

TIBCO Adapter™ for Oracle Applications

User's Guide

*Software Release 5.4.1
June 2011*

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document contains confidential information that is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO, The Power of Now, TIBCO ActiveMatrix BusinessWorks, TIBCO Adapter, TIBCO ActiveEnterprise, TIBCO Rendezvous, TIBCO Designer, TIBCO Administrator, TIBCO IntegrationManager and TIBCO Hawk are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 1999-2011 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures	xiii
Tables	xix
Preface	xxi
Related Documentation	xxii
TIBCO Adapter for Oracle Applications Documentation	xxii
Other TIBCO Product Documentation	xxii
Third-Party Documentation	xxiv
Typographical Conventions	xxv
Terminology and Acronyms	xxviii
Connecting with TIBCO Resources	xxx
How to Join TIBCOCommunity	xxx
How to Access All TIBCO Documentation	xxx
How to Contact TIBCO Support	xxx
Chapter 1 Introduction	1
Oracle Applications Overview	2
Product Overview	4
Features	4
Integration with Oracle Applications	7
Additional TIBCO Products	8
Adapter Services Overview	10
Publication Service Overview	12
Database Components	12
Materialized View Logs	13
Publication Architecture	15
Publication Configuration	17
Subscription Service Overview	18
Database Components	18
Subscription Architecture	19
Subscription with Reply	20
Schema Support	23
Before You Start	24

Chapter 2 Installation	25
Preparing your Environment for Installation	26
Operating System Requirements	26
Oracle Application Requirements	27
Pre-Installation Worksheet	29
Adapter Machine Information	29
Oracle Applications Information	30
Oracle Applications Software	31
Installer Overview	32
Upgrading an Adapter	32
Uninstalling the Adapter	33
Installation Registry	35
Microsoft Windows Platforms	35
UNIX Platforms	35
Installation History	36
Adapter Components and Compatible Software	37
Adapter Components	37
Required and Optional TIBCO Products	37
Installing on Microsoft Windows	39
Installing on Windows Server 2003 Terminal Server	40
Installing the Adapter on Microsoft Windows	41
Combining Options	42
Clients, Platforms, and Drivers	42
Post-Installation Tasks on Windows	44
Configure the Service Name	44
Configure the Database Account	51
Using the create_user.sql Script	51
Using the common_all.sql Script	53
Create an ODBC System Data Source	57
Installation on UNIX Systems	63
Installing the Adapter on UNIX	64
Combining Options	65
Clients, Platforms, and Drivers	66
Post Installation Tasks on UNIX	68
Permission Requirements on UNIX Systems	68
Configure the Service Name	68
Configure the Database Account	68
Create an ODBC Data Source	69
Set Environment Variables	70
Database Objects Created as a Result of Configuration	72
Installation FAQs and Trouble Shooting	73
Frequently Asked Questions	73

Running Out of Disk Space	75
Installation Errors on HP-UX 11.00 64 bit Platform	75
Configuring TIBCO Hawk	76
Cannot Install the Adapter	76
Chapter 3 Getting Started	79
Overview	80
Configuring the Adapter Components	81
Task A Configure an Adapter Instance	81
Task B Configure Oracle Applications Business Object Design-time Connection Settings	83
Task C Configure an Items Business Object	85
Task D: Configure the corresponding ActiveDatabase Service	86
Deploying the Adapter	94
Start the Adapter	95
Stop the Adapter	96
Chapter 4 Configuring the Adapter	97
Overview	98
Configure the Adapter Overview	99
Projects	99
Using the Adapter with a Revision Control System	106
Oracle Applications Business Object Configuration	109
Configuration Tab	109
Design-time Connection Tab	110
PreDefined Outbound (Publisher) Oracle Applications Business Object Configuration	112
The Configuration Tab	113
The Tables Tab	115
The Adapter Options Tab	117
The Tables and View Mappings Tab	120
The Views Tab	121
Custom Outbound (Publisher) Oracle Applications Business Object	124
The Configuration Tab	125
The Tables Tab	126
The Adapter Options Tab	128
The Tables and View Mappings Tab	131
The Views Tab	132
PreDefined Inbound Current Program (Subscriber) Oracle Applications Business Object Configuration	135
The Configuration Tab	136
The Tables Tab	138
The Subscriber Options Tab (see Figure 69)	139
PreDefined Inbound API (Subscriber) Oracle Applications Business Object Configuration	144

The Configuration Tab	145
The Tables Tab.	146
The Subscriber Options Tab	147
Custom Inbound (Subscriber) Oracle Application Business Object Configuration	149
The Configuration Tab	150
The Tables Tab.	151
Adding tables for an API Business Object	153
Adding Tables for a Non-API Business Object.	153
The Subscriber Options Tab (see Figure 75).	154
Oracle Applications Inbound Business Object (For R12 Only)	159
Prerequisites for Configuring an Oracle Applications Inbound Business Object (For R12 Only) Service . .	159
Generic Procedures for Creating an Oracle Applications Inbound Business Object (For R12 Only).	160
The PL/SQL API Option.	161
The Concurrent Program Option	167
Functionalities of Tabs	170
Oracle Applications Inbound Business Object (for R12 Only) Services Configuration Examples	175
TIBCO Log Table Publisher Business Object Configuration	192
The Configuration Tab	193
The Tables Tab.	194
The TIBCO Log Table Message Subjects Tab.	196
The Adapter Options Tab.	198
Database Components	199
Oracle Publisher Configuration	200
The Configuration Tab	203
The Tables Tab.	208
The Publisher Options Tab.	210
The Advanced Tab	211
Oracle Subscriber Configuration.	214
The Configuration Tab.	217
The Tables Tab.	221
The Child Table Mappings Tab.	223
The Child Exception Table Mappings Tab	224
The Subscriber Options Tab	225
The Advanced Tab	226
Deleting a Publisher or a Subscriber.	229
Publisher service	229
Subscription Service	229
Testing the Adapter	229
Customizing a Business Object	230
PreDefined Publisher Business Object	230
Custom Publisher Business Object	230
PreDefined Subscriber Business Object	231
Custom Subscriber Business Object.	233

TIBCO Log Table Publisher	235
Conditional Publishing	237
Impact of Edit Actions on Database Objects	243
Setting Encoding Options	245
Chapter 5 Deploying and Starting an Adapter Using TIBCO Administrator.	247
Create an EAR File in TIBCO Designer.	248
Deploy the Project	249
Deploying Oracle Application Adapter Projects Using TIBCO Administrator	250
Start or Stop the Adapter.	252
Monitor the Adapter	253
Chapter 6 Monitoring the Adapter Using TIBCO Hawk	255
Overview	256
Starting TIBCO Hawk Software	257
The Auto-Discovery Process	258
Invoking Microagent Methods	259
Available Microagents	260
Chapter 7 Setting Encoding Options	261
Overview	262
Setting TIBCO Messaging Encoding	264
Configuring the Adapter to Communicate with the Database	266
Configuring the Adapter at Design Time	266
Special Configurations for Supporting a UTF-8 Database	268
Run-time Environment Variables	269
Relevant Environment Settings	270
Appendix A Available Transactions	271
Publish Transactions (PreDefined Outbound)	272
Subscribe Transactions (PreDefined Inbound)	274
Appendix B Oracle Application Transactions	277
Structure of PL/SQL Packages	279
Mandatory Parameters for All Transactions	281
Customizing TIBCO Procedure for Normal Subscription	283
Auto Invoicing	299

AutoLockBox 301

Bill of Materials 304

BudgetUpload 306

CollectionImport 308

CustomerItem 310

CustomerItemXRef 312

CycleCountEntries 314

DemandSchedules 316

JournalImport 319

MassAdditions 321

MasterSchedule 323

Open Replenishment 325

OpenBudget 327

OpenForecast 329

OpenItem 331

OpenMove 333

OpenResource 334

OpenTransaction 336

OrderEntry 338

PayablesOpen 340

PeriodicCost 343

PurchasingOpenDocs 345

RecCustomer 348

Receiving 349

Requisition 352

WorkOrder 354

Oracle Application APIs 356

Appendix C Master Mapping Tables 357

Publication Mapping Details 358

API Subscription Mapping Details 365

Non-API Subscription Mapping Details 370

Appendix D Trace Messages 379

Overview 380

Trace Message Fields 382

Identifying the Origin of Error Messages	384
Error Messages in TIBCO Adapter for Oracle Applications	385
Log Table Structure	386
Status Messages	389
TIBCO Log Table Messages	419
Validations Performed	420
User-defined Validation	420
Appendix E Clean-up Scripts	423
Clean-up of Records from MV Log	424
Clean-up Strategy for Publishing	425
Clean-up Strategy for Intermediate Tables	426
Purge TIBCO Records from Oracle Interface Tables	427
Appendix F Registering a Concurrent Program	429
Procedure	430
Appendix G Frequently Asked Questions	435
Frequently Asked Questions	436
Appendix H Database Comparison Tool	441
Functionality	442
Setting Up and Executing DBTool	443
Editing the Property file	444
Editing dbtool.bat	445
Sample Report Generated from dbtool	446
Appendix I Components for Custom Inbound Business Objects	447
Creating TIBCO Intermediate Tables for Custom Inbound API Business Objects	448
TIBCO ADB Columns	448
Example	449
Calling the PreCommit Stored Procedure	450
Writing a PL/SQL Package for Custom API Inbound Business Objects	451
Writing a PL/SQL Package for Custom Non-API Inbound Business Objects	459
Appendix J User Names for PreDefined Oracle Outbound Business Objects	461
User Names for PreDefined Oracle Outbound Business Object	462

Appendix K Troubleshooting Guide for TIBCO Oracle Applications Adapter	465
Insertion into Oracle Interface table failed	466
Oracle Concurrent Program Submission Failed	467
Insertion into Oracle Source Table Failed	468
Oracle Application Concurrent Manager is Down	469
Insertion into TIBCO Intermediate Table Failed	470
Running adbagent.exe Failed	471
Running adorapps.dta/adb.dta Failed	474
Error When Running the Adapter in Publication Mode	475
Modifying C-Shell to Other Shell Environments	479
Unable to Set Environment Variables on a Solaris Box	480
Unable to Create Trigger on MLOG Tables	482
Errors During Subscription and Subscription with Reply Services	483
Appendix L End-to-End Traceability	485
End-to-End Traceability	486
Non-API Subscription Transactions	486
API Subscription Transactions	490
Appendix M Logging Enhancements	491
Overview	492
Usage scenarios for enhanced logging	493
Case 1: Normal Transaction Execution	493
Case 2: Concurrent Program submission Failure	493
Case 3: Exception while program is running	493
Case 4: Option to customize log messages	493
Case 5: Provision to alter the Concurrent Program priority	494
Case 6: Quicker overall execution completion	494
Changes for enhanced logging	495
Appendix N Dynamic Population of Concurrent Program Parameters	497
Dynamic Population of Concurrent Program Parameters	498
Appendix O Tips on Using Oracle JPublisher	503
An Introduction to Oracle JPublisher	504
Creating a .bat File for a Specific API	505
Command Line Parameters	506
Before Executing the .bat File	508

Ways for Using a PL/SQL Wrapper File.509

**Appendix P Selecting APIs and Concurrent Programs from Oracle EBS R12 Integration
Repository511**

Index515

Figures

Figure 1	Logical Architecture for Integration With Oracle Applications	7
Figure 2	TIBCO Adapter for Oracle Applications and TIBCO ActiveMatrix Adapter for Database.	9
Figure 3	TIBCO Adapter for Oracle Applications - Graphical User Interface	10
Figure 4	Typical Publication Service Flow	12
Figure 5	Publishing from a Business Object consisting of one table in the Oracle Applications Database . .	15
Figure 6	The publishing architecture for a more complex object with two related source tables	16
Figure 7	Typical Subscription Service Flow	18
Figure 8	Subscription Architecture	20
Figure 9	Subscription with Reply	21
Figure 10	Oracle Configuration Assistant's Welcome Page.	44
Figure 11	The Net Service Name Configuration Screen	45
Figure 12	The Net Service Name Configuration, Service Name Screen	45
Figure 13	The Net Service name Configuration, Select Protocols Screen.	46
Figure 14	The Net Service Name Configuration, TCP/IP Protocol Screen	47
Figure 15	The Net Service name Configuration, Test Screen	47
Figure 16	The Net Service Name Configuration, connection Screen.	48
Figure 17	The Net Service Name Configuration, Net Service Name Screen.	49
Figure 18	The Net Service Name Configuration, Another Net Service Configuration Screen	49
Figure 19	The Net Service Name Configuration Done Screen.	50
Figure 20	The Oracle Net Configuration Assistant's Welcome Screen	50
Figure 21	create_user.sql: Prompts and Responses	52
Figure 22	common_all.sql: Prompts and Responses	57
Figure 23	The ODBC Data Source Administrator Dialog	58
Figure 24	The Create New Data Source Dialog	59
Figure 25	The ODBC Oracle Wire Protocol Driver Setup Dialog Box	59
Figure 26	The Logon to Oracle Wire Protocol Dialog	60
Figure 27	The Successful Connection Message	60
Figure 28	The ODBC Data Source Administrator Dialog	61

Figure 29	The Create New Data Source Dialog	61
Figure 30	The Oracle ODBC Driver Setup Dialog	62
Figure 31	The TIBCO Designer GUI	81
Figure 32	The Save Project Dialog	82
Figure 33	The TIBCO Designer Screen.	82
Figure 34	The Rename Warning Message	83
Figure 35	The Design-time Connection Tab	84
Figure 36	The Configuration Tab.	85
Figure 37	The Items Configuration Panel	86
Figure 38	The Configuration Tab of the ActiveDatabaseAdapterConfiguration Object.	87
Figure 39	The Design-time Connection Tab of the ActiveDatabaseAdapterConfiguration Object	88
Figure 40	The Run-time Connection Tab of the ActiveDatabaseAdapterConfiguration Object.	89
Figure 41	The Three Blue Services Buttons in the Palette Panel (Adapter Services Folder)	90
Figure 42	The Two Red Services Buttons in the Palette Panel (Adapter Services Folder).	90
Figure 43	The Configuration of the Items Business Object.	92
Figure 44	The Export Project Dialog	93
Figure 45	The Create Project Confirmation Dialog	93
Figure 46	The TIBCO Designer Dialog	99
Figure 47	The Save Project Dialog	100
Figure 48	Create an Oracle Apps Business Object.	101
Figure 49	The Configuration Tab of a Business Object.	102
Figure 50	Add the ActiveDatabase Adapter Configuration Button.	103
Figure 51	The Adapter Services Node	104
Figure 52	Select a Previously Configured Oracle Inbound or Outbound Business Object	105
Figure 53	The Additional Files Creation Warning Message	107
Figure 54	The Configuration Tab of the OracleAppsBusinessObjectConfiguration Object.	109
Figure 55	The Design-time Connection Tab of the OracleAppsBusinessObjectConfiguration Object	110
Figure 56	PreDefined Outbound Publisher Configuration Tab	114
Figure 57	PreDefined Outbound Publisher Tables Tab.	117
Figure 58	PreDefined Outbound Publisher Adapter Options Tab - Publish by Value	119
Figure 59	PreDefined Outbound Publisher Adapter Options Tab - Publish by Reference	120
Figure 60	PreDefined Outbound Publisher Tables and View Mappings Tab	121

Figure 61	PreDefined Outbound Publisher View Tab	123
Figure 62	The Custom Outbound Publisher Configuration Tab	126
Figure 63	The Custom Outbound Publisher Tables Tab	128
Figure 64	The Custom Outbound Publisher Adapter Options Tab	131
Figure 65	The Custom Outbound Publisher Table and View Mappings Tab	132
Figure 66	The Custom Outbound Publisher Views Tab	134
Figure 67	The PreDefined Inbound Subscriber Configuration Tab	138
Figure 68	The PreDefined Inbound Subscriber Tables Tab	139
Figure 69	The PreDefined Inbound Subscriber Options Tab	143
Figure 70	The PreDefined Inbound API Subscriber Configuration Tab	144
Figure 71	The PreDefined Inbound API Subscriber Tables Tab	147
Figure 72	The PreDefined Inbound API Subscriber Options Tab	148
Figure 73	The Custom Inbound Subscriber Configuration Tab	151
Figure 74	The Custom Inbound Subscriber Tables Tab	154
Figure 75	The Custom Inbound Subscriber Options Tab	158
Figure 76	The Inbound Business Object Configuration Options	161
Figure 77	The Inbound Business Object Configuration Panel	162
Figure 78	The Inbound Business Object Without Wrapper	163
Figure 79	The Inbound Business Object With Wrapper	164
Figure 80	The Inbound Business Object Tables Tab	166
Figure 81	The Inbound Business Object Subscriber Options Tab	166
Figure 82	The Concurrent Programs Configuration	167
Figure 83	The Inbound Business Object Tables Tab	168
Figure 84	The Concurrent Program Options Tab	169
Figure 85	The Object Subscriber Options Tab	170
Figure 86	The Business Object Configuration Tab	171
Figure 87	The Business Object API Wrapper Options Tab	172
Figure 88	The Concurrent Program Options Tab	173
Figure 89	The Inbound Business Object Tables Tab	174
Figure 90	The Object Subscriber Options Tab	175
Figure 91	API Simple Service Configuration (1).	176
Figure 92	API Simple Service Configuration (2).	177

Figure 93	API Simple Service Configuration (3)	177
Figure 94	API Simple Service Configuration (4)	178
Figure 95	API Simple Service Configuration (5)	179
Figure 96	API Simple Service Configuration (6)	179
Figure 97	API Simple Service Configuration (7)	180
Figure 98	Configuring an Adapter	180
Figure 99	API Wrapper Service Configuration (1)	182
Figure 100	API Wrapper Service Configuration (2)	183
Figure 101	API Wrapper Service Configuration (3)	184
Figure 102	API Wrapper Service Configuration (4)	184
Figure 103	API Wrapper Service Configuration (5)	185
Figure 104	API Wrapper Service Configuration (6)	185
Figure 105	API Wrapper Service Configuration (7)	186
Figure 106	Concurrent Program Service Configuration (1)	187
Figure 107	Concurrent Program Service Configuration (2)	188
Figure 108	Concurrent Program Service Configuration (3)	188
Figure 109	Concurrent Program Service Configuration (4)	189
Figure 110	Concurrent Program Service configuration (5)	190
Figure 111	Concurrent Program Service Configuration (6)	190
Figure 112	The TIBCO Log Table Publisher Configuration Tab	194
Figure 113	The TIBCO Log Table Publisher Tables Tab	196
Figure 114	TIBCO Log Table Publisher - the TIBCO Log Table Message Tab.	197
Figure 115	The TIBCO Log Table Publisher Adapter Options Tab	199
Figure 116	Oracle Applications Publisher	201
Figure 117	Oracle Publisher Configuration- Rendezvous	204
Figure 118	Oracle Publisher Configuration - JMS	205
Figure 119	Oracle Publisher Configuration Tab	207
Figure 120	The Oracle Publisher Tables Tab	209
Figure 121	Oracle Publisher - The Publisher Options Tab	211
Figure 122	The Oracle Publisher Advanced Tab.	213
Figure 123	Oracle Applications Subscriber	215
Figure 124	Oracle Subscriber Configuration- Rendezvous	217

Figure 125	Oracle Subscriber Configuration- JMS	219
Figure 126	The Oracle Subscriber Tables Tab	222
Figure 127	The Oracle Subscriber Child Table Mappings Tab	223
Figure 128	The Oracle Subscriber Child Exception Table Mappings Tab	224
Figure 129	The Oracle Subscriber -Subscriber Options Tab	226
Figure 130	The Oracle Subscriber Advanced Tab	228
Figure 131	TIBCO Hawk Enterprise Monitor	258
Figure 132	Example of Unicode Conversion	262
Figure 133	End-To-End Traceability	486
Figure 134	Dynamic Population of Concurrent Program Parameters	498
Figure 135	Concurrent Program Parameters in Intermediate Table	499
Figure 136	The EBS R12 Login Page	511
Figure 137	The EBS Navigator Page	512
Figure 138	The Search Page	512
Figure 139	The Package Information Page: Procedures and Functions List	513
Figure 140	The Package Information Page: Open Interfaces Tables /Views List	513

Tables

Table 1	General Typographical Conventions	xxv
Table 2	Syntax Typographical Conventions	xxvi
Table 3	TIBCO Adapter components	37
Table 4	Required and Optional TIBCO Products	37
Table 5	Supported platforms, package names, and disk space for Microsoft Windows	39
Table 6	Oracle Client Matrix for Microsoft Windows Platforms	42
Table 7	Oracle Driver Matrix for Windows Platforms	43
Table 8	Supported platforms, package names, patches and disk space for UNIX systems	63
Table 9	Oracle Client Matrix for UNIX Platforms	66
Table 10	IANAAppCodePage	267
Table 11	TIBCO Adapter for Oracle Applications Adapter and Oracle NLS_Lang Values	270
Table 12	Publish Transactions	272
Table 13	Subscribe Transactions (Non-API)	274
Table 14	Subscribe Transactions (API)	275
Table 15	Data Types for Subscription - Auto Invoicing	300
Table 16	Data Types for Subscription - AutoLockBox	302
Table 17	Data Types for Subscription - Bill of Materials	304
Table 18	Data Types for Subscription - BudgetUpload	306
Table 19	Data Types for Subscription - CollectionImport	308
Table 20	Data Types for Subscription - CustomerItem	310
Table 21	Data Types for Subscription - CustomerItemXRef	312
Table 22	Data Types for Subscription - CycleCountEntries	314
Table 23	Data Types for Subscription - DemandSchedules	317
Table 24	Data Types for Subscription - JournalImport	320
Table 25	Data Types for Subscription - MassAdditions	321
Table 26	Data Types for Subscription - Open Replenishment	325
Table 27	Data Types for Subscription - OpenBudget	327
Table 28	Data Types for Subscription - OpenItem	332

Table 29 Data Types for Subscription - OrderEntry 339

Table 30 Data Types for Subscription - PayablesOpen 341

Table 31 Data Types for Subscription - PeriodicCost. 343

Table 32 Data Types for Subscription - PurchasingOpenDocs 346

Table 33 Data Types for Subscription - Receiving 350

Table 34 Data Types for Subscription - Requisition 353

Table 35 Data Types for Subscription - WorkOrder 355

Table 36 Publication Mapping Tables 358

Table 37 API Subscription Mapping Details 365

Table 38 Non-API Subscription Mapping Details 370

Table 39 Tracing Fields 382

Table 40 Log Table Structure 386

Table 41 TIBCO Log Table Messages 419

Table 42 Validations 420

Table 43 User Names for PreDefined Oracle Outbound Business Objects 462

Preface

This guide explains about the features, configuration, installation, deploying and troubleshooting of TIBCO Adapter for Oracle Applications.

