for the TID in the TID log file:
   - If the TID is not found, the CR3_SUBMIT service appends a new entry for the TID, writes the date-time stamp, TID, and state (CREATED) in the entry, and then proceeds to the next step.
   - If the TID is found, R/3-to-eLink requests a new TID from R/3 because it is already processing the current TID.
6. The CR3_SUBMIT service binds the TID to the IDOC message (by encoding the TID in the CR3_RFC_TID field in the buffer) and assigns the message to the CR3_IDOC_IN queue.
7. The CR3_SUBMIT service returns TPSUCCESS or TPFAIL, as appropriate, to TMQFORWARD.
8. TMQFORWARD closes the transaction, committing the transaction if TPSUCCESS was returned, or rolling back the transaction if TPFAIL was returned. If the transaction is rolled back, the IDOC message remains in the CR3_SUBMIT queue.

Second Transaction Boundary

For the second transaction boundary, the information flow proceeds in the following sequence:

1. TMQFORWARD starts a new transaction, dequeues an IDOC message from the CR3_IDOC_IN queue, and submits the IDOC message to the CR3_IDOC_IN service of the eLink-to-R/3 server.
2. The CR3_IDOC_IN service encodes the IDOC packet for R/3 and submits the IDOC packet to R/3.
3. If the IDOC packet is successfully sent, the CR3_IDOC_IN service opens the TID log file, finds the TID, and updates the date-time stamp and state (CONFIRMED) in the log file.
4. CR3_IDOC_IN returns the result of the send request (TPSUCCESS or TPFAIL) to TMQFORWARD.
5. TMQFORWARD closes the transaction, committing the transaction if TPSUCCESS was returned, or rolling back the transaction if TPFAIL was returned. If the transaction is rolled back, the IDOC message remains in the CR3_IDOC_IN queue.

Handling Problems with eLink-to-R/3 IDOCs

eLink-to-R/3 uses TUXEDO's transaction management capabilities to ensure transactional integrity for eLink-to-R/3 IDOCs. The following table lists problems that can occur with eLink-to-R/3 IDOCs:

Table 2-3 Handling Problems with eLink-to-R/3 IDOCs

Problem	Description
Invalid IDOC structure	If an IDOC packet fails validation, the CR3_SUBMIT service queues the FML32 message into the CR3_ERROR queue and returns TPSUCCESS to TMQFORWARD.
No CR3_TARGET_ID	If an IDOC message contains no CR3_TARGET_ID field, the CR3_SUBMIT service queues the FML32 message into the CR3_ERROR queue and returns TPSUCCESS to TMQFORWARD.

Table 2-3 Handling Problems with eLink-to-R/3 IDOCs

Problem	Description
TID Not received from R/3	If R/3 does not return a TID, `CR3_SUBMIT` returns `TPFAIL` to `TMQFORWARD`, and `TMQFORWARD` rolls back the transaction.
Send attempt to R/3 failed	If the `CR3_IDOC_IN` service does not successfully send the IDOC packet to R/3 (for example, the R/3 System is down), `CR3_IDOC_IN` returns `TPFAIL` to `TMQFORWARD`, and TMQFORWARD rolls back the transaction. The IDOC packet remains in the `CR3_IDOC_IN` queue until a subsequent send attempt succeeds.

Note: You must write an application to explicitly unqueue and handle messages in the `CR3_ERROR` queue.

Processing R/3-to-eLink IDOCs

The following topics describe how to process R/3-to-eLink IDOCs using the R/3-to-eLink server:

- R/3-to-eLink Server
- Information Flow for R/3-to-eLink IDOCs
- Splitting R/3-to-eLink IDOC Packets Into Individual IDOCs
- Queuing R/3-to-eLink IDOCs Into Multiple Queues
- Managing Data-Dependent Routing (DDR)
- Load Balancing High Volumes of R/3-to-eLink IDOCs
- Registering Multiple Program IDs
- Managing Transactional Integrity for R/3-to-eLink IDOCs
- Handling Problems with R/3-to-eLink IDOCs

The R/3-to-eLink server must be configured properly, before it can process R/3-to-eLink IDOCs. For information about setting up R/3-to-eLink, see Appendix B, "Configuration Quick Reference."

R/3-to-eLink Server

R/3-to-eLink is a TUXEDO server that receives IDOC packets from R/3 via transactional RFC (tRFC). R/3-to-eLink encodes each IDOC packet into an FML32 message buffer and queues it into a TUXEDO queue. R/3-to-eLink uses a TID log file to track the IDOC packets that it processes to ensure that it queues an IDOC packet from R/3 once and only once. The name of the executable for R/3-to-eLink is `cr3aleout`.

R/3-to-eLink uses the following FML32 field definitions in IDOC messages:

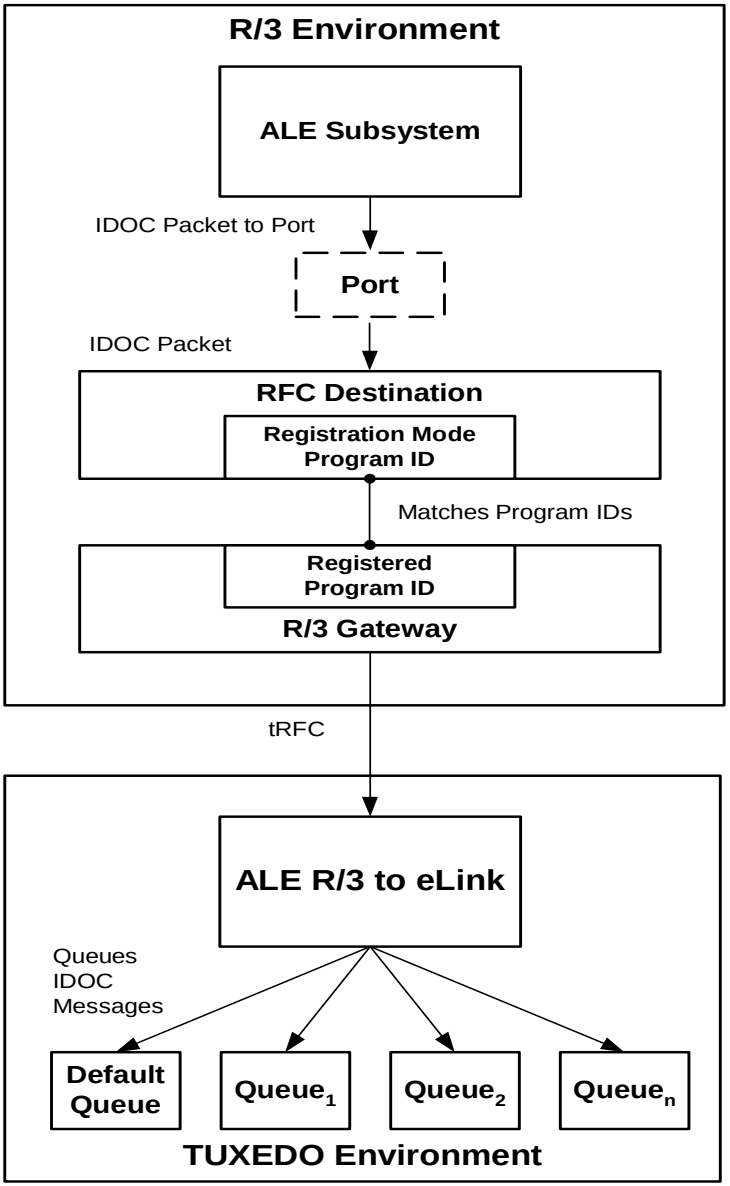
Table 2-4 FML32 Fields for R/3-to-eLink Messages

Field	Data Type	Description
`CR3_IDOC`	string	Contains a string of one or more IDOCs.
`CR3_TARGET_ID`	string	Data-dependent routing value.
`MERCATOR_FV_IN`	string	Either Y or N. See *the BEA eLink Data Integration Option v1.x* for more information.

Information Flow for R/3-to-eLink IDOCs

Figure 2-9 illustrates the information flow for R/3-to-eLink:

Figure 2-9 Information Flow for R/3-to-eLink IDOCs



The information flow proceeds in the following sequence:

1. One or more instances of the R/3-to-eLink server start up and register a program ID with the R/3 gateway. R/3-to-eLink runs in register mode and listens for IDOC packets associated with that program ID on the registered port. This program ID corresponds to a particular RFC destination.

2. R/3 submits an IDOC packet to a port (rather than to a file or another R/3 System) for a specific RFC destination.

3. R/3 sends the IDOC packet to an instance of R/3-to-eLink that is registered on the program ID of the RFC destination.

4. R/3-to-eLink receives the IDOC packet and processes the IDOC data according to the way that R/3-to-eLink is configured, or more specifically, according to the destination mapping environment variables in the environment file.

   If the `CR3_ALE_DEFAULT_IDOC_SPLIT` environment variable is set to "Y," then R/3-to-eLink splits IDOC packets containing multiple IDOCs into separate IDOC messages, each containing a single IDOC. If the `CR3_ALE_DEFAULT_IDOC_SPLIT` environment variable is set to "N," then R/3-to-eLink does not split IDOC packets. It sends packets to the default queue. This process is described in detail in "Splitting R/3-to-eLink IDOC Packets Into Individual IDOCs."

   If the IDOC packets are split (`CR3_ALE_DEFAULT_IDOC_SPLIT` = "Y"), R/3-to-eLink uses the settings in the environment file to determine the target queue for each IDOC as well as other processing options. See "Queuing R/3-to-eLink IDOCs Into Multiple Queues" for more information.

R/3-to-eLink uses a TID log file to manage transactional integrity. See "Managing Transactional Integrity for R/3-to-eLink IDOCs" later in this topic for more information.

Splitting R/3-to-eLink IDOC Packets Into Individual IDOCs

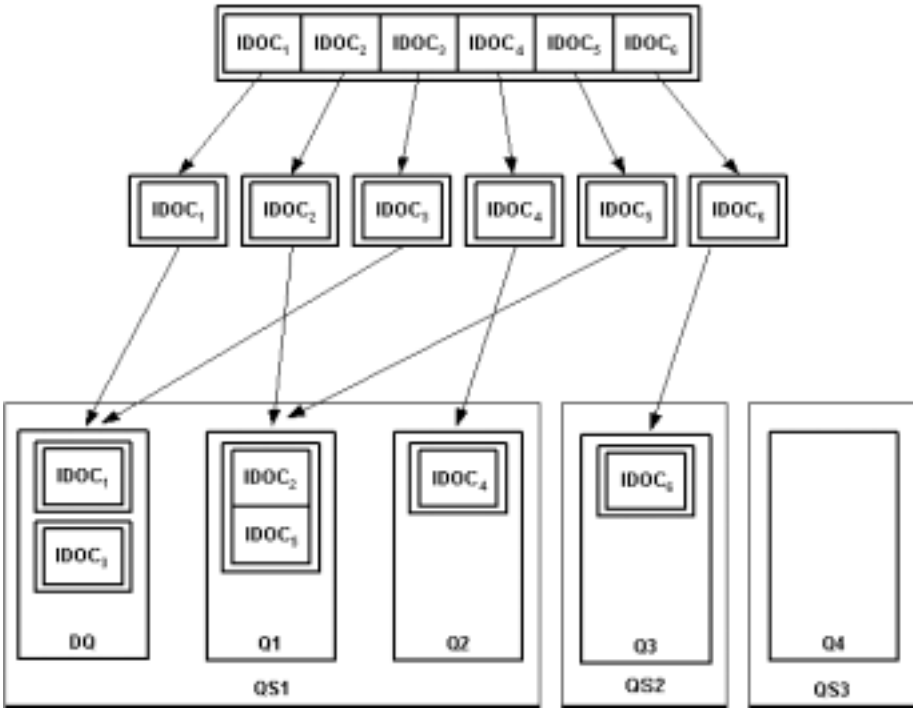
You can configure R/3-to-eLink to split IDOC packets containing multiple IDOCs into separate IDOC messages, each containing a single IDOC. By default, R/3-to-eLink encodes the entire IDOC packet into a single occurrence of the `CR3_IDOC` field in the message buffer, then queues the entire IDOC packet into a single message and places it in the default queue, which is defined in the `CR3_ALE_DEFAULT_TARGET_ID` environment variable. If you set the `CR3_ALE_DEFAULT_IDOC_SPLIT` environment variable to "Y," however, R/3-to-eLink will split the IDOC packet into individual IDOCs, then queue the individual IDOCs into queues as described in "Queuing R/3-to-eLink IDOCs Into Multiple Queues."

Queuing R/3-to-eLink IDOCs Into Multiple Queues

R/3-to-eLink is configured to use the R/3-to-eLink environment file (`cr3aleout.env`) so that it can place IDOC messages into different target queues, manage data-dependent routing, and group similar IDOCs into larger IDOCs. R/3-to-eLink makes routing and grouping decisions about individual IDOCs according to two settings specified in an IDOC's control record: the logical system ID of the target R/3 System and the IDOC message type.

If the logical system ID and message type in an individual IDOC's control record (an individual IDOC from the one that has been split) match the logical system ID and message type found in an IDOC section of `cr3aleout.env`, R/3-to-eLink queues it using the settings specified in the matching IDOC section of `cr3aleout.env`. If a matching IDOC section cannot be found, R/3-to-eLink will queue it in the default queue. This is shown in Figure 2-10:

Figure 2-10 Splitting IDOC Packets and Queuing IDOCs into Queues



See "Setting Environment Variables for R/3-to-eLink" in Chapter 5, "Configuring ALE Integration," in this guide for more information about setting the `CR3_ALE_DEFAULT_IDOC_SPLIT` environment variable.

About the Destination Mapping Variables in the Environment File

The environment file, `cr3aleout.env`, is used to configure the environment for the R/3-to-eLink server. IDOC types are listed in the `CR3_IDOC_LIST` environment variable, and specific settings are made for each IDOC type. Each known IDOC type is provided with its own section of the environment file. The destination mapping settings in these sections specify the following information for each IDOC type:

- Logical system ID of the target R/3 logical system (environment variable: `RECIEVER_PARTNER_NUMBER`).
- Message type specified in the IDOC (environment variable: `MESSAGE_TYPE`).

- Compress flag indicating whether to compress IDOC packets (with matching logical system and IDOC message type values) into a single IDOC message in the queue (environment variable: COMPRESS).
- Target ID representing the data-dependent routing value (environment variable: ROUTING).
- Destination queue space name (environment variable: QUEUE_SPACE).
- Destination queue name (environment variable: QUEUE_NAME).

For each outbound R/3-to-eLink IDOC, R/3-to-eLink searches the environment file for the logical system ID of the target R/3 System (RECIEVER_PARTNER_NUMBER) and IDOC message type (MESSAGE_TYPE) specified in the outbound IDOC's control record. If it finds an IDOC section that matches the logical system ID and message type specified in the IDOC's control record, R/3-to-eLink assigns the IDOC to the destination queue space and queue specified in the matching IDOC section of the environment file. R/3-to-eLink also processes the IDOC according to the COMPRESS flag and the Target ID (ROUTING) settings in that section.

Compressing R/3-to-eLink IDOCs

You use the COMPRESS environment variable specified in each IDOC section of the environment file to combine outbound IDOCs with matching combinations of logical system ID and IDOC message type into a single IDOC. For each outbound IDOC whose control record matches an IDOC section with regard to logical system ID and message type. If the COMPRESS variable is set to "Y", then R/3-to-eLink aggregates this IDOC with other matching IDOCs into a single, larger IDOC that it then assigns to the appropriate target queue. If the COMPRESS variable is set to "N", then R/3-to-eLink queues each outbound IDOC separately.

Setting the Data-Dependent Routing Value

You use the Target ID (ROUTING environment variable) specified in each IDOC's section of the environment file to associate an IDOC with a data-dependent routing (DDR) value. For each outbound IDOC whose control record matches an IDOC section with regard to logical system ID and message type, R/3-to-eLink adds the specified ROUTING value to the CR3_TARGET_ID FML32 field of the outbound IDOC. If no matching IDOC section is found in the environment file for that outbound IDOC, then R/3-to-eLink encodes the default DDR value, which is defined in the CR3_ALE_DEFAULT_TARGET_ID environment variable.

