



THE ENTERPRISE MIDDLEWARE SOLUTION

# BEA eLink Adapter for R/3 User Guide

BEA eLink Adapter for R/3, Version 1.4  
Document Edition 1.1.3  
August 1999

# Copyright

Copyright © 1998, 1999 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

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

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

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

## Trademarks or Service Marks

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

eSolutions<sup>TM</sup> and eLink<sup>TM</sup> are trademarks.

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

BEA eLink Adapter for R/3 User Guide

Document Edition	Part Number	Date	Software Version
1.1.3	870-001004-001	August 1999	BEA eLink Adapter for R/3 v1.4

---

# Contents

## About this Document

Purpose of This Document .....	vii
Who Should Read This Document .....	vii
Prerequisite Knowledge .....	viii
How This Document Is Organized .....	viii
Topics Not Covered in This Document.....	viii
Version Reference .....	ix
How to Use This Document .....	ix
Opening the Document in a Web Browser .....	x
Printing from a Web Browser .....	x
Documentation Conventions .....	x
Related Documentation .....	xii
BEA eLink Adapter for R/3 Documentation.....	xii
BEA Publications .....	xiii
Other Publications .....	xiii
Contact Information.....	xiii
Documentation Support .....	xiv
Customer Support .....	xiv

## 1. Introducing BEA eLink Adapter for R/3

About BEA eLink Adapter for R/3 .....	1-1
Key Benefits .....	1-2
The SAP R/3 Environment.....	1-2
Application-to-Application Integration.....	1-2
BEA Enterprise Application Integration Solution .....	1-3
BEA eLink Adapter for R/3 Business Applicability .....	1-3
Architectural Overview .....	1-4

---

Integration Strategies .....	1-4
Integrating Applications with R/3 .....	1-6
TUXEDO ATMI .....	1-6
FML32 Field Definitions .....	1-6

## 2. Integrating with RFCs

About RFC Integration .....	2-2
Invoking ABAP/4 Functions .....	2-3
About the RFC Inbound Server .....	2-3
Information Flow .....	2-4
About ABAP/4 Parameters .....	2-7
FML32 Field Definitions for RFC Inbound .....	2-8
Request Buffers .....	2-8
Response Buffers .....	2-9
Configuring Export Parameters and Export Tables .....	2-11
Handling Problems with RFCs .....	2-12
Handling R/3 Errors .....	2-12
Handling Adapter Errors .....	2-12
Checking the Error Log .....	2-13

## 3. Integrating with ALE

About ALE Integration .....	3-1
Usage Scenarios for ALE Integration .....	3-2
Integrating R/3 and Non-R/3 Systems .....	3-2
Communicating Among R/3 Logical Systems .....	3-2
Information Flow for ALE Integration .....	3-4
Key ALE Concepts .....	3-5
Intermediate Documents (IDOCs) .....	3-5
Transaction IDs (TIDs) .....	3-7
Processing Inbound IDOCs .....	3-7
About the ALE Inbound Server .....	3-8
ALE Inbound Services (CR3_SUBMIT and CR3_IDOC_IN) .....	3-8
FML32 Field Definitions for Inbound IDOCs .....	3-8
Information Flow for Inbound IDOCs .....	3-9
Splitting Inbound IDOC Packets .....	3-11

---

Load Balancing High Volumes of Inbound IDOCs .....	3-12
Managing Transactional Integrity for Inbound IDOCs .....	3-12
About the TID Log File Used for Inbound IDOCs .....	3-12
Processing TIDs with Inbound IDOCs .....	3-13
Handling Problems with Inbound IDOCs .....	3-16
Processing Outbound IDOCs .....	3-17
About the ALE Outbound Client .....	3-17
Information Flow for Outbound IDOCs .....	3-18
Splitting Outbound IDOC Packets Into Individual IDOCs .....	3-20
Enqueuing Outbound IDOCs Into Multiple Queues .....	3-21
About the Destination Map File .....	3-22
Examples of Using a Destination Map File .....	3-24
Managing Data-Depending Routing (DDR) .....	3-28
Load Balancing High Volumes of Outbound IDOCs .....	3-28
Registering Multiple Program IDs .....	3-29
Managing Transactional Integrity for Outbound IDOCs .....	3-30
About the TID Log File Used for Outbound IDOCs .....	3-31
Processing TIDs with Outbound IDOCs.....	3-32
Handling Problems with Outbound IDOCs .....	3-34

## Glossary

## Index



---

# About this Document

This preface includes the following sections:

- ◆ Purpose of This Document
- ◆ How to Use This Document
- ◆ Related Documentation
- ◆ Contact Information

## Purpose of This Document

This document describes the BEA eLink Adapter for R/3 product and gives instructions for using BEA eLink Adapter for R/3 to build TUXEDO client and server applications that integrate with SAP R/3 Systems.

## Who Should Read This Document

This document is intended for system integrators with programming backgrounds and an understanding of the R/3 product in at least some particular application space. Extensive basic knowledge of R/3 is not required, but may be helpful in learning BEA eLink Adapter for R/3. This document provides detail on using BEA eLink Adapter for R/3 to develop on-line interconnections to the R/3 application using the TUXEDO middleware. For system integrators concerned with the development of a client-server interface between R/3 and other applications, this user guide addresses the R/3 integration aspects. It does not cover any other applications or application wrappers.

---

## Prerequisite Knowledge

Users of this guide are expected to have the following skill set, in order to understand the context of this document:

- ◆ R/3 general product knowledge, including the SAPGUI interface and ALE and RFC concepts.
- ◆ Business application knowledge in some specific application area.
- ◆ Knowledge of the R/3 processes and data model in the required application area.
- ◆ General knowledge of client-server concepts.
- ◆ General knowledge of TUXEDO middleware architecture.

## How This Document Is Organized

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

- ◆ Chapter 1, “Introducing BEA eLink Adapter for R/3,” provides the conceptual information needed to understand how to use BEA eLink Adapter for R/3 to integrate applications with SAP R/3 systems.
- ◆ Chapter 2, “Integrating with RFCs,” describes how to integrate applications with R/3 using SAP’s Remote Function Call (RFC) interface.
- ◆ Chapter 3, “Integrating with ALE,” describes how to integrate applications with R/3 to process Intermediate Documents (IDOCs) using SAP’s Application Linking and Enabling (ALE) technology.
- ◆ Glossary

## Topics Not Covered in This Document

This document does not cover the following related topics:

- ◆ Interfacing to R/3 with file-based ("not on-line") processes

- 
- ◆ Business analysis of R/3
  - ◆ Interconnection architectures for specific businesses

## Version Reference

This document is validated for the following versions of the layered products:

**Table 0-1 Supported Platforms**

Product	Version(s)
SAP R/3	3.1H
TUXEDO	6.4
HP-UX	10.20, 11.00
Solaris	2.5, 2.6
AIX	4.1.4, 4.2.1, 4.3
Windows NT	4.0 (Service Pack 3)

## How to Use This Document

This document, BEA eLink Adapter for R/3 User Guide, is designed primarily as an online, hypertext document. If you are reading this as a paper publication, note that to get full use from this document you should install and access it as an online document via a Web browser.

The following sections explain how to view this document online, and how to print a copy of this document.

---

## Opening the Document in a Web Browser

To access the online version of this document, open the following HTML file in a Web browser:

`<your eLink app directory>/docs/index.htm`

**Note:** The online documentation requires a Web browser that supports HTML version 3.0. Netscape Navigator version 2.02 or Microsoft Internet Explorer version 3.0 or later are recommended.

## Printing from a Web Browser

You can print a copy of this document, one file at a time, from the Web browser. Before you print, make sure that the chapter or appendix you want is displayed and *selected* in your browser. (To select a chapter or appendix, click anywhere inside the chapter or appendix you want to print. If your browser offers a Print Preview feature, you can use the feature to verify which chapter or appendix you are about to print.)

The BEA eLink Adapter for R/3 Online Documentation CD also includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document.

## Documentation Conventions

The following documentation conventions are used throughout this document.

Convention	Item
<b>boldface text</b>	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
<i>italics</i>	Indicates emphasis or book titles.

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

---

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

---

## Related Documentation

The following sections list the documentation provided with the BEA eLink Adapter for R/3 software as well as other publications related to SAP's R/3 technology.

## BEA eLink Adapter for R/3 Documentation

The BEA eLink Adapter for R/3 information set consists of the following documents:

- ◆ *BEA eLink Adapter for R/3 Installation and Configuration Guide*
- ◆ *BEA eLink Adapter for R/3 User Guide*

**Note:** The BEA eLink Adapter for R/3 Online Documentation CD also includes Adobe Acrobat PDF files of all of the online documents. You can use the Adobe Acrobat Reader to print all or a portion of each document.

---

## BEA Publications

The following TUXEDO-related BEA publications are also available:

- ◆ *BEA TUXEDO Administrator's Guide*
- ◆ *BEA TUXEDO Installation and Configuration Guide* for UNIX
- ◆ *BEA TUXEDO Programmer's Guide*
- ◆ *FML Programmer's Guide*
- ◆ */Q Guide*

## Other Publications

For more information about SAP's R/3 technology, refer to the following SAP documents. R/3 documentation is provided typically on-line or on CD-ROM. The following headings, based on the SAP R/3 CD-ROM for Version 3.1H of R/3, are applicable:

- ◆ Application Link Enabling (ALE)
- ◆ Remote Communications

## Contact Information

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

---

# Documentation Support

If you have questions or comments on the documentation, you can contact the BEA Information Engineering Group by e-mail at **docsupport@beasys.com** or by telephone at **+1.408.542.4193**. (For information about how to contact Customer Support, refer to the following section.)

## Customer Support

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

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

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

# 1 Introducing BEA eLink Adapter for R/3

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

- About BEA eLink Adapter for R/3
- Integration Strategies
- Integrating Applications with R/3

## About BEA eLink Adapter for R/3

BEA eLink Adapter for R/3 is the infrastructure of choice for application integration of the SAP R/3 environment. BEA eLink Adapter for R/3 enables easy application integration using mission-critical, high-performance middleware to provide real-time access to SAP R/3 transactions, functions or data.

BEA eLink Adapter for R/3 is a production-proven solution that distributes access to SAP R/3 applications running in all industry-leading computer environments. BEA eLink Adapter for R/3 is an enterprise-based approach to integrating SAP R/3 with other applications.

# Key Benefits

- 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.

# The SAP R/3 Environment

Throughout the evolution of enterprise computing, corporations have deployed systems and software in an effort to improve business processes, reduce costs, and increase overall organizational effectiveness. SAP R/3 is now widely used by businesses as a complete information technology solution that provides an integrated suite of financial, distribution, human resources, and manufacturing applications which can be customized to meet customer needs.

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

# Application-to-Application Integration

The need for application-to-application integration has created the opportunity for a BEA middleware solution. BEA eLink Adapter for R/3 is a robust and comprehensive business solution that can be rapidly implemented and easily maintained, does not require custom interface programs, and supports a heterogeneous mix of platforms, architectures, databases, and third party applications.

## **BEA Enterprise Application Integration Solution**

BEA eLink Adapter for R/3 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 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 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.