Topics

- [Related Documentation, page xxii](#)
- [Typographical Conventions, page xxv](#)
- [Terminology and Acronyms, page xxviii](#)
- [Connecting with TIBCO Resources, page xxx](#)

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Adapter for Oracle Applications Documentation

The following documents form the TIBCO Adapter for Oracle Applications documentation set:

- *TIBCO Adapter Concepts*: This manual introduces adapters by explaining what they are, and explains how to install, configure, deploy and manage adapters.
- *TIBCO Adapter for Oracle Applications User's Guide*: Read this manual to gain an understanding of the product.
- *TIBCO Adapter for Oracle Applications Release Notes*: Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.
- *TIBCO Adapter for Oracle Applications readme.txt*: Read this document to check the current release number, supported platforms and required software.
- *TIBCO Adapter for ActiveDatabase User's Guide*: Read this document to gain an understanding of the TIBCO ActiveMatrix Adapter for Database product, the run-time component of TIBCO Adapter for Oracle Applications.
- *TIBCO ActiveMatrix Adapter for Database Release Notes*: Read this document for a summary of new features, changes since last release and a description of any open or closed issues that may affect installing or using the adapter.
- *TIBCO ActiveMatrix Adapter for Database readme.txt*: Read this document to check the current release number, supported platforms and required software.

Other TIBCO Product Documentation

You may find it useful to read the documentation for the following TIBCO products:

- TIBCO BusinessWorks™ Software
 - *TIBCO BusinessWorks Concepts*
 - *TIBCO BusinessWorks QuickStart*
 - *TIBCO BusinessWorks Business Palette Reference*
 - *TIBCO BusinessWorks Process Design Guide*
 - *TIBCO BusinessWorks Error Codes*
 - *TIBCO BusinessWorks Release Notes*
- TIBCO Designer™ software:
 - *TIBCO Designer User's Guide*
 - *TIBCO Designer Palette Reference*
 - *TIBCO Designer Release Notes*
- TIBCO Administrator™ software:
 - *TIBCO Administrator User's Guide*
 - *TIBCO Administrator Server Configuration Guide*
 - *TIBCO Administrator Release Notes*
- TIBCO IntegrationManager™ software:
 - *TIBCO IntegrationManager Concepts*
 - *TIBCO IntegrationManager Administrator's Guide*
 - *TIBCO IntegrationManager Process Design Guide*
 - *TIBCO IntegrationManager Reference*
 - *TIBCO IntegrationManager Release Notes*
- TIBCO Rendezvous™ software:
 - *TIBCO Rendezvous Concepts*
 - *TIBCO Rendezvous Administration*
 - *TIBCO Rendezvous Configuration Tools*
- TIBCO Enterprise Message Service™ software:
 - *TIBCO Enterprise Message Service User's Guide*
 - *TIBCO Enterprise Message Service Installation*
 - *TIBCO Enterprise Message Service Application Integration*
 - *TIBCO Enterprise Message Service Release Notes*

- TIBCO Hawk[®] software:
 - *TIBCO Hawk Installation and Configuration*
 - *TIBCO Hawk Administrator's Guide*
- TIBCO Adapter[™] SDK
 - *TIBCO Adapter SDK Concepts*
- TIBCO Runtime Agent[™]
 - *TIBCO Runtime Agent Release Notes*
 - *TIBCO Runtime Agent Installation*
 - *TIBCO Runtime Agent Domain Utility User's Guide*
- *TIBCO Runtime Agent Upgrading to Release 5.3*

Third-Party Documentation

Refer to the *Technical Reference Manual* (TRM) included with Oracle Applications 11i when modifying adapter publication and subscription transactions.

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
<i>TIBCO_HOME</i> <i>ENV_HOME</i> <i>ADORAPPS_HOME</i>	<p>Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as <i>TIBCO_HOME</i>. The value of <i>TIBCO_HOME</i> depends on the operating system. For example, on Windows systems, the default value is C:\tibco.</p> <p>Other TIBCO products are installed into an installation environment. Incompatible products and multiple instances of the same product are installed into different installation environments. The directory into which such products are installed is referenced in documentation as <i>ENV_HOME</i>. The value of <i>ENV_HOME</i> depends on the operating system. For example, on Windows systems the default value is C:\tibco.</p> <p>TIBCO Adapter for Oracle Application installs into a directory within <i>TIBCO_HOME</i>. This directory is referenced in documentation as <i>ADORAPPS_HOME</i>. The value of <i>ADORAPPS_HOME</i> depends on the operating system. For example on Windows systems, the default value is C:\tibco\adapter\adorapps\5.4.</p>
code font	<p>Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example:</p> <p>Use MyCommand to start the foo process.</p>
bold code font	<p>Bold code font is used in the following ways:</p> <ul style="list-style-type: none"> • In procedures, to indicate what a user types. For example: Type admin. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, MyCommand is enabled: MyCommand [enable disable]

Table 1 General Typographical Conventions (Cont'd)




Convention	Use
<i>italic font</i>	<p>Italic font is used in the following ways:</p> <ul style="list-style-type: none">• To indicate a document title. For example: See <i>TIBCO ActiveMatrix BusinessWorks Concepts</i>.• To introduce new terms For example: A portal page may contain several portlets. <i>Portlets</i> are mini-applications that run in a portal.• To indicate a variable in a command or code syntax that you must replace. For example: <code>MyCommand <i>PathName</i></code>
Key combinations	<p>Key name separated by a plus sign indicate keys pressed simultaneously. For example: <code>Ctrl+C</code>.</p> <p>Key names separated by a comma and space indicate keys pressed one after the other. For example: <code>Esc, Ctrl+Q</code>.</p>
	<p>The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.</p>
	<p>The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.</p>
	<p>The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.</p>

Table 2 Syntax Typographical Conventions

Convention	Use
[]	<p>An optional item in a command or code syntax.</p> <p>For example:</p> <pre>MyCommand [optional_parameter] required_parameter</pre>
	<p>A logical OR that separates multiple items of which only one may be chosen.</p> <p>For example, you can select only one of the following parameters:</p> <pre>MyCommand para1 param2 param3</pre>

Table 2 *Syntax Typographical Conventions*

Convention	Use
{ }	<p>A logical group of items in a command. Other syntax notations may appear within each logical group.</p> <p>For example, the following command requires two parameters, which can be either the pair param1 and param2, or the pair param3 and param4.</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command requires two parameters. The first parameter can be either param1 or param2 and the second can be either param3 or param4:</p> <pre>MyCommand {param1 param2} {param3 param4}</pre> <p>In the next example, the command can accept either two or three parameters. The first parameter must be param1. You can optionally include param2 as the second parameter. And the last parameter is either param3 or param4.</p> <pre>MyCommand param1 [param2] {param3 param4}</pre>

Terminology and Acronyms

The following terms and acronyms are used in this document:

Term	Definition
API	Application Program Interface.
Business Object	When a Business Transaction occurs, a set of database objects get affected by the transaction. Such a logical set of related database objects for a particular transaction is called a Business Object.
Outbound Business Object	A Business Object associated with the Publication Service is called an Outbound Business Object.
Inbound Business Object	A Business Object associated with the Subscription Service is called an Inbound Business Object.
Predefined Business Object	A Predefined Business Object consists of database objects that are pre-set according to standard Oracle Applications workflow.
Custom Business Object	A Custom Business Object allows you to configure database objects that reflect your business workflow
Key Column	When publishing by reference, a key column or substitute key column is required for populating the publishing table. If the source table has no primary key, it can be set up to publish by reference only if a non-key column is specified when adding the publication.
Views	A view is a tailored presentation of the data contained in one or more tables (or other views). A view takes the output of a query and treats it as a table. Hence, a view can be called a stored "query" or a "virtual table". You can use views in most places where a table can be used.
MVLog	Materialized View Log
DML	Data Manipulation Language
RV	TIBCO Rendezvous™ protocol; also in certain contexts refers to reliable message delivery quality of service, as opposed to certified message delivery quality of service.
JMS	Java Messaging Service
RVC	TIBCO Rendezvous Certified Message delivery quality of service

Term	Definition
RVDQ	TIBCO Rendezvous Distributed Queue
vpd	Vital Product Database

Connecting with TIBCO Resources

How to Join TIBCOCommunity

TIBCOCommunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOCommunity offers forums, blogs, and access to a variety of resources. To register, go to <http://www.tibcommunity.com>.

How to Access All TIBCO Documentation

After joining TIBCOCommunity, you can access the documentation for all supported product versions here:

<http://docs.tibco.com/TibcoDoc>

How to Contact TIBCO Support

For comments or problems with this document or the software it addresses, contact TIBCO Support as follows:

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

<http://www.tibco.com/services/support>

- If you already have a valid maintenance or support contract, visit this site:

<https://support.tibco.com>

Entry to this site requires a user name and password. If you do not have a user name, click Register with Support.

Chapter 1 **Introduction**

This manual describes the TIBCO Adapter for Oracle Applications product features, and explains how to install, configure, deploy and run the adapter.

Topics

- [Oracle Applications Overview, page 2](#)
- [Product Overview, page 4](#)
- [Adapter Services Overview, page 10](#)
- [Publication Service Overview, page 12](#)
- [Subscription Service Overview, page 18](#)
- [Schema Support, page 23](#)
- [Before You Start, page 24](#)

Oracle Applications Overview

The Oracle Applications product suite allows companies to run their worldwide operations from a single, centrally managed site. Oracle Applications Release 11i includes two principal product suites: Enterprise Resource Planning (ERP) products, and Customer Relationship Management (CRM) products. Oracle Self-Service Web Applications ensure smooth and user-friendly transactions through the Internet. Oracle Applications Release 11i also consists of Oracle Workflow and Business Intelligence products, which are specific applications to support business processes. Oracle Applications Interfaces are a part of Oracle Applications Release 11i, which plays an important role in interacting with external applications.

Enterprise Resource Planning (ERP) Products

The ERP products are divided into several product families, such as Financials, Human Resources, Manufacturing and Distribution, and Process Manufacturing. ERP products enable your business to manage important operations, including product planning, purchasing, inventory management, interacting with suppliers, order tracking, human resources, financial planning, and accounting.

Customer Relationship Management (CRM) Products

Customer Relationship Management products provide front office functions such as call center management, e-commerce, and Internet sales and marketing.

Oracle Self-Service Web Applications

Self-Service Web Applications provide a fast and cost-effective way to get information to and from people within an organization or business. For example, Self-Service Web Applications allow customers to enter their own orders without involving the sales staff, or employees to enter their own change of address without involving the Human Resources staff. The interface is familiar to Web users, easy to work with, and doesn't require any training.

Oracle Workflow

The Oracle Workflow monitors business processes, collects process data, and provides an e-mail and web page notification system. For example, when an employee uses Oracle Internet Procurement (an Oracle Self-Service Web product) to enter a requisition, Oracle Workflow automatically validates the requisition and routes it to the appropriate manager for approval.

Business Intelligence System (BIS) Products

Business Intelligence System is a decision support solution integrated with Oracle Applications. Using the BIS products, a manager can query the Oracle Applications database to monitor recent business performance across multiple organizations. For example, a manager can set a sales goal and then use BIS to determine how close actual sales amounts are to the goal. A manager can set tolerances and have the system inform people when those tolerances are exceeded. With the BIS Performance Manager Framework, some corrective actions can be performed automatically. If, for example, actual sales amounts are more than 10% below goals, BIS can send automatic notifications to regional sales managers.

Using the Oracle Database as the underlying storage for Enterprise Data

Release 11i leverages the power of Oracle to substantially increase performance speed and reduce network traffic. All the business objects present in the ERP, CRM, and other such modules are modeled with the help of a set of functionally related database tables. The relationships amongst the tables present in such a set represents a Business Object as far as the adapter is concerned. There are several advantages of using the Oracle Database such as Cost-Based Optimization (CBO), Database Resource Manager, Partitioned Tables, Oracle Parallel Server, Materialized Views, etc.

Oracle Applications Interfaces

Oracle Applications use the underlying Oracle database for integrating with the application. Oracle Applications 11i provides the following interfacing points:

- **Interface Tables** — These are used for pushing data into Oracle Applications source tables. A number of Oracle Applications 11i PL/SQL programs are invoked, which in turn are responsible for validating incoming data and moving it to the Oracle Applications 11i source tables.
- **APIs** — PL/SQL APIs available provide business logic and move data into Oracle Apps 11i source tables.
- **Materialized View Logs (MV Logs) on Source Tables** — Materialized View Logs (MV Logs) are created on each table involved in the Business Object. Thereafter, triggers are created on these MV Logs. These triggers follow a cascading design; that is, data from a table lowermost in the hierarchy is sequentially pushed into the MV Log of its parent table until it is pushed out of the system.

Product Overview

TIBCO Adapter for Oracle Applications has the following components:

- **Software**
 - **Core adapter application** This is the run-time adapter application. The TIBCO ActiveMatrix Adapter for Database engine is used as the run-time component.
 - **Oracle Applications Adapter Configuration Palette** This is an adapter specific configuration tool that is packaged with the adapter and loaded in TIBCO Designer. It presents an intuitive Graphical User Interface (GUI) that enables you to configure the adapter. The palette offers access to Oracle Applications Business Objects.
 - **ActiveDatabase Adapter Services** The Publication or Subscription service of TIBCO ActiveMatrix Adapter for Database is associated with the Business Object created by TIBCO Adapter for Oracle Applications. The TIBCO ActiveMatrix Adapter for Database run-time engine publishes data into or subscribes data from the tables specified at design time by TIBCO Adapter for Oracle Applications.
- Example programs that involve TIBCO IntegrationManager, TIBCO Business Works, and the adapter in general.



When a Business Transaction occurs, a set of database objects are affected by the transaction. Such a logical set of related database objects for a particular transaction is called a **Business Object**.

The adapter enables bi-directional and real-time communication between Oracle Applications and TIBCO environment.

Features

The following features are described in detail throughout this manual:

- **Support for Oracle Applications 11.5.10.x and 12.0.x** TIBCO Adapter for Oracle Applications supports Oracle Applications 11.5.10.x and 12.0.x. The Predefined Subscription Non-API transactions, Predefined Publication transactions and Predefined Subscription API transactions of the adapter support Oracle Applications 11.5.10.x and 12.0.x.
- **End-to-End Traceability** The adapter supports End-to-End Traceability for the Subscription service. A record inserted into the TIBCO log table `TIB_INT_LOG_SUB` by the TIBCO Adapter for Oracle Applications is mapped

onto the source from which it originated. See [End-to-End Traceability on page 486](#) for more information.

- **Dynamic Population of Concurrent Program Parameters** You can extract concurrent program parameters from the Header Intermediate table and pass the parameters at run-time. See [Dynamic Population of Concurrent Program Parameters on page 498](#) for more information.
- **Purge TIBCO records from Oracle Interface Tables** All records that were previously inserted by the adapter into the Oracle Interface tables can now be purged by executing the PreCommitStored procedure. Records inserted into the Interface tables by other sources will not be deleted. See [Purge TIBCO Records from Oracle Interface Tables on page 427](#) for more information.
- **Update of PreDefined API transactions for 11.5.10** A number of PreDefined Subscriptions APIs have been updated to Oracle Applications version 11.5.10.
- **Adapter Services** The adapter provides publication, subscription, and subscription with reply services. For more information on adapter services, refer to [Adapter Services Overview on page 10](#).
- **Oracle Applications Business Objects** The adapter can publish and subscribe to entire Business Objects (header data as well as all associated detail data). The adapter contains PreDefined and Custom Business Objects that can be used to configure adapter services.
- **Publishing from Views** If the adapter is configured to publish by reference, you can specify a View (a View is an Oracle Database Object) as the referred object. You can customize these views and thereby extend the functionality of publishing from database tables.
- **Exception Handling, Monitoring and Message Tracking** The adapter employs effective exception-handling techniques and extensive audit trails. The adapter can be configured to work in concert with the TIBCO Hawk monitoring component to monitor and handle exception situations. Each message that is processed by an adapter service is tagged with a tracking-ID that enables complete end-to-end tracking of messages in the TIBCO environment.
- **An easy-to-use GUI** The adapter is configured using a user-friendly, intuitive Graphical User Interface: TIBCO Designer. You can quickly specify operational parameters and change them as needed.

- **Messaging** The adapter supports multiple messaging technologies: Rendezvous and Java Messaging Service.
 - The TIBCO Rendezvous mode of transport supports the following quality of services:
 - Reliable (RV) — This ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and the loss of a message is detected.
 - Certified-delivery (RVCN, ensures at delivery of messages at least once) — This guarantees that every certified message reaches its intended recipient in the order sent.
 - Distributed Queue (RVDQ) — This is designed to deliver a message to one-of-many service listeners (workers). It has incorporated the features of both Certified Messaging and Fault Tolerance. Senders of RVDQ are ordinary Certified Message senders.
 - The JMS mode of transport supports the following connection types (TIBCO Enterprise for JMS must be installed to use the JMS transport):
 - Topic, where a service publishes to a topic or subscribes to a topic. This type of message protocol is also known as broadcast messaging because messages are sent over the network and received by all interested subscribers, similar to how radio or television signals are broadcast and received.
 - Queue, where a service sends to a queue or receives from a queue. This message protocol is known as point-to-point because one and only one receiver consumes messages sent to a queue. A queue retains all messages sent until such time the messages are consumed or expired. Each message has only one receiver, though multiple receivers can connect to the queue. JMS messages use the XML Message wire format only.
- The JMS mode of transport supports two type of delivery modes, Persistent and Non-Persistent in the case of Publisher, and Durable and Non Durable in the case of Subscriber.