See “Setting the Default Data-Dependent Routing Value” in Chapter 5, “Configuring ALE Integration,” in this guide for more information.

Examples of Using Destination Map Settings

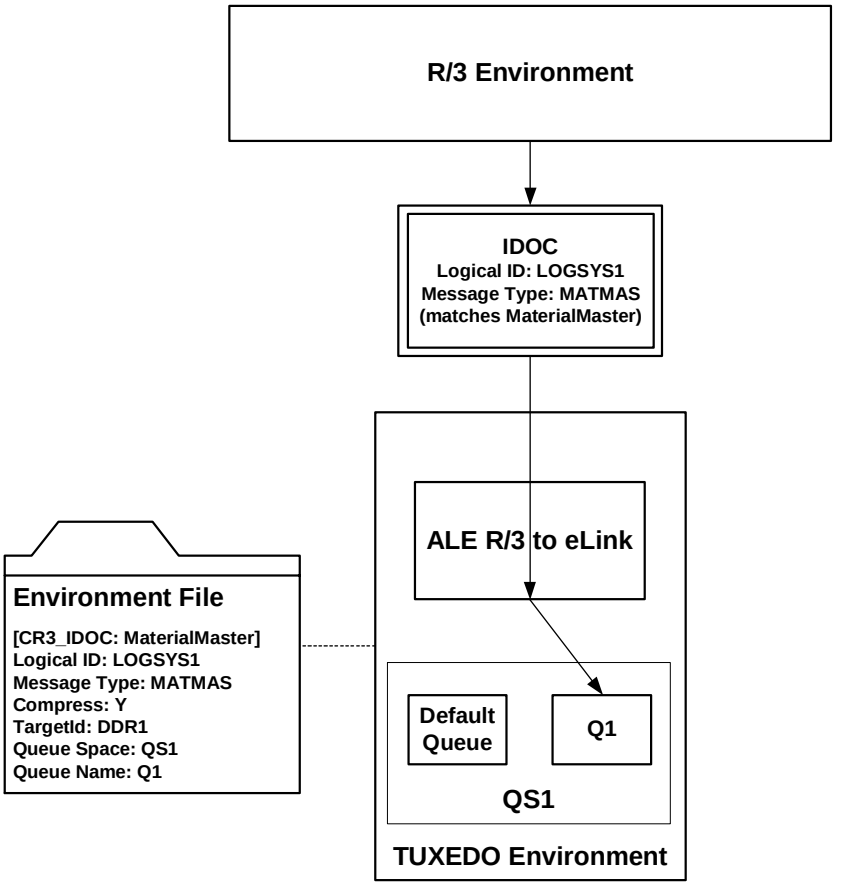
The first example shows how the R/3-to-eLink server processes an IDOC packet that contains a single IDOC. Suppose the environment file contains the settings shown in Listing 2-1.

Listing 2-1 Destination Map Settings in a Sample Environment File

```
CR3_IDOC_LIST=MaterialMaster
[CR3_IDOC=MaterialMaster]
RECIEVER_PARTNER_NUMBER=LOGSYS1
MESSAGE_TYPE=MATMAS
COMPRESS=Y
ROUTING=DDR1
QUEUE_SPACE=QS1
QUEUE_NAME=Q1
```

The control record in an R/3-to-eLink IDOC named `MaterialMaster` specifies a target logical ID (`RECIEVER_PARTNER_NUMBER`) of `LOGSYS` and a message type (`MESSAGE_TYPE`) of `MATMAS`. Figure 2-11 shows how R/3-to-eLink would process this IDOC packet according to the settings in the sample environment file section:

Figure 2-11 Queuing a Single IDOC According to the Environment File



In this scenario, R/3-to-eLink finds that the logical ID and message type specified in the outbound IDOC's control record match the logical ID and message type settings in the `MaterialMaster` IDOC section of the environment file, so it sends the IDOC message to the Q1 queue in queue space QS1. R/3-to-eLink encodes the specified TargetId value ("DDR1") in the `CR3_TARGET_ID` FML32 field. Compression does not apply in this case, because the IDOC packet contained only one IDOC.

The second example shows how R/3-to-eLink processes an IDOC packet that contains multiple IDOCs. Suppose the environment file contains the settings shown in Listing 2-2.

Listing 2-2 Complex Destination Map Settings in a Sample Environment File

```
CR3_IDOC_LIST=MaterialMaster,MaterialMaster2,CustomerMaster,
VendorMaster
[CR3_IDOC=MaterialMaster]
RECIEVER_PARTNER_NUMBER=LOGSYS1
MESSAGE_TYPE=MATMAS
COMPRESS=Y
ROUTING=DDR1
QUEUE_SPACE=QS1
QUEUE_NAME=Q1
[CR3_IDOC=MaterialMaster2]
RECIEVER_PARTNER_NUMBER=LOGSYS2
MESSAGE_TYPE=MATMAS
COMPRESS=Y
ROUTING=DDR2
QUEUE_SPACE=QS1
QUEUE_NAME=Q2
[CR3_IDOC=CustomerMaster]
RECIEVER_PARTNER_NUMBER=LOGSYS3
MESSAGE_TYPE=DEBMAS
COMPRESS=N
ROUTING=DDR3
QUEUE_SPACE=QS2
QUEUE_NAME=Q3
[CR3_IDOC=VendorMaster]
RECIEVER_PARTNER_NUMBER=LOGSYS4
MESSAGE_TYPE=CREMAS
COMPRESS=Y
ROUTING=DDR4
QUEUE_SPACE=QS3
QUEUE_NAME=Q4
```

An R/3-to-eLink IDOC packet contains six IDOCs with the following settings in the control record of each IDOC, as shown in Table 2-5:

Table 2-5 Sample IDOC Packet Containing Multiple IDOCs

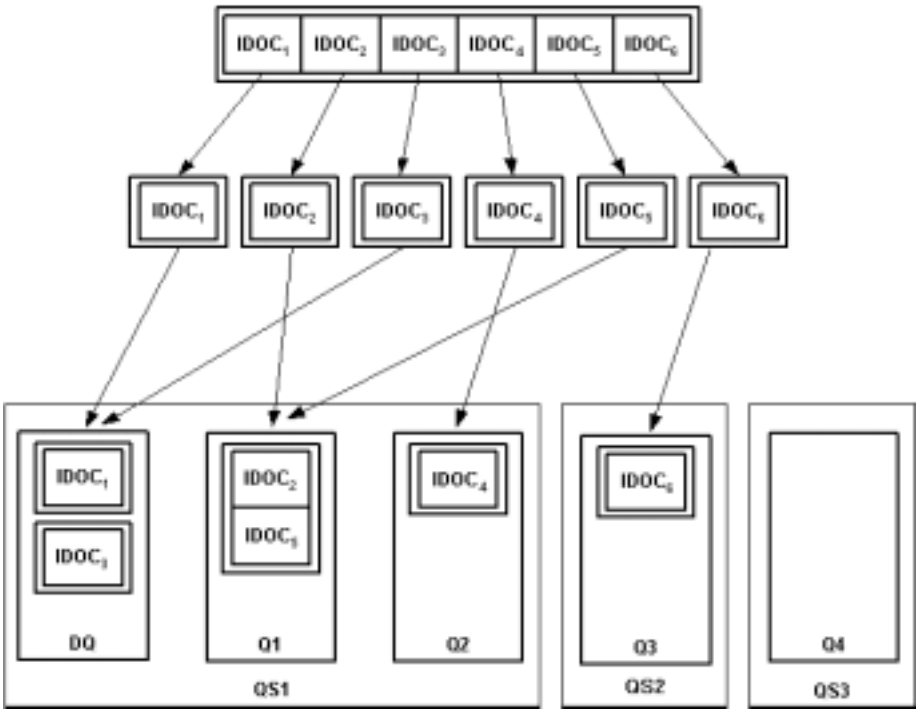
IDOC	LogicalId	MsgType
$IDOC_1$	LOGSYS1	CREMAS
$IDOC_2$	LOGSYS1	MATMAS

Table 2-5 Sample IDOC Packet Containing Multiple IDOCs

IDOC	LogicalId	MsgType
$IDOC_3$	LOGSYS3	MATMAS
$IDOC_4$	LOGSYS2	MATMAS
$IDOC_5$	LOGSYS1	MATMAS
$IDOC_6$	LOGSYS3	DEBMAS

Figure 2-12 shows how R/3-to-eLink would process this IDOC packet according to the settings in the environment file:

Figure 2-12 Queuing Multiple IDOCs According to the Environment File



R/3-to-eLink splits the IDOC packet into individual IDOCs and queues each IDOC in the following manner:

- R/3-to-eLink assigns $IDOC_1$ and $IDOC_3$ to the default queue, because no matching section was found in the environment file.
- R/3-to-eLink assigns $IDOC_2$, $IDOC_4$, $IDOC_5$, and $IDOC_6$ to their respective queues.
- R/3-to-eLink compresses $IDOC_2$ and $IDOC_5$ into a single IDOC packet in Q1, because `COMPRESS=Y` in the matching section of the environment file.

Managing Data-Dependent Routing (DDR)

You can configure the default data-dependent routing (DDR) value that R/3-to-eLink assigns to each IDOC message (in the `CR3_TARGET_ID` field) that it queues. R/3-to-eLink behaves as follows:

- If no matching section is found in the environment file for an IDOC, then R/3-to-eLink assigns a default DDR value to the IDOC messages (`CR3_ALE_DEFAULT_TARGET_ID` environment variable).
- If a matching section is found in the environment file for an IDOC (the logical ID and message type of the IDOC's control record match the logical ID and message type of the IDOC section), then R/3-to-eLink assigns the DDR value (TargetId) from the matching IDOC section of the environment file to the IDOC.

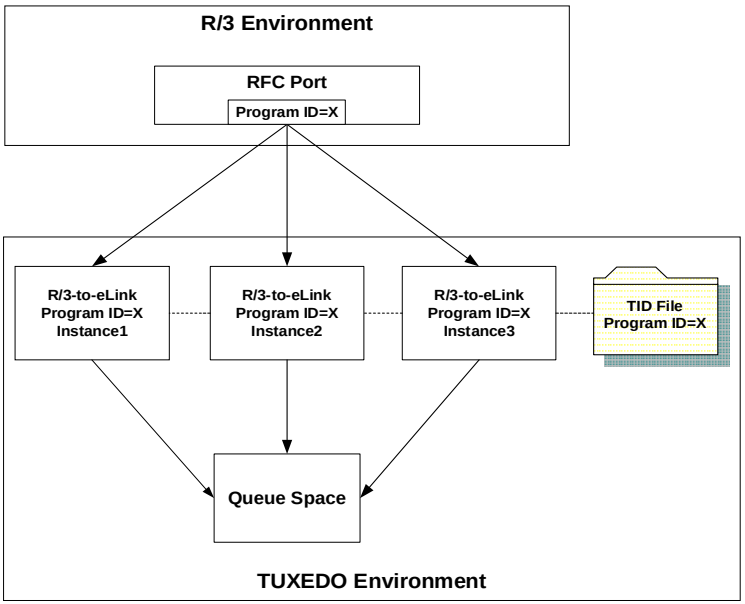
See "Setting the Default Data-Dependent Routing Value" and "Splitting R/3-to-eLink IDOC Packets" in Chapter 5, "Configuring ALE Integration," in this guide for instructions.

Load Balancing High Volumes of R/3-to-eLink IDOCs

Multiple instances of R/3-to-eLink can register using the same program ID. For deployments that involve high volumes of IDOC packets, you can enhance system performance by balancing the load across multiple instances of R/3-to-eLink.

Instances that register under the same program ID must also share the same TID file. Figure 2-13 shows multiple instances of R/3-to-eLink listening for IDOCs on the same program ID and sharing the same TID file:

Figure 2-13 Multiple Instances of R/3-to-eLink Sharing the Same Program ID



The number of R/3-to-eLink instances should match the anticipated number of IDOC packets that R/3 sends concurrently to port. For example, if R/3 sends five IDOC packets concurrently to port during peak loads, you should load five instances of R/3-to-eLink.

See “Configuring Load Balancing” in Chapter 6, Configuring ALE Integration,” in this guide.

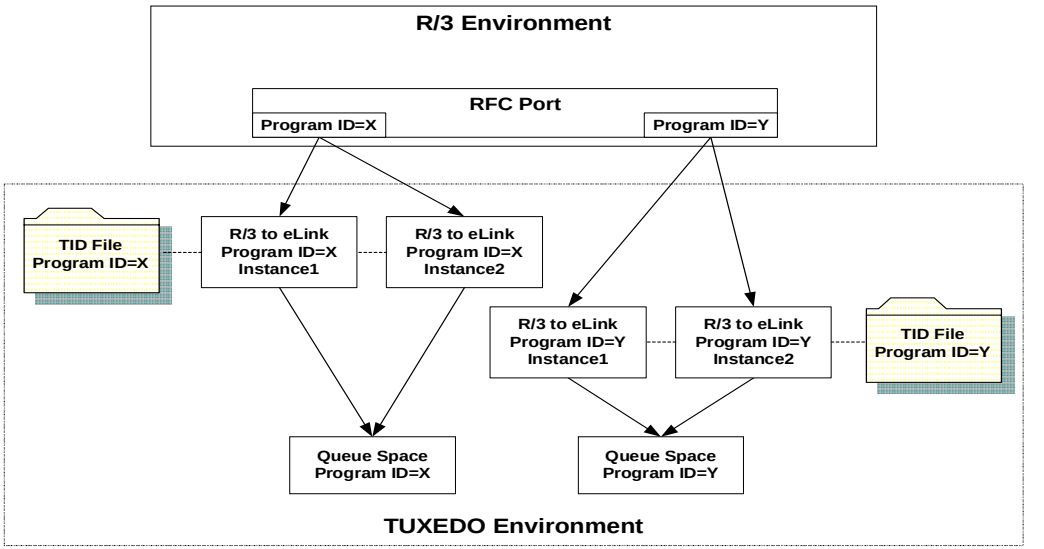
Registering Multiple Program IDs

If R/3 is configured to send R/3-to-eLink IDOCs to different program IDs, you can configure the R/3-to-eLink server to handle these IDOCs by running multiple instances of R/3-to-eLink using different program IDs. You must make sure that all instances sharing the same program ID also share the same TID file, and that all instances sharing the same TID file also share the same program ID.

Note: Instances that register under different program IDs must *not* share the same TID file.

Figure 2-14 shows two groups of instances of R/3-to-eLink, each of which is listening for IDOCs on a shared program ID and sharing the same TID file:

Figure 2-14 Multiple Instances of R/3-to-eLink Using Different Program IDs



See “Configuring Load Balancing for R/3-to-eLink” in Chapter 5, “Configuring ALE Integration,” of this guide.

Managing Transactional Integrity for R/3-to-eLink IDOCs

The R/3-to-eLink server manages transactional integrity for R/3-to-eLink IDOCs to ensure that an IDOC packet has been queued successfully. R/3-to-eLink uses a TID log file to track the transaction IDs (TIDs) associated with the IDOC packets it processes to ensure that it queues an IDOC packet from R/3 exactly once. See "Transaction IDs (TIDs)" later in this topic for an introduction to TIDs.

About the TID Log File Used for R/3-to-eLink IDOCs

R/3-to-eLink uses a TID file to track the IDOC packets it processes to ensure that it queues an IDOC packet once and only once. The R/3 System assigns a TID to each R/3-to-eLink IDOC packet.

The TID file for R/3-to-eLink is a log of all the TIDs that R/3-to-eLink has received and processed. Each row in the TID file represents the TID for a separate IDOC packet and contains three fixed-position columns of information:

Column	Description
Date-Time Stamp	Date and time at which the TID log file was last updated.
TID	TID that R/3 assigned to the IDOC packet.
State	The processing state. One of the following strings: ■ **CREATED** indicates that R/3-to-eLink has received the TID from R/3. ■ **EXECUTED** indicates that R/3-to-eLink has queued the IDOC message with the TID and has committed the transaction. ■ **ROLLBACK** indicates that R/3-to-eLink has rolled back the IDOC packet from the queue. ■ **CONFIRMED** indicates that R/3-to-eLink has confirmed that the IDOC message has been queued and the transaction has been committed.