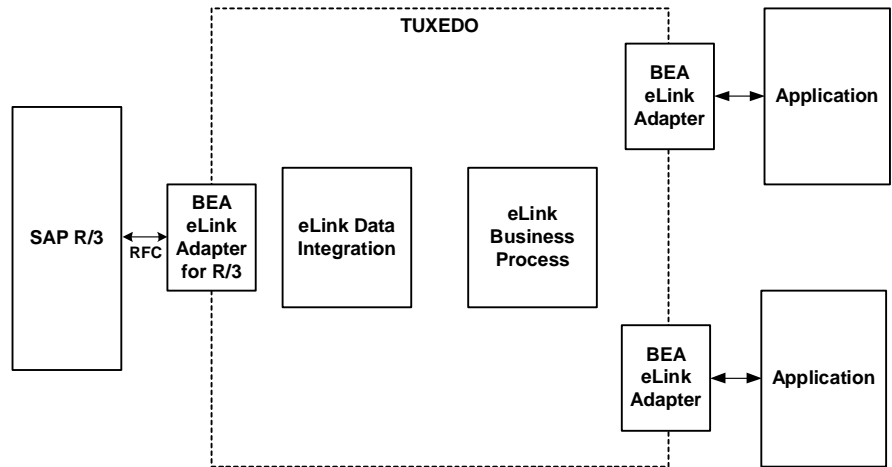
## **BEA eLink Adapter for R/3 Business Applicability**

BEA eLink Adapter for R/3 empowers the successful integration of applications with SAP R/3 in heterogeneous multi-platform environments and 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's integration efficiency has the robustness and high-performance required of a permanent element in the overall business solution.

# Architectural Overview

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

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



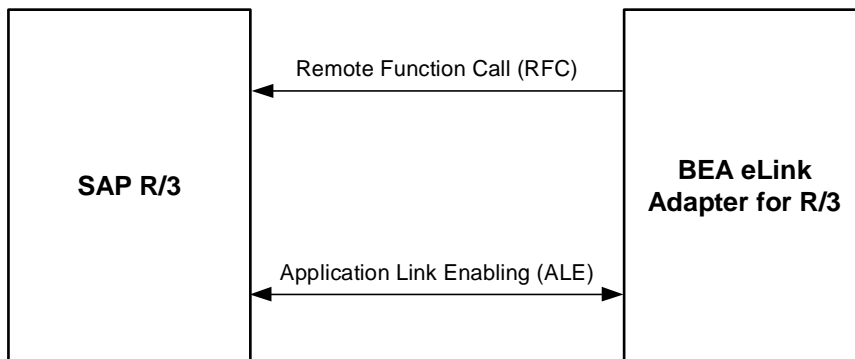
BEA eLink Adapter for R/3 includes the following components:

- **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 communicates with R/3 via SAP's Remote Function Call (RFC) protocol.

## Integration Strategies

Figure 1-2 shows the application integration strategies that BEA eLink Adapter for R/3 provides:

**Figure 1-2 Application Integration Strategies for BEA eLink Adapter for R/3**

Supported strategies include:

- **Remote Function Call (RFC).** Acting as clients to R/3, TUXEDO applications can invoke RFC-enabled ABAP/4 functions on a remote R/3 system (RFC Inbound). See Chapter 2, “Integrating with RFCs,” for more information.
- **Application Link Enabling (ALE).** TUXEDO applications can send (ALE Inbound) and receive (ALE Outbound) SAP Intermediate Documents (IDOCs) with R/3. ALE is SAP’s technology that supports the design, implementation, and operation of distributed business processes. See Chapter 3, “Integrating with ALE,” for more information.

See your SAP documentation for more information about SAP’s RFC and ALE technologies.

# Integrating Applications with R/3

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

- TUXEDO ATMI
- FML32 Field Definitions

## TUXEDO ATMI

BEA eLink Adapter for R/3 and TUXEDO applications communicate by exchanging TUXEDO FML32 buffers. Client applications that need access to the data and functionality of R/3 will use the TUXEDO ATMI to send request messages to the BEA eLink Adapter for R/3 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 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)*.

## FML32 Field Definitions

BEA eLink Adapter for R/3 comes with an FML field table (cr3.fml file) that defines the fields used in FML32 buffers. This file resides in the following directory:

```
<your eLink installation path>/elink/adapters/sapr3/config
```

Listing 1-1 shows the contents of this file:

**Listing 1-1 FML Field Table (cr3.fml)**


---

# eLink Adapter for R/3 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				
# name	number	type	flags	comments
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.fml File**

---

Field Name	Description
CR3_RFC_TID	Field that specifies the transaction ID. See “Transaction IDs (TIDs)” in Chapter 3, “Integrating with ALE.” <i>Internal Use Only.</i>
CR3_IDOC_CONTROL	Contains one or more control records for the IDOC packet. <i>Internal Use Only.</i>
CR3_IDOC_DATA	Contains one or more data records for the IDOC packet. <i>Internal Use Only.</i>

---

**Table 1-1 FML Field Definitions in the cr3.fml File**

Field Name	Description
CR3_IDOC	Field that contains IDOC packet data. See Chapter 3, “Integrating with ALE.”
CR3_TARGET_ID	Data Dependent Routing value. See Chapter 3, “Integrating with ALE.”
CR3_ERROR_TEXT	Error text.
ELINK_ADAPTER_ERR_CODE	eLink error category.
ELINK_ADAPTER_ERR	BEA eLink Adapter for R/3 error code and text.
ELINK_APP_ERR	SAP R/3-level error. See Chapter 2, “Integrating with RFCs.”

**Note:** BEA eLink Adapter for R/3 imposes no restriction on the length of string values sent and received using 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 RFCs

This chapter describes how to integrate with SAP Remote Function Calls (RFCs) in the BEA eLink Adapter for R/3 environment. It includes the following sections:

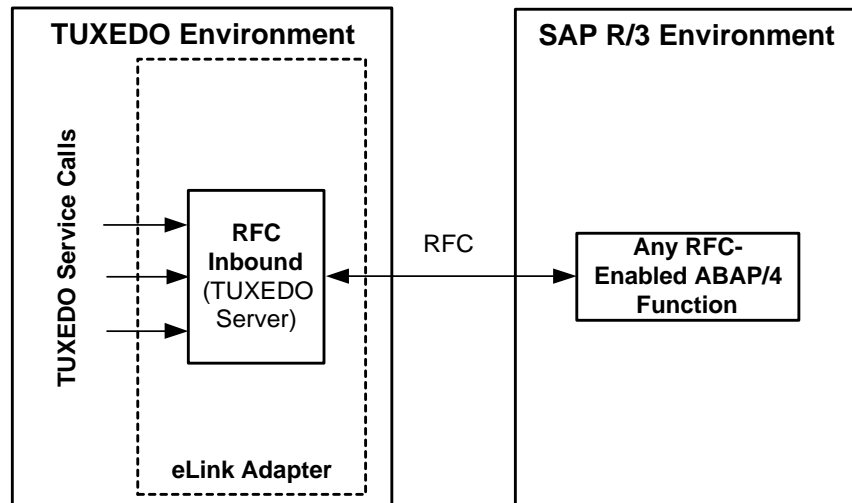
- About RFC Integration
- Invoking ABAP/4 Functions

For information about setting up RFC processing, see Chapter 4, “Configuring for RFC Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*.

# About RFC Integration

Figure 2-1 provides an overview of the information flow for RFC Inbound:

**Figure 2-1 Overview of Information Flow for RFC Processing**



RFC Inbound is a TUXEDO server that allows TUXEDO applications to invoke any ABAP/4 functions that have been RFC-enabled in R/3. See “Invoking ABAP/4 Functions” on page 2-3 for more information.

# Invoking ABAP/4 Functions

This section describes how to invoke RFC-enabled ABAP/4 functions in the TUXEDO environment using RFC Inbound, a TUXEDO server, via synchronous RFC. It includes the following topics:

- About the RFC Inbound Server
- Information Flow
- About ABAP/4 Parameters
- FML32 Field Definitions for RFC Inbound
- Configuring Export Parameters and Export Tables
- Handling Problems with RFCs

For information about setting up RFC processing, see the following chapters in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*:

- Chapter 4, “Configuring RFC Integration”
- Chapter 6, “Configuring R/3 Connections”

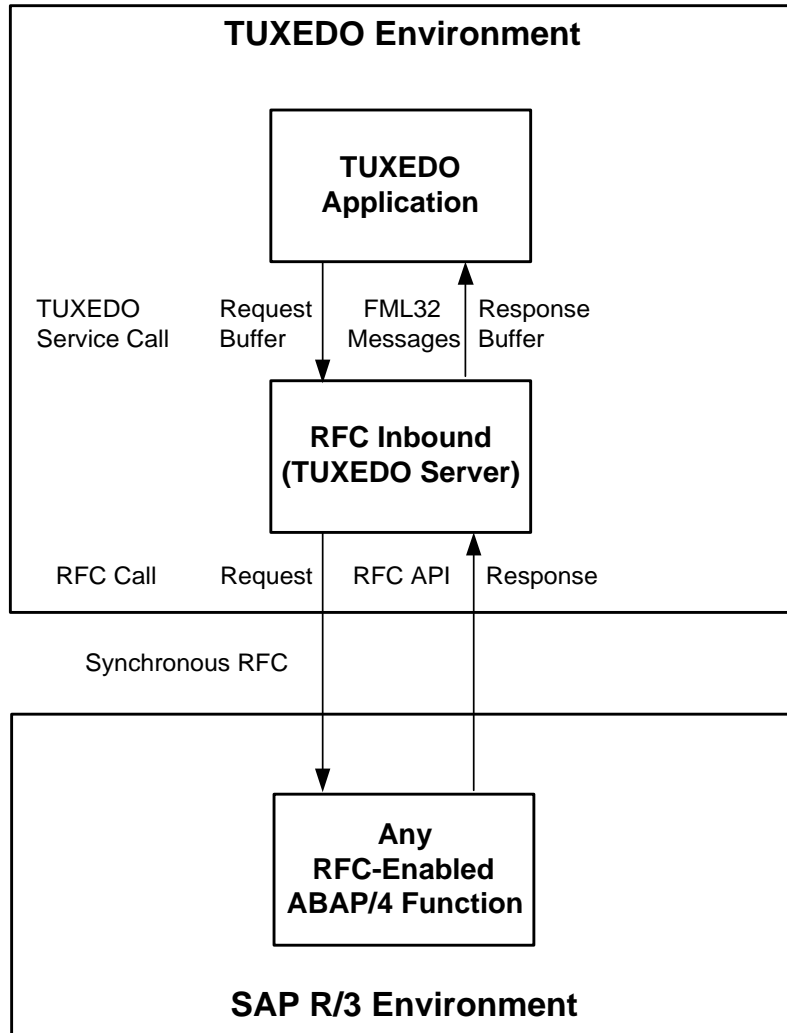
## About the RFC Inbound Server

RFC Inbound is a component of the BEA eLink Adapter for R/3 product that allows TUXEDO applications and other non-R/3 applications to invoke any ABAP/4 functions that have been RFC-enabled. RFC Inbound is a TUXEDO server that has a generic service, CR3\_RFC\_IN, which invokes any ABAP/4 function. The CR3\_RFC\_IN service is aliased to unique services corresponding to ABAP/4 functions. RFC Inbound advertises the ABAP/4 functions that it can invoke as TUXEDO services and aliases them to the CR3\_RFC\_IN service. Developers can configure which ABAP/4 functions can be invoked and the service names by which they are advertised.