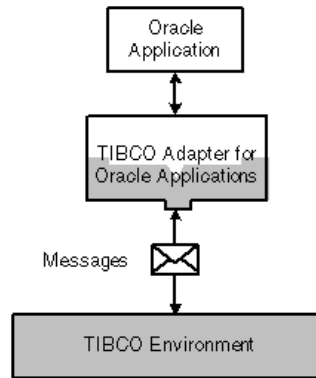


Migration For features like End-To-End traceability and passing concurrent program parameters at run-time, configure the TIBCO Adapter for Oracle Applications afresh.

Integration with Oracle Applications

The adapter is a bi-directional gateway between Oracle Applications and the TIBCO environment. The adapter can integrate with any Oracle Applications component. [Figure 1](#) shows a high level view of how the adapter is integrated with Oracle Applications in the TIBCO environment.

Figure 1 Logical Architecture for Integration With Oracle Applications



Each transaction in the TIBCO Adapter for Oracle Applications is modeled as a set of database objects also known as a *Business Object*. The database objects can be tables, triggers, or stored procedures. On receiving data from the TIBCO environment, the adapter inserts data into the required tables, and executes the triggers or stored procedures as specified in the Business Object. The runtime adapter interacts with these Business Objects.

In a Subscription scenario, a set of TIBCO Intermediate Tables forms the interface. The Intermediate table set for a PreDefined Open Item transaction is as follows:

Table	Child Tables
TIB_INT_MTL_SYS_ITEMS_IFACE	TIB_INT_MTL_ITEM_REV_IFACE

The runtime adapter interacts with this set of tables when it subscribes to data from the TIBCO environment. For more information, refer to the [Subscription Service Overview on page 18](#).

The adapter provides a few Oracle Applications API-based transactions for the subscription service. Oracle Applications APIs move data from TIBCO Intermediate Tables to Oracle Source Tables. For more information, refer to [Creating TIBCO Intermediate Tables for Custom Inbound API Business Objects on page 448](#).

In a Publication scenario, a set of database tables specific to the Business Object are created. During design time, the TIBCO Oracle Applications palette creates database objects in the database schema. The publishing table acts as the single point interface with which the adapter interacts. For more information, refer to [Publication Service Overview on page 12](#).

Additional TIBCO Products

TIBCO Designer is required to configure the adapter. TIBCO ActiveMatrix Adapter for Database is used at runtime to publish business objects to other applications or subscribe to business objects sent to Oracle Applications by other applications.

TIBCO Designer

TIBCO Designer is a GUI-based configuration utility for TIBCO applications. All adapter configuration including Publication Services, Subscription Services, Database Connection parameters, and parameters used to communicate with the TIBCO environment, is done through TIBCO Designer.

Each adapter package includes a *palette*, which is made available in TIBCO Designer when the adapter is installed. The *palettes* included with TIBCO Adapter for Oracle Applications include a TIBCO ActiveMatrix Adapter for Database *palette* and a TIBCO Adapter for Oracle Applications palette (a palette specialized for Oracle Applications providing pre-built transaction definitions). Resources from both palettes are required while configuring adapter services.

Additional TIBCO Designer operations, concepts, and terms are covered in the *TIBCO Designer User's Guide*. Details on performing tasks that are based on TIBCO ActiveMatrix Adapter for Database palette resources are described in the *TIBCO ActiveMatrix Adapter for Database User's Guide*. The instructions in the following chapters refer you to specific sections and procedures when necessary.

TIBCO ActiveMatrix Adapter for Database

TIBCO ActiveMatrix Adapter for Database is a database adapter that extends Publish, Subscribe, and Request-Response technology to databases.

TIBCO Adapter for Oracle Applications provides Oracle Applications specific functions. TIBCO ActiveMatrix Adapter for Database provides lower level database connectivity through its services.

Some tasks in this book refer to the *TIBCO ActiveMatrix Adapter for Database User's Guide* for database specific instructions.

Why TIBCO Adapter for Oracle Applications includes TIBCO ActiveMatrix Adapter for Database

TIBCO Adapter for Oracle Applications — This provides a *palette* to create a Business Object. Business Objects are a set of logically related database objects. The palette is used to create Business Objects for every business process.

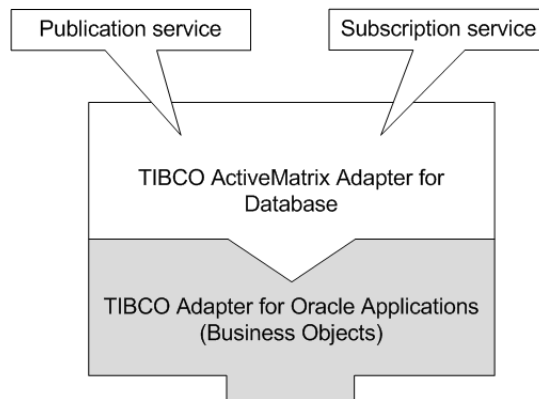
TIBCO ActiveMatrix Adapter for Database — This provides services. TIBCO ActiveMatrix Adapter for Database provides Publication and Subscription services.

A Business Object (created by TIBCO Adapter for Oracle Applications) is associated with a service (belonging to TIBCO ActiveMatrix Adapter for Database) in order to publish or subscribe. TIBCO Adapter for Oracle Applications specifies *where* and *how* the data has to be published (tables, triggers, or stored procedures) and TIBCO ActiveMatrix Adapter for Database publishes data using the publication or subscription service.

The TIBCO ActiveMatrix Adapter for Database run-time engine publishes data into or subscribes data from tables as configured by the TIBCO Adapter for Oracle Applications palette.

TIBCO Adapter for Oracle Applications is the Design-time component and TIBCO ActiveMatrix Adapter for Database is the Run-time component of the adapter.

Figure 2 TIBCO Adapter for Oracle Applications and TIBCO ActiveMatrix Adapter for Database



Adapter Services Overview

In TIBCO terminology, an Adapter offers *services* to the host application and to the TIBCO environment. A service broadly encapsulates routing rules for messages handled by the service and also custom configuration information. This section discusses services provided by the adapter with common usage scenarios drawn from real-life customer requirements.



The Publication or Subscription service of TIBCO ActiveMatrix Adapter for Database is associated with the Business Object created by TIBCO Adapter for Oracle Applications. The TIBCO ActiveMatrix Adapter for Database run-time engine publishes data into or subscribes data from tables specified by the TIBCO Adapter for Oracle Applications.

Communication parameters, database connectivity parameters, polling rate, and other parameters for these services can be configured using TIBCO Designer.

The following sections describe how these adapter services interact with an Oracle Applications environment.

Figure 3 TIBCO Adapter for Oracle Applications - Graphical User Interface

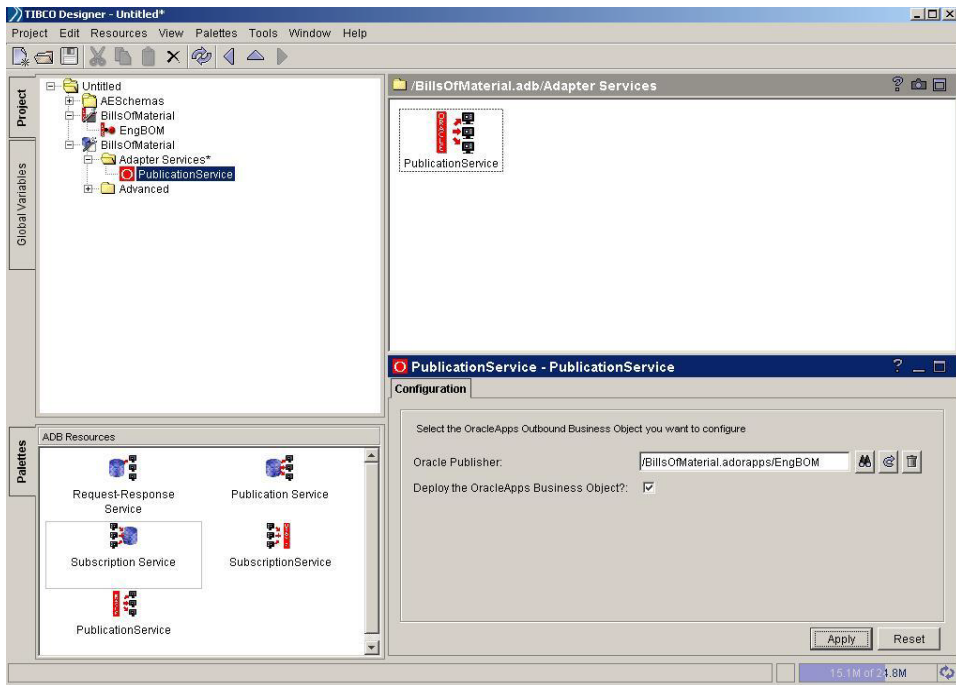
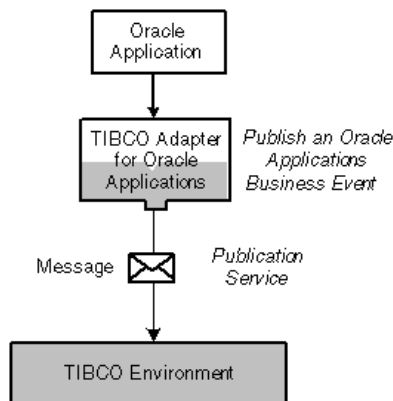


Figure 3 shows TIBCO Designer, TIBCO Adapter for Oracle Applications Business Objects (project panel), and TIBCO ActiveMatrix Adapter for Database services (palette panel). The blue buttons represent services provided by TIBCO ActiveMatrix Adapter for Database. The red buttons indicate TIBCO Adapter for Oracle Applications publication and subscription configuration objects.

Publication Service Overview

The adapter detects changes or creations of Business Objects in the Oracle Applications and publishes them to the TIBCO environment.

Figure 4 Typical Publication Service Flow



The adapter provides PreDefined and Custom Publication services. The predefined publication service publishes from Oracle source tables that are pre-set according to standard Oracle Applications workflow. The custom publication service allows you to configure custom adapter configurations that reflect your business workflow. After configuring the publication service, the adapter will publish any change to the business objects as they occur in the Oracle Applications system.

The current list of PreDefined publications supported in the adapter is shown in [Available Transactions on page 271](#).

The tables, views, triggers, and other items associated with each publish transaction are listed in [Master Mapping Tables on page 357](#).

Database Components

The following database components are required to publish a business object:

- A **materialized view log** (MV Log): MV Log is a table maintained by the Oracle database. It contains a trail of all records that are updated, inserted, or deleted for an associated source table. MV Logs are automatically refreshed by Oracle Applications when a change occurs in the database table.

The adapter uses MV Log tables to detect changes to business objects. For PreDefined Publications, the scripts to create MV Logs are bundled in the installation SQL script provided for the business object.

- An Oracle database **trigger** on an MV Log: The purpose of the trigger is to ensure that any changes to a business object cause an appropriate update to its publishing table. The trigger on the MV Log corresponding to the business object child table (Child MV Log) inserts a record into the Header MV Log whenever there is a change in the Child MV Log.
- A **publishing table**: The publishing table is an interface table that the run-time adapter polls at predefined intervals. The adapter gets notifications inserted by a trigger and publishes the corresponding business object to the TIBCO environment. This table is created at configuration time through TIBCO Designer.

This section describes the components used in publication and describes how publication works for complex objects.

Materialized View Logs

This section describes how the adapter uses materialized view logs (MV Logs) to publish objects. The MV Log is a table maintained by the Oracle database. It contains a trail of all records that are updated, inserted, or deleted from an associated source table. MV Logs are automatically refreshed by Oracle when there is a change in the base table.

Materialized views improve query performance by pre-calculating expensive join and aggregation operations on a database prior to execution time, and storing the results in MV Logs in the database. The query optimizer automatically recognizes when an existing materialized view should be used to satisfy a request. The query optimizer then rewrites the request and directs it to the MV Log instead of the underlying detail tables. The MV Log tables have DMLTYPE\$\$ fields that indicate the type of operation that took place: UPDATE, INSERT, or DELETE. The DMLTYPE\$\$ field will contain value I for INSERT, D for DELETE and U for UPDATE. DML stands for *Data Manipulation Language*, which belongs to the category of SQL statements that query and update the database. Common DML statements are SELECT, INSERT, UPDATE, and DELETE. Please refer to *Oracle SQL*Plus Documentation* accompanying the Oracle Applications setup for further details.

The adapter uses MV Logs directly, bypassing the need to create actual Materialized View tables. An MV Log is created for an object and a trigger is applied to it. Thus, when an object is modified, the trigger copies the object's key data to a publishing table with the appropriate DML code indicating whether it was an update, insertion, or deletion. Meanwhile, the adapter polls the publishing table for changes, and when the key data is inserted, the adapter accesses the object's data and publishes it.

This approach has a number of advantages:

- There is no need for triggers on source tables. (Oracle discourages the use of triggers on Oracle Applications tables).
- Changes are propagated in real-time.
- MV Log creation is very simple.
- Any change to the source table is logged to the MV Log table with the appropriate DML operation code.
- Fields to be included in the MV Log can be specified at creation time.
- Changes in the related header and line tables can be captured and published as a single message.
- MV Log monitors change to INSERT, UPDATE, DELETE on source tables automatically. This is an in-built Oracle database functionality. No external tool is required for monitoring changes.

The adapter uses MV Log tables to detect changes to business objects.

Each MV Log contains the following columns:

- The key fields of the table.
- The columns in the table that need to be tracked for changes.
- Default columns (such as M_ROW\$).

The MV Log for a table is created with the following statement:

```
CREATE MATERIALIZED VIEW LOG ON <TABLE_NAME>
WITH ROWID (Field1, Field2, Field3,...) INCLUDING NEW VALUES
```

where Field1, Field2, and Field3 are columns in the table <TABLE_NAME>.

If field names are specified, the specified fields are added at the beginning of the MV Log table. If fields are not specified, an MV Log is created with the following default fields:

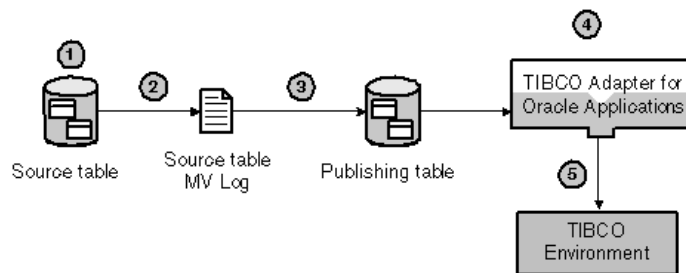
- M_ROW\$\$
- SNAPTMS\$\$
- DMLTYPE\$\$ contains information about the kind of operation that has occurred on the base table:
 - I (Insert)
 - U (Update)
 - D (Delete)

- OLD_NEW denotes whether the current record is the one before the change or after the change on the base table. These values can be:
 - N (New)
 - U (Old for the Update Operation)
 - O (Old for the Delete Operation)
- CHANGE_VECTOR\$\$

Publication Architecture

The following diagram illustrates the architecture for publishing data from a single source table.

Figure 5 Publishing from a Business Object consisting of one table in the Oracle Applications Database



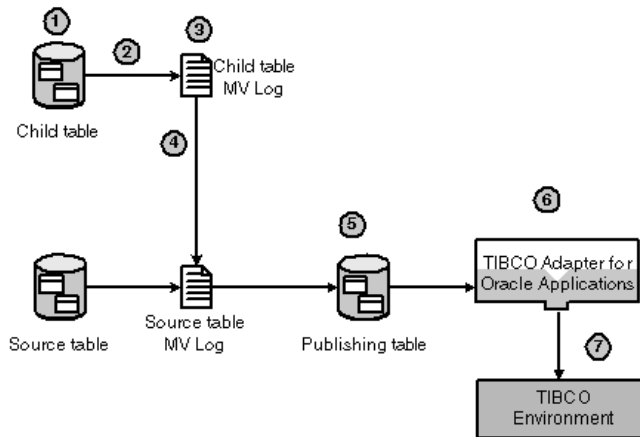
1. A row is inserted, updated, or deleted from the source table due to an operation performed. This can be any operation performed from the Oracle Applications GUI Layer or from SQL*Plus or any manual operation.
2. A corresponding record is automatically inserted into the MV Log of the source table. This automatic insertion is performed by the Oracle database serving the Oracle Applications installation. The DMLTYPE\$\$ column of the MV Log contains a value indicating the kind of operation that was performed on the source table. The record has a value of I for insert in the DMLTYPE\$\$ column and N in the OLD_NEW\$\$ column. The record has a value of D for delete in the DMLTYPE\$\$ column and O in the OLD_NEW\$\$ column. The record has a value of U for update in the DMLTYPE\$\$ column and the OLD_NEW\$\$ column.
3. If the Storage Mode is Publish by Reference, the insert trigger on the MV Log causes the primary key of the changed record to be copied from the MV Log to the publishing table. If the Storage Mode is Publish by Value, entire columns of the changed record are copied from MV log to the Publishing table. However, before the replication, the trigger first ensures that there are

no unpublished update rows in the publishing table with the same key field. This additional check prevents multiple publications with the same data.

4. The run-time adapter retrieves new rows (if any) whenever it polls the publishing table and checks if the status of the column `ADB_L_DELIVERY_STATUS` is N in the publishing table. If the status is N, the adapter publishes its data. You can specify the polling interval.
5. The run-time adapter creates a message and sends it on a pre-configured subject.

Figure 6 illustrates the publishing architecture for a more complex object with two related source tables. In this example, changes to the child are not propagated automatically to the parent table.

Figure 6 The publishing architecture for a more complex object with two related source tables



1. A row is inserted, updated or deleted from the child table.
2. A corresponding record is automatically inserted into the child table MV Log.
3. A trigger is invoked on the MV Log of the child table.
4. The trigger on the child table MV Log causes a record to be inserted into the MV Log of the parent table. The record has a value of U for update in the `DMLTYPE$$` column and the `OLD_NEW$$` column.
5. The insert trigger on the MV Log of the parent table causes the primary key of the changed record to be copied from the MV Log to the publishing table. However, before copying the data, the trigger first ensures that there are no unpublished update rows in the publishing table with the same key field. This additional check prevents multiple publications with the same data.

6. Once a row is written out to the publishing table, the run-time adapter retrieves the row along with any additional records from related tables the next time it polls the publishing table.
7. The runtime adapter creates a message and sends it out on a pre-configured subject.



The Storage Mode `Publish by Value` is not applicable for transactions involving multiple tables. It is applicable only if those transactions involve a single table so that minor updates done on the source table are published.

Publication Configuration

PreDefined Outbound (Publication) Business Object is configured and used for creating a service. Refer to [Configuring the Adapter on page 97](#) for further details on how to configure a PreDefined Outbound Business Object.

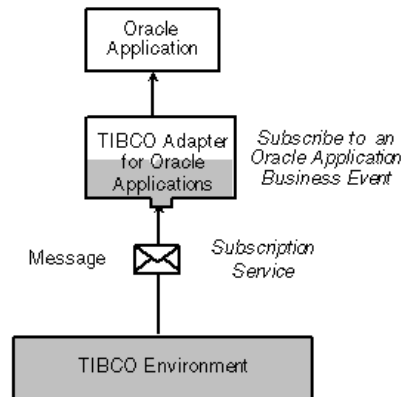
When you configure a publication service, the adapter will automatically perform the following actions on the database.

