

Oracle® Application Server Integration

Adapter for SAP R/3 User's Guide

10g (9.0.4)

Part No. B10299-01

September 2003

Oracle Application Server Integration Adapter for SAP R/3 User's Guide, 10g (9.0.4)

Part No. B10299-01

Copyright © 2003 Oracle Corporation. All rights reserved.

Primary Author: Krista Gervais

Contributing Authors: Seetha Rao, Clement Pellerin, Philip Bridger

Contributors: Arvind Jain, Bo Stern, Maneesh Joshi, Seshu Adunuthula

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Store, Oracle8i, Oracle9i, SQL*Plus, and PL/SQL are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
Preface.....	ix
Intended Audience	x
Organization.....	x
Related Documentation	xi
Conventions.....	xi
Documentation Accessibility	xvi
1 Introduction to Oracle Application Server Integration Adapter for SAP R/3	
Architecture: Oracle Application Server Integration Adapter for SAP R/3	1-2
Outbound Interactions: Oracle Application Server ProcessConnect Acting as a Client....	1-3
Inbound Interactions: Oracle Application Server ProcessConnect Acting as a Server	1-3
Required Software.....	1-4
Supported Platforms	1-4
Postinstallation.....	1-5
Installing Oracle AS ProcessConnect Utility Modules in Your R/3 System	1-5
Performance Enhancement for Defining Interactions with RFC Function Modules.....	1-6
Performance Enhancement for Defining Interactions with ALE IDOC	1-6
Checking If the Custom Function Modules Are Installed.....	1-6
Importing the Oracle Application Server ProcessConnect Utility Function Modules.....	1-6
Copying the Files to the Transport Directory	1-7
Running the Transport Request Commands on the R/3 System.....	1-7

2	Defining a Delivery Channel	
	Adding and Configuring a Delivery Channel	2-2
	Parameters Specific for RFC or BAPI.....	2-5
	Connection Pooling	2-8
	Parameters Specific for ALE IDOCs	2-9
	Troubleshooting R/3 Settings	2-12
3	Defining an Interaction	
	Adding an Interaction	3-2
4	Using Oracle Application Server Integration Adapter for SAP R/3 Datatypes	
	Basic Datatypes	4-2
	Numeric Strings (NUMC Type)	4-3
	Fixed-Point Numeric Strings (PACKED Type).....	4-4
	Date and Time.....	4-6
	Binary Data.....	4-6
	Complex Types	4-7
	Getlist Function Module	4-7
5	Using Application Link Enabling	
	Manually Downloading an IDOC.....	5-2
	Frequently Used ALE Transactions	5-5
	ALE Terminology	5-6
	Logical System.....	5-6
	Intermediate Documents (IDOC) Type	5-7
	Message Type	5-8
	Custom IDOC Behavior	5-8
	AdvancedSend Versus Send	5-8
6	Retrying Inbound Calls	
	Using a Transactional RFC	6-2
	Managing Incorrect Calls.....	6-2
	Scheduling an Automatic Retry	6-2

Deleting a Background Job..... 6-3

Index

Send Us Your Comments

Oracle Application Server Integration Adapter for SAP R/3 User's Guide, 10g (9.0.4)

Part No. B10299-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: appserverdocs_us@oracle.com
- FAX: 650-506-7407 Attn: Oracle Application Server Documentation Manager

- Postal service:

Oracle Corporation
Oracle Application Server Documentation
500 Oracle Parkway, M/S 2op3
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

This guide explains how to use Oracle Application Server ProcessConnect and the Oracle Application Server Integration Adapter for SAP R/3 to access R/3 modules. In this guide you will learn how to define a delivery channel for R/3 and add an interaction to generate native events, which are XML instances defined by XSD (XML payload defined by an XML Schema Definition instance). In this guide you will also find a chapter describing the datatype mapping between R/3 and XSD.

See Also: *Oracle Application Server ProcessConnect User's Guide* for more information on using native events with Oracle Application Server ProcessConnect business processes

This preface contains these topics:

- [Intended Audience](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)
- [Documentation Accessibility](#)

Intended Audience

Oracle Application Server Integration Adapter for SAP R/3 User's Guide is intended for anyone who performs the following tasks:

- Creates delivery channels and interactions with an R/3 system
- Maintains applications

To use this document, you need some knowledge of an R/3 system or have access to an R/3 system administrator. You must have development rights on the areas of the R/3 system you want to access.

Organization

This document contains:

Chapter 1, "Introduction to Oracle Application Server Integration Adapter for SAP R/3"

This chapter describes the Oracle Application Server Integration Adapter for SAP R/3 and the hardware and software requirements. It also provides instructions on adding custom function modules into an R/3 system. The custom modules enhance browsing performance during the selection of interactions from an R/3 system.

Chapter 2, "Defining a Delivery Channel"

This chapter provides instructions for using Oracle Application Server ProcessConnect to define a delivery channel for an R/3 system.

Chapter 3, "Defining an Interaction"

This chapter provides instructions for using Oracle Application Server ProcessConnect to add an R/3 Remote Function Call (RFC), Business Application Programming Interface (BAPI), and Application Link Enabling (ALE) interaction.

Chapter 4, "Using Oracle Application Server Integration Adapter for SAP R/3 Datatypes"

This chapter provides information on R/3 datatypes.

Chapter 5, "Using Application Link Enabling"

This chapter describes how to download an ALE IDOC to your local machine and provides basic ALE knowledge extracted from SAP documentation.

Chapter 6, "Retrying Inbound Calls"

This chapter describes how to set up an R/3 system to retry inbound calls.

Related Documentation

For more information, see these Oracle resources:

- *Oracle Application Server ProcessConnect User's Guide* in the Oracle Application Server Documentation Library
- *Oracle Application Server Installation Guide*

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com>

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/membership>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/docs>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Windows Operating Systems](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE Monospace (fixed-width) font	Uppercase Monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase Monospace (fixed-width) font	Lowercase Monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
<i>lowercase italic Monospace (fixed-width) font</i>	Lowercase italic Monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>Uold_release.SQL</i> where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a Monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
.	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fs1/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . /fs1/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>

Convention	Meaning	Example
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	<pre>SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;</pre>
lowercase	<p>Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files.</p> <p>Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</p>	<pre>SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;</pre>

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - <i>HOME_NAME</i> > Configuration and Migration Tools > Database Configuration Assistant.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.	<pre>c:\winnt\"system32 is the same as C:\WINNT\SYSTEM32</pre>

Convention	Meaning	Example
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.	C:\oracle\oradata>
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp scott/tiger TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\" C:\>imp SYSTEM/password FROMUSER=scott TABLES=(emp, dept)
<i>HOME_NAME</i>	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	C:\> net start Oracle <i>HOME_NAME</i> INSListener

Convention	Meaning	Example
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory. For Windows NT, the default location was <code>C:\orant</code>.</p> <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is <code>C:\oracle</code>. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is <code>C:\oracle\orann</code>, where <i>nn</i> is the latest release number. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle9i Database Getting Started for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Go to the <i>ORACLE_BASE\ORACLE_HOME\rdms\admin</i> directory.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle Corporation is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Introduction to Oracle Application Server Integration Adapter for SAP R/3

Oracle Application Server ProcessConnect connects to an R/3 system through the Oracle Application Server Integration Adapter for SAP R/3. The Oracle Application Server Integration Adapter for SAP R/3 provides connectivity and executes interactions on an R/3 system. This chapter discusses the following topics:

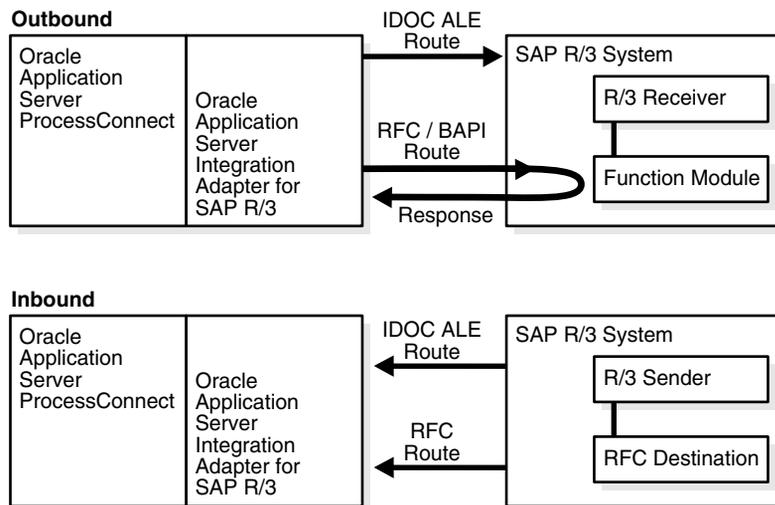
- [Architecture: Oracle Application Server Integration Adapter for SAP R/3](#)
- [Required Software](#)
- [Supported Platforms](#)
- [Postinstallation](#)
- [Installing Oracle AS ProcessConnect Utility Modules in Your R/3 System](#)

Architecture: Oracle Application Server Integration Adapter for SAP R/3

The Oracle Application Server Integration Adapter for SAP R/3 is a JCA-based component that plugs in to Oracle Application Server ProcessConnect. Using Oracle Application Server ProcessConnect, you can access R/3 remote function call modules (RFCs), BAPI modules, and Application Link Enabling (ALE) Intermediate DOCuments (IDOCs).

As shown in [Figure 1–1](#), the Oracle Application Server Integration Adapter for SAP R/3 runs on the machine running Oracle Application Server ProcessConnect and uses native R/3 protocols to download metadata from the R/3 system. The Oracle Application Server Integration Adapter for SAP R/3 uses R/3 libraries to send calls to, or receive inbound calls from, the R/3 system.

Figure 1–1 Architecture



The Oracle Application Server Integration Adapter for SAP R/3 supports both outbound and inbound interactions. Outbound interactions are supported for the adapter exchange protocols, RFC, BAPI, and ALE. Inbound interactions are supported for RFC and ALE.

ALE is an event-based protocol and the ALE interactions do not have reply records. For RFC, inbound interactions have request records and no reply records, because Oracle Application Server ProcessConnect supports only asynchronous

communication for inbound interactions. Each record has one record element associated with it. The following naming convention is used for the records.