The following example shows a sample TID file for R/3-to-eLink:

```
Tue Apr 27 14:27:36 1999  0A0201FD03F937262C5B0001  CONFIRMED
```

```
Tue Apr 27 14:29:38 1999  0A0201FD03E937262CD70004  CONFIRMED

Tue Apr 27 14:46:56 1999  0A0201FD03F9372630E60009  CONFIRMED

Tue Apr 27 15:50:21 1999  0A0201FD03E837263F98003F  CONFIRMED
```

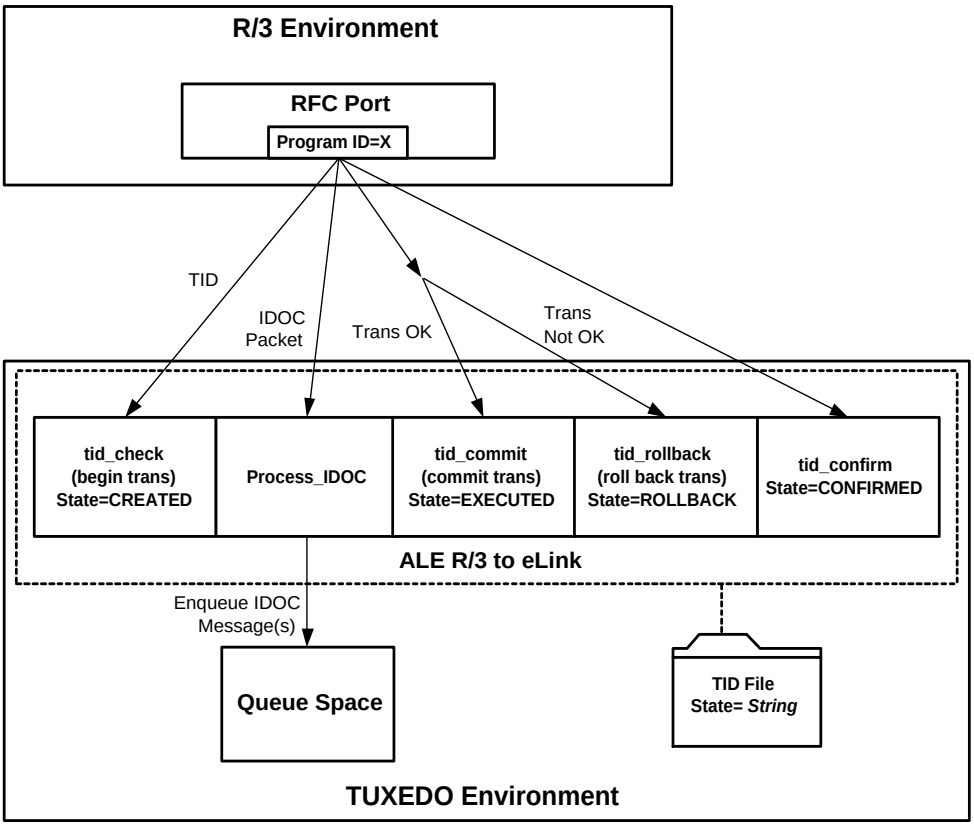
The `CR3_ALEOUT_TID_FILE` environment variable specifies the location of the TID log file for R/3-to-eLink. See "Setting Environment Variables for R/3-to-eLink" in Chapter 5, "Configuring ALE Integration," in this guide for more information.

Use the `cr3tidmanager` program to manage the size and number of entries kept in TID files. See "Configuring the TID File Manager" in Chapter 5, "Configuring ALE Integration," of this guide.

Processing TIDs with R/3-to-eLink IDOCs

Figure 2-15 shows how R/3-to-eLink uses the TID log file to manage transactional integrity for R/3-to-eLink IDOCs:

Figure 2-15 TID Processing for R/3-to-eLink



The information flow proceeds in the following sequence:

1. R/3 sends a TID to an instance of R/3-to-eLink that is registered on the matching program ID.
2. R/3-to-eLink receives the TID and checks the TID file to determine whether it has previously received this TID from R/3. If the TID is not found in the TID file, then R/3-to-eLink appends an entry to the TID file, specifying the date-time stamp, TID, and the state (CREATED). R/3-to-eLink returns a code to R/3 indicating whether the TID was found, and the TID state determines whether R/3 continues processing.
3. If R/3 continues processing, R/3-to-eLink starts a new transaction.

4. R/3 sends the IDOC packet associated with the TID to the same instance of R/3-to-eLink.
5. R/3-to-eLink receives the IDOC packet and processes the IDOC data according to the way that R/3-to-eLink is configured (such as splitting IDOC packets, making routing decisions based on the environment file, and so on). R/3-to-eLink encodes the IDOC data in one or more FML32 message buffers and queues the message(s) into one or more queues.
6. After processing the IDOC data, R/3-to-eLink returns success or an SAP exception (if, for example, the target queue is full) to R/3.
7. Based on the status returned from R/3-to-eLink, R/3 instructs the same instance of R/3-to-eLink to commit or roll back the transaction:
8. R/3-to-eLink takes one of the following actions:
   - R/3-to-eLink commits the transaction and updates the date-time stamp and state (EXECUTED) in the TID file.
   - R/3-to-eLink call rolls back the transaction and updates the date-time stamp and state (ROLLBACK) in the TID file.
9. If the transaction is successfully committed, R/3-to-eLink updates the date-time stamp and state (CONFIRMED) in the TID file.

Handling Problems with R/3-to-eLink IDOCs

R/3-to-eLink uses TUXEDO's transaction management capabilities to ensure transactional integrity for R/3-to-eLink IDOCs. Figure 2-6 lists problems that can occur with R/3-to-eLink IDOCs:

Table 2-6 Handling Problems with R/3-to-eLink IDOCs

Problem	Description
Unable to Lock the TID File	The TID file might be locked by another instance of R/3-to-eLink or the TID File Manager. R/3-to-eLink retries the lock attempt. After a configurable number of retry attempts, R/3-to-eLink returns a lock error to R/3. R/3 then attempts to retry the operation until it succeeds or stops trying.

Table 2-6 Handling Problems with R/3-to-eLink IDOCs

Problem	Description
Unable to Update the TID File	The file might be corrupted. If R/3-to-eLink can lock the TID file but cannot update it, R/3-to-eLink retries the lock attempt. After a configurable number of retry attempts, R/3-to-eLink returns a lock error to R/3. R/3 then attempts to retry the operation until it succeeds or stops trying.
R/3-to-eLink cannot queue an IDOC message(s)	One or more target queues might be full. R/3-to-eLink returns an SAP exception to R/3, and R/3 instructs R/3-to-eLink to roll back the transaction. R/3 will subsequently resubmit the IDOC packet to R/3-to-eLink.

3 Preinstallation Tasks

This topic describes important information that you need to review before installing BEA eLink Adapter for R/3 ALE. It includes the following main sections:

- Determining the Installation Node
- eLink Platform (TUXEDO) Requirements

Determining the Installation Node

Where you install eLink Adapter for R/3 ALE depends on the node configuration:

- **Single Node**. If the R/3 environment is a single node configuration, then you should install BEA eLink Adapter for R/3 ALE on that node.
- **Multiple Nodes**. If the R/3 environment is a multiple node configuration, then you should install BEA eLink Adapter for R/3 ALE on one or more of the application servers. BEA eLink Adapter for R/3 ALE must *not* be installed on the database server.

BEA eLink Adapter for R/3 ALE interacts with R/3 using the SAP RFC library. The RFC library uses IP sockets to communicate with the R/3 application servers. Installing BEA eLink Adapter for R/3 ALE on an R/3 application server node minimizes the risk of failure in this IP socket connection. If BEA eLink Adapter for R/3 ALE is installed on an R/3 application server node, it is recommended that no other TUXEDO applications run on that same node, in order to reduce the workload on the node.

However, BEA eLink Adapter for R/3 ALE can be installed on a node other than an R/3 application server. In this situation, the IP socket connection will transparently extend across the network to the R/3 application server.

eLink Platform (TUXEDO) Requirements

BEA eLink Adapter for R/3 ALE requires that you properly install and configure eLink Platform:

- **Installation Node**. eLink Platform must be installed on the node running BEA eLink Adapter for R/3 ALE. For installation instructions, see the *BEA eLink Platform Installation Guide.*
- **eLink Platform (TUXEDO) Configuration**. After you install BEA eLink Adapter for R/3 ALE, you must configure the eLink Platform according to the instructions in Appendix B, "Configuration Quick Reference." BEA eLink Adapter for R/3 ALE provides a semi-configured TUXEDO initialization file (`cr3_ale.ubb`), which requires only minimal configuration for the eLink environment.

4 Installing BEA eLink Adapter for R/3 ALE

This topic describes how to install BEA eLink Adapter for R/3 ALE on UNIX and Windows NT systems. It covers the following main sections:

- Installing on a UNIX System
- Installing on a Windows NT System

Installing on a UNIX System

Installing eLink Adapter for R/3 ALE on UNIX systems requires the three main steps:

- Creating the conr3 User
- Logging In as con3
- Running install.sh

Creating the conr3 User

A UNIX user with non-administrative privileges should be created on the R/3 application server to provide a context in which to run both eLink Platform (TUXEDO) and BEA eLink Adapter for R/3 ALE. This user should be used to run TUXEDO and BEA eLink Adapter for R/3 ALE *only* so that processes associated with TUXEDO and BEA eLink Adapter for R/3 ALE can be identified easily.

BEA suggests that you use `conr3` for the user name. This document refers to this UNIX user as `conr3`. If your organization uses a different user name, then substitute your user name whenever you encounter a reference to `conr3` in this document.

Note: BEA strongly recommends that you do *not* use the UNIX R/3 administration user (*nnn*adm, where *nnn* represents the R/3 system identifier) to run TUXEDO and BEA eLink Adapter for R/3 ALE.

Logging In as con3

After creating the `conr3` user name, log in as that user.

Running install.sh

Run the `install.sh` script to install the BEA eLink Adapter for R/3:

1. Insert the CD-ROM in the CD-ROM drive.

   If your system is not already configured to access data from a CD-ROM directory, you will need to create a mount directory (for example, `mkdir /cdrom`) and mount the CD-ROM device (with the `mount` command).

   Change directories to your CD-ROM directory using the following command:

   ```
   cd /cdrom
   ```

2. Type `ls` to view the contents of the CD.

   The CD should contain the following files and directories:

   ```
   ALPHA/   HP/   IBM/   install.sh   SUN5X/   WINNT/
   ```

3. Start the installation by typing the following at the command line prompt:

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

   Press **Enter**.

   This invokes the installation script.

   The UNIX system installation script provides a set of step-by-step instructions to help you quickly install the BEA eLink Adapter for R/3. This script lets you specify your platform, operating system, and the directory where you want to install. The installation script prompts you through the entire installation process. You can cancel the installation at any time by pressing `CTRL-C` simultaneously.

4. Type the number that corresponds to the name of the operating system you are using (for example, if using HPUX1020, type `2`).

   ```
   01) ALPHA/DUX40     02) HP/HPUX1020      03) HP/HPUX1100
   04) IBM/AIX431      05) SUN5X/SOL26      06) SUN5X/SOL27

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

   Press **Enter**.

5. You are prompted to confirm your choice. If correct, type `y` for "yes," or `n` for "no" or `q` to "quit." Press **Enter**.

   ```
   ** You have chosen to install from HP/HPUX1020 **

   BEA eLink Adapter for R/3 ALE Release 1.6

   This directory contains the BEA eLink Adapter for R/3 ALE for

   HP-UX 10.20 on 9000/800 series.

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

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

6. The script indicates which packages are available for the chosen operating system. Indicate which package to install and press **Enter**.

   ```
   The following packages are available:

   1  eLinkR3ALE  BEA eLink Adapter for R/3 ALE

   Select the package(s) you wish to install (or 'all' to install

   all packages) (default: all) [?,??,q]: 1
   ```

The following copyright information is displayed about the product you are about to install:

```
BEA eLink Adapter for R/3 ALE Release BEA eLink Adapter for R/3
ALE Release 1.6

Copyright (c) 2000 BEA Systems, Inc.

All Rights Reserved.

Distributed under license by BEA Systems, Inc.

BEA and eLink are trademarks of BEA Systems, Inc.
```

7. You are prompted for the directory location where you want to install the product. Type the install directory and press **Enter**.

   Note: It is recommended that you install BEA eLink Adapter for R3 under a directory named `eLink`. The host machine where you install and the prefix path to the `eLink` directory is up to you. For this example, we install on `/home/miller/elink/`.

   ```
   Directory where eLink files are to be installed [?,q]:
   /home/miller/elink
   ```

8. Watch the screen messages to verify that the installation is successful. The messages will be similar to the following.

   ```
   Creating /home/miller/elink

   Using /home/miller/elink as the eLink base directory

   Creating /home/miller/elink/adapters

   Creating /home/miller/elink/adapters/sapr3_ale

   Determining if sufficient space is available ...

   5320 blocks are required

   720596 blocks are available to
   /home/miller/elink/adapters/sapr3_ale

   Using /home/miller/elink/adapters/sapr3_ale

   as the eLink Adapter for R/3 ALE install directory

   Unloading /cdrom/HP/HPUX1020/ELINKR3/ELINKR3.Z ...

   bin/cr3alein

   bin/cr3aleout

   bin/cr3tidmanager

   config/cr3_queues.sh
   ```

```
config/cr3_tlog.sh
config/setenv_ale.sh
config/cr3alein.env
config/cr3aleout.env
config/cr3_ale.ubb
config/cr3_ale.fml
config/sideinfo
5320 blocks
... finished
Installation of BEA eLink Adapter for R/3 ALE was successful
The following packages are available:
  1  eLinkR3ALE  BEA eLink Adapter for R/3 ALE
```

9. When the installation completes successfully, exit the install script. (Or proceed with another installation, as appropriate.)

```
Select the package(s) you wish to install (or 'all' to install
all packages) (default: all) [?,??,q]: q
Please don't forget to fill out and send in your registration
card #
```

Table 4-1 describes the files and directories that are installed for BEA eLink Adapter for R3 on a UNIX system.

Table 4-1 Directory Structure of BEA eLink Adapter for R/3 ALE on a UNIX System

File or Directory Name	Description
`bin/cr3alein`	eLink-to-R/3 executable
`bin/cr3aleout`	R/3-to-eLink executable
`bin/cr3tidmanager`	TID manager executable
`config/cr3_queues.sh`	Semi-configured script for creating the TUXEDO queue space and queues
`config/cr3_tlog.sh`	Semi-configured script for creating the TUXEDO `tlog` file

Table 4-1 Directory Structure of BEA eLink Adapter for R/3 ALE on a UNIX System

File or Directory Name	Description
`config/setenv_ale.sh`	Semi-configured setup file for generic eLink for R/3 environment variables
`config/cr3alein.env`	Environment file for eLink-to-R/3
`config/cr3aleout.env`	Environment file for R/3-to-eLink
`config/cr3_ale.ubb`	TUXEDO UBB configuration file
`config/cr3_ale.fml`	CR3 FML field table
`config/sideinfo`	Example R/3 sideinfo file

Installing on a Windows NT System

The BEA eLink Adapter for R/3 product install program for the Windows NT platform is located in the `winnt\setup.exe` file on the CD-ROM.

To install the product, insert the CD into the CD-ROM drive and double click on the `winnt\setup.exe` file to launch the installation wizard.