1. Create separate MV Log for header and detail tables.
2. Create publishing tables.
3. Define the relationship between the parent and child tables.
4. Create trigger on MV Log of detail table to update the MV Log of the header table (if necessary).
5. Create triggers on MV Log of header table to insert data into publishing tables.
6. Create Views of the source tables, if required.

Subscription Service Overview

The adapter can subscribe to complex business objects from the TIBCO environment and insert, update, or delete them in Oracle Applications.

Figure 7 Typical Subscription Service Flow



The adapter provides PreDefined and Custom Subscription services. The PreDefined subscription service subscribes from Oracle source tables that are preset according to standard Oracle Applications workflow. The custom subscription service allows you to customize adapter configurations that reflect your business workflow. After configuring the subscription service, the adapter will be ready to receive messages containing business objects and apply those business objects to the target Oracle Applications system.

The list of PreDefined subscriptions supported in the adapter is shown in [Appendix A, Available Transactions, on page 271](#).

The tables, scripts, sequences, and other items associated with each subscribe transaction are listed in [Appendix C, Master Mapping Tables, on page 357](#).

You can modify the provided validation suite, and can create additional subscriptions for business objects not currently provided by the adapter using Custom Subscription Service.

Database Components

To subscribe to a Business Object, ensure that the following is created:

- The **Intermediate Table** is the table where the adapter inserts the business object received from the TIBCO environment. For PreDefined Subscription Business Objects, the adapter palette generates these tables based on the table names provided with the adapter. For a PreDefined Subscription Business Object based on an Oracle API, the script to generate this table is bundled in the SQL script for installing the business object. The palette thereafter executes these scripts during design time.
- The adapter provides a set of inbound business objects for which the database components, relationships, and business logic are compiled in the form of database tables, properties files, and PL/SQL procedures. These components perform the necessary actions for inbound messages for the Oracle Applications system.
- The **Log Table** is where the validation program logs the errors detected in the incoming message and successful records. It also logs the source from where the data is published. For a PreDefined subscription, this table is created by the SQL installation script for the business object. See [Log Table Structure on page 386](#), for more details on the Log Table Structure.
- The TIBCO PL/SQL program imports data from the TIBCO Intermediate tables. This PL/SQL program is generated by the adapter palette based on the property values supplied by you during design time. The TIBCO PL/SQL Program gives you an option to put custom validation code into user exits (pre and post) before the data is moved from the TIBCO Intermediate tables to the Oracle Interface tables. The structure of the PL/SQL package is available at [Oracle Application Transactions on page 277](#).

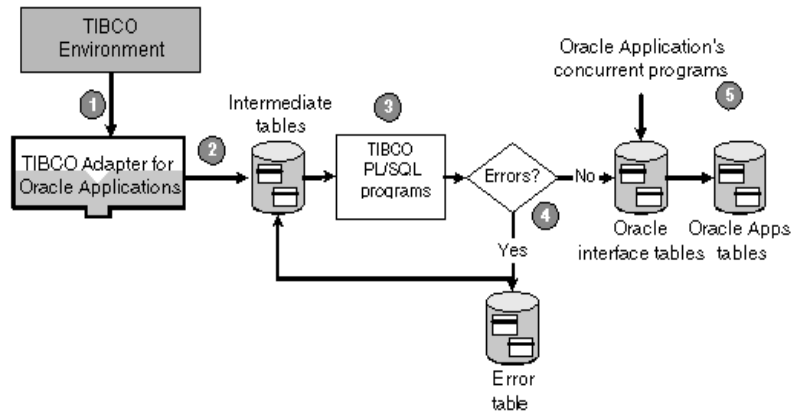
Subscription Architecture

There are two possible subscription scenarios; **simple subscription**, **subscription with reply**. These scenarios are described in the following sections.

Subscription

The subscription process is illustrated in [Figure 8](#).

Figure 8 Subscription Architecture



1. The adapter gets a transaction message from the TIBCO environment.
2. The adapter inserts the data into the intermediate tables.
3. The PL/SQL validation program provided with the adapter validates the data in the intermediate table.
4. Data is processed as follows:
 - a. The incoming data is first moved from the TIBCO Intermediate table to the Oracle Interface table.
 - b. If the insertion into the Oracle Interface table succeeds, then the error flag status becomes 5; else it is set to 1. The reason for the insertion having failed is also logged into the TIBCO Log table.
 - c. The TIBCO PL/SQL program invokes the Applications concurrent program, which copies the data from the Oracle Interface Tables into the Oracle source tables. In case the records error out, then the Oracle error is logged in the TIBCO Log table. In case of a successful insertion into the Oracle source tables, the corresponding success message is also logged in the TIBCO Log table.

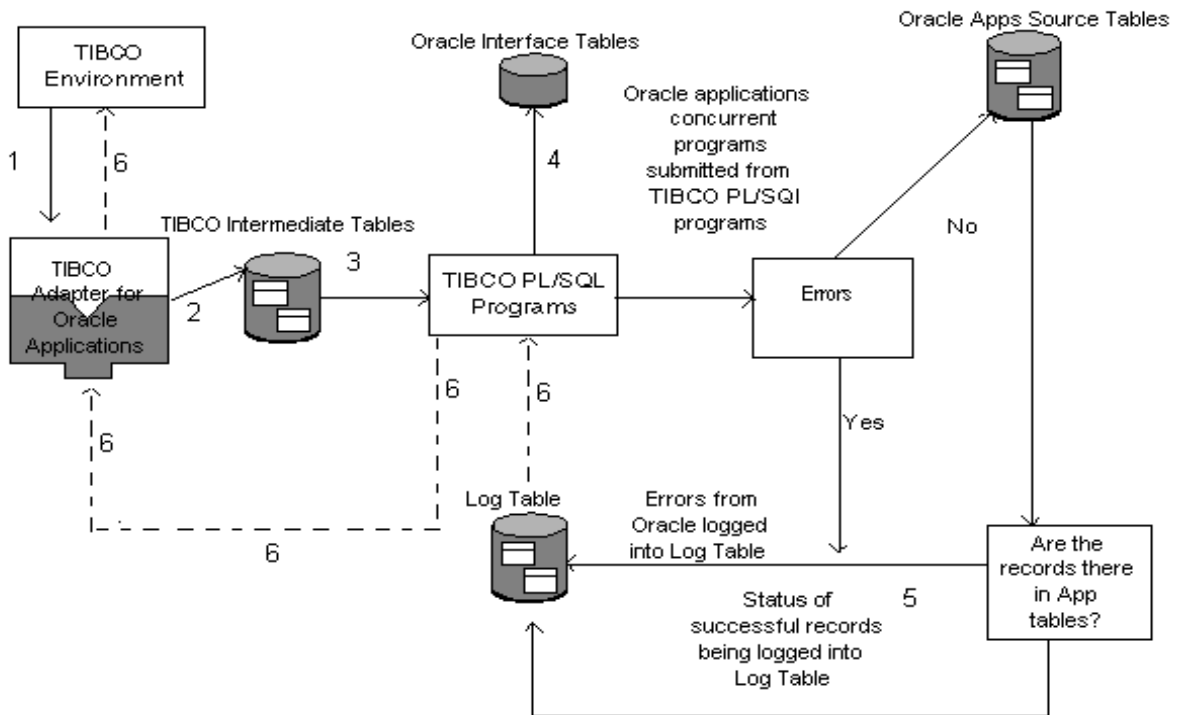


Concurrent programs is an Oracle Applications terminology that signifies managed programs administered by the Oracle Application Concurrent Manager. Each program has several dependencies that are managed and resolved by the *Concurrent Manager*.

Subscription with Reply

The subscription with reply process is illustrated in [Figure 9](#).

Figure 9 Subscription with Reply



1. The adapter gets a transaction message from the TIBCO environment.
2. The adapter then inserts the data into the intermediate tables.
3. The **Subscription with Reply** service invokes the TIBCO PL/SQL Program configured.
4. Data is processed as follows:
 - Records for each TIBCO Intermediate Table are inserted in the corresponding Oracle Interface Table.
 - The Oracle Application Concurrent Program that validates and inserts the data into the Oracle Source Tables is invoked.

5. A check is made on the Oracle Source tables to confirm whether the records have been put in the Oracle Source Tables.
 - If the data is not put into the Oracle Source Tables, an entry for the erroneous record is logged into the Log Table with status as **E**.
 - If the data is present in the Oracle Source Tables, an entry for this record is made into the Log table with status as **S**.
6. The **Subscription with Reply** feature involves confirmation of the message depending on the value of `DO_ROLLBACK` as returned by the `preCommitProcedure`. The TIBCO PL/SQL Programs of the TIBCO Adapter for Oracle Applications will NOT modify this value since the program updates the TIBCO Log Table for both the `Error` and `Success` scenarios. The adapter will therefore confirm the message.

Schema Support

The adapter uses schema to describe the data that the adapter receives from or sends to the TIBCO environment. The TIBCO Adapter for Oracle Applications will read all of the Oracle Application database tables/views and create standard TIBCO schemas for each one of them and store them in the repository. Then, other TIBCO applications will have a standard way of interfacing with the TIBCO Adapter for Oracle Applications.

Schemas are useful in a variety of situations, for example:

- Where several developers collaborate on an adapter application, a specification document normally defines the data model for several related adapter applications. Inside the TIBCO framework, you can define a schema to serve as the data model and update it as needed.
- When the data model of the adapter application changes, developers do not have to redesign the business process.
- Using schemas in conjunction with the various mapping tools available in the TIBCO environment, you can transfer data across applications with incompatible data formats.

The TIBCO Adapter for Oracle Applications palette contains existing Oracle Applications schemas that you can apply to an adapter service.

Before You Start

Before you start configuring the TIBCO Adapter for Oracle Applications software, read Chapter 1, *Introduction*, in the *TIBCO ActiveMatrix Adapter for Database User's Guide*.

Also read the *TIBCO Designer User's Guide* for instructions on using the TIBCO Designer GUI and explanations of terms used throughout this book.

Chapter 2 **Installation**

This chapter describes the procedures for installing and uninstalling the TIBCO Adapter for Oracle Applications on Microsoft Windows and UNIX platforms.



This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. Please see the readme.txt file for the availability of this software version on a specific operating system platform.

Topics

- [Preparing your Environment for Installation, page 26](#)
- [Pre-Installation Worksheet, page 29](#)
- [Installer Overview, page 32](#)
- [Installation Registry, page 35](#)
- [Adapter Components and Compatible Software, page 37](#)
- [Installing on Microsoft Windows, page 39](#)
- [Post-Installation Tasks on Windows, page 44](#)
- [Installation on UNIX Systems, page 63](#)
- [Post Installation Tasks on UNIX, page 68](#)
- [Installation FAQs and Trouble Shooting, page 73](#)

Preparing your Environment for Installation

The most time-consuming part of an adapter installation is the collection of environment information and parameters. This section helps you to complete this process. It provides a parameters checklist you should obtain from various system administrators within your organization before the adapter installation. Note that obtaining an Oracle Applications account can take some time depending on your corporate policies — so plan in advance!

Operating System Requirements

Perform the following tasks on the machine which you plan to install the adapter on:

- Obtain the credentials to run the installer on your machine.
 - On Microsoft Windows, it is essential to install administrator privileges.
 - On UNIX systems, you can install as root or a regular user. See [Installer Account on page 39](#) for details.
 - Note that the TIBCO Runtime Agent (TRA) must be installed prior to the adapter installation. The adapter installation always places files under the TIBCO root directory that was set when the TIBCO Runtime Agent was installed.
- Ensure that you have enough disk space in your temp area and the directory where the adapter is to be installed.
 - See [Installation Registry on page 35](#) for details about temp folder space requirements on Microsoft Windows and UNIX systems.
 - See [Table 5 on page 39](#) for Microsoft Windows installations.
 - See [Table 8 on page 63](#) for UNIX System installations.

You must have write permissions to these directories in order to install the adapter. In addition, on UNIX systems some permissions must be set to run the adapter. See [Permission Requirements on UNIX Systems on page 68](#) for details.

- Ensure that you have permissions to access the project where the adapter configuration is stored.
- Depending on whether TIBCO Administrator is used to set access permissions, you may need an account identified by an Administrator. See the *TIBCO Administrator User's Guide* for details.

- Determine how the adapter installation files are transferred to your system. The installation files can be downloaded from download.tibco.com (if you have set up an account to download).

Oracle Application Requirements

Determine which Oracle Applications versions are supported.

- See [Supported Oracle Applications on page 39](#) for Microsoft Windows.
- See [Supported Oracle Applications on page 63](#) for UNIX systems.

Obtain the following information from your Oracle Applications administrator. This is the Oracle Applications server parameters list that you will need to configure the adapter, along with the Oracle Applications credentials that the adapter requires.

Note that the parameters listed here use Oracle Applications terminology. Normally, you will be prompted with this information during an Oracle Applications client installation.

Test Connectivity

If you install the Oracle Applications client, start it and ensure you can connect to the Oracle Applications server with the username and password that you were provided by your administrator.

Oracle JPublisher

The adapter requires the output of the Oracle Applications Inbound Business Object (R12 only). To use this service, SQL wrapper files generated from Oracle's JPublisher must be made accessible to the adapter during configuration. For information on using Oracle JPublisher to generating SQL wrapper files, refer to [Appendix O, Tips on Using Oracle JPublisher, on page 503](#).

Oracle JPublisher can be installed on a system with access to the Oracle e-Business Suite server. The system can be on the machine where the TIBCO Adapter for Oracle Applications is installed or another machine that has access to the Oracle e-Business Suite server.

Oracle JPublisher is a tool provided by Oracle through the Oracle Technology Network. It is provided through Oracle Java Tools offerings in the JDBC/Java in the Database focus area. Through Oracle JPublisher, users can obtain information about Oracle's stored procedures and packages in different formats, such as SQL script, XML, and Java classes. For more information about Oracle JPublisher, visit Oracle's web site and search for JPublisher.

The configuration of the Oracle Inbound Business Service (R12 Only) in Chapter 4 describes the way for using TIBCO Adapter for Oracle Applications along with Oracle JPublisher.

For information on downloading and installing Oracle JPublisher, please visit Oracle's Technology Network web site, search for JPublisher, and follow the instructions for download and installation.

Pre-Installation Worksheet

Use this form to record the information you need prior to the Oracle Applications installation.

Adapter Machine Information

Field Name	Field Description	Field Value
Hostname (Example: adapter1.tibco.com)	Name of the machine which the adapter is installed on.	
IP address (Example: 192.168.12.12)	IP address of the machine which the adapter is installed on.	
User account (Example: administrator)	User account used for the installation.	
User domain (if Windows) (Example: ENGR2)	Network domain to which the user belongs.	
User password	User password	
Disk and path on which to install adapter (Example: /opt/tibco)		
Access mode	How will machine be accessed	<input type="checkbox"/> directly <input type="checkbox"/> terminal server <input type="checkbox"/> xterm <input type="checkbox"/> telnet <input type="checkbox"/> other:

Field Name	Field Description	Field Value
Transfer mode	How the installation files will be transferred to the machine.	<input type="checkbox"/> CD-drive <input type="checkbox"/> Internet download <input type="checkbox"/> FTP to machine <input type="checkbox"/> network disk mounting

Oracle Applications Information

Field Name	Field Description	Field Value
User account (Example: jsmith)	User account used for the installation.	
User password	Password for the above account.	
Application server hostname (Example: Oracleserver.tibco.com)	Name or IP address of the machine hosting the Oracle Application Server.	
Listener Port (Example: 1573)	The database port on which Oracle Application is listening.	
JDBC Driver Name (Example: tibcosoftwareinc.jdbc.oracle.OracleDriver)	Name of the JDBC driver.	
Server ID (SID) (Example: SID=ORCL)	Name of the Server Id to which your Oracle Application is connected.	
URL	URL to access the Oracle Application.	jdbc:tibcosoftwareinc:oracle://<servername>:1521;SID=ORCL

Oracle Applications Software

Specify the location of the software for Oracle Applications within your organization:

☐ It's already installed on adapter machine.

Location on disk:

☐ Installation files are available via :

FTP (Server: User: Password:)

Disk mount (full path:)

CD provided during install by (name/extension):

Installer Overview

The installer allows you to run in different modes. Each mode is supported on all platforms.

- GUI mode
- Console mode
- Silent mode

GUI Mode

In GUI mode, the installer presents panels that allow you to make choices about product selection, product location, and so on. When you invoke the installer by double-clicking the icon, GUI mode is used.

Console Mode

Console mode allows you to run the installer from a command prompt or a terminal window. This is useful if your machine does not have a Windows environment.

Silent Mode

Silent installation mode either uses default settings or a response file that was saved during an earlier installation. No information is required during this mode.

- If no response file has been recorded earlier and you have invoked the installer with the `-silent` argument, the default installation parameters will be used.
- If a response file exists, and the installer is started with `-options <responseFileName>` as an argument, the installer uses the values specified by the user when the response file was generated.

Upgrading an Adapter

Software from TIBCO uses three numbers to indicate whether the release is major, minor or a patch. For example, 5.0.0 indicates a major release, 5.2.0 indicates a minor release, and 5.2.1 indicates a hotfix release. A hotfix release performs an automatic upgrade by unzipping files contained in it. It then overwrites all the contents present in 5.2 lib directory.

For a major and minor release, the installer prompts whether you wish to upgrade, and informs you if incompatible products are on your system. If you proceed, major or minor releases are installed under a new directory that is named using the major or minor release numbers.

For example, if you have installed the 5.1.0 release and are upgrading to a 5.2.0 minor release, it will be installed under the 5.2 directory. This allows both the 5.1 and 5.2 releases to coexist on the same machine.

If you are upgrading the adapter, or reinstalling a clean version of the software, you can uninstall the product first or allow the installer to perform the upgrade or reinstall.

Note that, if you are reinstalling over the same adapter version:

- You are not prompted to supply the installation location. The software is automatically reinstalled where the previous version was installed.
- If any files are currently locked (that is, in use), the installer marks the file for deletion in the install location. After installation, the installer prompts you to reboot your system. You must reboot before using the software.

Uninstalling the Adapter

If another product is dependent on the product you want to uninstall, you will be informed that you must uninstall the other product first.

Microsoft Windows

Use one of the following options to uninstall the Adapter from the Microsoft Windows platform:

- Click **Start > Programs > TIBCO > TIBCO Adapter for Oracle Applications > Uninstall**.
- Navigate to the `_uninst` directory located in the Adapter installation directory and invoke the `Tibuninstall.exe` program.
- Click **Start > Programs > TIBCO > TIBCO Installation Manager**.
- Select **Add/Remove Programs** from the Control Panel.

UNIX

Use one of the following options to uninstall the Adapter from the supported UNIX platform:

- Navigate to the `_uninst` directory located in the Adapter installation directory and invoke the `Tibuninstall.bin` program.

- Run TIBCO Installation Manager which is located in the `<install-path>/tibco/TibcoInstallationManager.bin` directory.

Installation Registry

The installer maintains an installation registry. The registry location depends on the platform. This section explains where the registry files are located. The files have a prefix named vpd (Vital Product Database). Note that the installer does not recognize TIBCO ActiveEnterprise 4.x products.



Do not edit, modify, rename, move, or remove any of the registry vpd files.

Microsoft Windows Platforms

ActiveEnterprise 5.1 products maintain the installation registry in the SystemDrive:\Windows directory. The following files represent the installation registry:

```
SystemDrive:\Windows\vpd.properties
SystemDrive:\Windows\vpd.properties.tibco.systemName
```

Installer Disk Space Requirements in Temporary Area

The entire package is extracted into a temp folder, typically the SystemDrive:\Temp folder or SystemDrive:\Documents and Settings\\Local Settings\Temp folder.

The installer requires 54 MB of free space in the temp directory.