- Request records

`AEProtocolName_interactionGroupName_interactionName_Request`

- Reply records

`AEProtocolName_interactionGroupName_interactionName_Reply`

See Also: [Chapter 3, "Defining an Interaction"](#)

Outbound Interactions: Oracle Application Server ProcessConnect Acting as a Client

If you want to make your Oracle Application Server ProcessConnect application to act as a client sending IDOCs or invoking RFC or BAPI function modules, then you must create an outbound interaction for the R/3 system. When this interaction is triggered, an IDOC is sent to the R/3 system (in cases of ALE) or the interaction for the function module is invoked (in the case of RFC and BAPI).

For sending an IDOC to an R/3 system, you must set up the ALE-related settings in the delivery channel for the Oracle Application Server Integration Adapter for SAP R/3. When you browse the ALE in the Oracle Application Server ProcessConnect user interface, methods associated with each IDOC are displayed. These methods are called `Send` and `AdvancedSend`. Interactions can be built around either of these methods.

Inbound Interactions: Oracle Application Server ProcessConnect Acting as a Server

If you want your Oracle Application Server ProcessConnect application to act as a server receiving R/3 IDOCs or RFC calls, then you must create an inbound interaction for the R/3 system. To receive an inbound call, you must first assign the RFC program ID in the delivery channel of the Oracle Application Server Integration Adapter for SAP R/3. When the Oracle Application Server Integration Adapter for SAP R/3 starts, it registers itself as the destination for this program ID within the R/3 system. After registration, the R/3 system sends calls with the program ID to the Oracle Application Server Integration Adapter for SAP R/3. The type of RFC destination in R/3 is TCP/IP.

For ALEs, you must use the `AdvancedSend` method associated with the ALE IDOC to define your event/procedure. When an IDOC is sent or an RFC call is invoked from the R/3 system, the adapter raises an inbound interaction and the record is sent to Oracle Application Server ProcessConnect.

Required Software

The Oracle Application Server Integration Adapter for SAP R/3 supports an R/3 system, version 4.6B or 4.6C, using the R/3 RFC library, `librfccm.so` (Solaris) or `librfccm.sl` (HP).

To work with the Oracle Application Server Integration Adapter for SAP R/3 and Oracle Application Server ProcessConnect:

1. The R/3 system you are going to use must be running.

You must have developer rights to the sections of the R/3 system that you will access.

2. You can have the R/3 SAPGUI installed.

Although not required for standard use, the SAPGUI is recommended for tracking ALE messages. It is not required for standard use of RFC function modules. The SAPGUI is available in the SAP RFCSDK.

The R/3 system is not modified except for an optional module to improve the performance of metadata retrieval.

Oracle Application Server ProcessConnect can be used with existing R/3 systems without any changes. However, to speed up retrieval of function modules and IDOC definitions, you can use the Oracle Application Server ProcessConnect utility function modules. If you cannot install the R/3 system utility function modules for ALE, you can manually download ALE IDOCs from an R/3 system to the Oracle Application Server ProcessConnect computer to browse the ALE IDOCs locally.

See Also:

- ["Installing Oracle AS ProcessConnect Utility Modules in Your R/3 System"](#) on page 1-5
- ["Manually Downloading an IDOC"](#) on page 5-2

Supported Platforms

The Oracle Application Server Integration Adapter for SAP R/3 is supported on:

- Solaris 8 (2.8)
- HP-UX 11.0

Postinstallation

1. Verify you are connecting to an R/3 system, version 4.6B or 4.6C.
2. Copy the R/3 RFC library, `librfccm.so` (Solaris) or `librfccm.sl` (HP) and place it in `ORACLE_install/ip/adapters/lib`.
3. The R/3 system you are going to use must be running.
You must have developer rights to the sections of the R/3 system that you will access.
4. You can have the R/3 SAPGUI installed.
Although not required for standard use, the SAPGUI is recommended for tracking ALE messages. It is not required for standard use of RFC function modules. The SAPGUI is available in the SAP RFCSDK.
5. If using ALE IDOCs, you have either installed the utility modules or have manually downloaded IDOCs.

See Also:

- ["Installing Oracle AS ProcessConnect Utility Modules in Your R/3 System"](#) on page 1-5
- ["Manually Downloading an IDOC"](#) on page 5-2

Installing Oracle AS ProcessConnect Utility Modules in Your R/3 System

Oracle Application Server ProcessConnect includes files for installing the utility function modules on your R/3 system (these are called *transport requests* in R/3). These function modules only to be imported one time, and are only required on an R/3 development machine. (They are not required on an R/3 production machine.) A single import is used for both ALE and RFC enhancements. The following provides an overview on why you would use the function modules, and the steps to import the files into an R/3 system.

- [Performance Enhancement for Defining Interactions with RFC Function Modules](#)
- [Performance Enhancement for Defining Interactions with ALE IDOC](#)
- [Checking If the Custom Function Modules Are Installed](#)
- [Importing the Oracle Application Server ProcessConnect Utility Function Modules](#)

- [Copying the Files to the Transport Directory](#)
- [Running the Transport Request Commands on the R/3 System](#)

Performance Enhancement for Defining Interactions with RFC Function Modules

In an unoptimized R/3 environment, RFC table retrieval is slow because Oracle Application Server ProcessConnect downloads information for area, groups, and functions from one large table. To reduce the time spent downloading information, Oracle Application Server ProcessConnect provides function modules that allow selective retrieval of areas, groups, and functions from three separate tables.

Performance Enhancement for Defining Interactions with ALE IDOC

Installing the Oracle Application Server ProcessConnect function modules is optional for ALE IDOCs. There are two ways to use Oracle Application Server ProcessConnect and R/3 to explore IDOCs:

- You can use the Oracle Application Server ProcessConnect function modules to automatically search the R/3 system for existing IDOCs. You must have administrative rights to modify an R/3 system.
- You can manually download the ALE IDOC definitions (*.mtd files) to the Oracle Application Server ProcessConnect computer.

See Also: ["Manually Downloading an IDOC"](#) on page 5-2 to download IDOCs locally for browsing

Checking If the Custom Function Modules Are Installed

To check if the Oracle Application Server ProcessConnect utility modules have already been installed in your R/3 system, perform a simple check. In the R/3 system SAPGUI:

1. Navigate to transaction /nse37.
2. Search for the function module /ACTIONAL/*.

If it is not there, install the utility function modules.

Importing the Oracle Application Server ProcessConnect Utility Function Modules

The following outlines how to import specific Oracle Application Server ProcessConnect R/3 objects (the /ACTIONAL/namespace and

/ACTIONAL/DEV/devclass) in your R/3 system. The import file is compatible with 4.6B and 4.6C versions.

Copying the Files to the Transport Directory

To load the ACTIONAL namespace and devclass objects, copy the files contained within: ORACLE_HOME/ip/adapters/config/SAP into two subdirectories under the R/3 trans directory. The subdirectories are: cofiles and data.

1. Copy all the files starting with K*.* into the cofiles subdirectory on your R/3 system, for example, \usr\sap\trans\cofiles.
2. Copy the all the files starting with R*.* into the subdirectory data on your R/3 system, for example, \usr\sap\trans\data.

Running the Transport Request Commands on the R/3 System

1. Change directories to the transport\bin subdirectory.
2. Type `tp checkimpdp SID`.

Where *SID* is the system identification. The default is 00.

The response should be `RDDIMPDP is scheduled`. Contact your R/3 system administrator about transport system problems if you get another response.

3. Type `tp cleanbuffer SID`.
This empties the transport buffer for your system.
4. Type `tp showbuffer SID`.
This shows that the buffer is clean.
5. Type `tp addtobuffer SS3K900063 SID`.
This brings the first transport (number ranges) into the buffer.
6. Type `tp addtobuffer SS3K900041 SID`.
This brings the second transport (number ranges) into the buffer.
7. Type `tp showbuffer SID`.
This shows how the buffer has been loaded.
8. Type `tp import SS3K900063 SID u1`.
This command imports the first transport into your system.
9. Type `tp import SS3K900041 SID u1`.

This command imports the second transport into your system.

Defining a Delivery Channel

This chapter describes how to use Oracle Application Server ProcessConnect to define a delivery channel to connect to an R/3 system.

In this chapter, the delivery channel parameters for RFC and BAPI are described separately from the ALE parameters. When defining a delivery channel for RFC and BAPI, the ALE parameters are not used and can be left blank. Similarly, when defining a delivery channel for ALE, the RFC and BAPI parameters are not used and can be left blank. You can also choose to define a single delivery channel for all exchange protocols (RFC, BAPI, and ALE) by combining both sets of parameters.

This chapter discusses the following topics:

- [Adding and Configuring a Delivery Channel](#)
- [Parameters Specific for RFC or BAPI](#)
- [Parameters Specific for ALE IDOCs](#)
- [Troubleshooting R/3 Settings](#)

Adding and Configuring a Delivery Channel

Part of the application definition includes adding a delivery channel for the adapter. Setting up the delivery channel in Oracle Application Server ProcessConnect requires information which is specific to the adapter.

See Also: *Oracle Application Server ProcessConnect User's Guide* for details about adding an application delivery channel in Oracle Application Server ProcessConnect

1. Select the **Profiles > Applications** tabs.
2. Click **Create**.

Oracle Application Server
ProcessConnect

Home Help Logout

Modeling Profiles Deployment Reports Administration

Host Trading Partners Applications Agreements

Logged in as ip

Create Application

Please enter the application parameter(s) and choose Apply.
* Indicates required field

* Name

Description

Application Type

Cancel Apply

Modeling | Profiles | Deployment | Reports | Administration | Home | Help | Logout

Copyright © 2002, 2003, Oracle Corporation. All rights reserved.

3. Type an application name in the **Name** field.
4. Select the application type from the **Application Type** box and click **Apply**.
5. Click **Add** in the **Adapter Types** section to add the Oracle Application Server Integration Adapter for SAP R/3.

The Add Adapter Type page appears.



6. Select **SAP R/3 Adapter** in the **Type** selection box in the Add Adapter Type page and click **Apply**.

The Adapter Type Details: SAP R/3 Adapter page is displayed.