## Information Flow

RFC Inbound allows a TUXEDO application to act as a client to R/3. Figure 2-2 shows the information flow for invoking RFC-enabled ABAP/4 functions:

**Figure 2-2 Information Flow for Invoking ABAP/4 Functions**



The information flow for invoking ABAP/4 functions is as follows:

1. One or more instances of RFC Inbound (a TUXEDO server) start up and advertise each ABAP/4 function that can be invoked as a TUXEDO service.
2. The calling application prepares an FML32 request buffer that contains the applicable import parameters and import tables for the RFC.

It is the developer's responsibility to ensure that the names of import parameters and import tables are spelled exactly as they appear in the ABAP/4 function signature.

3. The calling application makes a service call to the TUXEDO service associated with the RFC, passing in the FML32 request buffer.
4. The RFC Inbound service (CR3\_RFC\_IN) determines the service name used to invoke it and therefore knows the associated ABAP/4 function.
5. RFC Inbound processes the FML32 request buffer and then makes the remote function call on R/3 using the synchronous RFC API, passing the import parameters to R/3.
6. The executed ABAP/4 function processes the input parameters and returns the results back to RFC Inbound.
7. RFC Inbound receives the results and, depending on the success of the RFC call, does one of the following:
  - If the RFC call succeeded, RFC Inbound populates the FML32 response buffer with the returned data. RFC Inbound includes only data for the export parameters and export tables that are listed in the CR3\_EXPORT\_PARAMS and CR3\_EXPORT\_TABLES environment variables. RFC Inbound returns the FML32 response buffer to the calling application, as well as TPSUCCESS with the tpurcode set to zero (0).
  - If the RFC call failed, RFC Inbound populates the FML32 response buffer with information about the error:

**R/3-level errors.** RFC Inbound populates the ELINK\_APP\_ERR field with the message text returned from R/3. RFC Inbound returns the FML32 response buffer to the calling application, as well as TPFAIL with the tpurcode set to 0.

**Adapter-level errors.** RFC Inbound populates the ELINK\_ADAPTER\_ERR\_CODE and ELINK\_ADAPTER\_ERR fields with

the appropriate adapter-level error code and BEA eLink Adapter for R/3-specific error code and message. RFC Inbound returns the FML32 response buffer to the calling application, as well as TPFAIL with the tpurcode set to -1. Alternatively, for uncorrectable errors such as a memory corruption problem, RFC Inbound returns TPEXIT (which causes the service to shut down) with the tpurcode set to -1.

8. The calling application processes the response buffer as appropriate.

## About ABAP/4 Parameters

ABAP/4 functions use the following types of parameters:

- **Import parameters** are passed into an ABAP/4 function.
- **Export parameters** are passed out of an ABAP/4 function.
- **Tables** are passed into, out of, or both into *and* out of an ABAP/4 function. Tables are similar to arrays, consisting of 1+ rows, where each row is a specific type (such as a string) or a structure. Tables are passed by reference.
- **Exceptions** are defined error messages that are passed out of an ABAP/4 function if R/3 raises an exception (a business-level error).

The exact parameters vary with each ABAP/4 function and are defined in the function's ABAP/4 function signature. Listing 2-1 defines a generic ABAP/4 function:

---

### Listing 2-1 ABAP/4 Function Signature

---

```
FUNCTION FUNCTION_NAME
IMPORT
    IMPORT_PARAM_1
    IMPORT_PARAM_2
EXPORT
    EXPORT_PARAM_1
    EXPORT_PARAM_2
TABLE
    TABLE_IN
    TABLE_OUT
    TABLE_IN_OUT
EXCEPTION
    EXCEPTION_1
    EXCEPTION_2
```

---

This function signature includes two import parameters and one import table that it reads as input, two export parameters and one export table that it writes as output, one table (TABLE\_IN\_OUT) that it reads on input and writes as output, and two exceptions that it can raise if an R/3 error occurs.

# FML32 Field Definitions for RFC Inbound

In TUXEDO messages, import parameters, export parameters, table, and exceptions are represented as named fields in FML32 buffers. A calling application uses two kinds of buffers:

- **Request Buffers** contain import parameters and import tables.
- **Response Buffers** contain export parameters and export tables as well as success/failure information (R/3-level errors or adapter errors).

The calling application constructs a request buffer containing import parameters and import tables. RFC Inbound allocates a response buffer and populates it with data returned from the ABAP/4 function according to the settings of the `CR3_EXPORT_PARAMS` and `CR3_EXPORT_TABLES` environment variables in the configuration file.

## Request Buffers

Before invoking a service associated with an RFC call, a calling application prepares an FML32 request buffer that contains the following information:

- **Import parameters.** The Field Name is the import parameter name as specified in the ABAP/4 function signature and the Field Value is the parameter value to pass into the ABAP/4 function. Each import parameter consists of only one field.
- **Import tables.** Tables consist of one or more rows of data. Each row is an occurrence of an FML32 field. The Field Name is the table name as specified in the ABAP/4 function signature and the Field Value is a string containing the row data to pass into the ABAP/4 function. The calling application is responsible for formatting the row data correctly in the fixed-position format that the ABAP/4 function recognizes.

The calling application passes this request buffer when it invokes the service associated with the RFC.

For example, for the function signature shown in Listing 2-1, “ABAP/4 Function Signature,” on page 2-7, a calling application might construct the following FML32 request buffer:

**Table 2-1 Request Buffer for a Sample ABAP/4 Function**

FML32 Field Name	Field Value	Data Type
IMPORT_PARAM_1	ParameterValue	String
IMPORT_PARAM_2	ParameterValue	String
TABLE_IN	TableRowValue	String
TABLE_IN	TableRowValue	String
TABLE_IN	TableRowValue	String
..	..	String
TABLE_IN_OUT	TableRowValue	String
TABLE_IN_OUT	TableRowValue	String
TABLE_IN_OUT	TableRowValue	String
..	..	String

In this example, the TABLE\_IN\_OUT table is both an import and export table. Therefore, the request buffer contains table rows to be passed into the ABAP/4 function. In addition, TABLE\_IN\_OUT is specified in the CR3\_EXPORT\_TABLES environment variable in the configuration file so that data from this table is returned in the response buffer.

**Note:** It is the responsibility of the calling application to ensure that the names of import parameters and import tables are spelled exactly as they appear in the ABAP/4 function signature.

## Response Buffers

The RFC Inbound allocates an FML32 response buffer. After executing the ABAP/4 function, RFC Inbound populates the response buffer with the results, which include:

- **Export parameters.** The Field Name is the export parameter name as specified in the ABAP/4 function signature and the Field Value is the parameter value returned from the ABAP/4 function. Each export parameter consists of only one field.
- **Export tables.** Tables consist of one or more rows of data. Each row is an occurrence of an FML32 field. The Field Name is the table name as specified in the ABAP/4 function signature and the Field Value is a string containing the row data returned from the ABAP/4 function. The calling application is responsible for anticipating the fixed-position format of the row data returned from R/3 and for retrieving and processing the row data accordingly.
- **R/3 error information.** The ELINK\_APP\_ERR field contains the text of an R/3 error, if any, that was raised by the ABAP/4 function.
- **Adapter error information.** Two FML32 fields, ELINK\_ADAPTER\_ERR\_CODE and ELINK\_ADAPTER\_ERR, contain information about adapter errors.

For example, for the function signature specified in Listing 2-1, “ABAP/4 Function Signature,” on page 2-7, RFC Inbound might return the following FML32 response buffer:

**Table 2-2 Response Buffer for Sample ABAP/4 Function**

FML32 Field Name	Field Value	Data Type
EXPORT_PARAM_1	ParameterValue	String
EXPORT_PARAM_2	ParameterValue	String
TABLE_OUT	TableRowValue	String
TABLE_OUT	TableRowValue	String
TABLE_OUT	TableRowValue	String
..	..	String
TABLE_IN_OUT	TableRowValue	String
TABLE_IN_OUT	TableRowValue	String
TABLE_IN_OUT	TableRowValue	String
..	..	String

**Note:** Error fields (ELINK\_APP\_ERR, ELINK\_ADAPTER\_ERR\_CODE, and ELINK\_ADAPTER\_ERR) appear in the response buffer only if an R/3 error or adapter error occurs.

## Configuring Export Parameters and Export Tables

Two environment variables control which export parameters and export tables are included in the response buffer:

- CR3\_EXPORT\_PARAMS specifies the list of export parameters that RFC Inbound includes in the response buffer.
- CR3\_EXPORT\_TABLES specifies the list of export tables that RFC Inbound includes in the response buffer.

After the ABAP/4 function is executed, RFC Inbound populates the response buffer with these values. If an export parameter or export table is not specified in this list, RFC Inbound discards it even if it is returned from the ABAP/4 function. See “Configuring RFCs” in Chapter 4, “Configuring RFC Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*.

# Handling Problems with RFCs

Two kinds of problems can arise when executing RFCs:

- **R/3 errors** indicate business-level errors that are raised by the ABAP/4 function, such as invalid import data. If an R/3 error occurs, RFC Inbound returns a response buffer containing *only* the `ELINK_APP_ERR` field.
- **Adapter errors**, which indicate any other infrastructure-level problems, such as an R/3 connection error. If an adapter error occurs, RFC Inbound returns a response buffer containing *only* the `ELINK_ADAPTER_ERR_CODE` and `ELINK_ADAPTER_ERR` fields.

It is the responsibility of the calling application to retrieve and process exception and error information accordingly.

## Handling R/3 Errors

If an exception is raised when executing the ABAP/4 function on R/3, RFC Inbound:

- Populates the `ELINK_APP_ERR` field in the response buffer with the text of the R/3 error.
- Returns `TPFAIL` with the `tpurcode` set to zero (0).

After making the service call, the calling application should check the return value and `tpurcode`. If `TPFAIL` is returned with `tpurcode` set to zero (0), it should parse the text in the `ELINK_APP_ERR` field in the response buffer and respond appropriately.

## Handling Adapter Errors

If an adapter error occurs when executing an ABAP/4 function, RFC Inbound:

- Populates the `ELINK_ADAPTER_ERR_CODE` field in the response buffer with the adapter error category.
- Populates the `ELINK_ADAPTER_ERR` field in the response buffer with the BEA eLink Adapter for R/3 error code and message.
- Returns `TPFAIL` with the `tpurcode` set to -1 or, if the error is an uncorrectable one (such as a memory allocation failure), then RFC Inbound returns `TPEXIT` with the `tpurcode` set to -1.

- Writes a message to the error log.

After making the service call, the calling application should check the return value and tpurcode. If TPFFAIL or TPEXIT is returned and tpurcode is -1, it should parse the text in the ELINK\_APP\_ERR field in the response buffer and respond appropriately.

## Checking the Error Log

If RFC Inbound encounters an adapter level error, it logs the following information to the TUXEDO userlog:

- The adapter error category. For example, if the RFC Inbound server cannot start up, the adapter error category in the log entry is ELINK\_ECONFIG.
- An error code and text that is specific to BEA eLink Adapter for R/3 (the same text that is returned in the ELINK\_ADAPTER\_ERROR field).
- A unique code identifying where in the code the error was encountered.

Refer to your TUXEDO documentation for more information about the TUXEDO userlog.