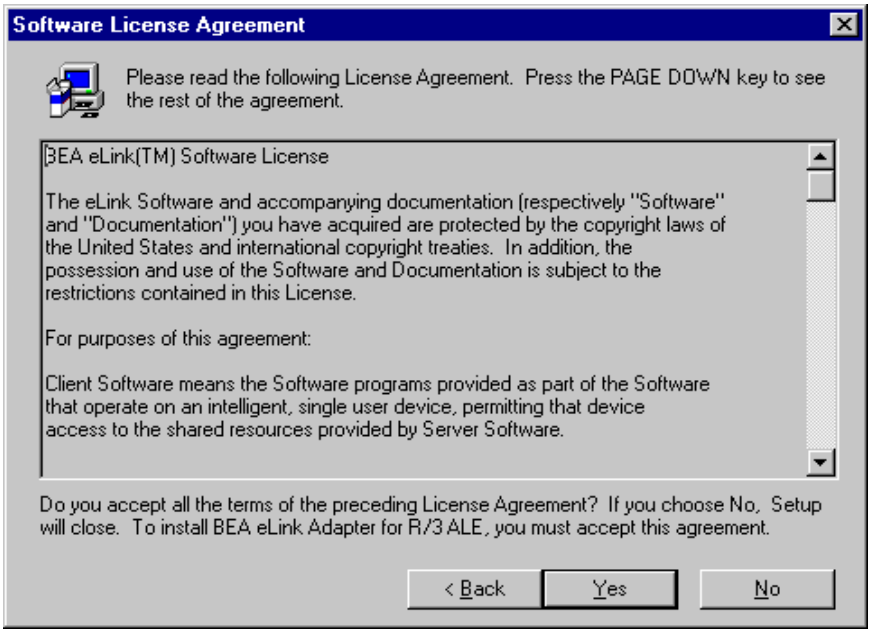
When the installation program launches, follow the prompts shown in the windows as described.

1. The Welcome window describes the product you are about to install. Click **Next** to continue (or click **Cancel** to cancel the installation process).

Figure 4-1 Welcome to BEA eLink Adapter for R3 Setup Program

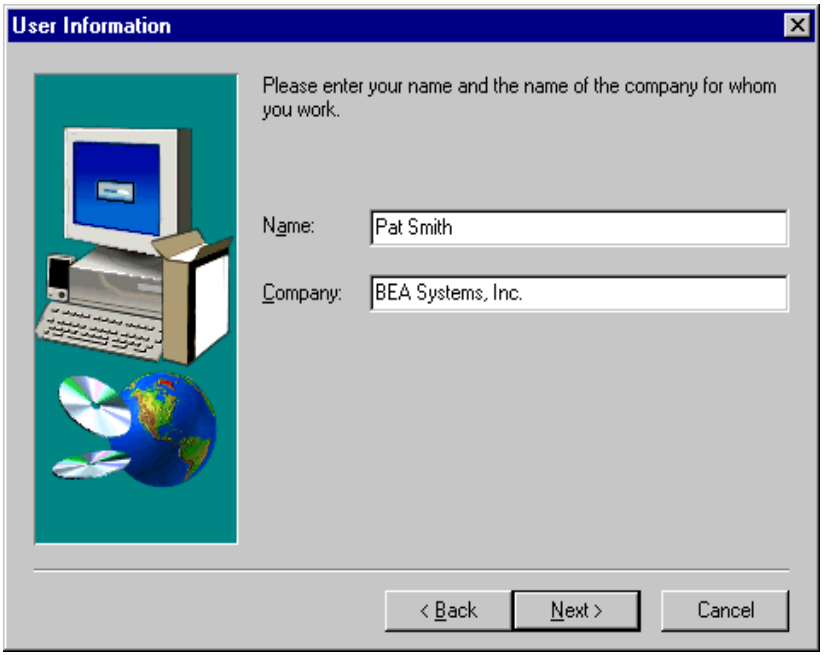
2. In order to proceed with the installation, you must accept the license agreement. Click **Yes** to accept (or click **No** to cancel the installation process).

Figure 4-2 License Agreement



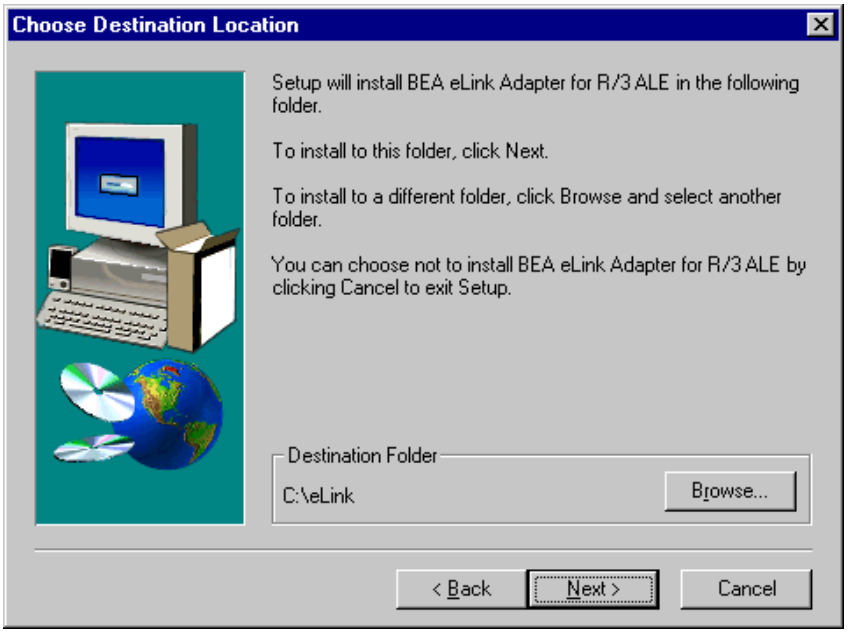
3. After you accept the license agreement, the User Information window is displayed. Provide the requested information, and click **Next** to continue (or **Back** to return to a previous window).

Figure 4-3 Provide User Information



4. The path shown for the "Destination Folder" on the Choose Destination Location window is the location where the BEA eLink product(s) will be installed. The default Destination Folder is *YourDrive*:\eLink\ as shown in Figure 4-4. To accept the default, click **Next** to continue. Otherwise, use the file browser to choose a different destination folder as explained below.

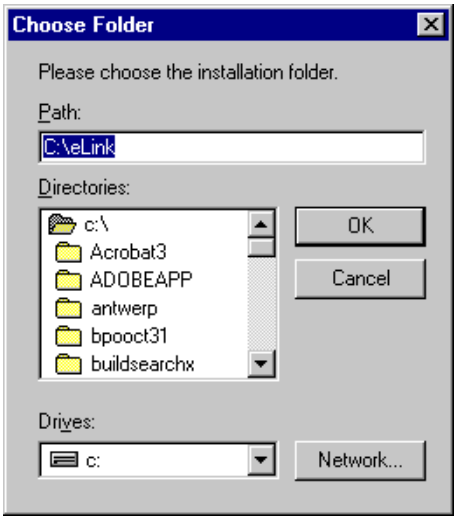
Figure 4-4 Choose Directory Location for Product Installation



If you want to install the BEA eLink product in a folder other than the default, click the **Browse** button to get the file browser. Use the file browser to locate the directory in which you want to install the BEA eLink Adapter for R/3 ALE and click **OK** to specify this new path name as the Destination Folder.

Note: It is recommended that you install BEA eLink Adapter for R/3 ALE under a directory named `eLink`. The host machine where you install and the prefix path to the `eLink` directory is up to you. For this example, we install on `C:\eLink\`. Installing on `C:\eLink\` actually installs the configuration files to the `C:\eLink\adapters\sapr3_ale\config` directory and the executable files and dynamic link library to the `C:\eLink\adapters\sapr3_ale\bin` directory.

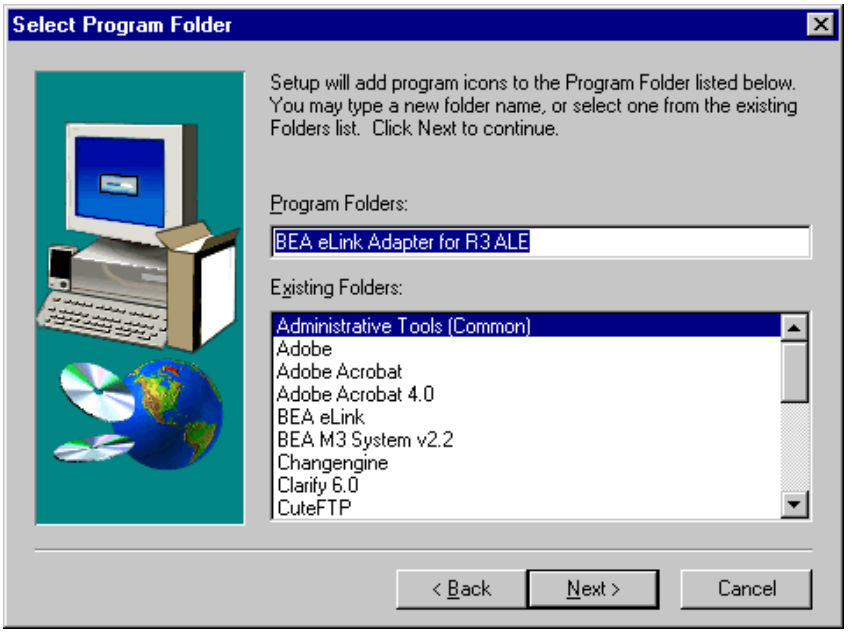
Figure 4-5 File Browser for Choosing an Install Location



If you do change the Destination Folder to something other than the default, you should now see the new path name reflected in the Choose Folder window.

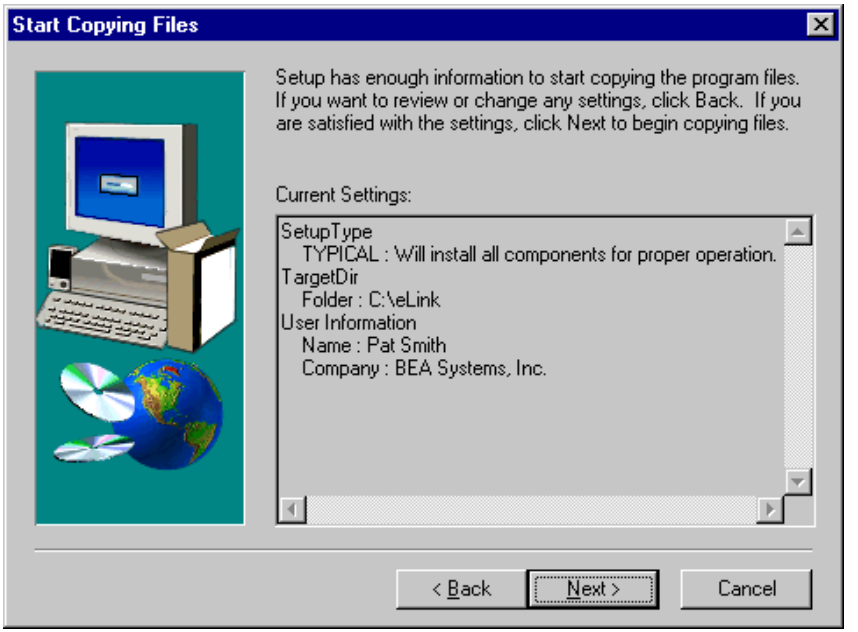
5. Select the program folder where you want to store the startup icon for BEA eLink Adapter for R/3 ALE. The default folder is already chosen for you. If you want to choose a different folder, do so. Click **Next** to continue (or **Back** to return to a previous window).

Figure 4-6 Select a Program Folder



6. If you are satisfied with the current settings, click **Next** to proceed with the installation. (Otherwise, click **Back** to return to a previous window.)

Figure 4-7 Current Settings



7. When all files have been successfully copied to the destination folders, the Setup Complete window is displayed. Click **Finish** to exit the installation.

Figure 4-8 Setup Complete

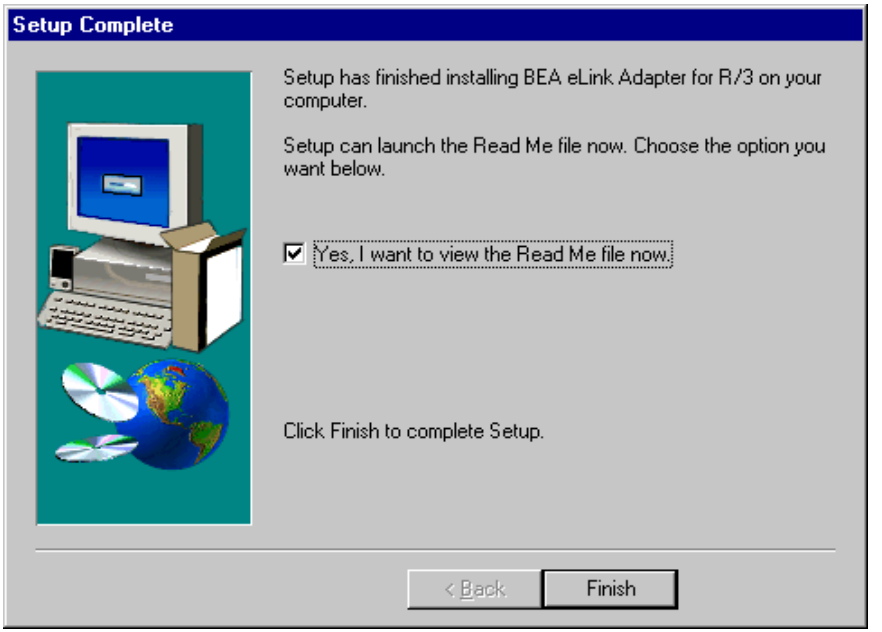


Table 4-2 describes the files and folders that are installed for BEA eLink Adapter for R/3 ALE on a Windows NT system.

Table 4-2 Directory Structure of BEA eLink Adapter for R/3 ALE on Windows NT

File or Folder Name	Description
`eLink\adapters\sapr3_ale\bin\`	Directory containing executables
`eLink\adapters\sapr3_ale\bin\cr3alein.exe`	eLink-to-R/3 executable
`eLink\adapters\sapr3_ale\bin\cr3aleout.exe`	R/3-to-eLink executable
`eLink\adapters\sapr3_ale\bin\cr3tidmanager.exe`	TID manager executable
`elink\adapters\sapr3_ale\bin\librfc.dll`	R/3 RFC Dynamic Link Library

Table 4-2 Directory Structure of BEA eLink Adapter for R/3 ALE on Windows NT

File or Folder Name	Description
`eLink\adapters\sapr3_ale\config`	Directory containing configuration files
`eLink\adapters\sapr3_ale\config\cr3_queues.nt`	Semi-configured makefile for creating TUXEDO queue space and queues
`eLink\adapters\sapr3_ale\config\cr3_tlog.nt`	Semi-configured makefile for creating the TLOG file
`eLink\adapters\sapr3_ale\config\setenv_ale.bat`	Semi-configured setup file for generic eLink for R3 environment variables
`eLink\adapters\sapr3_ale\config\cr3alein.env`	Environment file for eLink-to-R/3
`eLink\adapters\sapr3_ale\config\cr3aleout.env`	Environment file for R/3-to-eLink
`eLink\config\cr3_ale.ubb`	TUXEDO UBB configuration file
`eLink\adapters\sapr3_ale\config\cr3_ale.fml`	CR3 FML field table
`eLink\adapters\sapr3_ale\config\sideinfo`	Example R/3 sideinfo file

5 Configuring ALE Integration

This topic describes how to configure your BEA eLink Adapter for R/3 ALE installation for integration with SAP application linking and embedding (ALE) technology. It includes the following main sections:

- Configuring the eLink-to-R/3 Server
- Configuring the R/3-to-eLink Server
- Configuring Queues
- Configuring the TID File Manager
- Shutting Down the BEA eLink Adapter for R/3 ALE

Note: You need to complete the tasks in this section only if you require application integration with ALE.

Using BEA eLink Adapter for R/3 ALE, TUXEDO applications can exchange SAP Intermediate Documents (IDOCs) with R/3 Systems. See Chapter 2, “Integrating with ALE,” in this guide for more information about ALE integration.

Configuring the eLink-to-R/3 Server

The following topics describe how to configure the eLink-to-R/3 server:

- Configuring the UBB File for eLink-to-R/3
- Setting Environment Variables for eLink-to-R/3

In addition, see the following topics for further configuration instructions:

- Setting Up Queues for eLink-to-R/3 IDOCs
- "Configuring eLink-to-R/3 Connections" in Chapter 6, "Configuring R/3 Connections"

eLink-to-R/3 is a TUXEDO server that submits IDOC packets to R/3 via tRFC. eLink-to-R/3 receives IDOC packets forwarded from a TUXEDO queue, encodes the IDOC data for R/3, and submits the message to R/3 via tRFC. See "Processing eLink-to-R/3 IDOCs" in Chapter 2, "Integrating with ALE," of this guide for more information about eLink-to-R/3.