7. Click **Create** in the **Delivery Channels** section in the Adapter Type Details page.

The Create Delivery Channel page is displayed.

8. Add information to create a delivery channel.

For specific information on RFC and BAPI parameters, use [Table 2-1](#).

For specific information on ALE parameters, use [Table 2-2](#).

Oracle Application Server
ProcessConnect

Home Help Logout

Modeling Profiles Deployment Reports Administration

Host Trading Partners Applications Agreements

Logged in as ip

Create Delivery Channel

Application mySAPTest Cancel Apply

Please enter the delivery channel parameter(s) and choose Apply.
* Indicates required field

* Name	
Application Server	
System Number	100
Message Server	
System ID	
Message Server Group	
Router	
Client Number	
ALE IDOC Version	
ALE Partner Logical System	
OracleAS ProcessConnect Logical System	
Maximum Number of Concurrent Connections	15
Additional Outbound Connection Parameters	
RFC Program ID	
Additional Inbound Connection Parameters	
Maximum Number of Sessions	100
User ID	
Password	
Language	EN

Cancel Apply

Modeling | Profiles | Deployment | Reports | Administration | Home | Help | Logout

Note: For specific R/3 assistance, refer to your R/3 system administrator. You can also visit help.sap.com.

See Also: "[Troubleshooting R/3 Settings](#)" on page 2-12 for connection error information

9. Click **Apply** after entering your parameters to connect to an R/3 system.

Logged in as ip

 **Confirmation**
Delivery Channel mySAPDeliveryChannel successfully created.

Delivery Channel Details : mySAPDeliveryChannel Delete Update

Details

Application Server		ss2
System Number		00
Message Server		
System ID		
Message Server Group		
Router		
Client Number		810
ALE IDOC Version		
ALE Partner Logical System		
OracleAS ProcessConnect Logical System		
Maximum Number of Concurrent Connections		15
Additional outbound connection parameters		
RFC Program ID		
Additional inbound connection parameters		
Maximum Number of Sessions		100
User ID		dan
Password		<hidden>
Language		EN

[Return To List](#) Delete Update

[Modeling](#) | [Profiles](#) | [Deployment](#) | [Reports](#) | [Administration](#) | [Home](#) | [Help](#) | [Logout](#)

Copyright © 2002, 2003, Oracle Corporation. All rights reserved.

The delivery channel Confirmation screen appears. You can modify any parameters and click **Update** in the confirmation page to change your parameters or click **Delete** to remove the channel.

10. Click the **Return to List** link to return to the Adapter Details page.

The delivery channel is added for the adapter and you can now add interactions.

See Also: [Chapter 3, "Defining an Interaction"](#)

Note: The Oracle Application Server ProcessConnect creates the Cache and Queue directories at the time you create the Delivery Channel using ALE parameters.

Parameters Specific for RFC or BAPI

The delivery channel screen contains parameters specific to RFC and BAPI and other parameters specific to ALE. When defining a delivery channel specifically for

RFC and BAPI function modules, the ALE parameters are not used; therefore, their values can be left blank. You can define a delivery channel to handle both modules and IDOCs by combining both sets of parameters. You can also create two separate delivery channels; one for RFC and BAPI function modules, and one for ALE IDOCs.

In [Table 2–1](#) an asterisk (*) indicates a required field.

In [Table 2–1](#) a double asterisk (**) indicates that you can select either an application server or a message server. Depending on how your R/3 system administrator set up your R/3 system, you can have either an application server or a message server with any number of application servers. Verify with your R/3 system administrator which server you can use.

Table 2–1 RFC or BAPI Specific Parameters

Field	Description
Name*	Enter a name for the delivery channel.
Application Server **	Enter a string representing the machine on which the R/3 system is running and listening. The name defines a connection to the Application Server representing a single R/3 system and identifies the system, for example, ss1.
Application Server / System Number**	Enter a system number to identify the Host, for example, 00.
Message Server**	Enter the R/3 system host name, for example, hs0016.WDF.SAP-AG-DE. You get the value from an R/3 system administrator. This message server provides a list of the currently available application servers that are running on the selected system. Each R/3 system provides one Message Server. This information is stored in the R/3 SAPMSG.INI file.
Message Server / System ID**	Enter the System ID that identifies the R/3 system. The list of R/3 systems is retrieved from the R/3 SAPMSG.INI file. For example, D15.
Message Server / Message Server Group**	Enter the Message Server Group if your application servers belong to a logon group. The group is retrieved from a list of defined Groups in the R/3 system. The Message Server Group is used to log on to any application server defined in the group. The group is based on load balancing. For example, a group can be PUBLIC or JAPANESE. If you do not know which group you want to log on, you can use an empty string for this argument. The R/3 system selects one automatically to make the connection. The message server option is only valid for inbound calls.

Table 2–1 (Cont.) RFC or BAPI Specific Parameters

Field	Description
Router	Optional. Enter a destination router to connect to the Application Server or Message Server, for example, /H/UNICENTER/H/204.79.199.5/H. Destination router is used to connect to the message server as well as to the listed application servers. The list of available SAP™ routers is retrieved from the R/3 SAPROUTE.INI file.
Client Number*	Enter a number for the R/3 system. For example, 810.
Maximum Number of Concurrent Connections	<p>Enter a number of concurrent connections to an R/3 system. This can range from 1 to 999. The default value is 15 connections.</p> <p>Refer to "Connection Pooling" on page 2-8 for more information.</p> <p>Oracle Application Server ProcessConnect enables you to pool connections within particular logins but maintains critical control of the total number of connections into a system. Each connection to an R/3 system requires significant resources in R/3 to maintain the connection state. You must determine the optimum number for the maximum number of concurrent connections so that your R/3 system is not overloaded (while allowing as many simultaneous requests that can be served) to minimize the amount of time spent waiting for requests to be processed.</p> <p>The following equation is a rough <i>rule-of-thumb</i> to determine the value to use to ensure that the processes do not overload a system:</p> <ul style="list-style-type: none"> ■ (Maximum Connection Parameter) is less-than-or-equal-to the (number of R/3 Worker Processes) times(*) 1.2. <p>If you have more than one Delivery Channel definition for a particular R/3 host, you must use the total of the Max Connection Parameter values across all of the Delivery Channels in the equation.</p>
Additional Outbound Connection Parameters	<p>Enter any additional connection parameters. This setting passes additional connection parameters when Oracle Application Server ProcessConnect connects to an R/3 system. The format is a space-separated list of name=value pairs:ID1=value1 ... IDn=valuen. The IDs are not case-sensitive but the values are case-sensitive and must be set in "..." if blanks are included.</p> <ul style="list-style-type: none"> ■ Refer to the R/3 documentation for the function RfcOpenEx. For a list of IDs, consult the file <code>saprfc.h</code>, part of the RFC SDK.
RFC Program ID	Enter an RFC Program ID to be used by the adapter to register itself with the RFC Gateway in the R/3 system. This is mandatory for inbound calls (R/3 system is initiating the call).

Table 2–1 (Cont.) RFC or BAPI Specific Parameters

Field	Description
Additional Inbound Connection Parameters	<p>Enter any additional connection parameters. This setting passes additional connection parameters when Oracle Application Server ProcessConnect is listening for calls from an R/3 system. The parameters are case-sensitive. The format is a space-separated list of name=value pairs:ID1=value1 ... IDn=valuen. The IDs are not case-sensitive but the values are case-sensitive and must be set in "... " if blanks are included.</p> <ul style="list-style-type: none">Refer to the R/3 documentation for the <code>RfcAccept</code> function. For a list of IDs, consult the file <code>saprfc.h</code>, part of the RFC SDK.
Maximum Number of Sessions	Enter the maximum number of sessions. This is the number of simultaneous calls an R/3 system can make to the adapter.
User ID*	Enter an R/3 user name to login to the R/3 system.
Password*	Enter the password of the specified user. You must have developer rights to the areas of the R/3 system you want to access.
Language	A parameter that specifies the language to be used by the R/3 system. If not specified, the default language on the R/3 system is used.

Connection Pooling

The Oracle Application Server Integration Adapter for SAP R/3 creates pools of connections to an R/3 system by sharing login sessions and keeping the total number of sessions below a maximum number determined according to the capacity of the R/3 system. By sharing connections, callers avoid the delay of waiting for a login to complete, since they can use an existing session that had been opened for a previous request. Keeping the number of connections to a minimum avoids the severe performance degradation that can happen if too many sessions are opened on the R/3 system, and session swapping occurs. In such conditions, requests take a long time to execute.

Sessions can be shared across instances of a business process or across multiple business processes when they use the same login credentials into R/3. If several instances of a business process need to execute requests at the same time, additional sessions are opened for the pool, up to the maximum number of concurrent connections (**Max Concurrent Connections**). After a period of inactivity, idle sessions are released.

R/3 reserves approximately 1MB of memory for each R/3 connection. As a result, connection pooling and concentration (allocating calls and requests) across a

number of existing, open connections can substantially increase the level of resources available, thus increasing scalability. It is effective because not every call has pending requests, so the connections are reused for multiple calls.

For optimum performance, benchmarking should be performed by gradually increasing the number of connections.

See Also: ["Maximum Number of Concurrent Connections"](#) on page 2-7

Parameters Specific for ALE IDOCs

The delivery channel screen contains parameters specific to ALE and other parameters specific to RFC and BAPI. When defining a delivery channel specifically for ALE, the RFC and BAPI parameters are not used; therefore, their values can be left blank. You can define a delivery channel to handle both IDOCs and modules by combining both sets of parameters. You can also create two separate delivery channels; one for ALE IDOCs, and one for RFC and BAPI function modules.

Note: The Oracle Application Server ProcessConnect creates the Cache and Queue directories at the time you create the delivery channel using ALE parameters.

In [Table 2–2](#) an asterisk (*) indicates a mandatory field.

In [Table 2–2](#) a double asterisk (**) indicates you can select either an Application Server or a Message Server. Depending on how your R/3 system administrator set up your R/3 system, you can have either an application server or a message server with any number of application servers. Verify with your R/3 system administrator which server you can use.

In [Table 2–2](#) a triple asterisk (***) indicates you can select an ALE Partner Logical System and OracleAS ProcessConnect Logical System for outbound. (Select only RFC Program ID for inbound.)