# 3 Integrating with ALE

This chapter describes how to integrate with SAP's Application Linking and Embedding (ALE) technology in the BEA eLink environment. It includes the following sections:

- About ALE Integration
- Processing Inbound IDOCs
- Processing Outbound IDOCs

For information about setting up ALE processing, see the following chapters in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*:

- Chapter 5, "Configuring ALE Integration"
- Chapter 6, "Configuring R/3 Connections"

## About ALE Integration

This section provides important conceptual information about integrating with ALE. It includes the following topics:

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

## Usage Scenarios for ALE Integration

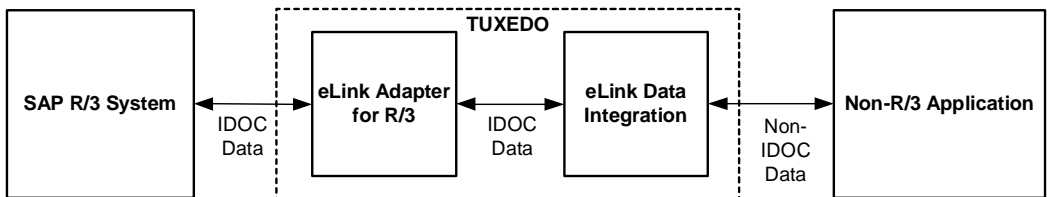
Common ALE integration implementations of BEA eLink Adapter for R/3 include:

- **Integrating R/3 and non-R/3 systems** using TUXEDO and BEA eLink Adapter for R/3 to exchange IDOCs across application and platform boundaries.
- **Communicating among R/3 systems** using TUXEDO and BEA eLink Adapter for R/3 to reliably and efficiently transport IDOCs among R/3 logical systems.

### Integrating R/3 and Non-R/3 Systems

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

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

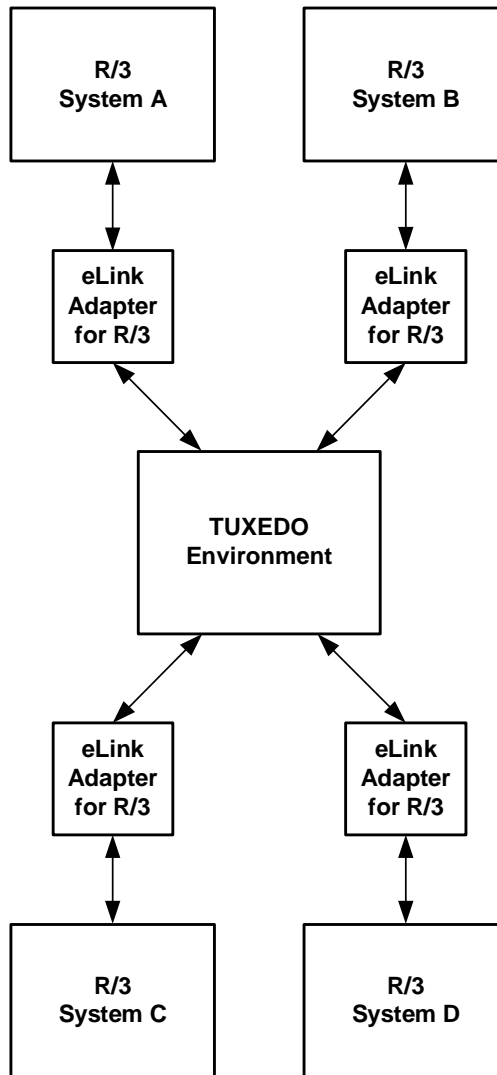


In this scenario, these BEA components provide communication and data transformation services that enable the exchange of IDOC and non-IDOC data between R/3 and non-R/3 systems.

### Communicating Among R/3 Logical Systems

Figure 3-2 shows how TUXEDO and BEA eLink Adapter for R/3 can be used to transport IDOCs among R/3 logical systems:

Figure 3-2 Communicating Among R/3 Logical Systems

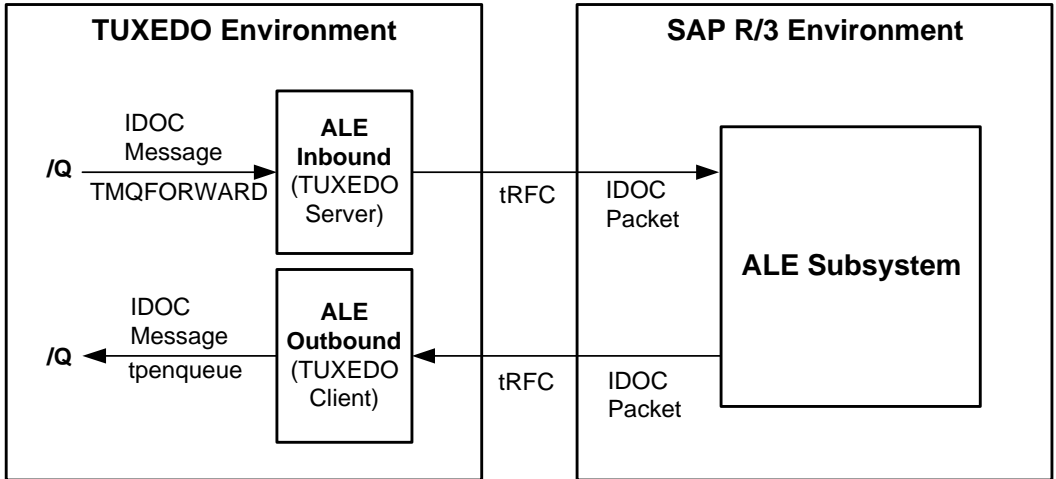


In this scenario, BEA eLink Adapter for R/3 and TUXEDO provide reliable and efficient communication services that enable the transport of IDOC packets between R/3 logical systems, thereby reducing the load on SAP communication services.

## Information Flow for ALE Integration

Figure 3-3 provides an overview of the information flow for the two main ALE processes, ALE Outbound and ALE Inbound:

**Figure 3-3 Overview of Information Flow for ALE Processing**



- **ALE Inbound** is a TUXEDO server that submits IDOC packets to R/3. ALE Outbound receives each IDOC packet as an FML32 message buffer (forwarded from a TUXEDO queue), encodes the IDOC data for R/3, and submits the IDOC packet to R/3 via tRFC. See “Information Flow for Inbound IDOCs” on page 3-9 for more information.
- **ALE Outbound** is a TUXEDO client that receives IDOC packets from R/3 via Transactional RFC (tRFC). ALE Outbound encodes each IDOC packet data into an FML32 message buffer and enqueues it into one or more TUXEDO queues. See “Information Flow for Outbound IDOCs” on page 3-18 for more information.

## Key ALE Concepts

This section describes the following key concepts used in ALE processing:

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

### Intermediate Documents (IDOCs)

In SAP's 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:

- *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.
- *Data records* uniquely identify a segment that contains application data. The length of a data record is 1055 bytes, 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 3-4:

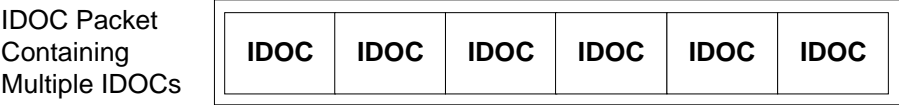
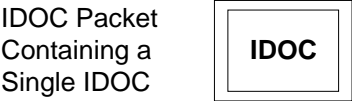
**Figure 3-4 Structure of an ALE IDOC**

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

For inbound IDOCs, ALE Inbound validates the size and structure of control records and data records. For each IDOC, ALE Inbound 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 3-5:

**Figure 3-5 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.fml file, as described in “FML32 Field Definitions” in Chapter 1, “Introducing BEA eLink Adapter for R/3.”

## 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:

- **Inbound.** For inbound IDOC packets, R/3 uses TIDs to guarantee receipt once and only once.
- **Outbound.** For outbound IDOC packets, R/3 uses TIDs to guarantee delivery once and only once.

In the TUXEDO environment, ALE Outbound and ALE Inbound monitor TIDs through the use of TID log files. See “Managing Transactional Integrity for Inbound IDOCs” on page 3-12 and “Managing Transactional Integrity for Outbound IDOCs” on page 3-30 for more information.

# Processing Inbound IDOCs

This section describes how to process inbound IDOCs using the ALE Inbound server. It includes the following topics:

- About the ALE Inbound Server
- Information Flow for Inbound IDOCs
- Splitting Inbound IDOC Packets
- Load Balancing High Volumes of Inbound IDOCs
- Managing Transactional Integrity for Inbound IDOCs
- Handling Problems with Inbound IDOCs

ALE Inbound must be properly configured before it can process IDOCs. See “Configuring ALE Inbound” in Chapter 5, “Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide* for more information.

## About the ALE Inbound Server

ALE Inbound is a TUXEDO server that submits IDOC packets to R/3. ALE Inbound 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. ALE Inbound uses a TID log file to track the TIDs associated with IDOC packets to guarantee delivery to R/3 once and only once. The name of the executable for ALE Inbound is cr3alein.

### ALE Inbound Services (CR3\_SUBMIT and CR3\_IDOC\_IN)

ALE Inbound provides two services that process inbound 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 enqueues 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 Inbound IDOCs

ALE Inbound uses the following FML32 field definitions in IDOC messages:

**Table 3-1 FML32 Fields for ALE Inbound 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. <i>Internal Use Only</i> .

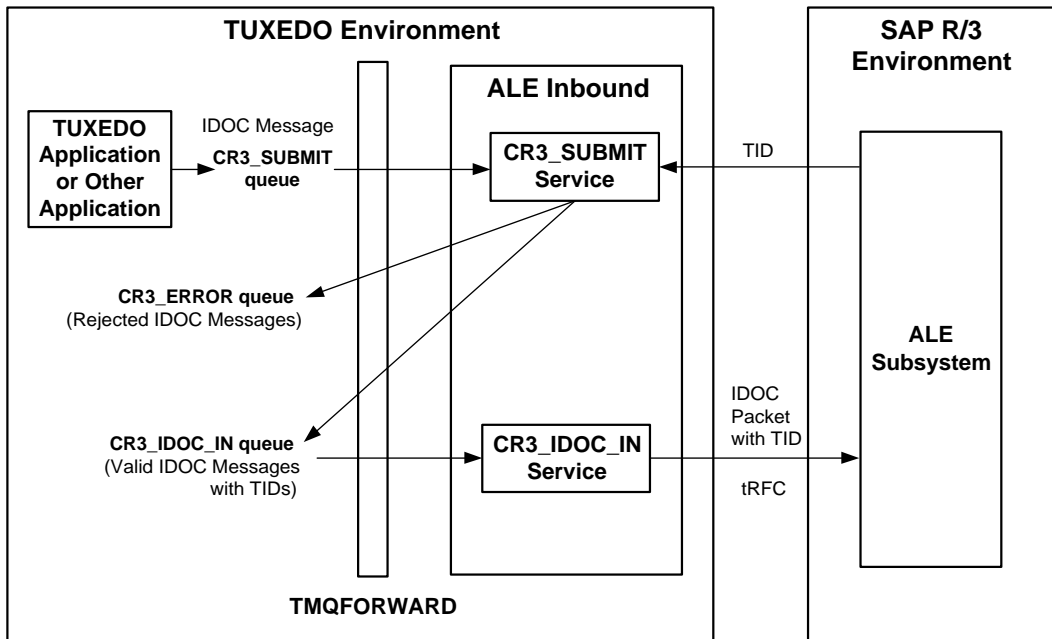
**Table 3-1 FML32 Fields for ALE Inbound Messages**

Field	Data Type	Description
CR3_IDOC_CONTROL	string	Contains one or more control records for the IDOC packet. <i>Internal Use Only.</i>
CR3_IDOC_DATA	string	Contains one or more data records for the IDOC packet. <i>Internal Use Only.</i>

These fields are defined in the cr3.fml file, as described in “FML32 Field Definitions” in Chapter 1, “Introducing BEA eLink Adapter for R/3.”