Configuring the UBB File for eLink-to-R/3

eLink-to-R/3 is a standard TUXEDO server that is defined in the UBB file. You must configure this file to specify the server group, server, environment file, and services associated with the eLink-to-R/3 server. See Appendix A, "Sample cr3_ale.ubb File," for a listing of the sample UBB file that comes with BEA eLink Adapter for R/3 ALE.

Defining the eLink-to-R/3 server (`cr3alein`)

In the SERVERS section of your UBB file, you must specify the eLink-to-R/3 server as `cr3alein`, then define it in the succeeding code lines by specifying its server group, server ID, and environment file

Do this using the following syntax:

```
cr3alein
       SRVGRP=GroupId SRVID=ServerID
       CLOPT = "-A -- -i unique_id -e env_filename"
```

where:

- *GroupID* is the unique ID of the server group to which the eLink-to-R/3 server belongs.
- *ServerID* is the unique ID of the eLink-to-R/3 server.
- -i *unique_id* is the unique ID that identifies the label section in the environment file containing the adapter's environment variables. You would use this value when you run multiple eLink adapters that share this configuration file. If it is not specified, then the unique ID defaults to the process name.
- -e *env_filename* is the environment file that specifies the detailed configuration information.

Listing 5-1 shows SERVERS settings for the eLink-to-R/3 server (cr3alein) in a sample UBB file for Windows NT:

Listing 5-1 Sample SERVERS Settings for eLink-to-R/3 Server

```
cr3alein
       SRVGRP=CR3 SRVID=3
       CLOPT="-A -- -i cr3alein -e cr3alein.env"
```

Defining eLink-to-R/3 Services

In the SERVICES section of your UBB file, define the CR3_SUBMIT and CR3_IDOC_IN services, as shown in the following example:

```
CR3_SUBMIT
CR3_IDOC_IN
```

CR3_SUBMIT checks the validity of the data meant for R/3. CR3_IDOC_IN sends it to R/3.

Setting Environment Variables for eLink-to-R/3

The eLink-to-R/3 server requires environment variables that specify connection information and the behavior of the server to R/3. These environment variables are defined in an environment file that is specified in the UBB file as the ENVFILE parameter for the server.

Environment Variables for eLink-to-R/3

Configure the required and optional environment variables for eLink-to-R/3 as shown in Table 6-1:

Table 5-1 Environment Variables for the eLink-to-R/3 Server

Category / Variable	Set To	Description
Connecting To R/3		
CR3_DESTINATION	System Name	Destination in the sideinfo that defines the R/3 application server to connect to. Default is undefined. This setting is case sensitive. *Required*.
CR3_CLIENT	Client Number	R/3 login client number. Default is undefined. *Required*.
CR3_USER	User	R/3 login user. Default is undefined. The user must be of type CPIC. *Required*.
CR3_PASSWORD	Password	R/3 login user password. Default is undefined. *Required*.
CR3_LANGUAGE	E	R/3 login language. Default is E for English. *Required*.
SIDE_INFO	Path and File	Full path to the sideinfo file. *Required*.
Managing the TID File		
CR3_ALEIN_TID_FILE	Path and File	Full path and filename for the eLink-to-R/3 TID file, *without* the .tid file extension. *Required.*

Table 5-1 Environment Variables for the eLink-to-R/3 Server

Category / Variable	Set To	Description
CR3_ALEIN_TID_PER_IDOC	Y or N	Specify "Y" to split IDOC packets containing multiple IDOCs and pass them as individual IDOCs to R/3. Specify "N" (the default) to pass multiple IDOCs to R/3 in one batch. "N," the default is also preferred because, very seldom, is there any need to split inbound IDOCs. *Optional*.
Managing Queues		
CR3_ALEIN_IDOCIN_QUEUE	Queue Name	Name of the queue for messages containing processed IDOCs and their corresponding TIDs. By default, should be set to CR3_IDOC_IN. TMQFORWARD dequeues messages from this queue and forwards them to the CR3_IDOC_IN service for submission to R/3. *Required*.
CR3_ALEIN_IDOCIN_QUEUE_SPACE	Queue Space	Queue space that contains the CR3_IDOC_IN queue. *Required*.
CR3_ALEIN_ERROR_QUEUE	Queue Name	Name of the queue for messages containing IDOCs that the CR3_SUBMIT has received and rejected due to IDOC format or consistency errors. By default, should be set to CR3_ERROR. *Required*.
CR3_ALEIN_ERROR_QUEUE_SPACE	Queue Space	Queue space that contains the CR3_ERROR_QUEUE. *Required*.
Logging		
CR3_ALEIN_PRINT_FILE	Path and File	Specifies the full path and file for logging and tracing output. If this variable is not used (value is null), output is sent to the TUXEDO USERLOG. *Optional*.
CR3_TRACE	Y or N	Specify "Y" to include detailed trace information in the log output. *Optional.*

Sample Environment File for eLink-to-R/3 (cr3alein.env)

BEA eLink Adapter for R/3 ALE provides a semi-configured TUXEDO environment file (`cr3alein.env`) for use with the eLink-to-R/3 server. You should create a separate environment file for each eLink-to-R/3 server specified in the `cr3_ale.ubb` configuration file.

Note the settings in the sample `cr3alein.env` file:

Listing 5-2 Sample Environment File (cr3alein.env)

```
CR3_DESTINATION=<R/3 system name>
CR3_CLIENT=<R/3 client id>
CR3_USER=<R/3 user>
CR3_PASSWORD=<R/3 password>
CR3_LANGUAGE=E
CR3_ALEIN_TID_FILE=<your eLink app directory>\cr3alein
CR3_ALEIN_TID_PER_IDOC=N
CR3_ALEIN_IDOCIN_QUEUE=CR3_IDOC_IN
CR3_ALEIN_IDOCIN_QUEUE_SPACE=QSPACE
CR3_ALEIN_ERROR_QUEUE=CR3_ERRQUE
CR3_ALEIN_ERROR_QUEUE_SPACE=QSPACE
CR3_ALEIN_PRINT_FILE=<your eLink app directory>\cr3alein.log
CR3_ALEIN_TRACE=Y
```

Note: You can use any directory for the eLink application directory. However, it is recommended that you use the same directory as the TUXEDO APPDIR.

Setting Up the eLink-to-R/3 Connection to R/3

To set up the eLink-to-R/3 connection to R/3, you must set environment variables and configure the `sideinfo` file. See “Configuring eLink-to-R/3 Connections” in Chapter 6, “Configuring R/3 Connections” for instructions.

Configuring Queues for eLink-to-R/3

eLink-to-R/3 requires that you set up two TUXEDO queues, `CR3_ALE_ERROR_QUEUE` and `CR3_ALE_IDOC_IN_QUEUE`. See “Setting Up Queues for eLink-to-R/3 IDOCs” later in this topic for instructions.

Specifying the TID File for eLink-to-R/3

eLink-to-R/3 uses a TID file to log TIDs that it has already processed to guarantee that it delivers an IDOC packet to R/3 once and only once. You must specify the path and file name of the TID file (*without* an extension) in the `CR3_ALEIN_TID_FILE` environment variable, as shown in the following example:

```
CR3_ALEIN_TID_FILE=<your eLink app directory>\cr3alein
```

Upon start-up, if the eLink-to-R/3 server does not find a TID file in the specified location, it creates one automatically. See "Configuring the TID File Manager" on page 5-27 for information about configuring and administering the TID file.

Splitting eLink-to-R/3 IDOC Packets Containing Multiple IDOCs

By default, eLink-to-R/3 passes an IDOC message containing multiple IDOCs to R/3 in a single packet. You can configure eLink-to-R/3 to split IDOC messages containing multiple IDOCs into individual IDOC messages, each with its own TID. For example, if an IDOC message contains six IDOCs, eLink-to-R/3 can create six IDOC separate packets, each containing a single IDOC and associated with a unique TID.

To split eLink-to-R/3 IDOC packets, set the `CR3_ALEIN_TID_PER_IDOC` environment variable to "Y", as shown in the following example:

```
CR3_ALEIN_TID_PER_IDOC=Y
```

See "Splitting eLink-to-R/3 IDOC Packets Containing Multiple IDOCs" in Chapter 2, "Integrating with ALE," of this guide for more information.

Configuring Logging for eLink-to-R/3

By default, eLink-to-R/3 writes logging output to the TUXEDO USERLOG. If sending output to a file, you must specify its name and location in the `CR3_ALEIN_PRINT_FILE` environment variable.

You can also include trace information in the log by setting the `CR3_TRACE` environment variable to "Y". You exclude trace information by omitting this environment variable or setting it to "N".

The following example sets logging options for eLink-to-R/3:

```
CR3_ALEIN_PRINT_FILE=<your eLink app directory>\cr3alein.log
CR3_TRACE=Y
```

Configuring the R/3-to-eLink Server

The R/3-to-eLink is a TUXEDO server. Configuration is required in the TUXEDO UBB file to run the R/3-to-eLink server within the TUXEDO domain. An environment file will define detailed configuration information required by the R/3-to-eLink.

The following topics describe how to configure the R/3-to-eLink server:

- Configuring the UBB File for R/3-to-eLink
- Setting Environment Variables for R/3-to-eLink
- Configuring Load Balancing for R/3-to-eLink
- Configuring Multiple Program IDs

See "Setting Up Queues for R/3-to-eLink IDOCs" later in this topic for configuration instructions.

The R/3-to-eLink server receives R/3-to-eLink IDOC packets from R/3 via Transactional RFC (tRFC); encodes each IDOC packet to an FML32 message buffer; and queues the IDOC message into a TUXEDO queue for subsequent processing. See "Processing R/3-to-eLink IDOCs" in Chapter 2, "Integrating with ALE," of this guide for more information about R/3-to-eLink.

Configuring the UBB File for R/3-to-eLink

R/3-to-eLink is a standard TUXEDO server that is defined in the UBB file. You must configure this file to specify the server group, server, and environment file associated with the R/3-to-eLink server. See Appendix A, "Sample cr3_ale.ubb File," for a listing of the sample UBB file that comes with the BEA eLink Adapter for R/3 ALE.

Defining the R/3-to-eLink Server (cr3aleout)

In the SERVERS section of your UBB file, you must specify the R/3-to-eLink server as `cr3aleout`, then define it in the succeeding code lines by specifying its server group, server ID, and environment file.

Do this using the following syntax:

```
cr3aleout
       SRVGRP=GroupId SRVID=ServerID
       CLOPT = "-A -- -i unique_id -e env_filename"
```

where:

- *GroupID* is the unique ID of the server group to which the R/3-to-eLink server belongs.
- *ServerID* is the unique ID of the R/3-to-eLink server.
- `-i` *unique_id* is the unique ID that identifies the label section in the environment file containing the adapter's environment variables. You would use this value when you run multiple eLink adapters that share this configuration file. If it is not specified, then the unique ID defaults to the process name.
- `-e` *env_filename* is the environment file that specifies the environment for the R/3-to-eLink server.

Listing 5-3 shows SERVERS settings for the R/3-to-eLink server (`cr3aleout`) in a sample UBB file for Windows NT:

Listing 5-3 Sample SERVERS Settings for R/3-to-eLink Server

```
cr3aleout
       SRVGRP=CR3 SRVID=4
       CLOPT="-A -- -i cr3aleout -e cr3aleout.env"
```

Setting Environment Variables for R/3-to-eLink

The detailed configuration information for the R/3-to-eLink is specified in an environment file. The configuration information is grouped into the following tables. Users must provide server connection variables, client connection variables, generic configuration variables, and some additional variables based on different TUXEDO actions.

The environment variables for an adapter can be specified in an environment file individually or as part of a combined environment file with environment variables for other adapters. See the "Description" for the [SERVER=..] adapter unique ID variable in Table 5-7 for more information.

Adapter Unique ID Variable

You can configure the following adapter unique ID variable for the R/3-to-eLink server:

Table 5-2 Adapter Unique ID Variable for the R/3-to-eLink Server

Task / Variable	Value	Description
[SERVER=..]	String The adapter unique ID will either be the value of the -i flag on the CLOPT line, or if not specified, then the default value of the process name.	Adapter unique id. The label section defines all the environment variables for this adapter with this adapter unique id. If the environment file specifies environment variables for more than one adapter, the [SERVER=..] environment variable must be presented to distinguish the environment variables for this adapter with this adapter unique id from the environment variables for other adapters. *Required.*

Server Connection Variables

You can configure the following server connection variables for the R/3-to-eLink server:

Table 5-3 Server Connection Variables for the R/3-to-eLink Server

Task / Variable	Value	Description
CR3_HOST	String	Destination switch in the sideinfo file. This setting is case sensitive. *Required.*
CR3_SOCKET	String	R/3 socket. *Required.*

Table 5-3 Server Connection Variables for the R/3-to-eLink Server

Task / Variable	Value	Description
CR3_PROGRAM_ID	String	R/3 program ID. *Required.*

R/3 Client Connection Variables

You can configure the following client connection variables for the R/3-to-eLink server:

Table 5-4 Client Connection Variables for the R/3-to-eLink Server

Task / Variable	Value	Description
CR3_DESTINATION	String	Destination switch in the sideinfo file. This setting is case sensitive. *Required.*
CR3_CLIENT	String	R/3 client. *Required.*
CR3_USER	String	R/3 user. *Required.*
CR3_PASSWORD	String	R/3 password. *Required.*
CR3_LANGUAGE	String	R/3 logon language. E for English is the default. This setting should not be changed because no other language is presently supported. *Required.*
SIDE_INFO	String	Path and file name of sideinfo file. This file defines the R/3 connection information that is used by the ALE library. Note: R/3-to-eLink will not read this environment variable; it is read by the ALE library — it is documented here for completeness only. *Required.*

Other Required and Optional Environment File Variables

You can configure the following required and optional environment variables for R/3-to-eLink:

Table 5-5 Environment Variables for the R/3-to-eLink Server

Variable	Set To	Description
Configuring IDOC Routing		
CR3_ALE_DEFAULT_TARGET_ID	Data-dependent routing (DDR) value	Default data-dependent routing value that is encoded in the CR3_TARGET_ID field in the FML32 message (if it cannot be determined for a particular IDOC in the environment file). This environment variable *must* be defined. *Required.*
Splitting IDOC Packets		
CR3_ALE_DEFAULT_IDOC_SPLIT	Y or N	If an IDOC section cannot be found that matches the outbound IDOC's control record with respect to logical ID and message type, this setting specifies whether IDOC packets containing multiple IDOCs are passed either together in a single message (if variable = "N") or split into separate IDOC packets, with one IDOC per message (if variable = "Y"). *Optional*.
Configuring Queues		
CR3_ALE_TARGET_QUEUE	Queue	Name of the default target TUXEDO queue. This is used if no IDOC section of the environment file matches the outbound IDOC with respect to logical ID and message type. *Required.*
CR3_ALE_TARGET_QUEUE_SPACE	Queue Space	Name of the TUXEDO queue space containing the default target queue. *Required.*
Configuring the TID Log File		
CR3_ALEOUT_TID_FILE	Path and File Name	Name and location of the TID file. The file name must *not* include an extension. *Required.*

Table 5-5 Environment Variables for the R/3-to-eLink Server

Variable	Set To	Description
Logging Options		
CR3_ALEOUT_PRINT_FILE	Path and File	Specifies the full path and file for logging and tracing output. If this variable is not used (value is null), output is sent to the TUXEDO USERLOG. *Optional*.
CR3_TRACE	Y of N	Specify "Y" to include detailed trace information in the log output. *Optional.*

These environment variables are defined in the server startup program.