Table 2–2 *ALE Specific Parameters*

Field	Description
Name*	Enter a name for the delivery channel.

Table 2–2 (Cont.) ALE Specific Parameters

Field	Description
Application Server **	Enter a string representing the machine on which the R/3 system is running and listening. The name defines a connection to the Application Server representing a single R/3 system and identifies the system, for example, <code>ss1</code> .
Application Server / System Number**	Enter a system number to identify the Host, for example, <code>00</code> .
Message Server**	Enter the R/3 system host name, for example, <code>hs0016.WDF.SAP-AG-DE</code> . You get the value from an R/3 system administrator. This message server provides a list of the currently available application servers that are running on the selected system. Each R/3 system provides one Message Server. This information is stored in the R/3 <code>SAPMSG.INI</code> file.
Message Server / System ID**	Enter the System ID that identifies the R/3 system. The list of R/3 systems is retrieved from the R/3 <code>SAPMSG.INI</code> file. For example, <code>D15</code> .
Message Server / Message Server Group**	Enter the Message Server Group if your application servers belong to a logon group. The group is retrieved from a list of defined Groups in the R/3 system. The Message Server Group is used to log on to any application server defined in the group. The group is based on load balancing. For example, a group can be <code>PUBLIC</code> or <code>JAPANESE</code> . If you do not know which group you want to log on, you can use an empty string for this argument. The R/3 system selects one automatically to make the connection. The message server option is only valid for inbound calls.
Router	Optional. Enter a destination router to connect to the Application Server or Message Server, for example, <code>/H/UNICENTER/H/204.79.199.5/H</code> . Destination router is used to connect to the message server as well as to the listed application servers. The list of available SAP™ routers is retrieved from the R/3 <code>SAPROUTE.INI</code> file.
Client Number*	Enter a number for the R/3 system. For example, <code>810</code> .
ALE IDOC Version*	Enter three upper-case letters to specify the version of R/3 that the IDOC definitions follow. For example, <code>46B</code> or <code>46C</code> . This is mandatory if you are using ALE.

Table 2–2 (Cont.) ALE Specific Parameters

Field	Description
ALE Partner Logical System***	<p>Enter the name of an ALE Logical System in R/3 that receives IDOCs. This is mandatory for outbound calls (Oracle Application Server ProcessConnect is initiating the call).</p> <p>For example, if the R/3 system is to receive an IDOC, enter the Logical System Name of the recipient registered in R/3. The name contains a maximum of 10 characters.</p> <ul style="list-style-type: none"> ▪ Note: The Logical System is called the System Base Logical System ID in R/3 documentation. The Logical System is named and created in an R/3 system by an R/3 system administrator and must be configured to receive IDOCs.
OracleAS ProcessConnect Logical System***	<p>Enter the name of a Logical System that sends the IDOC. This is mandatory for outbound calls (Oracle Application Server ProcessConnect is initiating the call).</p> <p>For example, if Oracle Application Server ProcessConnect sends an IDOC to R/3, enter the Logical System Name for Oracle Application Server ProcessConnect. The name contains a maximum of 10 characters.</p> <ul style="list-style-type: none"> ▪ Note: The name for the Logical System must be registered as an external source in the R/3 system by the R/3 system administrator.
Additional Outbound Connection Parameters	<p>Enter any additional connection parameters. This setting passes additional connection parameters when Oracle Application Server ProcessConnect connects to an R/3 system. The parameters are case-sensitive.</p> <ul style="list-style-type: none"> ▪ For more information on connection parameters, refer to the R/3 documentation on the <code>RfcOpenEx</code> function.
RFC Program ID***	<p>Enter an RFC Program ID to be used by the adapter to register itself with the RFC Gateway in the R/3 system. This is mandatory for inbound calls (R/3 system is initiating the call).</p>
Additional Inbound Connection Parameters	<p>Enter any additional connection parameters. This setting passes additional connection parameters when Oracle Application Server ProcessConnect is listening for calls from an R/3 system. The parameters are case-sensitive.</p> <ul style="list-style-type: none"> ▪ For more information on connection parameters, refer to R/3 documentation for <code>RfcAccept</code> function.
User ID*	<p>Enter an R/3 user name to login to the R/3 system.</p>
Password*	<p>Enter the password of the specified user. You must have developer rights to the areas of the R/3 system you want to access.</p>
Language	<p>A parameter that specifies the language to be used by the R/3 system. If not specified, the default language on the R/3 system is used.</p>

Troubleshooting R/3 Settings

Error ID	Possible Cause / Error Description	Possible Correction
E-SAP0110	Wrong user or wrong password or wrong client. RfcOpenEx returned, Communication error. Error message: You are not authorized to logon to the target system.	Verify your connection parameters and your log permissions into the R/3 system. Refer to "User ID*" on page 2-8.
	Wrong server name. A wrong host ss6 was given as the host name. RfcOpenEx returned, Communication error.	Verify the server and host name in your R/3 settings. Refer to "Application Server **" on page 2-6.
	Wrong system number. RfcOpenEx returned. Error message: Connect to SAP gateway failed. Connection refused.	Verify system number parameter setting. Refer to "Application Server / System Number**" on page 2-6.
	E-SAP0106	Wrong Server Name. RFC Communication error. Error message: Connect to SAP gateway failed.
	Wrong System Number. RFC Communication error. Error message: Connect to SAP gateway failed. Connection refused.	Verify system number parameter setting. Refer to "Application Server / System Number**" on page 2-6.
E-CORE0036	No librfccm.so in the library path. Could not load Oracle Application Server Integration Adapter for SAP R/3 or one of its dependencies may not be installed.	Verify the existence of the R/3 librfccm.so. Verify that the library path points to the librfccm.so. Refer to "Required Software" on page 1-4.

Defining an Interaction

This chapter describes how to configure Oracle Application Server ProcessConnect to access RFC and BAPI function modules, and ALE IDOCs in an R/3 system.

The Oracle Application Server Integration Adapter for SAP R/3 supports two Adapter Exchange Protocols in the inbound direction: ALE and RFC. For the inbound direction, R/3 makes a remote call and expects Oracle Application Server ProcessConnect to implement the function.

- ALE—ALE, or Application Link Enabling. This is a messaging protocol where the messages are big structures called Intermediate Documents or IDOCs. For the inbound direction, R/3 sends the IDOC into Oracle Application Server ProcessConnect.
- RFC—RFC, or Remote Function Call. This is a request/reply protocol which makes possible the remote execution of function modules.

In the outbound direction, there are three Adapter Exchange Protocols: ALE, RFC and BAPI.

- BAPI, or Business Application Programming Interface. This is similar to RFC except the hierarchy of functions is presented differently.

For ALE in the outbound direction, Oracle Application Server ProcessConnect sends the IDOC to R/3.

For RFC and BAPI in the outbound direction, Oracle Application Server ProcessConnect makes the remote call and expects R/3 to implement the call.

This chapter discusses the following topic:

- [Adding an Interaction](#)

Adding an Interaction

After defining a delivery channel for an R/3 system you can add interactions. Follow these instructions to add an RFC, BAPI, or ALE IDOC as an interaction in Oracle Application Server ProcessConnect.

See Also: *Oracle Application Server ProcessConnect User's Guide* for details about interactions in Oracle Application Server ProcessConnect

Before you can view ALE IDOCs in the browser you must have either:

- Downloaded the ALE IDOCs you want to use to an Oracle Application Server ProcessConnect machine

See Also: ["Manually Downloading an IDOC"](#) on page 5-2

- Imported the Utility Function Modules into an R/3 system

See Also: ["Installing Oracle AS ProcessConnect Utility Modules in Your R/3 System"](#) on page 1-5.

See Also: ["AdvancedSend Versus Send"](#) on page 5-8

All interactions have at most two arguments. The first argument is a structure that contains all input arguments. The name is fully scoped and ends with underscore Request. The second argument is a structure that contains all output arguments and the return value. The name is fully scoped and ends with underscore Reply.

1. Select **Modeling > Interactions**.

Oracle Application Server
ProcessConnect

Home Help Logout

Modeling Profiles Deployment Reports Administration

Business Processes Roles Event Types Datatypes Transformations **Interactions** Condition Expressions

Logged in as ip

Interactions

This shows the interactions defined in the system. Please choose Add to add an interaction.

Expand All | Collapse All

⊕ Adapter Providers

Focus Item	Delete
▼ Adapter Providers	
⊕ ▶ Oracle	
⊕ ▶ Oracle IP Development team	

Add

Shortcuts

- Create Native Event Type
- Native Event Types
- Application Event Types

Modeling | Profiles | Deployment | Reports | Administration | Home | Help | Logout

Copyright © 2002, 2003, Oracle Corporation. All rights reserved.

2. Click **Add** to add an interaction.
3. Expand **Oracle**.

Oracle Application Server
ProcessConnect

Home Help Logout

Modeling Profiles Deployment Reports Administration

Business Processes Roles Event Types Datatypes Transformations **Interactions** Condition Expressions

Logged in as ip

Add Interaction: Select Adapter Type

Please select an adapter type.

Expand All | Collapse All

⊕ Adapter Providers

Focus Item
▼ Adapter Providers
⊕ ▼ Oracle
AQ Adapter
Oracle DB Adapter
JMS Adapter
File/FTP Adapter
HTTP Adapter
Email Adapter
Webservice Adapter
SAP R/3 Adapter
PeopleSoft8 Adapter
Siebel2000 Adapter
JDE Adapter
⊕ ▶ Oracle IP Development team

Modeling | Profiles | Deployment | Reports | Administration | Home | Help | Logout

Copyright © 2002, 2003, Oracle Corporation. All rights reserved.

4. Select **SAP R/3 Adapter**.
5. Select a delivery channel.

The screenshot shows the Oracle Application Server ProcessConnect web interface. The page title is "Add Interaction: Select Delivery Channel". The main content area contains a message: "Please select the delivery channel for the adapter type selected. This delivery channel will be used to browse the application's interactions." Below this message are links for "Expand All" and "Collapse All". A tree view shows the following structure:

- Applications
 - Applications (Focus Item)
 - SAPApp1
 - mySAPTest
 - mySAPDeliveryChannel

The footer of the page includes navigation links: Modeling | Profiles | Deployment | Reports | Administration | Home | Help | Logout, and a copyright notice: Copyright © 2002, 2003, Oracle Corporation. All rights reserved.