## Information Flow for Inbound IDOCs

Figure 3-6 shows the information flow for inbound IDOCs:

**Figure 3-6 Information Flow for Inbound IDOCs**

The information flow for ALE Inbound proceeds in the following sequence:

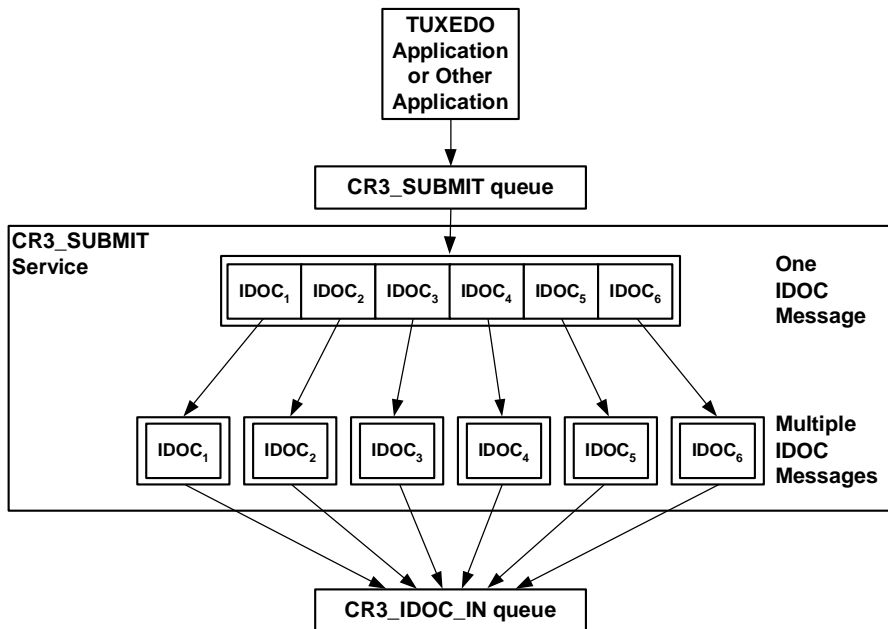
1. One or more instances of ALE Inbound (a TUXEDO server) start up.
2. An application (a TUXEDO application, eLink Data Integration, or some other tool) constructs an FML32 buffer containing IDOC data and enqueues it into the CR3\_SUBMIT queue.
3. TMQFORWARD dequeues the IDOC message from the CR3\_SUBMIT queue and submits it to the CR3\_SUBMIT service of the ALE Inbound 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 enqueues 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 ALE Inbound.
7. The CR3\_IDOC\_IN service submits the IDOC packet and TID to R/3.

ALE Inbound uses a TID log file to manage transactional integrity. See “Managing Transactional Integrity for Outbound IDOCs” on page 3-30 for more information.

## Splitting Inbound IDOC Packets

By default, ALE Inbound passes an IDOC message containing multiple IDOCs to R/3 in a single packet. You can configure ALE Inbound 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, ALE Inbound can create six IDOC separate packets, each containing a single IDOC and associated with a unique TID. Figure 3-7 shows splitting IDOC packets and enqueueing them into the CR3\_SUBMIT queue:

**Figure 3-7 Splitting Inbound IDOC Packets**



Splitting IDOC packets provides additional flexibility for processing inbound 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 Inbound IDOC Packets Containing Multiple IDOCs” in Chapter 5, “Configuring ALE Integration,” in the *BEA eLink Adapter for R/3 Installation and Configuration Guide* for instructions.

## Load Balancing High Volumes of Inbound IDOCs

Multiple instances of ALE Inbound can log onto R/3 and submit inbound IDOCs for processing. For deployments that involve high volumes of inbound IDOCs, you can enhance system performance by balancing the load across multiple instances of ALE Inbound. See “Configuring Load Balancing for Inbound IDOCs” in Chapter 5, “Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide* for instructions.

## Managing Transactional Integrity for Inbound IDOCs

ALE Inbound manages transactional integrity for inbound 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, ALE Inbound retries using the same TID. ALE Inbound uses a TID log file to track the transaction IDs (TIDs) that R/3 assigns to each inbound IDOC packet. See “Transaction IDs (TIDs)” on page 3-7 for an introduction to TIDs.

### About the TID Log File Used for Inbound IDOCs

The TID log file used with inbound IDOCs contains information about TIDs that ALE Inbound 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 3-2 Columns in the TID Log File for Inbound 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: <ul style="list-style-type: none"><li>■ <b>CREATED</b> indicates that ALE Inbound has successfully associated a TID with the IDOC packet and enqueued it into the CR3_IDOC_IN queue.</li><li>■ <b>CONFIRMED</b> indicates that ALE Inbound has successfully passed the IDOC packet onto R/3.</li></ul>

The following example shows a sample TID file for ALE Inbound:

```
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
```

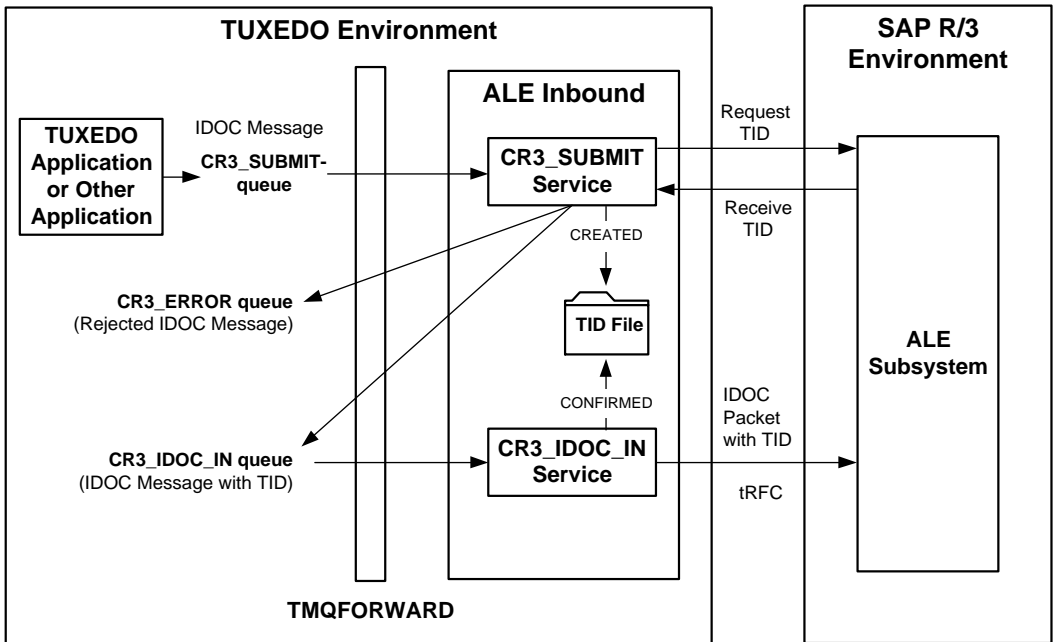
The CR3\_ALEIN\_TID\_FILE environment variable specifies the location of the TID log file for ALE Inbound. See “Setting Environment Variables for ALE Inbound” in Chapter 5, “Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide* for more information.

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,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*.

## Processing TIDs with Inbound IDOCs

Figure 3-8 shows how ALE Inbound uses the TID log file to manage transactional integrity for inbound IDOCs:

**Figure 3-8 TID Processing for ALE Inbound**



The inbound 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 enqueued 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 ALE Inbound.
- 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, dequeues an IDOC message from the CR3\_SUBMIT queue, and submits the IDOC message to the CR3\_SUBMIT service of the ALE Inbound 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, ALE Outbound 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 enqueues the message into 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 ALE Inbound 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 Inbound IDOCs

ALE Inbound uses TUXEDO's transaction management capabilities to ensure transactional integrity for inbound IDOCs. Figure 3-3 lists problems that can occur with inbound IDOCs:

**Table 3-3 Handling Problems with Inbound IDOCs**

Problem	Description
Invalid IDOC structure	If an IDOC packet fails validation, the CR3_SUBMIT service enqueues 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 enqueues the FML32 message into the CR3_ERROR queue and returns TPsUCCESS to TMQFORWARD.
TID Not received from R/3	If R/3 for some reason does not return a TID, CR3_SUBMIT 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 dequeue and handle messages in the CR3\_ERROR queue.

## Processing Outbound IDOCs

This section describes how to process outbound IDOCs using the ALE Outbound client. It includes the following topics:

- About the ALE Outbound Client
- Information Flow for Outbound IDOCs
- Splitting Outbound IDOC Packets Into Individual IDOCs
- Enqueuing Outbound IDOCs Into Multiple Queues
- Managing Data-Depending Routing (DDR)
- Load Balancing High Volumes of Outbound IDOCs
- Registering Multiple Program IDs
- Managing Transactional Integrity for Outbound IDOCs
- Handling Problems with Outbound IDOCs

ALE Outbound must be properly configured before it can process outbound IDOCs. For information about setting up ALE Outbound, see “Configuring ALE Outbound” in Chapter 5, “Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*.

## About the ALE Outbound Client

ALE Outbound is a TUXEDO client that receives IDOC packets from R/3 via Transactional RFC (tRFC). ALE Outbound encodes each IDOC packet into an FML32 message buffer and enqueues it into a TUXEDO queue. ALE Outbound uses a TID log

file to track the IDOC packets that it processes to ensure that it enqueues an IDOC packet from R/3 once and only once. The name of the executable for ALE Outbound is cr3aleout.