Generic Configuration Variables

You can configure the following generic configuration variables for the R/3-to-eLink server:

Table 5-6 Generic Configuration Variables for the R/3-to-eLink Server

Task / Variable	Value	Description
CR3_EXIT_R3_CONNECT_LOSS	Y or N	Specifies whether R/3-to-eLink is to exit when it detects that it has lost the ALE connection to R/3. If not specified or defined as a value other than Y, then R/3-to-eLink will not exit upon losing the ALE connection and will try to restore it. *Required.*

Table 5-6 Generic Configuration Variables for the R/3-to-eLink Server

Task / Variable	Value	Description
CR3_CONNECT_RETRY_ ATTEMPTS	Number	Number of connection retry attempts. If CR3_EXIT_R3_CONNECT_LOSS = "Y," the adapter will exit no matter what the value is for this variable. If CR3_EXIT_R3_CONNECT_LOSS = "N," then there are two options: 1) CR3_CONNECT_RETRY_ ATTEMPTS (this variable) and CR3_CONNECT_RETRY_ INTERVAL (next variable) have null values. With this option, the adapter will continue trying to make a connection until it is successful. 2) If values are provided for CR3_CONNECT_RETRY_ ATTEMPTS (this variable) and CR3_CONNECT_RETRY_ INTERVAL (next variable), the connection is attempted according to these values. *Required.*

Table 5-6 Generic Configuration Variables for the R/3-to-eLink Server

Task / Variable	Value	Description
CR3_CONNECT_RETRY_ INTERVAL	Number with minimum 1	Time is seconds. It is the length of time the system waits before attempting to make another connection to R/3. If CR3_EXIT_R3_CONNECT_LOSS = "Y," the adapter will exit no matter what the value is for this variable. If CR3_EXIT_R3_CONNECT_LOSS = "N," then there are two options: 1) CR3_CONNECT_RETRY_ ATTEMPTS (this variable) and CR3_CONNECT_RETRY_ INTERVAL (next variable) have null values. With this option, the adapter will continue trying to make a connection until it is successful. 2) User can provide values for CR3_CONNECT_RETRY_ ATTEMPTS (this variable) and CR3_CONNECT_RETRY_ INTERVAL (next variable), the connection is attempted according to these values. *Required.*

These environment variables are defined in the server startup program.

Configuration Variables for Destination Mapping

These configuration variables specify IDOC destination mapping so that R/3-to-eLink can queue IDOC messages into different target queues, manage data-dependent routing, and group similar IDOC messages into larger IDOC messages. R/3-to-eLink makes routing and grouping decisions about individual IDOCs according to two settings specified in an IDOC's control record: the logical system ID of the target R/3 System and the IDOC message type. If the logical system ID of the target R/3 System and the IDOC message type specified in an IDOC section (which starts with

[CR3_IDOC=..].) match these same settings specified in an IDOC's control record, R/3-to-eLink automatically splits IDOC packets containing multiple IDOCs into separate IDOC messages, with each message containing a single IDOC.

Table 5-7 Configuration Variables for Destination Mapping

Task / Variable	Value	Description
CR3_IDOC_LIST	String	Comma-separated list of all IDOC names.
[CR3_IDOC=..]	String	IDOC Name. The six configuration variables immediately below the bracketed line specify the destination mapping for IDOCs with this name.
RECIEVER_PARTNER_ NUMBER	String	Logical System ID of the target R/3 logical system.
MESSAGE_TYPE	String	Message type specified in the IDOC.
COMPRESS	Y or N	If Compress=Y, then R/3-to-eLink concatenates IDOC packets with matching logical system and IDOC message type values into a single IDOC message in the queue. If Compress=N, then R/3-to-eLink keeps the IDOCs in individual IDOC messages.
ROUTING	String	Data-dependent routing (DDR) value. The value is encoded in the FML field CR3_TARGET_ID for use with DDR.
QUEUE_SPACE	String	Name of the destination queue space for the IDOC message.
QUEUE_NAME	String	Name of the destination queue for the IDOC message.

Sample Destination Mapping Settings

Listing 5-4 shows destination mapping settings for the IDOCs in a sample cr3aleout.env file for Windows NT:

Listing 5-4 Sample Destination Mapping Settings for Outbound IDOCs

```
CR3_IDOC_LIST=MaterialMaster
[CR3_IDOC=MaterialMaster]
RECIEVER_PARTNER_NUMBER=ALETEST01
MESSAGE_TYPE=MATMAS
COMPRESS=N
ROUTING=CR3_ERROR
QUEUE_SPACE=QSPACE
QUEUE_NAME=CR3_ERRQUE
```

Specifying Default Target Queue Spaces and Queue Names in the Environment File

You use the QUEUE_SPACE and QUEUE_NAME settings in the environment file to associate an IDOC with a destination TUXEDO queue space and queue. If an IDOC section in the environment file matches the logical system ID of the target R/3 System and the IDOC message type specified in an IDOC's control record, then R/3-to-eLink queues the IDOC into the specified queue space and queue name. If no matching combination of logical system ID and IDOC message type is found in an IDOC section of the environment file, then R/3-to-eLink queues the IDOC message into the default queue space and queue, which are specified in the CR3_ALE_TARGET_QUEUE_SPACE and CR3_ALE_TARGET_QUEUE environment variables.

Note: You must create the default queue and any individual queues before processing R/3-to-eLink IDOCs. See "Setting Up Queues for R/3-to-eLink IDOCs" later in this topic for information.

Setting Data-Dependent Routing Values in the Environment File

You use the TargetId (ROUTING) settings in the environment file to associate an IDOC with a data-dependent routing value. If the logical system ID of the target R/3 System and the IDOC message type specified in an IDOC section of the environment file match the logical system ID and message type of an IDOC's control record, then R/3-to-eLink encodes the specified TargetId value in the CR3_TARGET_ID FML32 field of the IDOC message. If no matching section is found in the environment file, then R/3-to-eLink encodes the default DDR value, which is defined in the CR3_ALE_DEFAULT_TARGET_ID environment variable. See "Setting the Default Data-Dependent Routing Value" on page 5-18 for more information.

Compressing R/3-to-eLink IDOCs

You use the COMPRESS environment variable to combine IDOC messages with matching logical system IDs and IDOC message types into an IDOC message containing multiple IDOCs. If the logical system ID of the target R/3 System and the IDOC message type specified in an IDOC section of the environment file match the logical system ID and message type of an IDOC's control record, and its COMPRESS setting is "Y", then R/3-to-eLink aggregates the IDOCs with matching criteria into a single, larger IDOC message that it then queues into the appropriate target queue. If the Compress setting is "N", then R/3-to-eLink queues each IDOC message separately.

Configuring Queues for R/3-to-eLink

R/3-to-eLink requires that you set up at least one default TUXEDO queue and additional queues as needed. See "Setting Up Queues for R/3-to-eLink IDOCs" later in this topic for instructions.

Setting the Default Data-Dependent Routing Value

You can configure the default data-dependent routing (DDR) value that R/3-to-eLink assigns to each IDOC message (in the CR3_TARGET_ID field) that it queues.

Set the CR3_ALE_DEFAULT_TARGET_ID environment variable to a DDR value in the cr3aleout.env file as follows:

```
CR3_ALE_DEFAULT_TARGET_ID=DDR_VALUE
```

The following is an example from that file setting the default target ID to CR3_ERROR:

```
CR3_ALE_DEFAULT_TARGET_ID=CR3_ERROR
```

See "Setting Data-Dependent Routing Values in the Environment File" later in this topic and "Managing Data-Dependent Routing" in Chapter 2, "Integrating with ALE," in this guide for more information.

Splitting R/3-to-eLink IDOC Packets

You can configure R/3-to-eLink to split IDOC packets containing multiple IDOCs into separate IDOC messages, each containing a single IDOC. By default, R/3-to-eLink queues the entire IDOC packet. If you set the CR3_ALE_DEFAULT_IDOC_SPLIT

environment variable to "Y", R/3-to-eLink will split the IDOC packet into individual IDOC packets. See "Splitting R/3-to-eLink IDOC Packets Into Individual IDOCs" in Chapter 2, "Integrating with ALE," of this guide for more information.

Specifying the TID File for R/3-to-eLink

R/3-to-eLink uses a TID file to log TIDs that it has already processed to guarantee that it queues an IDOC packet from R/3 once and only once. You must specify the path and file name of the TID file (*without* an extension) in the `CR3_ALEOUT_TID_FILE` environment variable, as shown in the following example (Windows NT):

```
CR3_ALEOUT_TID_FILE=<your eLink app directory>\cr3aleout
```

Upon start-up, if R/3-to-eLink does not find a TID file in the specified location, it creates one automatically. See "Configuring the TID File Manager" later in this topic for information about configuring and administering the TID file.

Configuring Logging for R/3-to-eLink

By default, R/3-to-eLink writes logging and tracing output to the TUXEDO USERLOG. If your are sending output to a file, you must specify its name and location in the `CR3_ALEOUT_PRINT_FILE` environment variable.

You can also include trace information in the log by setting the `CR3_TRACE` environment variable to "Y". You exclude trace information by omitting this environment variable or setting it to "N".

The following example from the R/3-to-eLink environment file, `cr3aleout.env`, sets logging options for R/3-to-eLink:

```
CR3_ALEOUT_PRINT_FILE=
CR3_ALEOUT_TRACE=Y
```

Configuring Load Balancing for R/3-to-eLink

Multiple instances of R/3-to-eLink can log onto R/3 and submit R/3-to-eLink IDOCs for processing. For deployments that involve high volumes of R/3-to-eLink IDOCs, you can enhance system performance by balancing the load across multiple instances of R/3-to-eLink.

Use TUXEDO's load balancing capabilities to manage the number of R/3-to-eLink instances running. See your BEA TUXEDO documentation regarding TUXEDO administrative tools for more information.

Note: Each instance of R/3-to-eLink requires a separate R/3 login, so be sure to tune the number of concurrent instances to optimize system performance without requiring excessive R/3 resources. Instances can log in using the same user name and password.

Example:

More than one instance of R/3-to-eLink can be specified in `cr3_ale.ubb`. The number of instances you can specify is limited only by the capacity of the network on which you are running. Two instances have been defined in Listing 5-5. For more information on the meaning of the various commands and switches used in this example, see "Defining the R/3-to-eLink Server (cr3aleout)."

Listing 5-5 Configuring Two Instances of R/3-to-eLink in cr3_ale.ubb

```
cr3aleout
    SRVGRP=CR3 SRVID=5
    CLOPT="-A -- -i cr3aleout1 -e cr3aleout.env"
cr3aleout
    SRVGRP=CR3 SRVID=6
    CLOPT="-A -- -i cr3aleout2 -e cr3aleout.env"
```

Then, `cr3aleout.env` must be appended so there are two matching sections, one for `cr3aleout1` and one for `cr3aleout2`. All of the settings specified in "Setting Environment Variables for R/3-to-eLink" must be set. The first few lines of each server section are shown in Listing 5-6.

Listing 5-6 Settings for cr3aleout1 and cr3aleout2 in cr3aleout.env

```
[SERVER=cr3aleout1]
CR3_DESTINATION=delia
CR3_SOCKET=sapgw00
...
[SERVER=cr3aleout2]
CR3_DESTINATION=delia
```

```
CR3_SOCKET=sapgw00
...
```

Configuring Multiple Program IDs

If R/3 is configured to send R/3-to-eLink IDOCs to different program IDs, you can configure R/3-to-eLink to handle these IDOCs by running multiple instances of R/3-to-eLink using different program IDs. You must make sure that all instances sharing the same program ID also share the same TID file, and that all instances sharing the same TID file also share the same program ID. Instances that register under different program IDs must *not* share the same TID file.

The example shown in Listing 5-7 uses the same `cr3_ale.ubb` file as the load balancing example shown in "Configuring Load Balancing for R/3-to-eLink." Notice that the `CR3_PROGRAM_ID` for the `cr3aleout1` server is `bea.cr3.tim.ale`, but the `CR3_PROGRAM_ID` for the `cr3aleout2` server is `bea.cr3.malay.ale`.

Listing 5-7 Configuring for Multiple Program IDs in cr3aleout.env

```
[SERVER=cr3aleout1]
CR3_DESTINATION=delia
CR3_SOCKET=sapgw00
CR3_PROGRAM_ID=bea.cr3.tim.ale
...
[SERVER=cr3aleout2]
CR3_DESTINATION=delia
CR3_SOCKET=sapgw00
CR3_PROGRAM_ID=bea.cr3.malay.ale
...
```

See "Registering Multiple Program IDs" in Chapter 2, "Integrating with ALE," in this guide for more information.

Generating a Transaction Log

The eLink-to-R/3 and R/3-to-eLink servers use TUXEDO transactions. For these components, you need to create a transaction log (TLOG). The `cr3_tlog` file for your platform generates the transaction log (TLOG) when it is run with the following command:

```
> nmake -f cr3_tlog.nt (cr3_tlog.sh on UNIX)
```

Normally, this files does not need to be edited. Of course, there are always exceptions to the rule. For information on how this file should be edited, see the topic “crlog: Create the Transaction Log” in the “BEA Tuxededo /Q Guide.” using TUXEDO administrative tools.

Configuring Queues

The following topics describe how to create and configure TUXEDO queues for ALE processing:

- TUXEDO Queues
- Setting Up Queues for eLink-to-R/3 IDOCs
- Setting Up Queues for R/3-to-eLink IDOCs

TUXEDO Queues

You must set up queues and queue spaces for both eLink-to-R/3 and R/3-to-eLink. You need to create and configure the Server Group (`TMS_QM`), Message Queue Server (`TMQUEUE`) and Message Forwarding Server (`TMQFORWARD`).

Using the cr3_queues Makefiles to Add Queues

The queue space QSPACE and the queues CR3_IDOC_IN, CR3_SUBMIT, and CR3_ERRQUE. The queues are generated inside the queue space QSPACE when the makefile described in Listing 5-8 is run with the following command:

```
> nmake -f cr3_queues.nt (cr3_queues.sh on UNIX)
```

Listing 5-8 cr3_queues.nt Makefile (Sample Supplied with Software)

```
all:   que
que:
       del /q $(APPDIR)\QUE
       (set QMCONFIG=$(APPDIR)\QUE) &
       qmadmin $(APPDIR)\QUE < <<
echo
crdl $(APPDIR)\QUE 0 400
# The following section adds queue spaces and queues.

qspacecreate
QSPACE
62849
100
6
4
9
20
y
16

qopen QSPACE
qcreate
CR3_IDOC_IN
fifo
none
2
30
80%
0%

qcreate
CR3_SUBMIT
fifo
none
2
```

```
30
80%
0%

qcreate
CR3_ERRQUE
fifo
none
2
30
80%
0%
#end of section that needs to be copied to create custom queues
q
<<NOKEEP
rmipc:
       (set QMCONFIG=$(APPDIR)\QUE) &
       qmadmin $(APPDIR)\QUE < <<
echo
ipcrm -y QSPACE
<<NOKEEP
```

Actually, this script uses the TUXEDO `qmadmin` utilities called `qspacecreate` and `qcreate` to create this queue space and these queues. If you need to create additional queues, you can do so by copying a portion of this script to a point in the `cr3_queues` makefile just above the single `q`, then changing the queue space and queue names to the names of the custom queue spaces and queues that you want to create.

In the following example, the queue space `QS1` is created, and the queues `Q1` and `Q2` are created inside it (text that must be changed is marked in **BOLD**:

```
qspacecreate
QS1
62849
100
6
4
9
20
y
16

qopen QS1
qcreate
Q1
```

```
fifo
none
2
30
80%
0%

qcreate
Q2
fifo
none
2
30
80%
0%
```

For an explanation of queue space and queue parameters for the queue spaces and queues being created see the "Creating Queue Space and Queues" section of the */Q Guide* in your BEA TUXEDO documentation.

Note: You can also create queue spaces and queues manually using the instructions described in the *BEA TUXEDO /Q Guide*.

Setting Up Queues for eLink-to-R/3 IDOCs

eLink-to-R/3 uses the following queues:

Queue Name	Description
CR3_SUBMIT	IDOC submit queue. Handles IDOC packets that a TUXEDO application or other application has submitted to the CR3_SUBMIT service for processing.
CR3_ERROR	IDOC error queue. Handles invalid IDOC packets that have failed validation by the CR3_SUBMIT service.
CR3_IDOC_IN	IDOC eLink-to-R/3 queue. Handles IDOC packets that the CR3_SUBMIT service has validated, assigned a TID, and queued. TMQFORWARD forwards each IDOC packet to the CR3_IDOC_IN service for submission to R/3.