6. Expand either the Inbound or Outbound folders depending on the type of interaction you would like to create. For this discussion, click **Outbound**.

Outbound interactions generate a request and a reply. They are used when Oracle Application Server ProcessConnect is communicating with an R/3 system.

Inbound interactions generate a request and are used when an R/3 adapter exchange protocol is sending messages to Oracle Application Server ProcessConnect.

The Add Interaction: Select Interaction page displays the Remote Function Call (RFC), BAPI, and ALE adapter exchange protocols that you can access.

Oracle Application Server
ProcessConnect

Home Help Logout

Modeling Profiles Deployment Reports Administration

Business Processes Roles Transformations Event Types Datatypes Interactions Condition Expressions

Logged in as ip

Add Interaction: Select Interaction

Please select the interaction to add.

[Expand All](#) | [Collapse All](#)

⊕ Adapter Exchange Protocols

Focus Item	
▼	Adapter Exchange Protocols
⊕	▼ Inbound
⊕	▶ ALE
⊕	▶ RFC
⊕	▼ Outbound
⊕	▶ ALE
⊕	▶ RFC
⊕	▶ BAPI

Modeling | [Profiles](#) | [Deployment](#) | [Reports](#) | [Administration](#) | [Home](#) | [Help](#) | [Logout](#)

Copyright © 2002, 2003, Oracle Corporation. All rights reserved.

- Click **ALE**, **RFC**, or **BAPI** to display modules that you can access.

ALE—Under ALE you can find a long list of available IDOCs. Every IDOC has the same two methods: Send and AdvancedSend. They function the same way, but the arguments for AdvancedSend are more complex and, therefore, give more control. In general, you should use Send for ALE outbound and AdvancedSend for ALE inbound.

RFC—The RFC hierarchy starts with a letter, then a GroupName, and then the list of function modules in that group. This hierarchy reflects the R/3 native hierarchy.

BAPI—The BAPI hierarchy is organized in groups that have more significant names. Within BAPI, the groups can also contain inner groups, with the interactions at the lowest level.

The RFC and BAPI branches are provided as a convenience, depending on whether you are more familiar with the Business Object Repository (BOR) hierarchy or with the RFC groups. Both contain function modules capable of performing an operation.

- The RFC folder contains function modules organized by group (BAPI modules). In the RFC folder, function modules are organized by function module groups. The areas are denoted by a letter, for example, A,B, ..., Z.

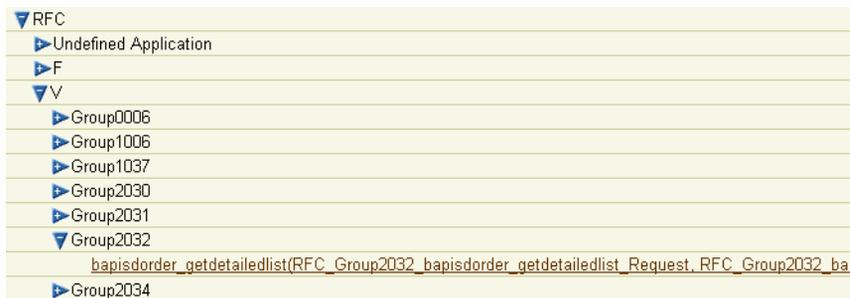
Expanding the area folder displays its groups, and expanding a function group displays its function modules.

- The BAPI folder contains function modules used by Business Object Repository (BOR), and is organized according to the BOR hierarchy.

BAPI - BAPI Full Content contains an Application Components folder. Expanding Application Components displays other folders and application components. An application component contains function modules and corresponds to a business object.

The ALE folder displays ALE IDOCs (if you loaded the enhancement Utility or downloaded the IDOCs manually).

8. Select a group and choose an interaction. For this discussion, expand **RFC > V > Group2032** nodes and select the **bapisdorder_getdetaileddlist** transaction.



If you were selecting an ALE Inbound interaction, you must select the AdvancedSend method.

See Also: ["AdvancedSend Versus Send"](#) on page 5-8

Add Interaction: Review

Please verify the interaction you are about to add. Choose **Apply** to add the interaction. Please note that you will be asked to specify native formats and extractors after adding the interaction.

Create Native Event Type

After adding the interaction and specifying the native formats and extractors, you will be able to create the native event and event body elements.

Interaction

Name **bapisdorder_getdetailedlist**
Is Inbound **False**

In Record Type

Name **RFC_Group2032_bapisdorder_getdetailedlist_Request**

Out Record Type

Name **RFC_Group2032_bapisdorder_getdetailedlist_Reply**

Cancel Apply

The Add Interaction: Review page displays the details. For an Outbound selection, there is an InRecord Type and an OutRecord Type. For an Inbound selection, there is only an InRecord Type.

9. Click **Apply**.

The Confirmation screen appears, allowing you to specify the Native Format of the request.

 **Confirmation**

Interaction bapisdorder_getdetailedlist(RFC_Group2032_bapisdorder_getdetailedlist_Request, RFC_Group2032_bapisdorder_getdetailedlist_Reply) successfully added.

Specify Native Format

Apply

Please specify a native format and extractor for each record type element and choose **Apply**. Since a value is set by default, please verify that the correct native format and extractor are specified.

Record Type Element	Native Format	Extractor
RFC_Group2032_bapisdorder_getdetailedlist_Request	XSD	XSD

Apply

10. Verify that the selection for the Native Format and the Extractor of the request is **XSD** and click **Apply**.

The Confirmation screen appears, allowing you to specify the Native Format of the reply.

 **Confirmation**
Successfully specified native formats and extractors.

Specify Native Format Apply

Please specify a native format and extractor for each record type element and choose Apply. Since a value is set by default, please verify that the correct native format and extractor are specified.

Record Type Element	Native Format	Extractor
RFC_Group2032_bapisdorder_getdetaileddlist_Reply	XSD	XSD

Apply

11. Verify that the selection for the Native Format and the Extractor of the reply is **XSD** and click **Apply**.
12. The interaction continues into the Create Native Event Type wizard.

See Also: *Oracle Application Server ProcessConnect User's Guide* for instructions on using the wizard

On completion, the new interaction appears in the Interactions list.

Interactions

This shows the interactions defined in the system. Please choose Add to add an interaction.

[Expand All](#) | [Collapse All](#)

⊕ Adapter Providers

Focus Item

- ▼ Adapter Providers
 - ⊕ ▼ Oracle
 - ⊕ ▼ SAP R/3 Adapter
 - ⊕ ▶ ALE
 - ⊕ ▼ RFC
 - ⊕ ▼ V
 - ⊕ ▼ Group2032
 - [bapisdorder_getdetaileddlist\(RFC_Group2032_bapisdorder_getdetaileddlist_Request RFC_Group2032_bapisdorder_getde](#)

You can click the interaction link for a complete view of the interaction details.

Oracle Application Server
ProcessConnect

Home Help Logout

Modeling Profiles Deployment Reports Administration

Business Processes Roles Event Types Datatypes Transformations **Interactions** Condition Expressions

Logged in as ip

Interaction Details : bapisdorder_getdetailedlist

Delete

Details

Name **bapisdorder_getdetailedlist**
 Adapter Exchange Protocol **RFC**
 Group Name **Group2032**
 Is Inbound **False**
 In Record Type [RFC_Group2032_bapisdorder_getdetailedlist_Request](#)
 Out Record Type [RFC_Group2032_bapisdorder_getdetailedlist_Reply](#)

Interaction Parameters

Parameter	Value
InteractionVerb	1
AdapterExchangeProtocolName	RFC
ID	bapisdorder_getdetailedlist@R/3://RFC/Group2032

[Return To List](#)

Delete

Modeling | [Profiles](#) | [Deployment](#) | [Reports](#) | [Administration](#) | [Home](#) | [Help](#) | [Logout](#)

Using Oracle Application Server Integration Adapter for SAP R/3 Datatypes

This chapter provides information on the Oracle Application Server Integration Adapter for SAP R/3 datatypes. This chapter discusses the following topics:

- [Basic Datatypes](#)
- [Numeric Strings \(NUMC Type\)](#)
- [Fixed-Point Numeric Strings \(PACKED Type\)](#)
- [Date and Time](#)
- [Binary Data](#)
- [Complex Types](#)
- [Getlist Function Module](#)

Basic Datatypes

Table 4–1 describes the list of basic types in R/3 and how they map to XML schema types in Oracle Application Server ProcessConnect. The `xsd` prefix stands for the namespace `http://www.w3.org/2001/XMLSchema`.

Table 4–1 Basic Datatypes

R/3	Oracle Application Server ProcessConnect
Byte LIKE INT1.	xsd:unsignedByte
Char(i) TYPE C.	xsd:string
Date TYPE D.	xsd:date
Float TYPE F.	xsd:double
Integer TYPE I.	xsd:int
NUMC(i) TYPE N. ; i = 1 to 2	xsd:byte
NUMC(i) TYPE N. ; i = 3 to 4	xsd:short
NUMC(i) TYPE N. ; i = 5 to 9	xsd:int
NUMC(i) TYPE N. ; i = 10 to 14	xsd:double
NUMC(i) TYPE N. ; i > 14	xsd:string
Packed(i) TYPE P DECIMALS j ; i = 1; j = 0	xsd:byte
Packed(i) TYPE P DECIMALS j ; i = 2; j = 0	xsd:short
Packed(i) TYPE P DECIMALS j ; i = 3 to 5; j = 0	xsd:int
Packed(i) TYPE P DECIMALS j ; i = 6 to 7; j = 0	xsd:double
Packed(i) TYPE P DECIMALS j ; i <= 7; j > 0	xsd:double
Packed(i) TYPE P DECIMALS j ; i > 7	xsd:string
Raw(1) TYPE X.	xsd:unsignedByte
Raw(i) TYPE X.	xsd:base64Binary
Short LIKE INT2.	xsd:short
Time TYPE T.	xsd:time

Numeric Strings (NUMC Type)

The R/3 numeric string type is expressed as:

```
Numc (n)
```

Where

n is a positive integer specifying the number of decimal digits

Depending on n, this type can be translated as:

- xsd:byte
- xsd:short
- xsd:int
- xsd:double
- xsd:string

Take care when using NUMC precision, specifically when these types are for input parameters; otherwise Oracle Application Server ProcessConnect returns a conversion failure at runtime.