ALE Outbound uses the following FML32 field definitions in IDOC messages:

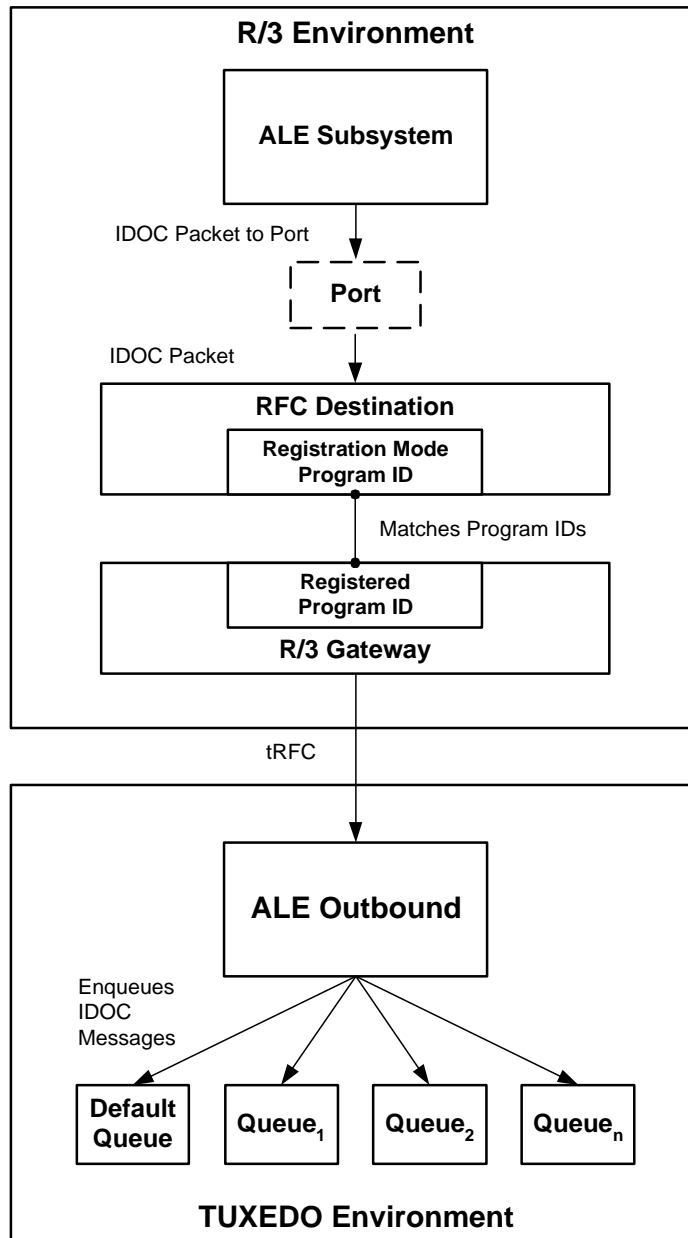
**Table 3-4 FML32 Fields for ALE Outbound Messages**

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. <i>See the BEA eLink Adapter for Mercator User Guide</i> for more information.

## Information Flow for Outbound IDOCs

Figure 3-9 illustrates the information flow for ALE Outbound:

Figure 3-9 Information Flow for Outbound IDOCs



The information flow proceeds in the following sequence:

1. One or more instances of ALE Outbound (a TUXEDO client) start up and register a program ID with the R/3 Gateway. ALE Outbound 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 ALE Outbound that is registered on the program ID of the RFC destination.
4. ALE Outbound receives the IDOC packet and processes the IDOC data according to the way that ALE Outbound is configured, with or without a destination map file:
  - If a destination map file is *not* configured, then ALE Outbound sends the IDOC packet to the default queue. If the `CR3_ALE_DEFAULT_IDOC_SPLIT` environment variable is set to “Y”, then ALE Outbound splits IDOC packets containing multiple IDOCs into separate IDOC messages, each containing a single IDOC, as described in “Splitting Outbound IDOC Packets Into Individual IDOCs” on page 3-20.
  - If a destination map file is configured, then ALE Outbound automatically splits IDOC packets containing multiple IDOCs into separate IDOC messages, each containing a single IDOC. ALE Outbound uses the settings in the destination map file to determine the target queue for each IDOC as well as other processing options. See “Enqueuing Outbound IDOCs Into Multiple Queues” on page 3-21 for more information.

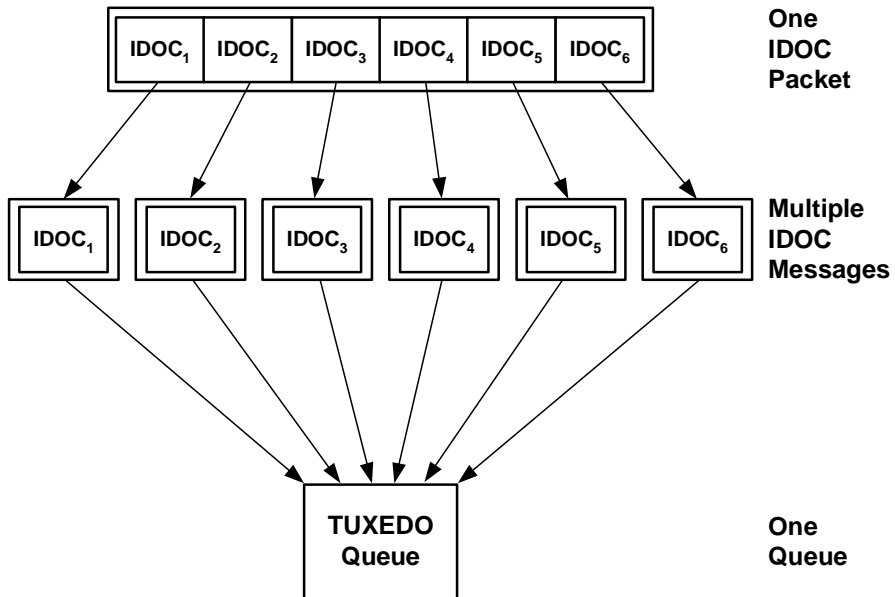
ALE Outbound uses a TID log file to manage transactional integrity. See “Managing Transactional Integrity for Outbound IDOCs” on page 3-30 for more information.

## Splitting Outbound IDOC Packets Into Individual IDOCs

You can configure ALE Outbound to split IDOC packets containing multiple IDOCs into separate IDOC messages, each containing a single IDOC. By default, ALE Outbound encodes the entire IDOC packet into a single occurrence of the `CR3_IDOC` field in the message buffer and enqueues the entire IDOC packet into a single message. If you set the `CR3_ALE_DEFAULT_IDOC_SPLIT` environment variable to “Y”,

However, ALE Outbound will split the IDOC packet into individual IDOC packets and then enqueue all IDOC packets into a single queue. Figure 3-10 shows splitting IDOC packets and enqueueing them into a single queue:

**Figure 3-10 Splitting IDOC Packets and Enqueueing to a Single Queue**



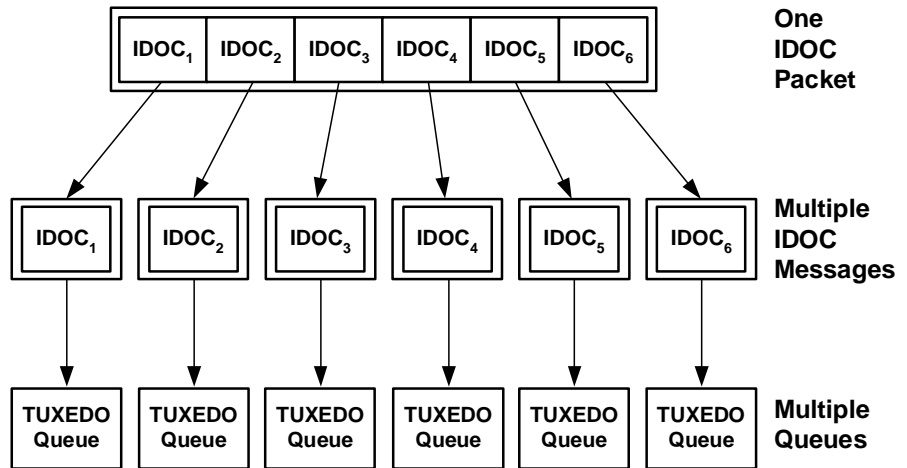
See “Setting Environment Variables for ALE Outbound” in Chapter 5, “Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide* for more information about setting the CR3\_ALE\_DEFAULT\_IDOC\_SPLIT environment variable.

## Enqueueing Outbound IDOCs Into Multiple Queues

ALE Outbound can be configured to use a destination map file so that it can enqueue IDOC messages into different target queues, manage data-dependent routing, and group similar IDOC messages into larger IDOC messages. ALE Outbound 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 a destination map file is used, ALE Outbound automatically splits IDOC packets containing multiple IDOCs into separate IDOC messages, each one containing a single IDOC, and enqueues them into target queues. Figure 3-11 shows splitting IDOC packets and enqueueing them into multiple queues (using the settings in the destination map file):

**Figure 3-11 Splitting IDOC Packets and Enqueueing to Multiple Queues**



To use a destination map file, you must set up the target queue spaces and queues (including the default queue space and queue), create the destination map file, and specify the name and location of the destination map file in the CR3\_ALEOUT\_DEST\_MAP\_FILE environment variable. See “Configuring the Destination Map File” in Chapter 5, “Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide* for more information.

## About the Destination Map File

The destination map file is an ASCII text file. Each row in the file denotes a separate entry and specifies the following information:

- Logical system ID of the target R/3 logical system.
- Message type specified in the IDOC.

- 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.
- Target ID representing the data-dependent routing value.
- Destination queue space name.
- Destination queue name.

For each outbound IDOC, ALE Outbound searches the destination map file for the target R/3 system and IDOC message type specified in an IDOC's control record. If it finds a matching entry, ALE Outbound enqueues the IDOC into the destination queue space and queue specified in the entry. ALE Outbound also processes the IDOC according to the Compress flag and the Target ID settings in the entry.

## Compressing Outbound IDOCs

You use the Compress column in the destination map file to combine IDOC messages with matching logical system IDs and IDOC message types into an IDOC message containing multiple IDOCs. For each matching entry, if the Compress column is "Y", then ALE Outbound aggregates this IDOC with other matching IDOCs in a single, larger IDOC message that it then enqueues into the appropriate target queue. If the Compress column is "N", then ALE Outbound enqueues each IDOC message separately.

## Setting the Data-Dependent Routing Value

You use the TargetId column in the destination map file to associate an IDOC with a data-dependent routing (DDR) value. For each matching entry, ALE Outbound encodes the specified TargetId value in the CR3\_TARGET\_ID FML32 field of the IDOC message. If this DDR is unspecified, or if no matching entry is found in the destination map file, then ALE Outbound encodes the default DDR value, which is defined in the CR3\_ALE\_DEFAULT\_TARGET\_ID environment variable.

**Note:** To use DDR, the DDR value must match the field ID configured in the ROUTING section of the cr3.ubb configuration file.

See "Setting the Data-Dependent Routing Value" in Chapter 5, "Configuring ALE Integration," in the BEA eLink Adapter for R/3 *Installation and Configuration Guide* for more information. See your TUXEDO documentation for more information about data-dependent routing.