Use the TUXEDO Queue manager administration utility (`qmadmin`) to create these queue spaces and queues. See the */Q Guide* in your BEA TUXEDO documentation for instructions.

Defining the TMQUEUE and TMQFORWARD Servers in the UBB File

In the SERVERS section of your UBB file, define the `TMQUEUE` and `TMQFORWARD` servers:

Listing 5-9 Sample SERVERS Section of UBB File

```
TMQUEUE
   SRVGRP=QUE SRVID=1
   GRACE=0  RESTART=Y CONV=N  MAXGEN=10
   CLOPT="-s QSPACE:TMQUEUE --  "
TMQFORWARD
   SRVGRP=QUE SRVID=2
   GRACE=0  RESTART=Y CONV=N  MAXGEN=10
   CLOPT=" -- -i 10 -q CR3_IDOC_IN,CR3_SUBMIT"
```

Setting Environment Variables

After creating the error and eLink-to-R/3 queue spaces and queues, specify their names in environment variables:

Listing 5-10 Setting Environment Variables for Queues

```
CR3_ALEIN_IDOCIN_QUEUE=CR3_IDOC_IN
CR3_ALEIN_IDOCIN_QUEUE_SPACE=QSPACE
CR3_ALEIN_ERROR_QUEUE=CR3_ERROR
CR3_ALEIN_ERROR_QUEUE_SPACE=QSPACE
```

See Table 5-1 earlier in this topic for a description of these environment variables.

Setting Up Queues for R/3-to-eLink IDOCs

R/3-to-eLink can queue R/3-to-eLink IDOC messages into one or more queues. At a minimum, you must create at least one default queue. If you are using the `cr3aleout.env` file to queue IDOC messages to multiple queues, you must create all the target queues, including the default queue.

Use the TUXEDO Queue manager administration utility (`qmadmin`) to create these queue spaces and queues. See the */Q Guide* in your BEA TUXEDO documentation for instructions.

After creating the R/3-to-eLink queues, specify the default queue space and queue name in environment variables by adding the following lines to your `cr3aleout.env` file:

```
CR3_ALE_TARGET_QUEUE=QUEUE_NAME
CR3_ALE_TARGET_QUEUE_SPACE=QUEUE_SPACE_NAME
```

Example:

```
CR3_ALE_TARGET_QUEUE=CR3_SUBMIT
CR3_ALE_TARGET_QUEUE_SPACE=QSPACE
```

See Table 5-2 earlier in this topic for a description of these environment variables.

Configuring the TID File Manager

The following topics describe the TID File Manager, a command line utility that manages the size of the TID file in the run-time environment:

- Command Line Options
- Running the TID File Manager
- How the TID File Manager Works

Command Line Options

The TID File Manager uses the command line options shown in Table 6-4:

Table 5-8 Command Line Options for the TID File Manager

Option	Description
`-h` or `-?`	Help text that describes each option.
`-f`	Name of the TID file (path is optional). *Required*.
`-n`	Number of lines to keep in the TID file. Default is 10 lines. If the date (`-d`) option is also specified, this setting overrides the date setting. The TID File Manager retains at least this number of lines in the original TID file.
`-t`	Frequency (in seconds) with which the TID File Manager periodically checks the TID file. Default is 60 seconds. If this option is omitted, then the TID File Manager runs once and exits. If this option is specified, the TID File Manager keeps running. *Optional*.
`-d`	Last save date. The TID File Manager removes any CONFIRMED entries that are older than the specified date. Valid date formats include: ■ Jan/1/2000 ■ 2000,1,jan ■ "1 JAN 2000"

Running the TID File Manager

To run the TID File Manager, enter the following command:

```
cr3tidmanager -f FileName [OptionalParameters]
```

where *FileName* is required and *OptionalParameters* include any of the other options in Table 5-8, “Command Line Options for the TID File Manager,” on page 5-28.

Note: You must run a *separate* TID File Manager for *each* TID file used by instances of eLink-to-R/3 and R/3-to-eLink.

How the TID File Manager Works

Upon startup, the TID File Manager does the following:

- Locks the specified TID file.
- Creates a backup file using the original file name plus the date-time and a `.bck` extension.
- Removes from the TID log file only CONFIRMED entries that meet the configured criteria.
- Appends each removed entry to the backup file.
- Saves both files.
- Releases the file lock.

Because the TID File Manager locks the file while editing, be careful to specify a large enough frequency to contain the file size without slowing run-time performance.

Shutting Down the BEA eLink Adapter for R/3 ALE

The BEA eLink Adapter for R/3 ALE is a series of eLink Platform servers that are constantly running. If you change your configuration, you must stop or shutdown these servers, then restart them, or the servers will continue to operate according to the configuration under which they were started. For more information on reconfiguring and restarting the BEA eLink Adapter for R/3 ALE, see Chapter B, "Configuration Quick Reference." Shutdown the eLink Platform servers by performing the following series of commands:

1. Shutdown all R/3-to-eLink (outbound) servers with the following command:

   ```
   > tmshutdown -i ServerID -k KILL
   ```

   where *ServerID* is the unique ID of the R/3-to-eLink server. This would be whatever is set as the `SRVID` parameter for all `cr3aleout` servers in `cr3_ale.ubb`. In the sample `cr3_ale.ubb` listed in Appendix A, "Sample Configuration Files," the *ServerID* value is "4."

2. Shutdown the remaining eLink Platform servers with the command:

   ```
   > tmshutdown -y
   ```

3. Repeat the command to make sure that all of the eLink Platform servers are shut down:

   ```
   > tmshutdown -y
   ```

6 Configuring R/3 Connections

This topic describes how to configure your BEA eLink Adapter for R/3 ALE installation for eLink-to-R/3 and R/3-to-eLink communication with R/3. It includes the following main sections:

- Configuring eLink-to-R/3 Connections
- Configuring R/3-to-eLink Connections
- Troubleshooting Connection Problems

Configuring eLink-to-R/3 Connections

The eLink-to-R/3 server uses the sideinfo method of connecting to R/3. The following topics describe how to configure eLink-to-R/3 connections:

- Configuring the sideinfo File

Configuring the sideinfo File

The eLink-to-R/3 server uses the `sideinfo` method of connecting to R/3. The `sideinfo` file is an ASCII text file that specifies the connection point for external programs to communicate with the R/3 application server. See your SAP R/3 documentation for more information about the `sideinfo` file.

The `sideinfo` file is associated with the parameters provided to the eLink-to-R/3 server when it is started. A sample `sideinfo` file comes with BEA eLink Adapter for R/3 ALE, but you must configure this sample file for your particular R/3 environment and BEA eLink Adapter for R/3 ALE installation. Errors in this file will prevent the eLink-to-R/3 server from connecting to R/3.

Contents of the sideinfo File

The following table describes the attributes in the `sideinfo` file:

Table 6-1 Attributes in the sideinfo File

Attribute	Description
`DEST`	Connection identifier. You must set this to the name of your SAP application server. The name is case sensitive, so pay close attention to the case of the letters in the application server name. The eLink-to-R/3 server uses this identifier to determine the R/3 application server to connect to. The eLink-to-R/3 server gets the identifier name from the `CR3_DESTINATION` environment variable that is defined in the server start-up script. The identifier links its setting to the connection.
`LU`	TCP/IP name of the node on which the R/3 application server is running. The eLink-to-R/3 server will open a TCP/IP socket to the R/3 application node. The LU can be specified as the full TCP/IP name or as an alias. This name or alias must match exactly what is specified in the hosts file, or it must be translated by the directory name service. Test this by using the `ping` command and the exact string used in the LU line item (`ping node`).
`TP`	Name of the R/3 user presentation connection socket (the network socket to which the SAPGUI connects). This name is configured in R/3 and is specific to the instance of R/3. The last two characters in the name specify the instance number. This name must appear in the services file on the node on which eLink-to-R/3 is running, and the value of the socket number must match the same number in the services file on the R/3 application server.
`PROTOCOL`	This will always be I when using the BEA eLink Adapter for R/3 ALE.

Table 6-1 Attributes in the sideinfo File

Attribute	Description
GWHOST	Node name for the R/3 Gateway. As with the LU attribute, it must be translated correctly into a TCP/IP address (test it by using the `ping` command to ping the node). The gateway host is usually the same node as the R/3 application server, although this is not required. The R/3 Gateway process might reside on a different node. If the gateway must be moved, then the gateway identifier must be changed in the R/3 profile. Be sure to check with the R/3 system manager to ensure that this value matches the gateway location.
GWSERV	Name of the TCP/IP socket that is the R/3 Gateway connection. All processes connecting to R/3 in the background use this socket number as an entry point. As with the TP attribute, this name must be defined in the services file on the node on which the eLink-to-R/3 server is running, and the definition must match the definition of the R/3 application server.

In an R/3 environment with multiple R/3 application servers, this list of attributes is simply repeated for different node names, different instances of R/3, and possibly different gateways.

Sample Settings

The following listing shows sample settings in a `sideinfo` file:

Listing 6-1 Sample Settings for sideinfo File

```
DEST=SAPNODE
LU=nodename
TP=sapdp00
PROTOCOL=I
GWHOST=nodename
GWSERV=sapgw00
```

Specifying the Location of the sideinfo File

The `sideinfo` file is usually located in the directory where the eLink-to-R/3 server is running (for example, the TUXEDO `appdir`). If you cannot use this location in your environment (for example, a number of other processes also connect to R/3 and use the same `sideinfo` information), then you can avoid duplicating the file by assigning the `SIDE_INFO` environment variable to a path containing the `sideinfo` table, as shown in the following examples:

UNIX C shell (csh):

```
setenv SIDE_INFO <your eLink app directory>/sideinfo
```

Windows NT:

```
set SIDE_INFO=<your eLink app directory>\sideinfo
```

The `sideinfo` file must have protections set so that all processes can read it.

For the eLink-to-R/3 server, this environment variable is defined in the server start-up script. See "Setting Environment Variables for eLink-to-R/3" in Chapter 5, "Configuring ALE Integration," for more information.

Configuring R/3-to-eLink Connections

The R/3-to-eLink server uses the register mode method of connecting to R/3. When the R/3-to-eLink server is started, it connects to R/3 and registers with a specific Program ID. This Program ID is defined in the RFC destination, as defined in transaction SM59. When an RFC is executed with the specified destination, the request is passed to the server registered on the Program ID. If no server is registered, then an error occurs. The server *must* be started and registered on the Program ID before the RFC is executed.

To register with R/3, the following information is passed to R/3-to-eLink automatically:

Setting	Description
Program-Id	Program ID (text string). It must match the `program-id` defined in transaction SM59.
Gateway Host Name	Name of the host running the SAP gateway server. It must match the GWHOST parameter specified in the `sideinfo` file.
Gateway Service Name	Service name of the SAP gateway server. It must match the GWSERV parameter specified in the `sideinfo` file.

Troubleshooting Connection Problems

R/3 provides extensive help in isolating and testing connection problems involving the R/3 Gateway. The R/3 gateway is sometimes called the CPIC gateway because the CPIC protocol is the lowest-level protocol used on the TCP/IP socket when connecting to R/3. RFCs are essentially layered on CPIC.

To troubleshoot connection problems:

- Verify that the TCP/IP network is functioning by using network testing.
- Examine your R/3 installation. Common problems arise from typographical errors in the `sideinfo` file, the `hosts` file, and the `services` file. Examine these files, bearing in mind that UNIX is case-sensitive (including TCP/IP translation and socket name translation).
- If problems persist, thoroughly examine the CPIC implementation on the R/3 application server. The R/3 documentation provides extensive material about the CPIC setup under the heading "BC - SAP Communication: Configuration." If necessary, the CPIC implementation can be modified to accommodate the BEA eLink Adapter for R/3 ALE installation.

A Sample Configuration Files

This topic describes the two preconfigured sample `cr3_ale.ubb` files and the `setenv` files ("cr3_ale.ubb for UNIX" and "cr3_ale.ubb for Windows NT") that come with BEA eLink Adapter for R/3 ALE. It includes the following main sections:

- cr3_ale.ubb for UNIX
- cr3_ale.ubb for Windows NT
- setenv_ale.sh (UNIX)
- setenv_ale.bat (Windows NT)

cr3_ale.ubb for UNIX

```
########################################################################
*RESOURCES
########################################################################

IPCKEY        40000
DOMAINID      CR3
MASTER        SITE1
MAXACCESSERS  30
MAXSERVERS    20
MAXSERVICES   30
SCANUNIT      10
SANITYSCAN    12
BLOCKTIME     30
MAXGTT        200
MODEL         SHM
LDBAL         Y

########################################################################
*MACHINES
########################################################################

<system name>
       LMID=SITE1
       TYPE="HP-UX"
       TUXDIR="<your Tuxedo install directory>"
       TUXCONFIG="<your eLink app directory>/tuxconfig"
       TLOGDEVICE ="<your eLink app directory>/TLOG"
       TLOGSIZE=10
       APPDIR="<your eLink app directory>"
       ULOGPFX="<your eLink app directory>/ULOG"
       MAXWSCLIENTS=2
       UID=278

########################################################################
*GROUPS
########################################################################

QUE
       LMID=SITE1  GRPNO=1
       TMSNAME=TMS_QM TMSCOUNT=2
       OPENINFO="TUXEDO/QM:<your eLink app directory>/QUE:QSPACE"

CR3
```

```
        LMID=SITE1  GRPNO=2
        TMSNAME=TMS TMSCOUNT=2

##################################################################
*SERVERS
##################################################################

DEFAULT:
        CLOPT="-A"

TMQUEUE
        SRVGRP=QUE SRVID=1
        GRACE=0  RESTART=Y CONV=N  MAXGEN=10
        CLOPT="-s QSPACE:TMQUEUE --  "

TMQFORWARD
        SRVGRP=QUE SRVID=2
        GRACE=0  RESTART=Y  CONV=N  MAXGEN=10
        CLOPT=" -- -i 10 -q CR3_IDOC_IN,CR3_SUBMIT"

cr3alein
    SRVGRP=CR3 SRVID=3
        CLOPT="-A -- -i cr3alein -e cr3alein.env"

cr3aleout
    SRVGRP=CR3 SRVID=4
        CLOPT="-A -- -i cr3aleout -e cr3aleout.env"

##################################################################
*SERVICES
##################################################################

CR3_SUBMIT
CR3_IDOC_IN
CR3_ALE_OUT
```

1. Replace <your Tuxedo install directory> with the path to the location where you installed the eLink Platform.
2. Replace <your eLink app directory> with the path to the location containing the edited versions of the BEA eLink Adapter for R/3 ALE configuration files. The directory containing these files is named bin. For more information, see Step 1 of "Configuration Quick Reference for UNIX" in Appendix B, "Configuration Quick Reference."

3. This value (278) that is set for the UID parameter is not always correct depending on the machine upon which the Adapter for R/3 ALE is running. Having this parameter incorrectly set causes an error while running the tmloadcf command. This parameter setting should be changed to a valid UID for your environment. See the Tuxedo Administration Guide for more information.