For example, suppose you have two function modules with a NUMC parameter only differing in length:

```
function y_test_numc20.
*"-----
*" IMPORTING
*" VALUE (ARG_NUMC20) LIKE NUMC (20)
*"-----
function y_test_numc10.
*"-----
*" IMPORTING
*" VALUE (ARG_NUMC10) LIKE NUMC (10)
*"-----
```

The generated schema translates the NUMC types to 2 different types:

```
<xsd:element name='arg_numc20' type='xsd:string' />
<xsd:element name='arg_numc10' type='xsd:double' />
```

In this example, even though NUMC (20) is exposed as a string, a value with more than 20 digits causes a conversion failure.

```
<!-- This fails because it is not a number -->
```

```
<arg_numc20>This is not a number</arg_numc20>

<!-- This fails because it has too many digits -->
<arg_numc20>12345678901234567890111</arg_numc20>

<!-- This works -->
<arg_numc20>12345678901234567890</arg_numc20>
```

In a similar way, even though NUMC (10) is exposed as a double, it should not contain more than 9 digits in the integral part. Any value in the fractional part is rounded upwards and discarded.

```
<!-- This fails because the precision is too high -->
<arg_numc10>12345678901</arg_numc10>

<!-- This passes in the value 12. -->
<arg_numc10>12.34</arg_numc10>

<!-- This works -->
<arg_numc10>123456789</arg_numc10>
```

Fixed-Point Numeric Strings (PACKED Type)

The R/3 version of a fixed point numeric string is:

```
Packed(b, s)
```

Where

b is the number of bytes occupied by instances of the type, which can range from 1 to 16. In a Packed datatype, two (2) characters are stored in each byte, and the sign is stored in half of a byte. 16 bytes give you 31 positions for numbers, with the precision telling you how many digits are in the fractional part.

s is the number of decimal digits after the decimal point. *s* is short for *scale*.

For example, instances of PACKED (7, 4) can range from -999 999 999.9999 to 999 999 999.9999, and the values are exact (as opposed to the IEEE floating point numeric type, where values are stored internally in base 2 and do not all have exact correspondence to decimal values).

Just like NUMC types, depending on *b* and *s*, the PACKED type can be translated as:

- xsd:byte

- xsd:short
- xsd:int
- xsd:double
- xsd:string

Take care when using Packed types as input parameters. If input length exceeds the specified Packed length, failures occur at runtime.

For example, the following three function modules show a PACKED parameter only differing in its number of decimals and length:

```
function y_test_pack_p4_d2
*"-----
*" IMPORTING
*"   VALUE(ARG_P4_D2) Packed(4) Type P Decimals 2
*"-----
function y_test_pack_p10_d0
*"-----
*" IMPORTING
*"   VALUE(ARG_P10_D0) Packed(10) TYPE P DECIMALS
*"-----
function y_test_pack_p10_d2
*"-----
*" IMPORTING
*"   VALUE(ARG_P10_D2) Packed(10) TYPE P DECIMALS 2
*"-----
```

The generated schema using the above three function modules translates the PACKED types to 3 different types:

```
<xsd:element name='arg_p4_d2' type='xsd:double' />
<xsd:element name='arg_p10_d0' type='xsd:string' />
<xsd:element name='arg_p10_d2' type='xsd:string' />
```

Even though the Packed(4, 2) is translated as a double, the total number of digits is $4 \times 2 - 1$ or 7, and no more than 5 digits in the integral part and the fractional part are rounded upwards to 2 digits.

```
<!-- This fails because the integral part exceeds 5 digits. -->
<arg_p4_d2>123456.78</arg_p4_d2>

<!-- This passes in the value 1234.57. -->
<arg_p4_d2>1234.567</arg_p4_d2>
```

```
<!-- This works -->
<arg_p4_d2>12345.67</arg_p4_d2>
```

The packed (10, 0) is being translated as a string. The integral part must not exceed 19 digits, and the fractional part is rounded up and discarded.

```
<!-- This passes in the value 1235. -->
<arg_p10_d0>1234.56</arg_p10_d0>
<!-- This fails because there are too many digits (20). -->
<arg_p10_d0>12345678901234567890</arg_p10_d0>
<!-- 19 Digits. This is valid. -->
<arg_p10_d0>-1234567890123456789</arg_p10_d0>
```

The packed (10, 2) is translated as a string. The integral part must not exceed 17 digits and the fractional part is rounded up to 2 digits.

```
<!-- This passes in the value 1234567890123456.79. -->
<arg_p10_d2>1234567890123456.789</arg_p10_d2>
<!-- 19 digits, 2 after the decimal points. This is valid. -->
<arg_p10_d2>12345678901234567.89</arg_p10_d2>
```

Date and Time

The R/3 Date type is exposed as `xsd:date` and the R/3 Time type is exposed as `xsd:time`.

The concept of null date values is useful when calling R/3. When a null date instance is used when invoking R/3, it translates as 00000000 in R/3. This special data value indicates that it should not be processed. To pass null dates in a call, you need to pass in a null Date instance.

In this example, the `xsi` prefix stands for the namespace `http://www.w3.org/2001/XMLSchema-instance`.

```
<myDate xsi:nil='true' />
```

Binary Data

R/3 Raw types, or `Raw(n)`, are types containing raw binary data. `Raw(1)` is a single byte value exposed as `xsd:unsignedByte`. `Raw(n)` for `n` greater than 1 is translated as `xsd:base64Binary`.

For example, the following function module has a Raw parameter:

```
function y_test_raw10.
*"-----
*" IMPORTING
*" VALUE (ARG_RAW10) Raw(10) TYPE X.
*"-----
```

The generated schema translates the RAW(10) type to:

```
<element name='arg_raw10' type='xsd:base64Binary' />
```

This type is rarely used. If it is, special care must be taken if inputting values. The sequence of bytes encoded in base64 must be exactly of length n.

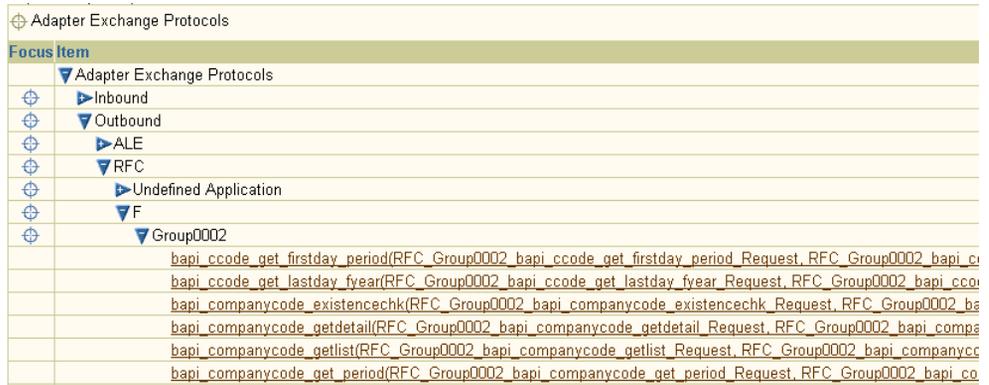
```
<!-- This works -->
<arg_raw10>ABCDabcd0123xy==</arg_raw10>
```

Complex Types

An R/3 structure translates to a schema complexType with a sequence of local elements. An R/3 table translates to a schema complexType containing a local element with minOccurs and maxOccurs.

Getlist Function Module

The following provides an example of how an RFC function module, `bapi_companycode_getlist`, is viewed in a generated XML schema format; one for input and one for output. This standard API function module returns a list of companies in the R/3 system database. The RFC function module can be located in the R/3 System in the area F, in the Function Group Group0002.



```

* "-----
FUNCTION BAPI_COMPANYCODE_GETLIST.
* "-----
* " * "Local interface:
* "     EXPORTING
* "         VALUE(RETURN) LIKE BAPIRETURN STRUCTURE BAPIRETURN
* "     TABLES
* "         COMPANYCODE_LIST STRUCTURE BAPI0002_1
* "-----

```

The function module `bapi_companycode_getlist` has 2 parameters:

- 1 output BAPIRETURN structure
- 1 inout table containing COMPANYCODE_LIST structures.

The BAPI0002_1 structure is a transfer structure for Object 0002: Company Code Get List. The following is the definition of Structure BAPI0002_1 in R/3.

Component	Component Type	DTyp	Length	Dec.	Short Text
COMP_CODE	BUKRS	CHAR	4	0	Company Code
COMP_NAME	BUTXT	CHAR	25	0	Name of the company code or company

The BAPIRETURN structure returns the status of the call and associated log messages. The following is the definition of Structure BAPIRETURN in R/3.