UNIX Platforms

The installation registry is maintained in the following files in the user's home directory:

```
User_Home_Directory/vpd.properties
User_Home_Directory/vpd.properties.tibco.systemName
```

If the installation is performed by a super-user (root), the installation registry is maintained as the following:

- On Solaris and HP-UX, the root user's home directory (which is /) contains four vpd files.
- On AIX, the /usr/lib/objrepos directory contains two vpd files.
- On Linux, the /root directory contains two vpd files.

Installer Disk Space Requirements in Temporary Area

The installer launcher first extracts a JVM (Java Virtual Machine) in a temporary directory and uses this JVM to launch the installer GUI. The size of the extracted JVM differs from platform to platform.

On UNIX platforms, the following disk space is required in the temporary area:

- On Solaris, 77 MB of free disk space in /var/tmp
- On HP-UX, 132 MB of free disk space in /var/tmp
- On AIX, 102 MB of free disk space in /var/tmp
- On Linux 102 MB of free disk space in /var/tmp

If your system does not have sufficient free disk space in the above temporary area, you can still run the installer with a different temporary area by using the following option when starting the installer:

```
TIB_adorapps-simple_<version_num>_s4_57_CC.bin -is:tempdir
/new_tmp
```

if /new_tmp has sufficient free disk space.

Disk Space Requirement in User's Home Directory

When a regular (non-root) user installs a TIBCO product on UNIX platforms, the installation registry (two vpd files) is maintained in the user's home directory. As more products are installed, entries are added into these vpd files.

The user's home directory must at least have 500 KB of free disk space.

Installation History

The installer and uninstaller create a file called `TIBCOInstallationHistory.xml` in the same location where the installation registry is created. Each time an installation and uninstallation is performed, entries are appended to the file.

On Microsoft Windows:

```
SystemDrive:\WINNT\TIBCOInstallationHistory.xml
```

On UNIX: `Users_Home_Directory/TIBCOInstallationHistory.xml`

The file `TIBCOInstallationHistory.xml` therefore contains a record of all installation and uninstallation activities of all products, features and components.



Do not edit, modify, rename, move, or remove the `TIBCOInstallationHistory.xml` file.

Adapter Components and Compatible Software

In the Typical installation mode, both the design time and runtime components are installed on one machine. If you want to install the design time and runtime components on separate machines, you must install the adapter in the Custom mode. For example, you can run the runtime adapter on one machine and install the design-time components on another machine. This allows you to configure an adapter on one machine and run it on another.

Adapter Components

The next table describes the adapter components on the adapter installation package.

Table 3 TIBCO Adapter components

Component	
Run-time adapter	TIBCO Adapter for Oracle Applications run-time engine publishes data into or subscribes data from the tables specified by the TIBCO Adapter for Oracle Applications.
Adapter palette	TIBCO Adapter for Oracle Applications provides a GUI component (known as a <i>palette</i>) to create Business Objects.

Required and Optional TIBCO Products

Depending on the tasks you want to perform, you may need to install other relevant TIBCO products. [Table 4](#) describes required and optional products and their purposes. See the *Readme.txt* file for the supported versions.

Table 4 Required and Optional TIBCO Products

Component	Purpose
TIBCO Runtime Agent	<p>Required. TIBCO Runtime Agent supplies a number of TIBCO and third-party libraries used by the adapter and other TIBCO products both at design time and runtime. This includes, for example, TIBCO Rendezvous software.</p> <p>You must install TIBCO Runtime Agent on each machine that hosts an adapter. TIBCO Runtime Agent must be installed before you install the adapter.</p>

Component	Purpose
TIBCO Administrator	<p>Required. TIBCO Administrator includes the following modules:</p> <ul style="list-style-type: none">• User Management. Management of authentication, roles and users, that is, connecting roles (groups) and users to access control lists (ACLs). This includes security for server-based projects at design time and for deployed applications at runtime.• Resource Management. Monitoring of machines and of all running applications in a TIBCO administration domain. Alerts can be created, for example, to notify an administrator if the number of processes or disk usage exceed a certain number.• Application Management. Uploading of Enterprise Archive (EAR) files, creation, configuration, deployment, and monitoring of applications. This console is also used to start and stop applications. <p>TIBCO Administrator is available as a separate installation and can be installed after installing the adapter.</p>
TIBCO BusinessWorks	<p>Optional. TIBCO BusinessWorks is a scalable, extensible, and easy to use integration platform that allows you to develop integration projects. TIBCO Adapters are typically part of integration projects created using BusinessWorks.</p> <p>TIBCO BusinessWorks is available as a separate installation and can be installed after installing the adapter.</p>

Installing on Microsoft Windows

Before starting the installation procedure, review the topics in this section to determine if your system meets the basic requirements and all the prerequisite software is installed.

Memory requirement for the adapter installation is 256 MB.

[Table 5](#) is a list of adapter installation prerequisites for Microsoft Windows systems. See [Installer Disk Space Requirements in Temporary Area on page 35](#) for additional disk space requirements.

Table 5 Supported platforms, package names, and disk space for Microsoft Windows

Platform	Hardware Platform	Package Names	Disk Space
Microsoft Windows XP Professional	x86	TIB_adorapps-simple_<version_num>_w32.exe	84.8 MB
Microsoft Windows Server 2003	x86	TIB_adorapps-simple_<version_num>_w32.exe	84.8 MB

Supported Oracle Applications

The adapter supports the Oracle Applications 11i (11.5.x) family and Oracle Database Server 9i and 10g.

TIBCO Runtime Agent Must be Installed Before the Adapter

Before you can install the adapter, you must install TIBCO Runtime Agent. If you choose the typical installation mode for TIBCO Runtime Agent, the installer places all libraries and other products required by the adapter into the TIBCO HOME directory.

During the installation, the adapter installer checks the availability of all required products in the system. The installer does not proceed with the installation if any of these are not available.

Installer Account

You must have administrator privileges for the machine on which the adapter is installed.

If you do not have administrator privileges, the installer will exit. You must then log out of the system and log in as a user with the required privileges, or request your system administrator to assign the privileges to your account.

Installing from Network Drive

If you intend to install the product on a network drive, you must ensure that the account used for installation has permission to access the network drive.

Installing on Windows Server 2003 Terminal Server

There are two modes in Windows Terminal Server: *Execute* and *Install*. Users are logged on by default in *Execute* mode, which allows them to run applications. To install an adapter so that everyone can use it, log on as administrator in *Install* mode. When the adapter is installed in the *Install* mode, the installation registry is maintained in *SystemDrive:\WINNT*.



Windows Terminal Server must run in Remote Admin mode, not Application Sharing mode. The adapter is not supported on the machine if it is using Windows Terminal server in Application Sharing mode.

The best way to install the adapter on Windows Terminal Server is to use the Add/Remove Programs control panel applet. This automatically sets your mode to *Install* during the installation and sets the mode back to *Execute* afterwards. Alternatively, you can manually change your mode to *Install* before starting the installation by typing the following at a command prompt:

```
C:\> change user /install
```

To switch to the *Execute* mode:

```
C:\> change user /execute
```

You can check your server mode by typing the following command:

```
C:\> change user /query
```

If the installation is performed in the *Execute* mode, the installation registry is maintained in your user home directory. If the installation is performed in the *Install* mode, the installation registry is maintained in the *SystemDrive:\WINNT* folder.

Installing the Adapter on Microsoft Windows

You can either download the adapter package or install the package from a CD. The installer prompts you to accept the license agreement, then to choose to perform a typical install or custom install.

- A typical install has minimal prompts and installs standard components in default locations.
- A custom install prompts you to choose which components of the product suite to install and installs only those components.

The installer checks your system for the installation home directory that was established when TIBCO Runtime Agent was installed. The adapter is installed under the installation home directory. Use one of the following modes to install the software.

Install Using GUI Mode

GUI Mode allows you to input values in panels. Double-click the following executable:

```
TIB_adorapps-simple_<version_num>_w32.exe
```

Install Using Console Mode

Console mode allows you to install the software from a command line. The installer will prompt you for values. Type the following at the command prompt:

```
TIB_adorapps-simple_<version_num>_w32.exe -is:javaconsole -console
```

When running in console mode you can move through the installation process as described below:

```
Enter Key or 1 = Moves forward in the installer
2 = Goes back to previous screen
3 = Cancels the Wizard and exits the installation or uninstallation
4 = Redisplays the current screen
```

Install Using Silent Mode

Silent mode allows you to install the software without any prompts. Type the following at the command prompt:

```
TIB_adorapps-simple_<version_num>_w32.exe -silent
```

Install and Generate a Response File

You can generate a response file during any installation of the product and can use the same file in future installations. For all installation modes using response file, the options in the file determines what will be installed.

To install and generate a response file, type the following at the command prompt:

```
TIB_adorapps-simple_<version_num>_w32.exe -options-record
C:\directory\<responseFile>
```

Install Using a Response File

You can use a previously generated response file for installation. For all installation modes, the response file determines what will be installed.

To install using a response file, type the following at the command prompt:

```
TIB_adorapps-simple_<version_num>_w32.exe -options
C:\directory\<responseFileName> -silent
```

Combining Options

You can combine the different available options. For example, to install in silent mode using a response file, use:

```
TIB_adorapps-simple_<version_num>_w32.exe -silent -options
<responseFileName>
```

To install using Console mode and generate a response file, use:

```
TIB_adorapps-simple_<version_num>_w32.exe -is:javaconsole -console
-options-record <responseFileName>
```

Clients, Platforms, and Drivers

This section identifies the drivers and clients supported by Windows platforms. The following table lists the supported platforms for the Oracle client, along with the required ODBC and JDBC drivers for each platform.

Table 6 Oracle Client Matrix for Microsoft Windows Platforms

Platform	Oracle 9i	Oracle 10g
Windows XP Professional	yes	yes
Windows Server 2003	yes	yes

Table 7 Oracle Driver Matrix for Windows Platforms

Connectivity	Oracle 9i	Oracle 10g
ODBC	Connect 5.10	Connect 5.10
JDBC	Connect JDBC 3.1	Connect JDBC 3.1

The adapter has been fully tested and certified against the JDBC and ODBC drivers listed in this section. Consult with your TIBCO representative before using the adapter with a driver not specifically listed here.

JDBC Drivers

TIBCO Adapter for Oracle Applications uses JDBC to access your database during design time. The correct JDBC driver is supplied with the TIBCO Adapter for Oracle Applications installation package. When configuring TIBCO Adapter for Oracle Applications, specify the JDBC URL so the correct driver will be automatically loaded.

ODBC Drivers

The TIBCO Adapter for Oracle Applications installation package includes the required ODBC drivers for use during runtime.

In addition, the adapter supports the following drivers that are provided separately by the various database vendors.



Consult with your TIBCO representative before using TIBCO Adapter for Oracle Applications with an ODBC driver not specifically listed here.

- **ODBC Drivers for Oracle** All of the Oracle databases listed in [Table 6](#) can use the ODBC driver included on the Oracle CD for that particular Oracle software version, or the DataDirect Connect 5.10 ODBC driver.

ODBC Driver
Documentation

The *DataDirect Connect ODBC Installation Guide* and *DataDirect Connect ODBC Reference* are available in the *install-path/odbc/books/odbc* directory.

Post-Installation Tasks on Windows

Complete the following tasks after installation:

1. [Configure the Service Name](#)
2. [Configure the Database Account](#)
3. [Create an ODBC System Data Source](#)

Configure the Service Name

The service name must be configured using Oracle Net 8 Easy Config:

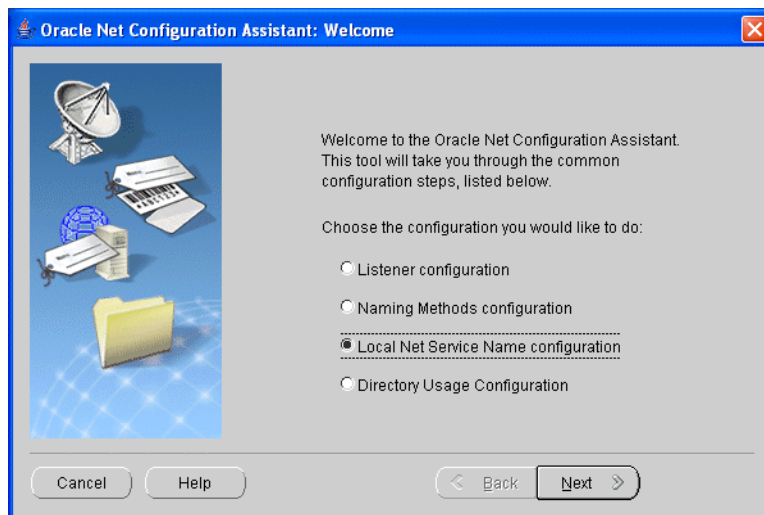
1. Click **Start > Programs > Oracle - OraDb10g_home1 > Configuration and Migration Tools > Database Configuration Assistant**.



These menu options are different for different versions of Oracle. Click **Start > Programs**, select the Oracle installation shortcut for your version of Oracle, and then look for **Net Configuration Assistant**.

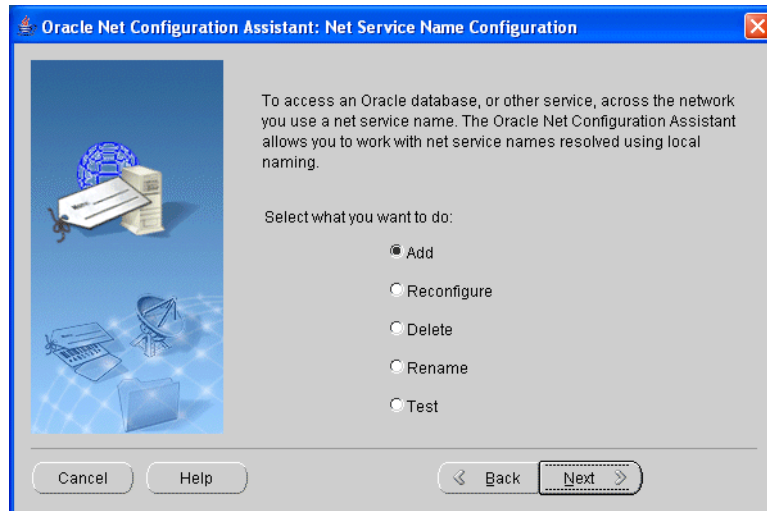
2. Select the **Local Net Service Name configuration** radio button from the configuration assistant's welcome page (see [Figure 10](#)). Click the **Next** button to continue.

Figure 10 Oracle Configuration Assistant's Welcome Page



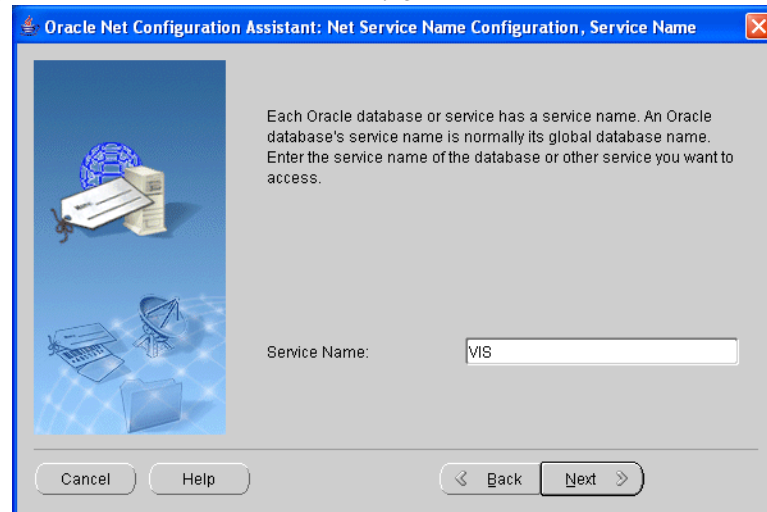
3. Select the **Add** option as shown in [Figure 11](#). Click the **Next** button to continue.

Figure 11 The Net Service Name Configuration Screen



4. Input a name for the service (see Figure 12), then click the **Next** button to continue.

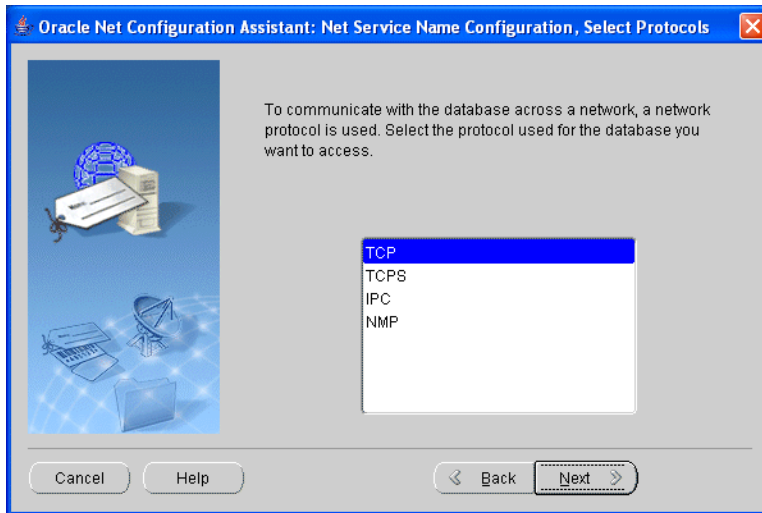
Figure 12 The Net Service Name Configuration, Service Name Screen



The name of the service is not a standard one and will change depending upon your installation. From your System Administrator, find out the name of the service for the database that you intend to use.

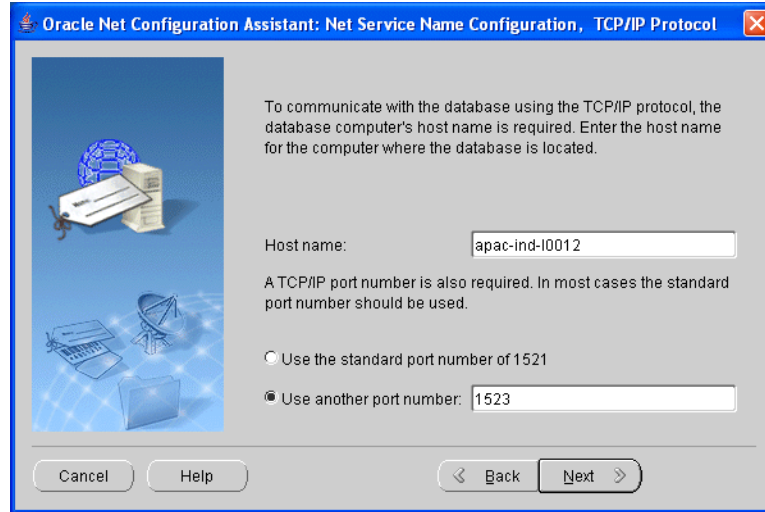
5. Select the **TCP** option as the Protocol option then click the **Next** button to continue (see [Figure 13](#)).

Figure 13 The Net Service name Configuration, Select Protocols Screen



6. Input the host name and the port number of the machine where the database is located. Click the **Next** button to continue (see [Figure 14](#)).

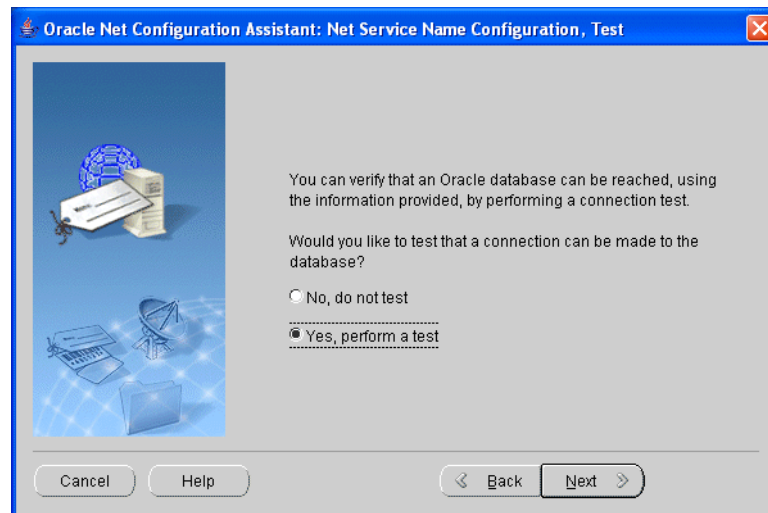
Figure 14 The Net Service Name Configuration, TCP/IP Protocol Screen



The host name and port number will change depending on your installation. Find out the hostname and port number of the machine where the database is installed from your System Administrator.