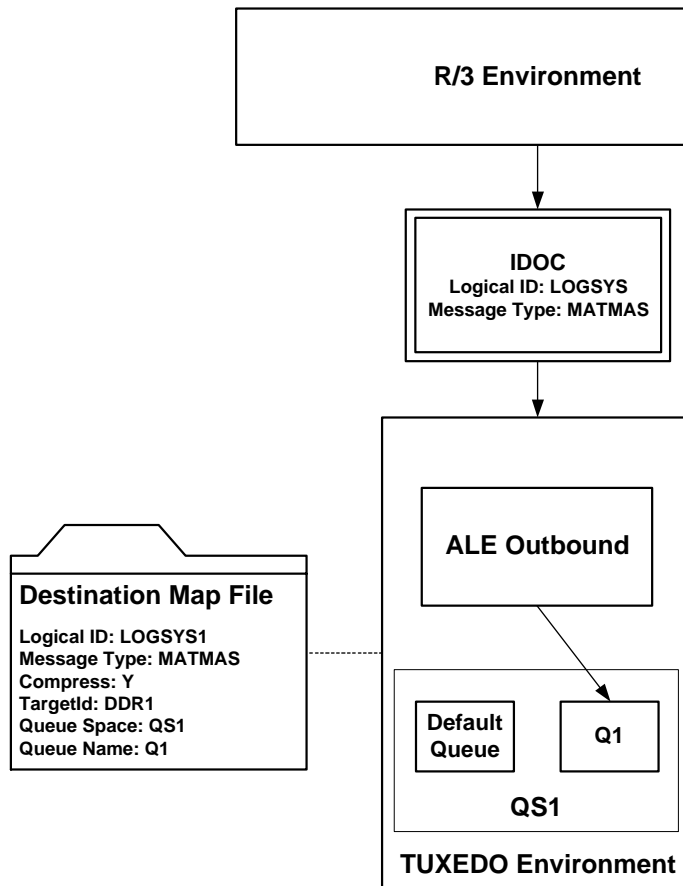
## Examples of Using a Destination Map File

The first example shows how ALE Outbound processes an IDOC packet that contains a single IDOC. Suppose the destination map file contains the following entry:

**Table 3-5 Settings in a Sample Destination Map File**

LogicalId	MsgType	Compress	TargetId	QueueSpace	QueueName
LOGSYS1	MATMAS	Y	DDR1	QS1	Q1

Suppose also that the control record in an outbound IDOC specifies a target logical ID of “LOGSYS” and a message type of “MATMAS”. Figure 3-12 shows how ALE Outbound would process this IDOC packet according to the settings in the destination map file:

**Figure 3-12 Enqueuing a Single IDOC According to the Destination Map File**

In this scenario, ALE Outbound finds the matching entry in the destination map file and sends the IDOC message to the Q1 queue in queue space QS1. ALE Outbound 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 ALE Outbound processes an IDOC packet that contains multiple IDOCs. Suppose the destination map file contains the following entries:

**Table 3-6 Settings in a Sample Destination Map File**

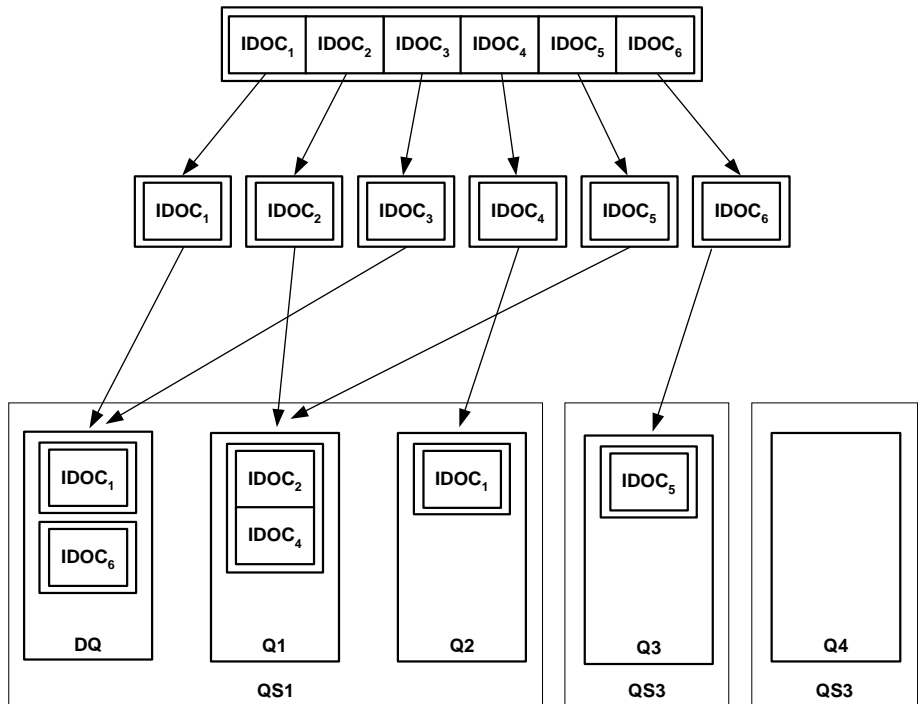
LogicalId	MsgType	Compress	TargetId	QueueSpace	QueueName
LOGSYS1	MATMAS	Y	DDR1	QS1	Q1
LOGSYS2	MATMAS	Y	DDR2	QS1	Q2
LOGSYS3	DEBMAS	N	DDR3	QS2	Q3
LOGSYS4	CREMAS	Y	DDR4	QS3	Q4

Suppose also that an outbound IDOC packet contains six IDOCs with the following settings in the control record of each IDOC:

**Table 3-7 Sample IDOC Packet Containing Multiple IDOCs**

IDOC	LogicalId	MsgType
IDOC <sub>1</sub>	LOGSYS1	CREMAS
IDOC <sub>2</sub>	LOGSYS1	MATMAS
IDOC <sub>3</sub>	LOGSYS3	MATMAS
IDOC <sub>4</sub>	LOGSYS2	MATMAS
IDOC <sub>5</sub>	LOGSYS1	MATMAS
IDOC <sub>6</sub>	LOGSYS3	DEBMAS

Figure 3-13 shows how ALE Outbound would process this IDOC packet according to the settings in the destination map file:

**Figure 3-13 Enqueuing Multiple IDOCs According to the Destination Map File**

ALE Outbound splits the IDOC packet into individual IDOCs and enqueues each IDOC in the following manner:

- ALE Outbound enqueues IDOC<sub>1</sub> and IDOC<sub>3</sub> into the default queue because no matching entry was found in the destination map file.
- ALE Outbound enqueues IDOC<sub>2</sub>, IDOC<sub>4</sub>, IDOC<sub>5</sub>, and IDOC<sub>6</sub> into their respective queues.
- ALE Outbound compresses IDOC<sub>2</sub> and IDOC<sub>5</sub> into a single IDOC packet in Q1 because Compress="Y" in the matching entry in the destination map file.

## Managing Data-Depending Routing (DDR)

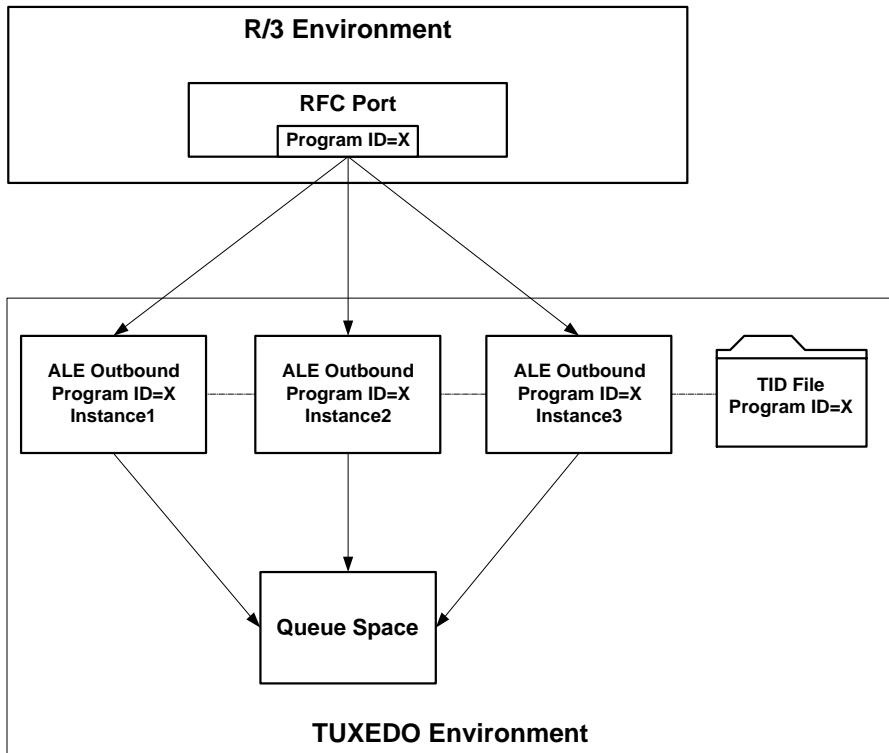
You can configure the default data-dependent routing (DDR) value that ALE Outbound assigns to each IDOC message (in the CR3\_TARGET\_ID field) that it enqueues. The behavior of ALE Outbound depends on whether a DDR value is defined in the destination map file:

- If no destination map file is used, or if no matching entry is found in the destination map file for an IDOC, then ALE Outbound assigns a default DDR value to the IDOC messages (CR3\_ALE\_DEFAULT\_TARGET\_ID environment variable).
- If a matching entry is found in the destination map file for an IDOC, then ALE Outbound assigns the DDR value (TargetId) from the map file.

See “Setting the Default Data-Dependent Routing Value” and “Setting Data-Dependent Routing Values in the Map File” in Chapter 5, “Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide* for instructions. See your TUXEDO documentation for more information about data-dependent routing.

## Load Balancing High Volumes of Outbound IDOCs

Multiple instances of ALE Outbound 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 ALE Outbound. Instances that register under the same program ID must also share the same TID file. Figure 3-14 shows multiple instances of ALE Outbound listening for IDOCs on the same program ID and sharing the same TID file:

**Figure 3-14 Multiple Instances of ALE Outbound Sharing the Same Program ID**

The number of ALE Outbound 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 ALE Outbound.

See “Configuring Load Balancing” in Chapter 5, Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*.

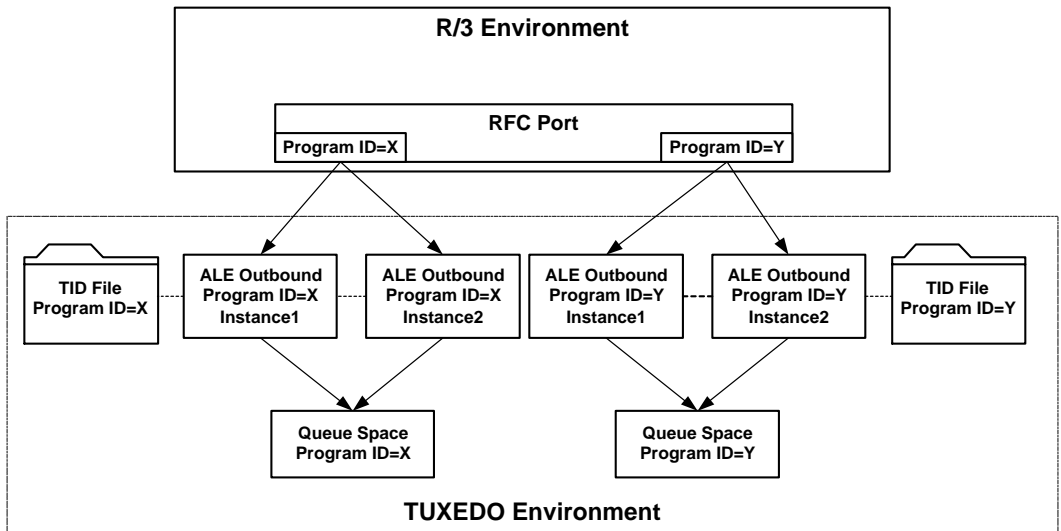
## Registering Multiple Program IDs

If R/3 is configured to send outbound IDOCs to different program IDs, you can configure ALE Outbound to handle these IDOCs by running multiple instances of ALE Outbound 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.

Figure 3-15 shows two groups of instances of ALE Outbound, each of which is listening for IDOCs on a shared program ID and sharing the same TID file:

**Figure 3-15 Multiple Instances of ALE Outbound Using Different Program IDs**



See “Configuring Multiple Program IDs” in Chapter 5, *Configuring ALE Integration*,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*.