Component	Component Type	DTyp	Length	Dec.	Short Text
TYPE	BAPI_MTYPE	CHAR	1	0	Message type
CODE	BAPI_RCODE	CHAR	5	0	Message Code
MESSAGE	BAPI_MSG	CHAR	220	0	Message text
LOG_NO	BALOGNR	CHAR	20	0	Application log: log number
LOG_MSG_NO	BALMNR	NUMC	6	0	Application log: Internal Message serial number
MESSAGE_V1	SYMSGV	CHAR	50	0	Messages, message variables
MESSAGE_V2	SYMSGV	CHAR	50	0	Messages, message variables
MESSAGE_V3	SYMSGV	CHAR	50	0	Messages, message variables
MESSAGE_V4	SYMSGV	CHAR	50	0	Messages, message variables

The generated schema for the input record looks like this:

```
<xsd:schema targetNamespace='http://www.oracle.com/appadapters'
  elementFormDefault='unqualified'
  xmlns:exposed='http://www.oracle.com/appadapters'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'>

  <xsd:complexType name='RFC_BAPI0002_1'>
    <xsd:sequence>
      <xsd:element name='comp_code' nillable='true' type='xsd:string'/>
      <xsd:element name='comp_name' nillable='true' type='xsd:string'/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name='RFC_BAPI0002_1Seq'>
    <xsd:sequence>
      <xsd:element name='item' nillable='true'
        type='exposed:RFC_BAPI0002_1'
        minOccurs='0' maxOccurs='unbounded'/>
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name='RFC_Group0002_bapi_companycode_getlist_Request'
    nillable='true'>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name='companycode_list' nillable='true'
```

```
        type='exposed:RFC_BAPI0002_1Seq' />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

The generated schema for the output record looks like this:

```
<xsd:schema targetNamespace='http://www.oracle.com/appadapters'
  elementFormDefault='unqualified'
  xmlns:exposed='http://www.oracle.com/appadapters'
  xmlns:xsd='http://www.w3.org/2001/XMLSchema'>

  <xsd:complexType name='RFC_BAPIRETURN'>
    <xsd:sequence>
      <xsd:element name='type' nillable='true' type='xsd:string' />
      <xsd:element name='code' nillable='true' type='xsd:string' />
      <xsd:element name='message' nillable='true' type='xsd:string' />
      <xsd:element name='log_no' nillable='true' type='xsd:string' />
      <xsd:element name='log_msg_no' nillable='true' type='xsd:int' />
      <xsd:element name='message_v1' nillable='true' type='xsd:string' />
      <xsd:element name='message_v2' nillable='true' type='xsd:string' />
      <xsd:element name='message_v3' nillable='true' type='xsd:string' />
      <xsd:element name='message_v4' nillable='true' type='xsd:string' />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name='RFC_BAPI0002_1'>
    <xsd:sequence>
      <xsd:element name='comp_code' nillable='true' type='xsd:string' />
      <xsd:element name='comp_name' nillable='true' type='xsd:string' />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:complexType name='RFC_BAPI0002_1Seq'>
    <xsd:sequence>
      <xsd:element name='item' nillable='true'
        type='exposed:RFC_BAPI0002_1'
        minOccurs='0' maxOccurs='unbounded' />
    </xsd:sequence>
  </xsd:complexType>

  <xsd:element name='RFC_Group0002_bapi_companycode_getlist_Reply'
    nillable='true'>
    <xsd:complexType>
```

```
<xsd:sequence>
  <xsd:element name='return' nillable='true'
    type='exposed:RFC_BAPIRETURN' />
  <xsd:element name='companycode_list' nillable='true'
    type='exposed:RFC_BAPI0002_1Seq' />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Using Application Link Enabling

An R/3 Application Link Enabling (ALE) system handles the exchange of data messages across applications, ensuring data is consistent. When developing interactions with ALE IDOCs, you must be able to view ALE IDOCs. There are two ways to use Oracle Application Server ProcessConnect and R/3 to search for IDOCs:

- Import the Utility Function Modules to automatically explore the R/3 system for existing IDOCs.

See Also: ["Importing the Oracle Application Server ProcessConnect Utility Function Modules"](#) on page 1-6

- Manually download the ALE IDOC definitions (*.mtd files) to the Oracle Application Server ProcessConnect computer.

If you do not include the custom browsing function modules for ALE in your R/3 system, you must download the IDOC you wish to view.

This chapter discusses the following topics:

- [Manually Downloading an IDOC](#)
- [Frequently Used ALE Transactions](#)
- [ALE Terminology](#)

Manually Downloading an IDOC

If you work with only two or three IDOCs, it is recommended that you download the IDOCs manually so you can view the IDOCs when selecting an interaction.

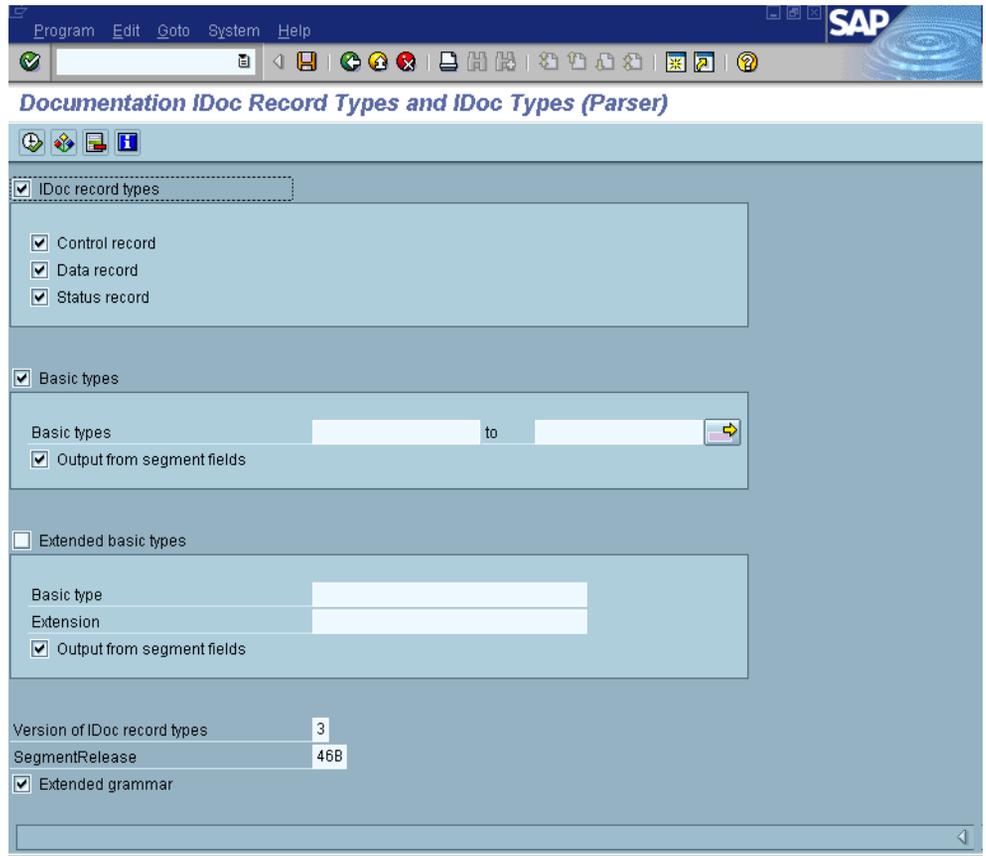
Note: If you are using IDOCs extensively, it is recommended that you use the enhanced browsing function modules to facilitate the browsing of the ALE IDOCs in the R/3 system.

See Also: ["Installing Oracle AS ProcessConnect Utility Modules in Your R/3 System"](#) on page 1-5 to install the custom function modules for browsing purposes

To manually download IDOC definitions from an R/3 system, complete the following steps:

1. Log on to an R/3 system.
The SAPGUI Easy Access dialog box is displayed.
2. From the main R/3 menu, expand **Tools > Business Communications > IDOC**.
The **Process technology** tree is displayed.
3. Expand **IDOC > IDOC Basis**.
4. Expand **Documentation > IDOC type (parser)**.

Figure 5–1 Documentation IDoc Record Types and IDoc Types (Parser)



5. Click the **Basic Types** browse button, and select *MATMAS01* from the **Selection of IDOC types**.
6. Click the check mark to accept and load your selection.
7. Press **F8** or **Execute** to run.
8. Select **List > Download**.
9. Select **unconverted** in the Save dialog.
10. Type the following in the Transfer List to a Local File type matmas02_46B.mtd:

`ORACLE_HOME/ip/adapters/config/SAP/Cache/IDOCName_SAPVers.mtd`

11. Click **OK**.
12. Exit the R/3 session.

You do not have to reestablish a session with R/3 to view ALE IDOCs, because the information has been downloaded and saved on your local computer.

Note: To refresh existing IDOCs on your Oracle Application Server ProcessConnect machine, delete any existing `.ido` files for that IDOC from your cache directory.

An MTD is the IDOC structure and an IDO is the binary version of an MTD.

For Your Information If an IDOC definition is needed, but no `*.ido` file exists, Oracle Application Server ProcessConnect downloads the `*.ido` file from an available R/3 system. However, if the R/3 system is down, nothing works. In this case, predownload the IDOC definitions to an `.mtd` file.

- `*.mtd` files create the `*.ido` files
- `*.ido` files are compiled versions of the `*.mtd` file

Calling the IDOC, either by viewing in Oracle Application Server ProcessConnect or being called by the Oracle Application Server Integration Adapter for SAP R/3, creates the `*.ido` file from the `*.mtd` ONLY on the initial call. If an `*.ido` exists in your Cache directory for a specific IDOC, Oracle Application Server ProcessConnect uses the existing `*.ido` file. If the `*.mtd` is then updated, the `*.ido` does NOT automatically update. You must first delete the old `*.ido` file, then call the IDOC to cause the updated `*.mtd` to generate a new `*.ido`.

The Cache directory contains local descriptions of ALE messages.

- Files named `".mtd"` (`IDOCName_SAPVersionNumber.mtd`) are created when downloaded using the R/3 `RSEIDOC3` file (with only **Display structure** and **Display segment** fields set and with only one IDOC type generated). Files of this name are automatically converted to `*.ido` files. However, `*.mtd` files do not convert to `*.ido` files if an `*.ido` file (with the same name) exists.
- Files named `"*.ido"` are binary files containing the local representation of IDOC messages. These are either downloaded from the R/3 system or they are built from `*.mtd` files as needed. IDOCs can be accessed at runtime only if the Enhanced Browsing Function Modules have been uploaded.

See Also: ["Installing Oracle AS ProcessConnect Utility Modules in Your R/3 System"](#) on page 1-5

The preferred method is to use the R/3 `SAPPARSER` method and download the `*.mtd` files locally to generate the `*.ido`.

See Also: ["Manually Downloading an IDOC"](#) on page 5-2

Frequently Used ALE Transactions

Table 5–1 displays a list of frequently used ALE transactions.