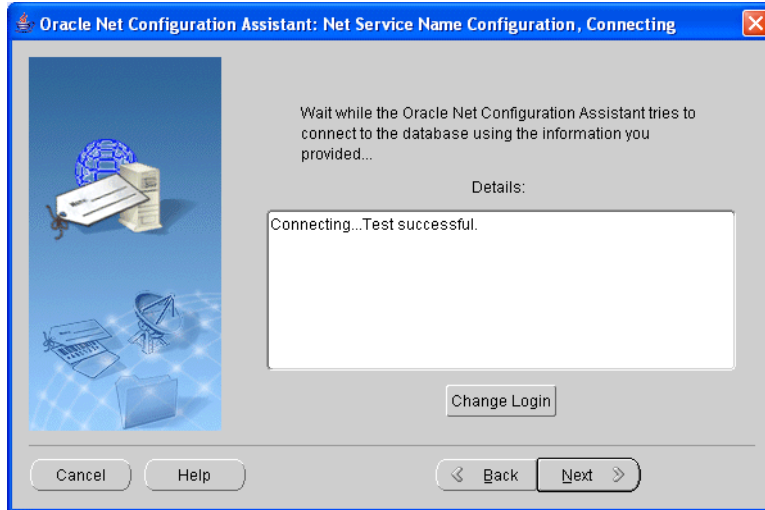
7. Select the **Yes, perform a test** radio button and click the **Next** button to check if the connection is established successfully (see [Figure 15](#)).

Figure 15 The Net Service name Configuration, Test Screen



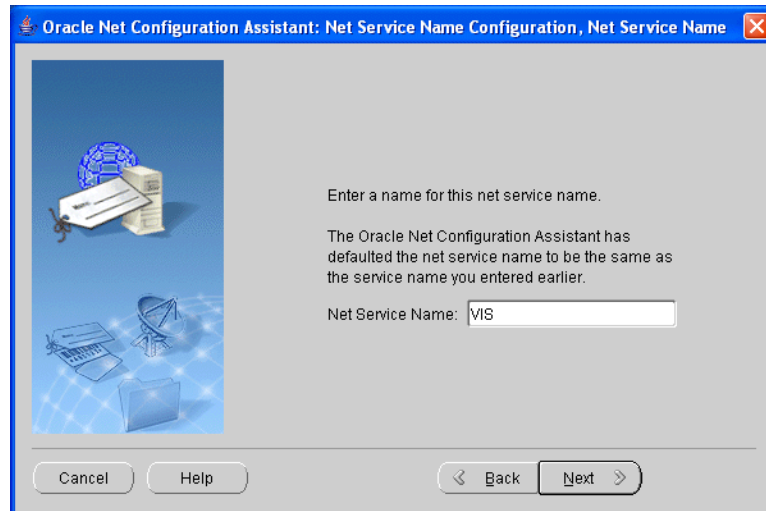
8. The Net Service Name Configuration Assistant will display Connecting...Test successful (as shown in [Figure 16](#)) if the connection is successful. If the connection is not successful, you can click the **Change Login** button to change the login details. After a successful connection, click the **Next** button to continue.

Figure 16 The Net Service Name Configuration, connection Screen



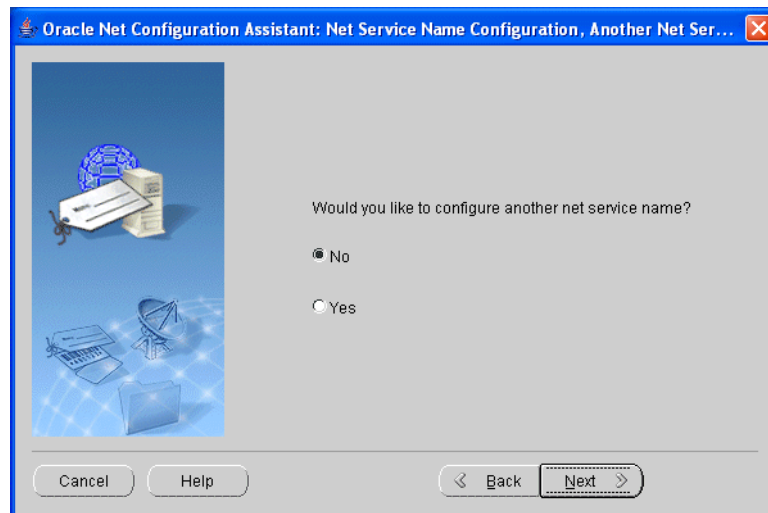
9. The default name displayed in the Net Service Name field is the one that you specified earlier (see [Figure 17](#)). However, you can change the name of the service. Click the **Next** button to continue.

Figure 17 The Net Service Name Configuration, Net Service Name Screen

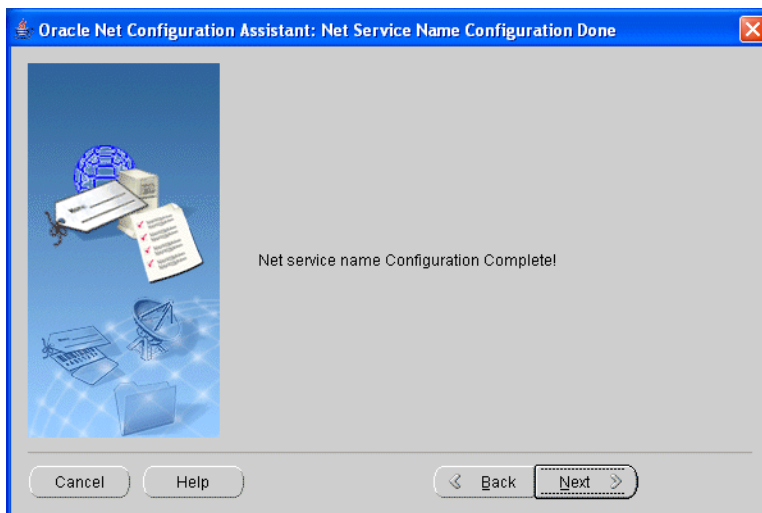


10. Select the **No** option if you do not want to configure another service (see Figure 18). Click the **Next** button to continue.

Figure 18 The Net Service Name Configuration, Another Net Service Configuration Screen



11. The Net Service Name Configuration Done screen (as shown in Figure 19) will appear. Click the **Next** button to continue.

Figure 19 The Net Service Name Configuration Done Screen

12. The Welcome dialog displays again as shown in [Figure 20](#). Click the **Finish** button to exit if you do not want to configure another service or select an option to start another configuration.

Figure 20 The Oracle Net Configuration Assistant's Welcome Screen

Configure the Database Account

The following SQL scripts are provided to set up the database account used by the adapter.

- `create_user.sql`

The `create_user.sql` script needs to be launched as the **APPLSYS user** (for example: `applsyst/apps@vis`). Once connected, it creates a user using the name that you will be prompted for.

This script is located in the `<ADORAPPS_HOME>\config\sub` folder.

- `common_all.sql`

The `common_all.sql` script does the following:

- Connects to the database as the **adapter user** created using the `create_user.sql` script and creates a log table, `TIB_INT_LOG_SUB`, and a sequence.
- Connects to the database as the **sysdba user** and provides grants to `SYS.MLOG$`.
- Connects to the database as the **Oracle Applications database user** and grants permissions on the required objects to the adapter user created using the `create_user.sql` script. It also creates the required synonyms and package, `pk_tib_log`, which handles the logging of the status into the log table.

This script is located in the `<ADORAPPS_HOME>\config\sub` folder.

- `alerter.sql`

The `alerter.sql` script sets up the tables used by the Oracle alerter. You will need to use this script only if you plan to use the Oracle alerter.

This script is located in the `<ADORAPPS_HOME>\adb\config\oracle` folder.

Using the create_user.sql Script

The `create_user.sql` script needs to be launched as the **APPLSYS user** (for example: `applsyst/apps@vis`). Once connected, it creates a user account with the username and password that you set up. In addition, several TIBCO database objects are created for this new user.

The account created using this script is used to connect to the database, store catalog tables, and access tables created when configuring an adapter.

Executing the create_user.sql Script

1. Open a command window and change to the following directory:

```
cd <ADORAPPS_HOME>\config\sub
```

For example:

```
cd c:\tibco\adapter\adorapps\<version_num>\config\sub
```

2. Connect to the database as the APPLSYS user:

```
c:\tibco\adapter\adorapps\<version_num>\config\sub> sqlplus  
applsys/apps@vis
```

3. Execute the create_user.sql script:

```
SQL> @create_user.sql
```

4. Choose a username and a password that you want to use with the adapter and enter these values when prompted. In this example, the value **test** has been chosen.

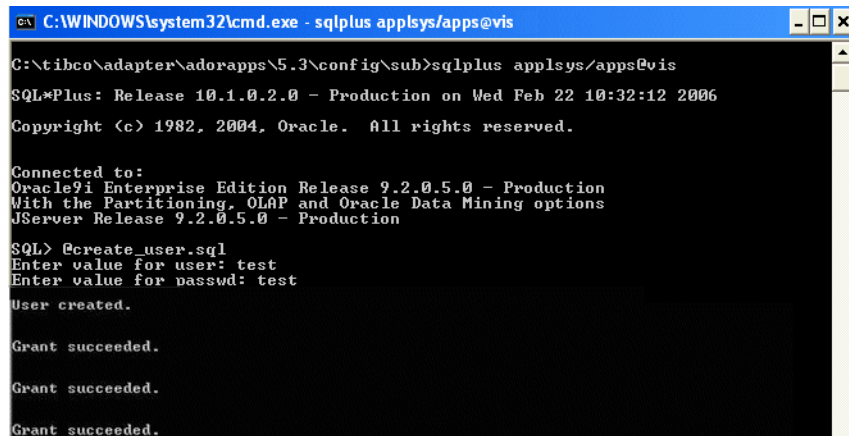
```
Enter value for user: test
```

```
Enter value for password: test
```

A user with the provided values is created.

Figure 21 shows an example of the prompts and responses. Note that the responses vary depending on the machine.

Figure 21 create_user.sql: Prompts and Responses



```
C:\WINDOWS\system32\cmd.exe - sqlplus applsys/apps@vis

C:\tibco\adapter\adorapps\5.3\config\sub>sqlplus applsys/apps@vis
SQL*Plus: Release 10.1.0.2.0 - Production on Wed Feb 22 10:32:12 2006
Copyright (c) 1982, 2004, Oracle. All rights reserved.

Connected to:
Oracle9i Enterprise Edition Release 9.2.0.5.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.5.0 - Production

SQL> @create_user.sql
Enter value for user: test
Enter value for passwd: test
User created.

Grant succeeded.

Grant succeeded.

Grant succeeded.
```

Using the common_all.sql Script

The `common_all.sql` script does the following:

As an Adapter User

1. Run the `common_all.sql` script, the system will prompt you for the user ID and password of the **adapter user** created using the `create_user.sql` script (For example: `test`).
2. Connects to the database (For example: `test`).
3. Creates a log table, `TIB_INT_LOG_SUB`, in the **adapter user** schema.



Before creating the log table, the script checks for the existence of the log table. If a log table already exists, the script checks for the existence of the `TIBCO_SOURCE` and `TIBCO_SOURCE_ID` columns in the table. This check is done, specifically, for users migrating from adapter versions prior to 5.2.0 as these columns were added in version 5.2.0.

4. Creates a sequence.
5. Disconnects from the database.

As a Sysdba User

1. Prompts you for the **sysdba user** ID and password.

Based on the configuration at your site, you can enter any combination of username and password that has **sysdba user** privileges. Contact your system administrator for the ID and password at your site or use one of the default **sysdba user** IDs: `internal`, `sys`, or `system`. The default password for these user IDs is `manager`.

2. Connects to the database and provides grants to `SYS.MLOGS$`.
3. Disconnects from the database.

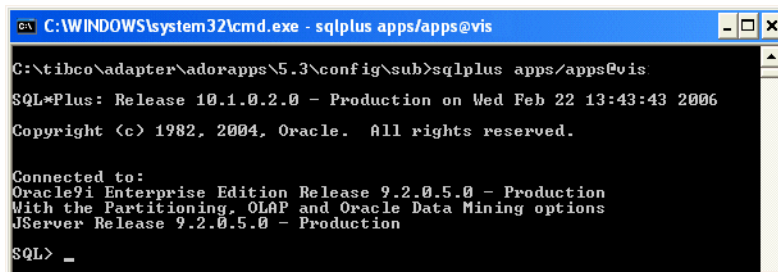
As an Oracle Applications Database User

1. Prompts you for the **Oracle Applications database user** ID and password. (For example: `apps/apps`). You are not prompted for the `tnsname` (`db_connectstring`) again. The same `db_connectstring` that you specified when you were prompted for the **adapter user** details is used.
2. Connects to the database.

3. Grants permissions on the Oracle Applications database objects to the **adapter user (test)**. The following grants are given:
 - Grant `SELECT` for the Interface tables in case of Subscription transactions.
 - Grant `SELECT` for the Source tables and for tables that are part of the Views to be defined, in case of the Publication transactions.
 For example: `GRANT SELECT ON AR_PAYMENTS_INTERFACE_ALL TO &&user;`
4. Creates required synonyms.
5. Creates a package, `pk_tib_log`, which handles the logging of the status into the log table.
6. Disconnects from the database.

Executing the `common_all.sql` Script

1. Open a command window and change directory as follows:
`cd <ADORAPPS_HOME>\config\sub`
2. Connect to the database. For example:
`c:\tibco\adapter\adorapps\<version_num>\config\sub> sqlplus apps/apps@vis`



```

C:\WINDOWS\system32\cmd.exe - sqlplus apps/apps@vis

C:\tibco\adapter\adorapps\5.3\config\sub>sqlplus apps/apps@vis
SQL*Plus: Release 10.1.0.2.0 - Production on Wed Feb 22 13:43:43 2006
Copyright (c) 1982, 2004, Oracle. All rights reserved.

Connected to:
Oracle9i Enterprise Edition Release 9.2.0.5.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.5.0 - Production
SQL> _
  
```

3. Execute the `common_all.sql` script on the Oracle database before configuring the TIBCO Adapter for Oracle Applications Business Objects. The script needs to be executed only once after the adapter is installed.
`SQL> @common_all.sql`

4. When prompted for the **adapter user** details, enter the following:
 - **user_schema_userid**: user created using the `create_user.sql` script (**test** in our example).
 - **user_schema_password**: password created using the `create_user.sql` script (**test** in our example).
 - **db_connectstring**: The database connection string (**vis** in our example).
 - **table_tblspace**: name of the tablespace on which the database objects used by the **adapter user** are to be created. (We have used the default tablespace **user_data** in our example).

Note: This tablespace is specific to your installation of Oracle Applications. Contact your Oracle Applications database administrator to get the name of your tablespace or use the default tablespace **user_data**.

```
SQL> @common_all.sql
Enter value for user_schema_userid: test
Enter value for user_schema_password: test
Enter value for db_connectstring: vis
Connected.
Enter value for table_tblspace: user_data

Table created.

Sequence created.

Commit complete.
```

5. When prompted for the **sysdba user** details, enter the following:
 - **sysdba_schema_userid**: **sysdba user** ID. Contact your system administrator for the **sysdba user** ID or use one of the defaults: **internal**, **sys**, or **system**.
 - **sysdba_schema_password**: password for the **sysdba user** ID. Contact your system administrator the password. If you have used one of the default **sysdba user** IDs, use **manager** as the password.

```
Enter value for sysdba_schema_userid: sys
Enter value for sysdba_schema_password: manager
Connected.

Grant succeeded.
```

6. When prompted for the **Oracle Applications database user** details, enter the following:
 - **apps_schema_userid**: Oracle Applications database user ID (**apps** in our example).
 - **apps schema password**: password for the **Oracle Applications database user ID** (**apps** in our example).

```
Enter value for apps_schema_userid: apps
Enter value for apps_schema_password: apps
Grant succeeded.

Grant succeeded.

Synonym created.

Synonym created.

Package created.

Package body created.

Commit complete.

SQL>
```

The next figure shows an example of all the prompts and responses. Note that the responses may be different in your case.

Figure 22 *common_all.sql: Prompts and Responses*

```

SQL> @common_all.sql
Enter value for user_schema_userid: test
Enter value for user_schema_password: test
Enter value for db_connectstring: vis
Connected.
Enter value for table_thlspace: user_data

Table created.

Sequence created.

Commit complete.

Disconnected from Oracle9i Enterprise Edition Release 9.2.0.5.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.5.0 - Production
Enter value for sysdba_schema_userid: sys
Enter value for sysdba_schema_password: manager
Connected.

Grant succeeded.

Disconnected from Oracle9i Enterprise Edition Release 9.2.0.5.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.5.0 - Production
Enter value for apps_schema_userid: apps
Enter value for apps_schema_password: apps
Grant succeeded.

Grant succeeded.

Synonym created.

Synonym created.

Package created.

Package body created.

Commit complete.

SQL>

```

Create an ODBC System Data Source

You must create an ODBC system data source for the adapter to send and receive information. When configuring the system data source, you must include the database instance name where the adapter catalog resides. The adapter uses the data source name to request a connection to an ODBC system data source, which specifies the computer name and the database to which the data source maps.

First you identify the service name for the database to which you are connecting, then configure the ODBC driver for your system. See your *Oracle Database* documentation for details.

You can choose between the Oracle ODBC driver or the DataDirect ODBC driver shipped with the TIBCO ActiveMatrix Adapter for Database. TIBCO recommends the DataDirect ODBC driver.

DataDirect ODBC Driver

To configure the DataDirect ODBC driver:

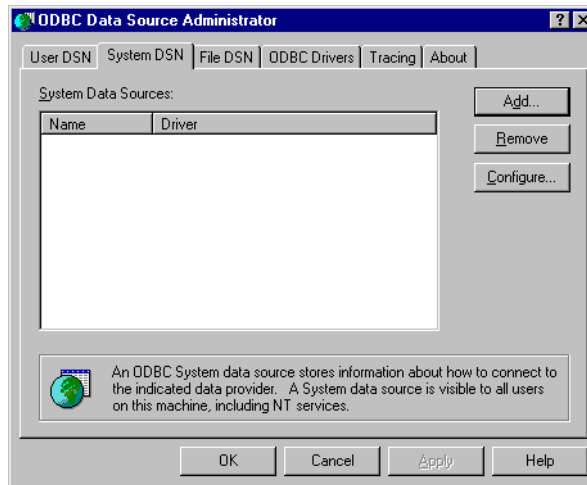
1. Execute the following file:
2. Select option 1 to install the drivers.
3. When prompted, enter the location of the DataDirect ODBC drivers as below:

```
$adapter_home\adb\odbc\Drivers
```

The DataDirect ODBC drivers are successfully installed and can be selected from the ODBC Data Source Administrator dialog.