## Managing Transactional Integrity for Outbound IDOCs

ALE Outbound manages transactional integrity for outbound IDOCs to ensure that an IDOC packet has been enqueued successfully. ALE Outbound uses a TID log file to track the transaction IDs (TIDs) associated with the IDOC packets it processes to ensure that it enqueues an IDOC packet from R/3 exactly once. See “Transaction IDs (TIDs)” on page 3-7 for an introduction to TIDs.

## About the TID Log File Used for Outbound IDOCs

ALE Outbound uses a TID file to track the IDOC packets it processes to ensure that it enqueues an IDOC packet once and only once. The R/3 system assigns a TID to each outbound IDOC packet.

The TID File for ALE Outbound is a log of all the TIDs that ALE Outbound 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	<p>The processing state. One of the following strings:</p> <ul style="list-style-type: none"> <li>■ <b>CREATED</b> indicates that ALE Outbound has received the TID from R/3.</li> <li>■ <b>EXECUTED</b> indicates that ALE Outbound has enqueued the IDOC message with the TID and has committed the transaction.</li> <li>■ <b>ROLLBACK</b> indicates that ALE Outbound has rolled back the IDOC packet from the queue.</li> <li>■ <b>CONFIRMED</b> indicates that ALE Outbound has confirmed that the IDOC message has been enqueued and the transaction has been committed.</li> </ul>

The following example shows a sample TID file for ALE Outbound:

```
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
```

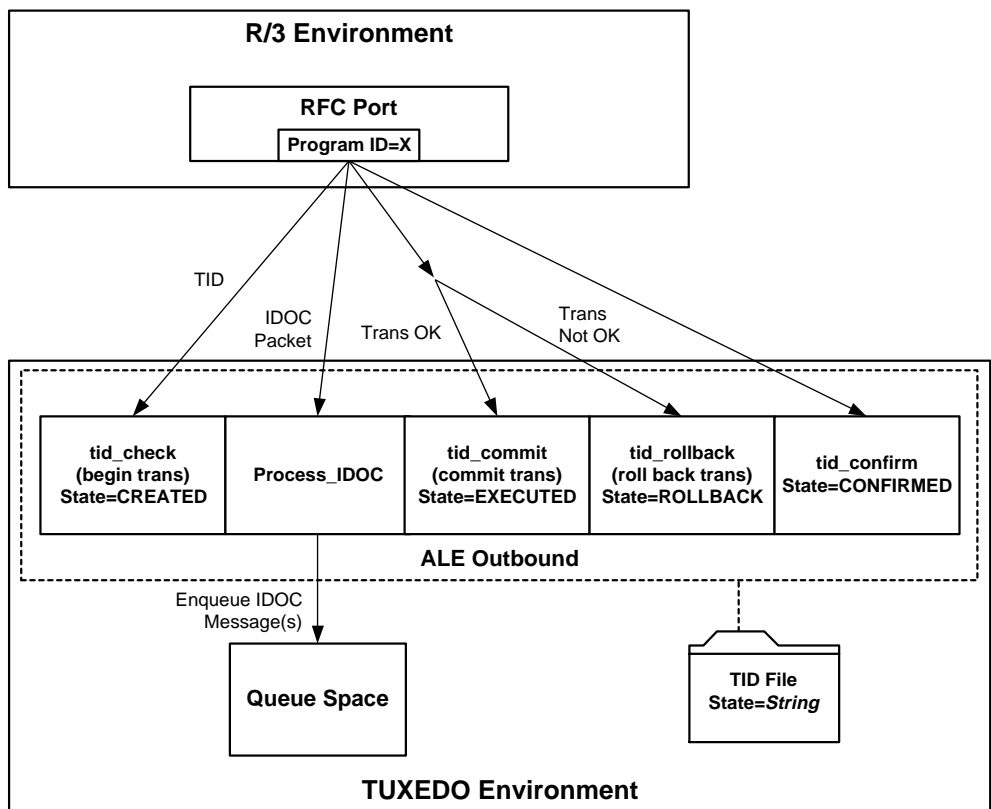
The CR3\_ALEOUT\_TID\_FILE environment variable specifies the location of the TID log file for ALE Outbound. See “Setting Environment Variables for ALE Outbound” in Chapter 5, “Configuring ALE Integration,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide* for more information.

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,” in the BEA eLink Adapter for R/3 *Installation and Configuration Guide*.

## Processing TIDs with Outbound IDOCs

Figure 3-16 shows how ALE Outbound uses the TID log file to manage transactional integrity for outbound IDOCs:

**Figure 3-16 TID Processing for ALE Outbound**



The information flow proceeds in the following sequence:

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

## Handling Problems with Outbound IDOCs

ALE Outbound uses TUXEDO's transaction management capabilities to ensure transactional integrity for outbound IDOCs. Figure 3-8 lists problems that can occur with outbound IDOCs:

**Table 3-8 Handling Problems with Outbound IDOCs**

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

# Glossary

**ABAP/4**

SAP's 4GL for 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).

**ALE Inbound**

The BEA eLink Adapter for R/3 component that processes and submits inbound IDOCs to R/3 for ALE processing.

**ALE Outbound**

The BEA eLink Adapter for R/3 component that receives and processes outbound IDOCs from R/3.

**BAPI**

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

**BDC**

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

**BDC Transaction Executor**

The BEA eLink Adapter for R/3 component that allows non-R/3 application to submit BDC data to R/3.

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

## **FML**

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

## **IDOC**

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

## **R/3**

SAP's client-server product that provides access to the SAP system via a three-tier architecture consisting of database, application, and presentation components.

## **RFC**

SAP's Remote Function Call interface that allows non-R/3 systems to invoke remotely callable ABAP/4 functions.

## **RFC Inbound**

The BEA eLink Adapter for R/3 component that allows non-R/3 applications to execute RFC-enabled APAB/4 functions.

## **RFC Outbound**

The BEA eLink Adapter for R/3 component that allows R/3 to access data and functionality in non-R/3 applications.

## **SAPGUI**

SAP's graphical user interface utility.

## **TID**

Transaction ID for tracking IDOC transactions.

---

# Index

## A

### ABAP/4

- function signature 2-7
- parameters 2-7

### about

- ABAP/4 parameters 2-7
- ALE Inbound 3-8
- ALE integration 3-1
- ALE Outbound 3-17
- BEA eLink Adapter for R/3 1-1
- destination map files 3-22
- FML field definitions 1-6
- intermediate documents (IDOCs) 3-5
- RFC Inbound 2-3
- RFC integration 2-2
- transaction IDs (TIDs) 3-7

### ALE

- about ALE integration 3-1
- information flow 3-4
- key concepts 3-5

### ALE Inbound

- about ALE Inbound 3-8
- CR3\_IDOC\_IN service 3-8
- CR3\_SUBMIT service 3-8
- FML32 field definitions 3-8
- handling problems 3-16
- information flow 3-9
- load balancing 3-12
- splitting IDOC packets 3-11
- TID file 3-12
- transactional integrity 3-12

### ALE Outbound

- about ALE Outbound 3-17
  - compressing outbound IDOCs 3-23
  - data-dependent routing
    - environment variable 3-28
    - in destination map file 3-23
  - destination map files 3-22
  - enqueueing into multiple queues 3-21
  - handling problems 3-34
  - information flow 3-18
  - load balancing 3-28
  - registering multiple program IDs 3-29
  - splitting IDOC packets 3-20
  - TID file 3-31
  - transactional integrity 3-30
- audience vii

## B

### BEA eLink Adapter for R/3

- about 1-1
- benefits 1-2

## C

- compressing outbound IDOCs 3-23
- CR3\_ERROR\_TEXT 1-8
- CR3\_IDOC 1-8
- CR3\_IDOC\_CONTROL 1-7
- CR3\_IDOC\_DATA 1-7
- CR3\_IDOC\_IN service 3-8
- CR3\_RFC\_IN service 2-3

---

CR3\_RFC\_TID 1-7  
CR3\_SUBMIT service 3-8  
CR3\_TARGET\_ID 1-8  
customer support xiv

## D

data-dependent routing  
    environment variable 3-28  
    in destination map file 3-23  
destination map files  
    about destination map files 3-22  
    examples 3-24  
documentation  
    audience vii  
    browsing in a web browser x  
    conventions x  
    how to use this document ix  
    organization viii  
    prerequisite knowledge viii  
    printing from a web browser x  
    related documentation xii  
    support xiv

## E

ELINK\_ADAPTER\_ERR 1-8  
ELINK\_ADAPTER\_ERR\_CODE 1-8  
ELINK\_APP\_ERR 1-8  
errors  
    adapter errors  
        about adapter errors 2-10  
        handling 2-12  
    error log 2-13  
    R/3 errors  
        about R/3 errors 2-10  
        handling 2-12  
export parameters  
    about export parameters 2-10  
    configuring 2-11  
export tables

about export tables 2-10  
configuring 2-11

## F

FML field definitions  
    about FML field definitions 1-6  
    ALE Inbound 3-8  
    CR3\_ERROR\_TEXT 1-8  
    CR3\_IDOC 1-8  
    CR3\_IDOC\_CONTROL 1-7  
    CR3\_IDOC\_DATA 1-7  
    CR3\_RFC\_TID 1-7  
    CR3\_TARGET\_ID 1-8  
    ELINK\_ADAPTER\_ERR 1-8  
    ELINK\_ADAPTER\_ERR\_CODE 1-8  
    ELINK\_APP\_ERR 1-8  
    RFC Inbound 2-8  
function signature 2-7

## I

import parameters, about 2-8  
import tables, about 2-8  
information flow  
    ALE 3-4  
    ALE Inbound 3-9  
    ALE Outbound 3-18  
    RFC Inbound  
        2-4  
intermediate documents (IDOCs), about 3-5

## K

key concepts  
    ALE 3-5

## L

load balancing  
    ALE Inbound 3-12  
    ALE Outbound 3-28

---

## N

notation x

## P

parameters

- export parameters 2-10

- import parameters 2-8

prerequisite knowledge viii

problems

- ALE Inbound 3-16

- ALE Outbound 3-34

- RFC Inbound 2-12

program IDs

- registering multiple 3-29

## R

request buffers, in RFC Inbound 2-8

response buffers, in RFC Inbound 2-9

RFC

- about RFC integration 2-2

RFC Inbound

- ABAP/4 parameters 2-7

- about RFC Inbound 2-3

- CR3\_RFC\_IN service 2-3

- error log 2-13

- export parameters

  - about export parameters 2-10

  - configuring 2-11

- export tables

  - about export tables 2-10

  - configuring 2-11

- FML32 field definitions 2-8

- handling adapter errors 2-12

- handling R/3 errors 2-12

- import parameters 2-8

- import tables 2-8

- information flow 2-4

- request buffers 2-8

- response buffers 2-9

## S

services

- CR3\_IDOC\_IN service 3-8

- CR3\_RFC\_IN service 2-3

- CR3\_SUBMIT service 3-8

splitting IDOC packets

- ALE Inbound 3-11

- ALE Outbound 3-20

support

- customer support xiv

- documentation xiv

- technical xiv

## T

tables

- export tables 2-10

- import tables 2-8

technical support xiv

TID file

- ALE Inbound 3-12

- ALE Outbound 3-31

transaction IDs (TIDs), about 3-7

transactional integrity

- ALE Inbound 3-12

- ALE Outbound 3-30

## U

usage scenarios

- ALE integration 3-2

## V

version reference ix