Table 5–1 *Frequently Used Application Link Enabling Transactions*

Transaction	Description
SALE	Application Link Enabling Customizing
BD21	Analyze change pointers - create IDOCs from change pointer
BD12	Send customer master
BD61	Activate change pointer generally
BD54	Maintain logical systems
BD64	Maintain distribution model
BD71	Distribute customer model
BDM2	Cross-system IDOCs reporting
WE02	IDOC Display
WE05	IDOC List
WE20	Maintain partner profile
WE21	Maintain port definition
WE30	Develop IDOC types
WE31	Maintain IDOC segment
WE60	IDOCs Documentation - IDOC types
BDM7	Application Link Enabling Audit - statistical analyses
WE14	Process (dispatch) IDOCs through port - RSEOUT00
WE16	Inbound file

Table 5–1 (Cont.) Frequently Used Application Link Enabling Transactions

Transaction	Description
WE42	Process code inbound
SARA	Central IDOCs archive
WE47	Status code maintenance
WE82	Assign IDOCs to message type
SM59	Maintain Remote Function Call destinations
SM37	Display batch jobs - job overview
SM50	Process overview
SLG1	Evaluate application log
SM21	System log
SM58	Transactional Remote Function Call monitoring
RZ12	Remote Function Call Server Group maintenance

ALE Terminology

The following terms are described:

- [Logical System](#)
- [Intermediate Documents \(IDOC\) Type](#)
- [Message Type](#)
- [Custom IDOC Behavior](#)
- [AdvancedSend Versus Send](#)

Logical System

A logical system is your R/3 address where you can distribute data to and from an R/3 system. Logical systems start with a base logical system.

A base system uses the case sensitive Remote Function Call (RFC). To browse the Remote Function Call destinations from the SAPGUI interface:

1. Click **Tools > Administration**.
2. Select **Network**.

3. Select **RFC destination**.
4. Select **TCP/IP Connections**.
5. Select the RFC destination to use.

Make sure the RFC destination points to the correct computer using the **System Information > Target System**. You can also verify your connection using the **Test Connection** button. Ask the administrator of the logical system which RFC destination to use.

Intermediate Documents (IDOC) Type

An IDOC type represents the structure of the data associated with a message type. An IDOC is a component with the data of a particular message type. Each IDOC contains only one business type.

Before you can send or receive IDOCs of a certain type, the IDOC structure must be defined. An IDOC consists of the following types:

- **Control Record**—Every IDOC has one control record. The control record contains information about the IDOC. For example, it contains the type of IDOC, the message type, sender and receiver information, and direction (inbound or outbound). This information provides control data on an outbound IDOC and processing options on an inbound IDOC.
- **Data Record**—An IDOC contains one or more data records containing application data and consists of one or more data records. Its sequence and structure are dictated by the sequence and structure of segments in a given IDOC type. For an outbound interface, ALE function modules populate these segments with application data. For inbound ALE interfaces, the application modules process the data contained in the segments.
- **Status Record**—With a length of 2 bytes, the status record contains information about the state of the IDOC as it passes through various stages of processing. R/3 assigns values between 01 to 41 for outbound IDOCs and assigns values between 50 to 73 for inbound IDOCs. The status record is a history of the IDOC states containing dates and time stamps.

IDOCs are identified by a unique IDOC number (`IDOCNUM`) assigned by R/3; however, it is possible to manually assign a number range of IDOCs.

Message Type

The message type represents the data exchanged between R/3 and an external system. A message type characterizes the data being sent across systems and relates to the structure of the data: an IDOC type. For example, `MATMAS` is a message type for Material Master, and `IVOIC` is a message type for an Invoice. Over 200 message types are supported by ALE in an R/3 system.

Access logical message types using the `/nwedi` transaction or by completing the following steps:

1. Select **Development**.
2. Select **IDOC types**.

Using **Environment > Message types** retrieves a list of available message types.

To access an assignment of logical message types to IDOC types, complete the following steps:

1. Select **Environment**.
2. Select **IDOC types/message**.

The main transaction in the R/3 system for IDOCs handling is `/nwedi`. R/3 documentation is available for IDOC types and IDOC segment types. The **IDOC > IDOC lists** menu accesses the list of IDOCs created and received in an R/3 system.

Custom IDOC Behavior

The segment type name in a custom IDOC must be compliant with R/3 specifications. For more information on using the R/3 transaction `/nWE31`, see online help on the field **segment type**: *All R/3 segment types begin with 'E1'; all customer-specific types must begin with 'Z1' or a customer specific prefix (namespace).*

AdvancedSend Versus Send

Each IDOC contains two methods: `AdvancedSend` and `Send`. Both methods perform the same function; however, the arguments are different.

- `Send` requires a `messageType` and an IDOC.
- `AdvancedSend` requires `controlData` and an IDOC

The Oracle Application Server Integration Adapter for SAP R/3 always triggers the `AdvancedSend` method when it receives an IDOC from an R/3 system; therefore, you must always choose the `AdvancedSend` method for inbound interactions.

For outbound interactions, you can choose `Send` or `AdvancedSend` depending on the method signature that suits your needs. If you use the `Send` method in your code, Oracle Application Server ProcessConnect takes the ALE Partner Logical System and the Oracle Application Server ProcessConnect Partner Logical System from the values set in the delivery channel. If you use the `AdvancedSend` method in your code, you have more control, but you must fill in the fields of the `controlData` argument yourself.

The following are the most useful fields in the `controlData` structure. Many of the other fields must be left blank.

Field	Description
<code>mestyp</code>	Message type
<code>idoctyp</code>	Basic structure of the IDOC.
<code>cimtyp</code>	Structure of customer extension if applicable.
<code>sndprt</code>	Partner type of the sender; LS (logical system) for ALE.
<code>sndprn</code>	Partner number of the sender; the logical system for ALE.
<code>rcvprt</code>	Partner type of the receiver; LS (logical system) for ALE.
<code>rcvprn</code>	Partner number of the receiver; the logical system for ALE.

See Also: *ALE Programming Guide* in the SAP Library for more details.

Retrying Inbound Calls

You can configure your R/3 system to retry inbound calls to Oracle Application Server ProcessConnect. When an inbound call fails, it goes in a list of incorrect calls. Within an R/3 system, you can delete the incorrect call, retry the incorrect call manually, or schedule a background job to retry incorrect calls automatically. When the retry fails again, the incorrect call stays in the list. When a call is successful, it disappears from the list.

This chapter discusses the following topics:

- [Using a Transactional RFC](#)
- [Scheduling an Automatic Retry](#)

Using a Transactional RFC

For the R/3 system to identify incorrect calls, it must receive a proper error status for each Oracle Application Server ProcessConnect inbound call. The protocol for all ALE interactions automatically handles the error status; whereas the protocol for RFC will not unless you explicitly select the transactional RFC (tRFC) protocol. This is the same protocol used by ALE. In ABAP, if the keywords “in background task” are appended to the call statement, then the function module is called asynchronously. This has the side effect of selecting the tRFC protocol. The tRFC protocol automatically reports errors. For example, the call to the function module `Z_MY_FUNC_MODULE` would look like `call function 'Z_MY_FUNC_MODULE' in a background task.`

Managing Incorrect Calls

To list incorrect calls, log on to SAPGUI. Using the R/3 system standard menu, select **Tools > Administration > Monitor > SM58 Transactional RFC**. Enter your search criteria and click **Execute**. The list of incorrect tRFC calls appears. On this page, you can delete an incorrect call by selecting a row and clicking **Delete Entry**. You can also reexecute a call by selecting a row and clicking **Edit > Execute LUW**. The list contains both invalid ALE calls and invalid asynchronous RFC calls.

To reexecute incorrect calls in batch, log on to SAPGUI. Using the R/3 system standard menu, select **Tools > ALE > ALE Administration > Services > Communication > Transactional RFC > BDA1 Invoke Calls Again**. Enter your selection criteria. It is recommended that you unselect the option **Currently being processed**. As soon as you click **Execute**, the calls immediately reexecute. For automatic retry, save your selection criteria as a variant. Select **Goto > Variants > Save As Variant** and give your variant a name.

Scheduling an Automatic Retry

To retry incorrect calls automatically, you must schedule a background job. This uses the built-in ABAP program, RSARFCEX. Log on to SAPGUI. Within the SAP standard menu select: **Tools > CCMS > Jobs > SM36 Definition**. The screen to define a background job appears.

1. Enter a new job name, for example, TRFC_RETRY.
2. Enter `job class C`.
3. Click **Start Condition**.
4. Click **Immediate**.

5. Select **Periodic job**.
6. Click **Periodic values** and enter the period.
Make sure the period is long enough to allow completion of each iteration.
7. Click the diskette icon to save the period values.
8. Click the diskette icon to save the start time.
9. Click **Step**.
10. For the ABAP program name, enter `RSARFCEX`.
11. For the **Variant**, enter the name of the variant you created on the BDA1 screen.
12. Click the diskette icon to save the step.
13. Click the diskette to save the background job.

As soon as you save, the ABAP program runs the job.

If a call is retried and fails again, it stays in the queue and it is retried at the next iteration.

Deleting a Background Job

The periodic job schedules a new one-time job for each iteration. If you delete the periodic job, the R/3 system stops running more iterations. To delete a background job, follow these steps:

1. Go to the screen **SM36 Define background job**.
2. Click **Own jobs**.
3. Select the job marked **Released**.
4. Click the **Delete** icon.
5. Click the **Refresh** icon to make sure the jobs are finished.

Index

A

ALE

- downloading intermediate documents
 - manually, 5-2
- frequently used transactions, 5-5
- interactions, 3-2
- logical system, 5-6
- parameters, 2-9
- terminology, 5-6

C

- custom modules for R/3 importing, 1-6

D

datatypes

- basic R/3, 4-2
 - R/3 complex, 4-7
 - R/3 date and time, 4-6
 - R/3 fixed point numeric string (packed type), 4-4
 - R/3 numeric string, 4-3
- ### delivery channels
- adding R/3, 2-2
 - configuring R/3 ALE, 2-9
 - configuring R/3 RFC, 2-5

I

- importing custom modules for R/3, 1-6
- interactions
 - adding R/3, 3-2

ALE, 3-2

RFC/BAPI, 3-2

intermediate documents type, 5-7

M

message types, 5-8

P

parameters

ALE, 2-9

RFC/BAPI, 2-5

platforms

supported for R/3, 1-4

R

R/3

importing custom modules, 1-6

retry inbound calls, 6-1

supported platforms, 1-4

requirements

software for R/3, 1-4

RFC/BAPI

interactions, 3-2

parameters, 2-5