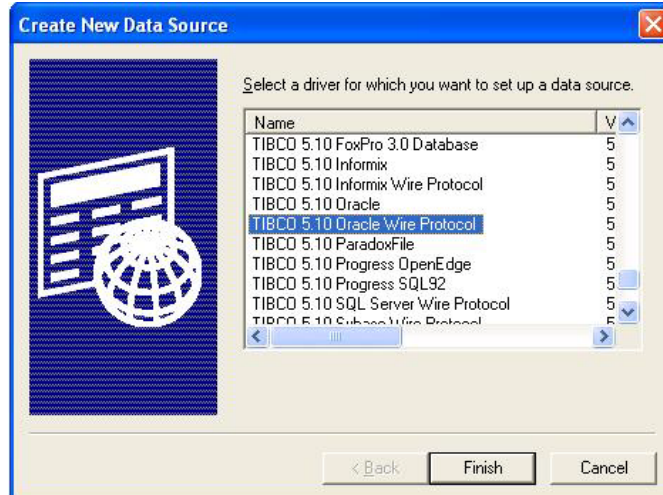
4. Click **Start > Settings > Control Panel > Administrative Tools**, and double-click the **ODBC** button to open the ODBC Data Source Administrator dialog (see Figure 23). Select the **System DSN** tab and then click the **Add...** button.

Figure 23 The ODBC Data Source Administrator Dialog



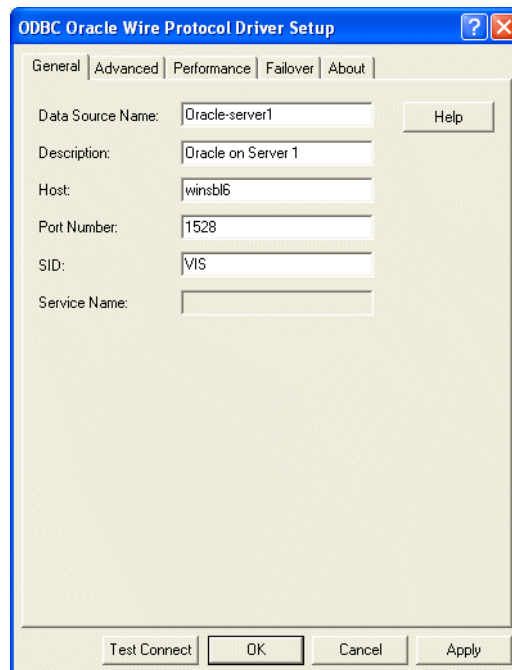
5. You will see a list of drivers (as shown in Figure 24). Select the **TIBCO 5.10 Oracle Wire Protocol** option and click the **Finish** button to open the ODBC Oracle Wire Protocol driver Setup dialog.

Figure 24 The Create New Data Source Dialog



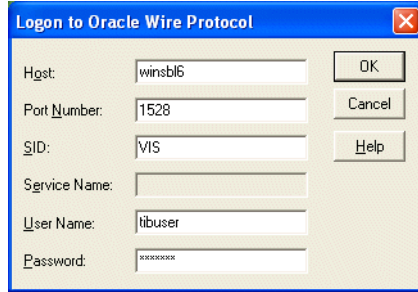
6. Input the appropriate values in the Data Source Name, Description, Host, Port Number, and SID fields. Click the **Help** button for a detailed explanation of each field. After that, click the **Test Connect** button to check the database connection (see Figure 25).

Figure 25 The ODBC Oracle Wire Protocol Driver Setup Dialog Box



7. Enter the pre-specified user's username and password in the User Name and the Password field respectively, click the **OK** button to start the test (see [Figure 26](#)).

Figure 26 The Logon to Oracle Wire Protocol Dialog



8. If the connection is successful, a message box will appear, as shown in [Figure 27](#). Click the **OK** button to close this dialog and click the **OK** button in the ODBC Oracle Wire Protocol Driver Setup dialog to complete the setup.

Figure 27 The Successful Connection Message

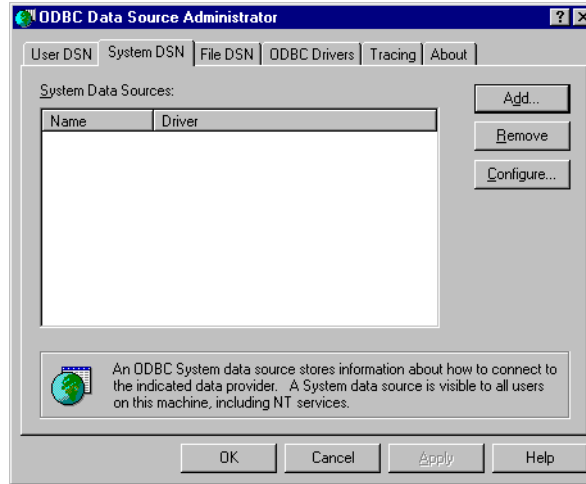


Oracle ODBC Driver

To configure the Oracle ODBC driver:

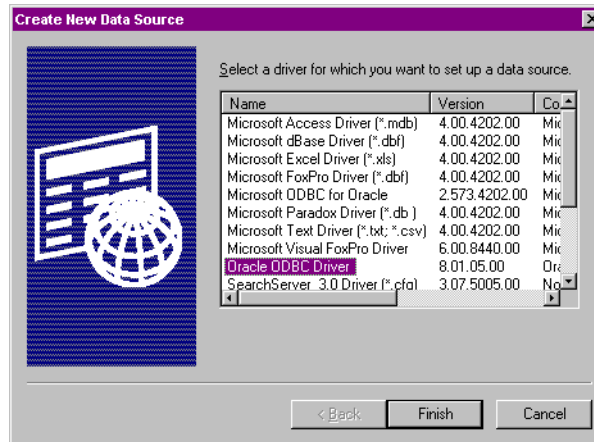
1. Click **Start > Settings > Control Panel > Administrative Tools**, and double-click the **ODBC** button to open the ODBC Data Source Administrator dialog (see [Figure 28](#)). Select the **System DSN** tab and click the **Add...** button to add new system data sources.

Figure 28 The ODBC Data Source Administrator Dialog



2. Select the **Oracle ODBC Driver** option from the drivers' list and click the **Finish** button to continue (see Figure 29).

Figure 29 The Create New Data Source Dialog



3. Type the appropriate values in the Data Source Name, Description, Service Name, and UserID fields. Click the **Help** button for a detailed explanation of each field. Click the **OK** button to apply these values (see Figure 30).

Figure 30 The Oracle ODBC Driver Setup Dialog

Oracle8 ODBC Driver Setup

Data Source Name: OK

Description: Cancel

Help

Data Source

Service Name:

UserID:

Database Options

Connect to database in Read only mode ☐

Fetch Count:

WorkAround Options

Force Retrieval of Long Columns ☐

Disable MTS Support ☒

Application Options

Enable Thread Safety ☒ Enable LOBs ☒ Enable Result Sets ☒

Enable Failover ☒ Retry Count: Delay:

Enable Query Timeout ☒ Enable Closing Cursors ☐

Translation Options

Option:

Library:

4. Click the **OK** button to complete the setup.



If you are using an Oracle ODBC Driver on the Windows platform, make sure that the **Disable MTS Support** checkbox is checked. Otherwise, the **Savepoint found in a distributed transaction environment** error will display when using the **Subscription with Reply** feature in the adapter.

Installation on UNIX Systems

Memory requirement for the adapter installation is around 256 MB or more (512 MB is recommended). Your operating system must meet the minimum patch requirements listed in [Table 8](#). See [Installer Disk Space Requirements in Temporary Area on page 36](#) for additional disk space requirements.

Table 8 Supported platforms, package names, patches and disk space for UNIX systems

Platform	Hardware Platform	Package Names	Minimum Patch(es)	Disk Space
Solaris 8	SPARC	TIB_adorapps-simple_<version_num>_s4_57_CC.bin	J2SE cluster patches are required for all Solaris platforms:	97.4 MB
Solaris 9	SPARC	TIB_adorapps-simple_<version_num>_s4_57_CC.bin	http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/J2SE	97.4 MB
Solaris 10	SPARC	TIB_adorapps-simple_<version_num>_s4_57_CC.bin	Font packages are required for all Solaris platforms: http://java.sun.com/j2se/1.4.2/font-requirements.html	97.4 MB
HP-UX 11	PA-RISC	TIB_adorapps-simple_<version_num>_h7_110_aCC.bin	J2SE patches are required for all HP-UX platforms:	116 MB
HP-UX 11i	PA-RISC	TIB_adorapps-simple_<version_num>_h7_110_aCC.bin	http://www.hp.com/products1/unix/java/patches/index.html	116 MB
HP-UX 11i v2	IA64 Itanium	TIB_adorapps-simple_<version_num>_odbc_h7_ia64.bin		105 MB
AIX 5.2	POWER	TIB_adorapps-simple_<version_num>_rs_51_xlrc.bin		102 MB
AIX 5.3	POWER	TIB_adorapps-simple_<version_num>_rs_51_xlrc.bin		102 MB

Supported Oracle Applications

The adapter supports the Oracle Applications 11i family and Oracle Database Server 9i and 10g.

TIBCO Runtime Agent Must be Installed Before the Adapter

Before you install the adapter, you must install TIBCO Runtime Agent. If you choose the Typical installation mode for TIBCO Runtime Agent, the installer places all libraries and other products required by the adapter into the *TIBCO HOME* directory.

During the installation, the adapter installer checks the availability of all dependent products in the target system. If any of the dependencies are not available, the installer will not proceed with the installation.

Installer Account

TIBCO 5.x products can be installed by a regular (non-root) user and super-user (root). Different users can install the same product at different locations.

Product dependencies at install time are resolved at user level through the installation registry maintained at user's home directory. See [Installation Registry on page 35](#) for more information.

Windows Environment

A windows environment such as CDE (that is, X Windows) is required to run the installer in GUI mode. It is not required for a console installation.

Installing the Adapter on UNIX

After unpacking the software and accepting the license agreement, you can choose to perform a typical install or custom install.

- A typical install has minimal prompts and installs standard components in default locations.
- A custom install allows you to choose which components you want to install and where you want to install them.

The installer checks your system for the installation home directory that was established when TIBCO Runtime Agent was installed. The adapter is installed under the installation home directory.

Use one of the following modes to install the software. The examples below assume you are installing the adapter on Solaris 2.8.

Install Using GUI Mode

The GUI Mode allows you to input values in panels. Type the following command in a terminal window and press the **ENTER** key:

```
% ./TIB_adorapps-simple_<version_num>_s4_57_cc.bin
```

Install Using Console Mode

Console mode allows you to install the software from a command line. The installer will prompt you for values. Type the following command in a terminal window:

```
% ./TIB_adorapps-simple_<version_num>_s4_57_cc.bin -is:javaconsole  
-console
```

You can move through the installation process when running in console mode using the number keys described below:

```
Enter Key or 1 = Moves forward in the installer  
2 = Goes back to previous screen  
3 = Cancels the Wizard and exits the installation or uninstallation  
4 = Redisplays the current screen
```

Install Using Silent Mode with Default Values

Silent mode allows you to install the software without any custom prompts and apply default settings automatically. Type the following in a terminal window:

```
% ./TIB_adorapps-simple_<version_num>_s4_57_cc.bin -silent
```

Install and Generate a Response File

You can generate a response file during the installation process and can use the same file in future installations. When using the response file for all installation modes, the options in the file determine what will be installed.

To install and generate a response file, type the following at the command prompt:

```
% ./TIB_adorapps-simple_<version_num>_s4_57_cc.bin -options-record  
/dir/<responseFile>
```

Combining Options

You can combine the different available options. For example, to install in silent mode using a response file, use:

```
% ./TIB_adorapps-simple_<version_num>_s4_57_cc.bin -silent -options  
<responseFileName>
```

To install using Console mode and generate a response file, use:

```
% ./TIB_adorapps-simple_<version_num>_s4_57_cc.bin -is:javaconsole  
-options-record <responseFileName>
```

Clients, Platforms, and Drivers

This section identifies the drivers and clients supported by different UNIX platforms.

The following table lists the supported platforms for the Oracle client, along with the required ODBC and JDBC drivers for each platform.

Table 9 Oracle Client Matrix for UNIX Platforms

Platform	Oracle 9i	Oracle 10g
Sun Solaris 8	yes	yes
Sun Solaris 9	yes	yes
Sun Solaris 10	yes	yes
HP-UX 11	yes	yes
HP-UX 11i	yes	yes
HP-UX 11i V2	yes	yes
AIX 5.2	yes	yes
AIX 5.3	yes	yes

Connectivity	Oracle 9i	Oracle 10g
ODBC	Connect 5.00	Connect 5.00
JDBC	Connect JDBC 3.1	Connect JDBC 3.1

The adapter has been fully tested and certified against the JDBC and ODBC drivers listed in this section. Consult with your TIBCO representative before using the adapter with a driver not specifically listed here.

JDBC Drivers

TIBCO Adapter for Oracle Applications access your database using JDBC drivers. The correct JDBC driver is supplied with each adapter package. When configuring an adapter, the JDBC URL and the correct package specified is automatically loaded.

ODBC Drivers

The TIBCO Adapter for Oracle Applications installation package includes the required ODBC drivers for use during runtime.

In addition, the adapter supports the following drivers which are provided separately by various database vendors.



Consult with your TIBCO representative before using TIBCO Adapter for Oracle Applications with an ODBC driver not specifically listed here.

ODBC Drivers for Oracle

- **On AIX and Red Hat Linux** DataDirect Oracle ODBC 5.00 (wire format and non-wire format). Note that DataDirect non-wire format drivers will not work with the Oracle 9.2.0 client on UNIX platforms.
- **On Solaris and HP-UX** DataDirect Oracle ODBC 5.00 (wire format and non-wire format). Note that DataDirect non-wire format drivers will not work with the Oracle 9.2.0 client on UNIX platforms.

ODBC Driver
Documentation

The *DataDirect Connect ODBC Installation Guide* and *DataDirect Connect ODBC Reference* are available in the *install-path/adb/odbc/books* directory.

Post Installation Tasks on UNIX

Permission Requirements on UNIX Systems

All adapter users must have read, write, and execute permissions for the following directories:

```
$TIBCO_HOME/adapter/adorapps/<version_num>/bin
$TIBCO_HOME/adapter/adorapps/<version_num>/logs
$TIBCO_HOME/adapter/adorapps/<version_num>/ledger
$TIBCO_HOME/adapter/adorapps/<version_num>/sql
$TIBCO_HOME/tra/<version_num>/logs
$TIBCO_HOME/logs
```

For example, if you have installed the adapter in the `/opt/tibco` directory, you need to make these directories writable for all other users by executing the following commands:

```
% chmod a+w /opt/tibco/adapter/adorapps/<version_num>/bin
% chmod a+w /opt/tibco/adapter/adorapps/<version_num>/logs
% chmod a+w /opt/tibco/adapter/adorapps/<version_num>/ledger
% chmod a+w /opt/tibco/adapter/adorapps/<version_num>/sql
% chmod a+w /opt/tibco/tra/<version_num>/logs
```

Configure the Service Name

The service name must be configured. This is done by putting an appropriate entry in the `tnsnames.ora` file which is located in the `<ORACLE_HOME>/network/admin` directory.

```
<service_name> = (DESCRIPTION=
                  (ADDRESS=(PROTOCOL=tcp)(HOST=<host>)(PORT=<port>))
                  (CONNECT_DATA=(SID=<service_name>))
                  )
```

where `<service_name>` is the name of the service to be configured; `<host>` is the host name; and `<port>` is the port name.

Configure the Database Account

A set of scripts are provided to configure the database account used by the adapter. These scripts are very similar to the ones available on Microsoft Windows; refer to [Configure the Database Account, page 51](#) for more information.

- `create_user.sql`—Refer to [Using the create_user.sql Script, page 51](#) for more information.

- `common_all.sql`—Refer to [Using the common_all.sql Script, page 53](#) for more information.
- `alerter.sql`—Refer to [, page 57](#) for more information.

Create an ODBC Data Source

To configure your ODBC driver:

1. Set the ODBCINI environment variable to point to a valid `odbc.ini` file. For example:

```
setenv ODBCINI /tsi/home/tcact3/.odbc.ini
```
2. Add a data source name (dsn) in the file pointed to by ODBCINI. For example, to add the data source name ORACLE8:
 - a. Append this dsn under Section [ODBC Data Sources]:

```
[ODBC Data Sources]
...
...
...
Oracle Wire Protocol=DataDirect 5.1 Oracle Wire Protocol
```

- b. Add the following section for the above data source name in the same file pointed to by [Oracle]

```
Driver=<ODBC_HOME>/lib/TIora20.so
Description=DataDirect 5.1 Oracle Wire Protocol
LogonID=uid
password=pwd
HostName=gomati
SID=Oracle SID
CatalogOptions=0
ProcedureRetResults=1
EnableDescribeParam=0
EnableStaticCursorsForLongData=0
ApplicationUsingThreads=1
```

Driver—absolute path of the ODBC driver library for Oracle.

Description—string that describes the driver.

HostName—connect string used to connect to the Oracle Server.

SID—Database Instance name.

LogonID—account name used to log into the Oracle database.

password—password for the account name used to log into the Oracle database.

Set Environment Variables

Certain environment variable settings are required for running the adapter. Each TIBCO product or product component contains the `adorappssetenv` file which you must execute before using the product. Execute the appropriate file to make these changes automatically.

- For `csh`, execute `<ADORAPPS_HOME>/bin/adorappssetenv.csh` using the command below:

```
% source adorappssetenv.csh
```

- For `sh` and `ksh`, execute `<ADORAPPS_HOME>/bin/adorappssetenv.sh` using the command below:

```
$ . ./adorappssetenv.sh
```

The `adorappssetenv` file adds information about the product component to the following environment variables on UNIX platforms:

`PATH`


```

CLASSPATH
SHLIB_PATH (HP-UX)
LD_LIBRARY_PATH (Solaris)
LIBPATH (AIX)
LD_LIBRARY_PATH (on Solaris and Red Hat Linux)

```

For Solaris, append the `<ORACLE_HOME>/lib` to the `LD_LIBRARY_PATH` (`SHLIB_PATH` in case of HP-UX).



When using the Oracle non-wire protocol driver, `ODBC_HOME/lib` must always appear in `LD_LIBRARY_PATH` before any database vendor's `lib` directory, such as `ORACLE_HOME/lib`.

Thereafter, export the path where the `tnsnames.ora` file is located, which is the `<ORACLE_HOME>/network/admin` directory. This can be done by adding the following command `export TNS_ADMIN=<ORACLE_HOME>/network/admin` in the `adorappssetenv.sh` file.

Database Objects Created as a Result of Configuration

Publication Transaction

The following tasks are done as a result of configuration:

1. Creation and Alteration of MV Logs. Grant `select`, `update`, `delete` on the MV Logs to the **adapter user** (for example: **test**) schema. This is done from the schema where the Source tables are existing.

For example, the schema is **BOM** for the **EngBOM** transaction.

Note that these MV Logs are created in the same schema as the Source tables.

2. Creation of Views. This is done from the **adapter user** schema.
3. Creation of the Publishing table along with the indexes on the Publishing table. This is done from the **adapter user** schema.
4. Creation of a Sequence on the Publishing table. This is done by the **adapter user** schema.
5. Creation of triggers on the MV Logs. This is done by the **adapter user** schema.

Subscription Transaction

The following tasks are done as a result of configuration:

1. Creation of Intermediate tables along with the Indexes. Sequence will also be created for the Header Intermediate table. This is done from the **adapter user** schema.
2. Creation of synonyms for the Intermediate Tables created above. These synonyms are created in the **Oracle Applications database user** schema.
3. Creation of the package related to the transaction. This will also be done from the **Oracle Applications database user** schema.
4. Grant `Execute` on the package created as above to the **adapter user** schema. This will also be done from the **Oracle Applications database user** schema.

Installation FAQs and Trouble Shooting

This section lists some common errors along with their causes and solutions.

Frequently Asked Questions

Where is the installation log file located?

Install and uninstall log files are created in the TIBCO_HOME\log directory.

What should I do, if JVM crashes when I run the installer?