4. The value set for the Type variable should correspond to the system used. The value presently set in the sample version of cr3_ale.ubb (HP-UX) is appropriate only for HP systems. If you are using other UNIX systems, the appropriate value for your particular system should be used. For example, if you are using Solaris, this value should be SUNOS. If you don't know what the value is for your particular system, you can determine it with the following command:

   ```
   > uname
   ```

   The system will return a string that should be set as the Type value.

cr3_ale.ubb for Windows NT

```
########################################################################
*RESOURCES
########################################################################

IPCKEY        40000
DOMAINID      CR3
MASTER        SITE1
MAXACCESSERS  30
MAXSERVERS    20
MAXSERVICES   30
SCANUNIT      10
SANITYSCAN    12
BLOCKTIME     30
MAXGTT        200
MODEL         SHM
LDBAL         Y

########################################################################
*MACHINES
########################################################################

<system name>
       LMID=SITE1
```

```
        TYPE="Win32"
        TUXDIR="<your Tuxedo install directory>"
        TUXCONFIG="<your eLink app directory>\tuxconfig"
        TLOGDEVICE ="<your eLink app directory>\TLOG"
        TLOGSIZE=10
        APPDIR="<your eLink app directory>"
        ULOGPFX="<your eLink app directory>\ULOG"
        MAXWSCLIENTS=2

########################################################################
*GROUPS
########################################################################

QUE
        LMID=SITE1  GRPNO=1
        TMSNAME=TMS_QM TMSCOUNT=2
        OPENINFO="TUXEDO/QM:<your eLink app directory>\QUE;QSPACE"

CR3
        LMID=SITE1  GRPNO=2
        TMSNAME=TMS TMSCOUNT=2

########################################################################
*SERVERS
########################################################################

DEFAULT:
        CLOPT="-A"

TMQUEUE
        SRVGRP=QUE SRVID=1
        GRACE=0  RESTART=Y CONV=N  MAXGEN=10
        CLOPT="-s QSPACE:TMQUEUE --  "

TMQFORWARD
        SRVGRP=QUE SRVID=2
        GRACE=0  RESTART=Y  CONV=N  MAXGEN=10
        CLOPT=" -- -i 10 -q CR3_IDOC_IN,CR3_SUBMIT"

cr3alein
    SRVGRP=CR3 SRVID=3
        CLOPT="-A -- -i cr3alein -e cr3alein.env"

cr3aleout
    SRVGRP=CR3 SRVID=4
        CLOPT="-A -- -i cr3aleout -e cr3aleout.env"

########################################################################
*SERVICES
```

```
#################################################################

CR3_SUBMIT
CR3_IDOC_IN
CR3_ALE_OUT
```

1. Replace <your Tuxedo install directory> with the path to the location where you installed the eLink Platform.
2. Replace <your eLink app directory> with the path to the location containing the edited versions of the BEA eLink Adapter for R/3 ALE configuration files. The directory containing these files is named bin. For more information, see Step 1 of "Configuration Quick Reference for Windows NT" in Appendix B, "Configuration Quick Reference."

setenv_ale.sh (UNIX)

This section provides a code listing corresponding to the template that is placed in <installation directory>/adapters/sapr3_ale/config during installation of the BEA eLink Adapter for R/3 ALE. <installation directory> is whatever path that you specified during installation. (Of course, this and other configuration files are copied to the bin directory during configuration, and this bin directory is <your eLink app directory> .)

```
#!/bin/sh
TUXDIR=<your Tuxedo directory> ;          export TUXDIR
APPDIR=<your eLink app directory> ;       export APPDIR
TUXCONFIG=$APPDIR/tuxconfig ;             export TUXCONFIG
QMCONFIG=$APPDIR/QUE ;                    export QMCONFIG
FLDTBLDIR32=$APPDIR:$TUXDIR/udataobj ;    export FLDTBLDIR32
FIELDTBLS32=cr3_ale.fml ;                export FIELDTBLS32
```

1. Replace <your Tuxedo directory> with the path to the location where you installed the eLink Platform.

2. Replace <your eLink app directory> with the path to the location containing the edited versions of the BEA eLink Adapter for R/3 ALE configuration files. The directory containing these files is named bin. For more information, see Step 1 of "Configuration Quick Reference for UNIX" in Appendix B, "Configuration Quick Reference."

setenv_ale.bat (Windows NT)

This section provides a code listing corresponding to the template that is placed in <installation directory>\adapters\sapr3_ale\config during installation of the BEA eLink Adapter for R/3 ALE. <installation directory> is whatever path that you specified during installation. (Of course, this and other configuration files are copied to the bin directory during configuration, and this bin directory is <your eLink app directory> .)

```
rem
set TUXDIR=<your Tuxedo directory>
set APPDIR=<your eLink app directory>
set TUXCONFIG=%APPDIR%\tuxconfig
set QMCONFIG=%APPDIR%\QUE
set FLDTBLDIR32=%APPDIR%:%TUXDIR%\udataobj
set FIELDTBLS32=cr3_ale.fml,Usysfl32
```

Note: The value Usysf132 set as a FIELDTBLS32 value is optional and is only used if you intend to use the ud32 application.

1. Replace <your Tuxedo directory> with the path to the location where you installed the eLink Platform.

2. Replace <your eLink app directory> with the path to the location containing the edited versions of the BEA eLink Adapter for R/3 ALE configuration files. The directory containing these files is named bin. For more information, see Step 1 of "Configuration Quick Reference for Windows NT" in Appendix B, "Configuration Quick Reference."

B Configuration Quick Reference

This section provides system administrators with a quick reference for getting the BEA eLink Adapter for R/3 ALE up and running quickly. It provides a step-by-step, chronological summary of the entire configuration process for UNIX and Windows NT. It includes the following main sections:

- Configuration Quick Reference for Windows NT
- Configuration Quick Reference for UNIX

Configuration Quick Reference for UNIX

Perform the following steps when configuring your BEA eLink Adapter for R/3 ALE:

1. Copy the following files from

   `<installation directory>/adapters/sapr3_ale/config`

   to

   `<installation directory>/adapters/sapr3_ale/bin:`

   ```
   setenv_ale.sh
   sideinfo
   cr3_ale.ubb
   cr3alein.env
   cr3aleout.env
   cr3_ale.fml
   cr3_tlog.sh
   cr3_queues.sh
   ```

   Note: `<installation directory>` is whatever path that you specified during installation of the BEA eLink Adapter for R/3 ALE.

2. Edit these files as appropriate to match the settings in your environment. The following are guidelines for editing these files:

   - `setenv_ale.sh`: This file sets your the environment variables necessary to run the eLink Platform and R/3. Replace the text enclosed in angle brackets (making sure to remove the angle brackets) with the appropriate value. For more information on `setenv_ale.sh`, see the "setenv_ale.sh (UNIX)" section of Appendix A, "Sample Configuration Files."
   - `sideinfo`: This file specifies the connection point for programs external to R/3 (like the BEA eLink Adapter for R/3 ALE) to communicate with the R/3 application server. For more information on how this file should be edited, see "Configuring the sideinfo File" in Chapter 6, "Configuring R/3 Connections."
   - `cr3_ale.ubb`: This file is used to configure the eLink Platform. For more information on how this file should be edited, see "Defining the eLink-to-R/3 server (cr3alein)," "Defining eLink-to-R/3 Services," and "Defining the R/3-to-eLink Server (cr3aleout)" in Chapter 5, "Configuring ALE Integration."

- `cr3alein.env`: This is the environment file that specifies detailed configuration information for the eLink-to-R/3 server. For more information on how this file should be edited, see "Setting Environment Variables for eLink-to-R/3" in Chapter 5, "Configuring ALE Integration."
- `cr3aleout.env`: This is the environment file that specifies detailed configuration information for the R/3-to-eLink server. For more information on how this file should be edited, see "Setting Environment Variables for R/3-to-eLink" in Chapter 5, "Configuring ALE Integration."
- `cr3_ale.fml`: This file configures the FML fields used by the BEA eLink Adapter for R/3 ALE. For most uses of the adapter, this file will not need to be edited. If you have special requirements and you do need to edit it, see "FML32 Field Definitions" in Chapter 1, "Introducing BEA eLink Adapter for R/3 ALE."
- `cr3_tlog.sh`: This file creates the transaction log (TLOG) file when it is run. Normally, this file does not need to be edited. For information on how this file should be edited, see the topic "crlog: Create the Transaction Log" in the "BEA Tuxededo /Q Guide."
- `cr3_queues.sh`: This file creates default queue spaces and queues when it is run. If you are using custom queue spaces and queues, you can either edit this file or use the `queuespacecreate` and `qcreate` utilities. Instructions for using these utilities are provided in the "Creating Queue Space and Queues" topic in the "BEA TUXEDO /Q Guide." For information on how this file should be edited, see "Using the cr3_queues Makefiles to Add Queues" in Chapter 5, "Configuring ALE Integration."

3. Set your environment variables by running `setenv_ale.sh` from the UNIX prompt as follows:

```
> setenv_ale.sh
```

   Make sure that you are in the `bin` directory, the directory containing the edited versions of the configuration files from Step 2.

4. Generate your eLink Platform configuration file (`tuxconf`) from `cr3_ale.ubb` using the following command:

```
> tmloadcf cr3_ale.ubb
```

5. Generate your transaction log (TLOG) file by running `cr3_tlog.sh` from the UNIX prompt as follows:

```
> cr3_tlog.sh
```

6. Generate your queue spaces and queues by running `cr3_queues.sh` from the UNIX prompt as follows:

```
> cr3_queues.sh
```

7. Start the eLink Platform with the following command:

```
> tmboot -y
```

8. If you need to change your configuration, repeat Step 2 (editing the configuration files only as necessary to implement the configuration change you want to make), shutdown the eLink Platform servers, then perform Steps 3 through 7. Shutdown the eLink Platform servers by performing the following series of commands:

   a. Shutdown all R/3-to-eLink (outbound) servers with the following command:

      ```
      > tmshutdown -i ServerID -k KILL
      ```

      where *`ServerID`* is the unique ID of the R/3-to-eLink server. This would be whatever is set as the `SRVID` parameter for all `cr3aleout` servers in `cr3_ale.ubb`. In the sample `cr3_ale.ubb` listed in Appendix A, "Sample Configuration Files," the *`ServerID`* value is "4."

   b. Shutdown the remaining eLink Platform servers with the command:

      ```
      > tmshutdown -y
      ```

   c. Repeat the command to make sure that all of the eLink Platform servers are shut down:

      ```
      > tmshutdown -y
      ```

Configuration Quick Reference for Windows NT

Perform the following steps when configuring your BEA eLink Adapter for R/3 ALE:

1. Copy the following files from

   ```
   <installation directory>\adapters\sapr3_ale\config
   ```

   to

```
<installation directory>\adapters\sapr3_ale\bin:

setenv_ale.bat
sideinfo
cr3_ale.ubb
cr3alein.env
cr3aleout.env
cr3_ale.fml
cr3_tlog.nt
cr3_queues.nt
```

Note: `The <installation directory>` is whatever path that you specified during installation of the BEA eLink Adapter for R/3 ALE. The path `C:\eLink` was used in the Chapter 4, "Installing BEA eLink Adapter for R/3 ALE," examples.

2. Edit these files as appropriate to match the settings in your environment. The following are guidelines for editing these files:

   - `setenv_ale.bat`: This file sets your the environment variables necessary to run the eLink Platform and R/3. Replace the text enclosed in angle brackets (making sure to remove the angle brackets) with the appropriate value. For more information on `setenv_ale.bat`, see the "setenv_ale.bat (Windows NT)" section of Appendix A, "Sample Configuration Files."

   - `sideinfo`: This file specifies the connection point for programs external to R/3 (like the BEA eLink Adapter for R/3 ALE) to communicate with the R/3 application server. For more information on how this file should be edited, see "Configuring the sideinfo File" in Chapter 6, "Configuring R/3 Connections."

   - `cr3_ale.ubb`: This file is used to configure the eLink Platform. For more information on how this file should be edited, see "Defining the eLink-to-R/3 server (cr3alein)," "Defining eLink-to-R/3 Services," and "Defining the R/3-to-eLink Server (cr3aleout)" in Chapter 5, "Configuring ALE Integration."

   - `cr3alein.env`: This is the environment file that specifies detailed configuration information for the eLink-to-R/3 server. For more information on how this file should be edited, see "Setting Environment Variables for eLink-to-R/3" in Chapter 5, "Configuring ALE Integration."

   - `cr3aleout.env`: This is the environment file that specifies detailed configuration information for the R/3-to-eLink server. For more information on how this file should be edited, see "Setting Environment Variables for R/3-to-eLink" in Chapter 5, "Configuring ALE Integration."

- `cr3_ale.fml`: This file configures the FML fields used by the BEA eLink Adapter for R/3 ALE. For most uses of the adapter, this file will not need to be edited. If you have special requirements and you do need to edit it, see "FML32 Field Definitions" in Chapter 1, "Introducing BEA eLink Adapter for R/3 ALE."
- `cr3_tlog.nt`: This file creates the transaction log (TLOG) file when it is run. Normally, this file does not need to be edited. For information on how this file should be edited, see the topic "crlog: Create the Transaction Log" in the "BEA Tuxededo /Q Guide."
- `cr3_queues.nt`: This file creates default queue spaces and queues when it is run. If you are using custom queue spaces and queues, you can either edit this file or use the `queuespacecreate` and `qcreate` utilities. Instructions for using these utilities are provided in the "Creating Queue Space and Queues" topic in the "BEA TUXEDO /Q Guide." For information on how this file should be edited, see "Using the cr3_queues Makefiles to Add Queues" in Chapter 5, "Configuring ALE Integration."

3. Set your environment variables by running `setenv_ale.bat` from the Command prompt as follows:

```
> setenv_ale.bat
```

   Make sure that you are in the `bin` directory, the directory containing the edited versions of the configuration files from Step 2.

4. Generate your eLink Platform configuration file (`tuxconf`) from `cr3_ale.ubb` using the following command:

```
> tmloadcf cr3_ale.ubb
```

5. Generate your transaction log (TLOG) file by running `cr3_tlog.nt` from the DOS prompt as follows:

```
> nmake -f cr3_tlog.nt
```

6. Generate your queue spaces and queues by running `cr3_queues.nt` from the DOS prompt as follows:

```
> nmake -f cr3_queues.nt
```

7. Start the eLink Platform with the following command:

```
> tmboot -y
```

8. If you need to change your configuration, repeat Step 2 (editing the configuration files only as necessary to implement the configuration change you want to make), shutdown the eLink Platform servers, then perform Steps 3 through 7. Shutdown the eLink Platform servers by performing the following series of commands:

   a. Shutdown all R/3-to-eLink (outbound) servers with the following command:

      ```
      > tmshutdown -i ServerID -k KILL
      ```

      where *ServerID* is the unique ID of the R/3-to-eLink server. This would be whatever is set as the SRVID parameter for all cr3aleout servers in cr3_ale.ubb. In the sample cr3_ale.ubb listed in Appendix A, "Sample Configuration Files," the *ServerID* value is "4."

   b. Shutdown the remaining eLink Platform servers with the command:

      ```
      > tmshutdown -y
      ```

   c. Repeat the command to make sure that all of the eLink Platform servers are shut down:

      ```
      > tmshutdown -y
      ```

Glossary

ABAP/4

The SAP internal programming language of R/3.

ALE

SAP's application link enabling technology that provides distributed processing for R/3 Systems and third-party systems through the broadcast and guaranteed delivery of IDOCs to their appropriate destination(s).

eLink-to-R/3

The BEA eLink Adapter for R/3 ALE component server that processes and submits eLink-to-R/3 IDOCs to R/3 for ALE processing. Since any IDOC movement is from an R/3 point of view, this is known as the inbound server.

R/3-to-eLink

The BEA eLink Adapter for R/3 ALE component server that receives and processes R/3-to-eLink IDOCs from R/3. Since any IDOC movement is from an R/3 point of view, this is known as the outbound server.

BAPI

SAP's Business Application Programming Interface, which provides remotely callable RFCs (methods) that are associated with R/3 Business Objects.

BDC

SAP Batch Data Control, which processes screen-oriented data in transactions.

BO

SAP Business Object framework that provides an object-oriented model of the enterprise. For example, a Sales Order is represented in R/3 as a Business Object.

BOR

SAP Business Object Repository that contains the definitions of R/3 Business Objects and their associated BAPIs.

eLink Adapter for R/3 ALE

BEA eLink Adapter for R/3 ALE (application link enabling technology) is the infrastructure of choice for application integration of the SAP R/3 environment. It works with mission-critical, high-performance middleware to enable easy application integration with and real-time access to SAP R/3 transactions, functions and data.

FML

BEA Field Manipulation Language, a type of message buffer in ATMI. In BEA eLink Adapter for R/3 ALE documentation, FML *always* refers to FML32.

IDOC

SAP Intermediate Document (a flat file record of data) that is distributed via ALE.

R/3

SAP client-server product that provides access to the R/3 System via a three-tier architecture consisting of database, application, and presentation components.

RFC

SAP Remote Function Call interface that allows non-R/3 Systems to invoke remotely callable ABAP/4 functions. Synchronous RFC, transactional RFC (tRFC), and asynchronous RFC (aRFC) styles are supported.

SAPGUI

SAP graphical user interface utility.

TID

Transaction ID for tracking IDOC transactions.

Index

D

E

F

G

I

K

L

N

P

Q