The installer first extracts this bundled JVM into a temporary area and then uses it to launch itself. If for some reason, the JVM crashes, you could still run the installer using another JVM, preferably JVM 1.3.1 or higher. The syntax is:

```
<TIB_adorapps-simple_<version_num>_w32.exe -is:javahome
C:\j2sdk1.4.0 (for Windows)
<TIB_adorapps-simple_<version_num>_s4_57_CC.bin -is:javahome
/opt/jre140 (for Sun Solaris 2.8)
```

The javahome directory must contain bin/java.exe or bin/java.

The installer will use the externally supplied JRE to launch itself.

Will 5.1 installer recognize a 3.x or 4.x installation?

TIBCO products follow a three digit release numbering scheme:

Major.Minor.Maintenance

Product releases that differ in either Major or Minor numbers will be a separate installation, and will not recognize the old installation. In this case, 5.0 is a major release and hence will not recognize either 3.x or 4.x product installations.

Why and how should I set the DISPLAY variable on UNIX platforms for GUI mode?

The installer in GUI mode on UNIX must open an additional window, generally for graphics. It uses the DISPLAY environment variable to let you know on what computer the window is to be opened. If the environment variable is not set, the installer will either wait or abort after displaying:

```
InstallShield Wizard
Initializing InstallShield Wizard...
Preparing Java(tm) Virtual Machine...
```

```
.....
.....
.....
```

The DISPLAY variable must be set to the IP address or the computer name (on which the installer graphics window are to be displayed), followed by a screen address, which can be :0.0. For example:

```
# Bourne shell
DISPLAY=<ip_address>:0.0; export DISPLAY

# Korn shell
export DISPLAY=<ip_address>:0.0

# C-shell
setenv DISPLAY <ip_address>:0.0
```

For example, consider a scenario where you need to install the adapter on a remote HP-UX machine (named *itaska*). Assume you have a Solaris 5.6 machine (named *alaska*) that has a video card and monitor installed and you can run an X-window application on it. Therefore, you decide to telnet to *itaska* from *alaska*.

When you telnet to *itaska*, you will not get access to *itaska*'s monitor and will not be able to display an X-window application. Therefore, you must set the DISPLAY variable which instructs the X-server to redirect all windows to the computer set in the variable. Before doing so, the computer (specified in the DISPLAY variable) must give permissions to share its monitor.

```
alaska> xhost + # give permission for all to its share monitor
alaska> telnet itaska
Welcome to HP-UX itaska 11.00
User:
Password:
itaska> export DISPLAY=alaska:0.0 # set display on alaska
itaska> TIB_adorapps-simple_<version_num>__h7_110_aCC.bin
```

What is uninst2 directory?

If the original uninstall directory is in use at uninstall time, it cannot be removed by the installer program. The installer will create a second uninstall directory for the second installation. To remove the second installation, you must invoke the uninstall program from the second uninstall directory. The original uninstall directory can also be manually removed, if empty.

Running Out of Disk Space

The installer calculates the disk space required in product home location for the selected components. The calculation is performed prior to the actual installation (copying of files to the system). The installer will proceed only if sufficient free disk space is available in the product home location.

However, if the disk space is consumed by another process while the installer is transferring files, which reduces the required disk space, the installer will fail and give a failure message.

Solution

Do not run other processes that consume disk space in the product home location during the process of installation.

Installation Errors on HP-UX 11.00 64 bit Platform

Error message Installation on a HP-UX 11.00 64 bit system will crash with the error message below:

```
Pid nnn killed due to trashed stack.
Pid nnn was killed due to failure in writing the signal context.
```

This happens only on HP-UX 11.00 64 bit systems. However, it does not happen on HP-UX 11.00 32 bit system and HP-UX 11.11 (or 11.i) system.

To determine the OS version on your system, run:

```
uname -a
```

To determine the kernel bits on your system, run:

```
getconf KERNEL_BITS
```

Resolution HP-UX kernel patch PHKL_27282, resolves the above crash.

To determine if your system has the kernel patch, run:

```
/usr/sbin/swlist -l product PHKL_27282
```

or

```
what /stand/vmunix | grep PHKL_27282
```

If your system is an HP-UX 11.00 64 bit system and does not have the patch, first install the HP-UX kernel patch PHKL_27282 and then proceed with the installation. Installation of patch PHKL_27282, will reboot your system.

Message When installing as root user on HP-UX, you will get the following error:

```
Assertion failed: so->so_hard_contents, file softobj.c, line 363
```

Resolution Install as a non-root user.

Configuring TIBCO Hawk

Error

TIBCO Runtime Agent includes the TIBCO Hawk Agent only. If you install the full TIBCO Hawk package after installing TIBCO Runtime Agent without installing a Java Runtime Environment (except TIBCO JRE), the TIBCO Hawk Configuration tool will not be able to determine the Java home location and the JVM executable. The TIBCO Hawk services will not start correctly and you will not be able to start the TIBCO Hawk Display.

Resolution

1. Start the TIBCO Hawk Configuration tool. For example, on Microsoft Windows:

Start > TIBCO > TIBCO Hawk > Hawk Configuration

2. Under the General tab, click the **Advanced** button.
3. In the Java Home Directory field, provide the Java path. For example:
C:\tibco\jre\<version_num>
4. In the JVM Executable field, provide the JVM executable. For example:

java.exe

The services will start properly and the TIBCO Hawk Display will run after the completion of the processes above.

Cannot Install the Adapter

On HP-UX and AIX platforms, even though the correct version of TIBCO Runtime Agent version is already installed on the system, installation of an adapter that depends on TIBCO Runtime Agent can fail in the dependency resolution.

The TIBCO product installer maintains the registry information in the `vpd.properties.tibco.systemName` file. The value for *systemName* is determined by `InetAddress.getLocalHost().getHostName()`. However, the method `getHostName()`, returns different values based on the JRE versions used. For example, on AIX, JRE 1.3.1 returns only *systemName*, whereas JRE 1.4.0 returns *systemName.domainName*. For this reason, the installer cannot load the correct registry file.

Resolution

On UNIX platforms, the installer registry file `vpd.properties.tibco.systemName` is located in the user's home directory.

Case 1: If the `vpd.properties.tibco.systemName` file exists:

```
$ cd user's_home_directory
$ ln -s vpd.properties.tibco.systemName
                                vpd.properties.tibco.systemName.domainName
```

For example:

```
$ cd ~
$ ln -s vpd.properties.tibco.upside
vpd.properties.tibco.upside.tibco.com
```

upside is *systemName* and *tibco.com* is *domainName*

Case 2: If the `vpd.properties.tibco.systemName.domainName` file exists:

```
$ cd user's_home_directory
$ ln -s vpd.properties.tibco.systemName.domainName
vpd.properties.tibco.systemName
```

For example:

```
$ cd ~
$ ln -s vpd.properties.tibco.upside.tibco.com
vpd.properties.tibco.upside
```

upside is *systemName* and *tibco.com* is *domainName*.

Chapter 3 **Getting Started**

This chapter gives you a hands-on exercise on using TIBCO Adapter for Oracle Applications.

Topics

- [Overview, page 80](#)
- [Configuring the Adapter Components, page 81](#)
- [Deploying the Adapter, page 94](#)
- [Start the Adapter, page 95](#)
- [Stop the Adapter, page 96](#)

Overview

In this exercise you will configure, start, run, deploy, and stop the adapter. You will perform the Create, Update, and Delete operations on the Items Business Object.

For more examples, refer to the *TIBCO Adapter for Oracle Applications Examples Guide*.

Configuring the Adapter Components

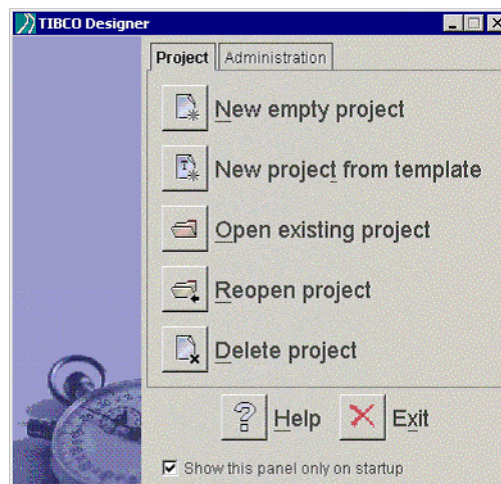
This section will take you through the Publisher options on Items Business Object.

Task A Configure an Adapter Instance

The TIBCO Designer GUI is used to configure adapter instances. When starting the TIBCO Designer, you must create or select a project. A project contains configuration files that define options used by a runtime adapter. After a project is configured, it is converted to a repository file and is available for use by the runtime adapter.

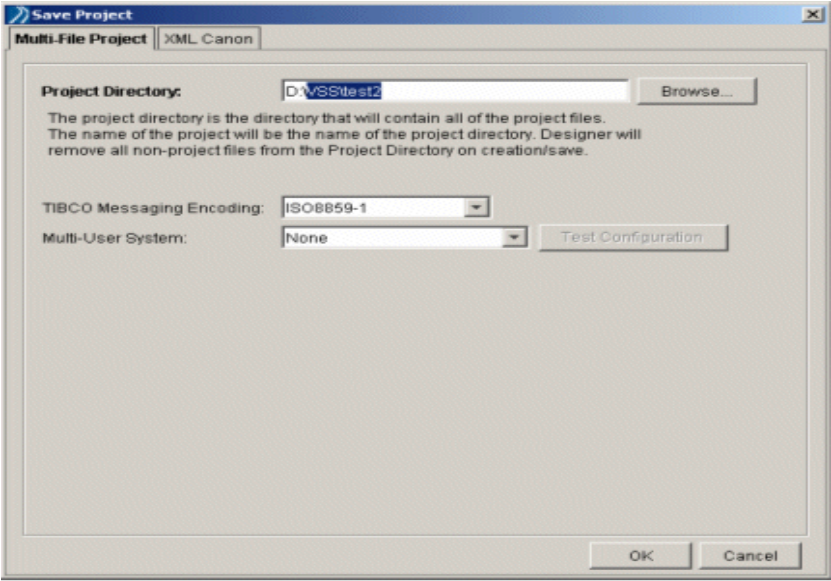
1. Open TIBCO Designer and choose the **New empty project** Option (see [Figure 31](#)).

Figure 31 The TIBCO Designer GUI



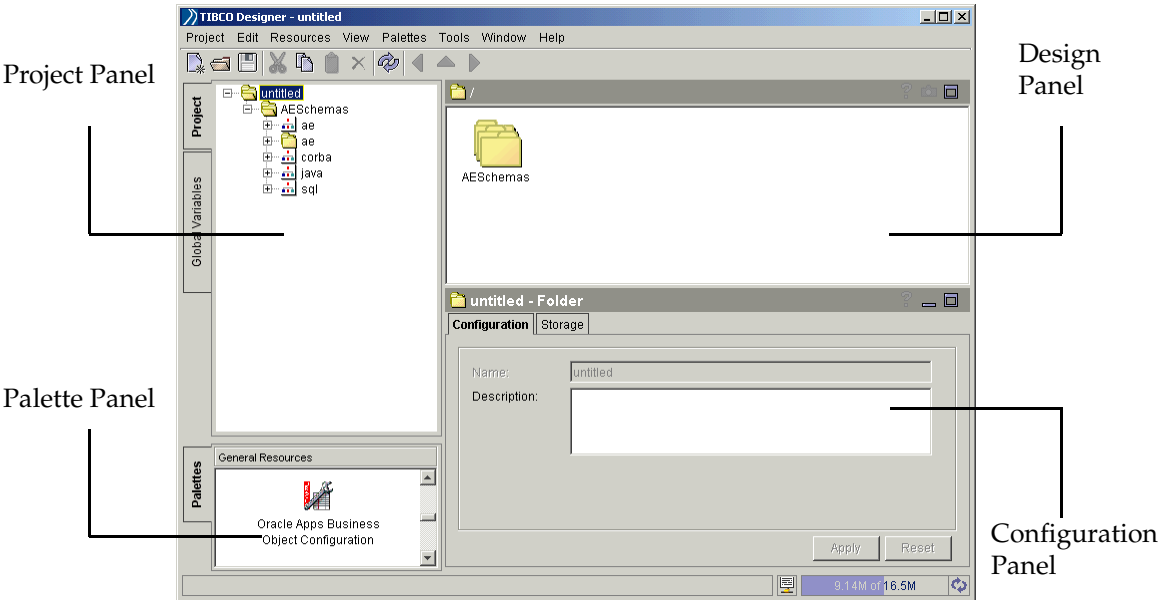
2. Click the **Multi-File Project** tab. Click the  button to select the location of the project. Click the **OK** button to continue (see [Figure 32](#)).

Figure 32 The Save Project Dialog



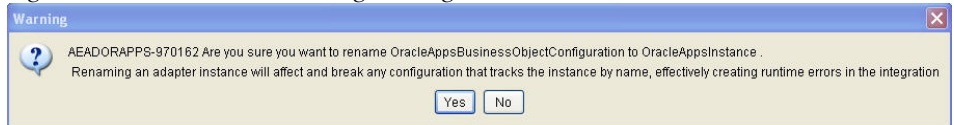
A new Multi-File project is created and the TIBCO Designer screen appears as shown in [Figure 33](#).

Figure 33 The TIBCO Designer Screen



3. Drag the **Oracle Apps Business Object Configuration** button from the palette panel and drop it into the design panel.
4. In the Configuration tab of the Configuration Panel, rename the Name field to `OracleAppsInstance` and click the **Apply** button.
5. A rename warning message is shown as in [Figure 34](#). Click the **Yes** button to confirm and continue.

Figure 34 The Rename Warning Message



Task B Configure Oracle Applications Business Object Design-time Connection Settings


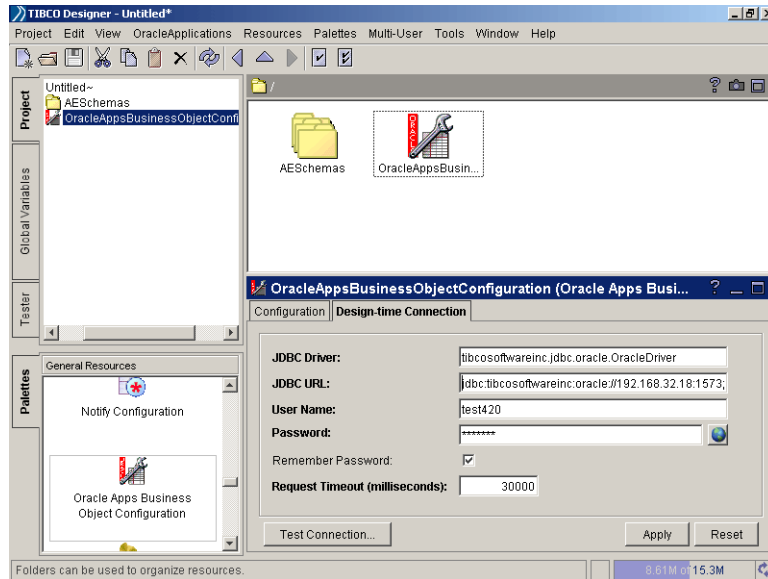
1. Click the **Design-time Connection** tab in the Configuration panel.
2. Enter the connection parameters.
 - a. Specify the JDBC Driver and JDBC URL that points to the Oracle Applications Database.
 - b. Specify the Username and the Password of the user account where the TIBCO Publisher Database objects are created. See [Post-Installation Tasks on Windows on page 44](#), for details.
 - c. Specify the Password for the user account mentioned above. The  button discards the current value in the field and you can replace its value with a global variable (see [Figure 35](#)).

Figure 35 The Design-time Connection Tab

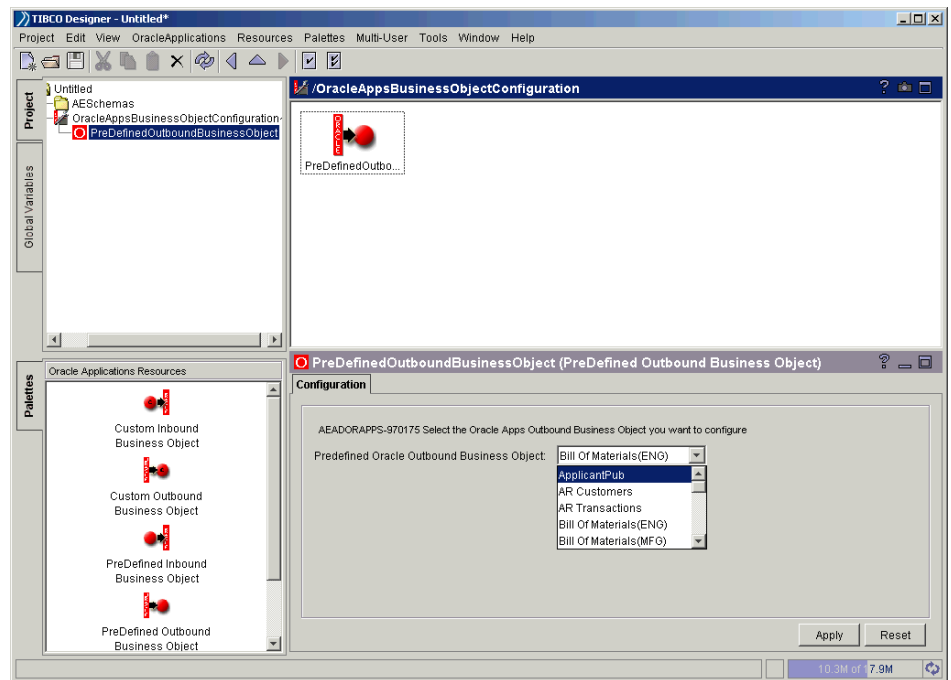


3. Click the **Test Connection** button.
4. Click the **Apply** button after a successful connection.

Task C Configure an Items Business Object

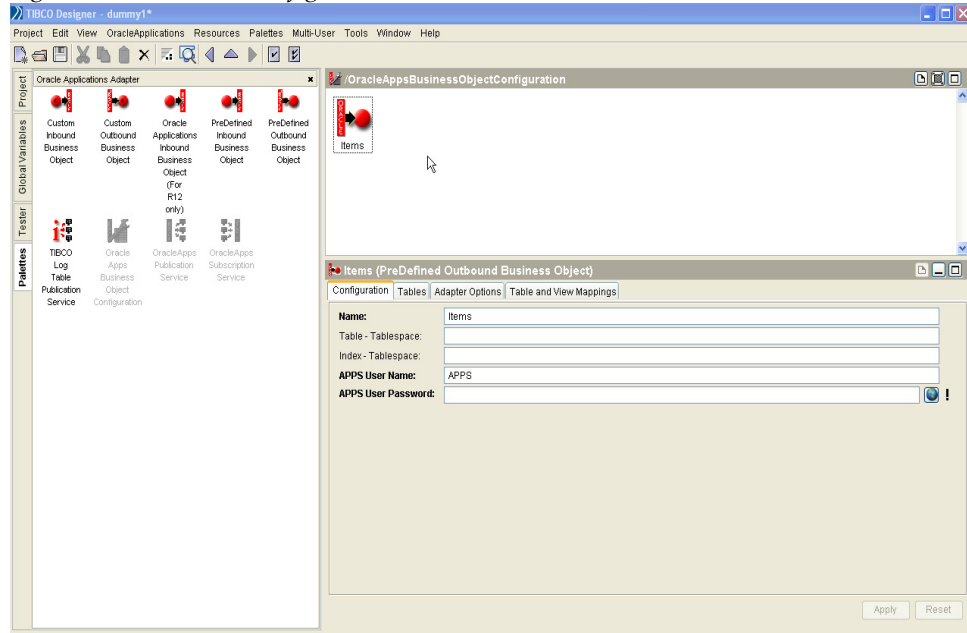
1. Click the **OracleAppsBusinessObjectConfiguration** resource in the Project Panel (top-left panel). You will see a set of five clickable buttons in the palette panel.
2. Drag the **PreDefined Outbound Business Object** button from the palette panel and drop it into the Design Panel (top-right panel). You will see a list of available PreDefined Business Objects in the **Configuration** tab.
3. Click the PreDefined Business Objects drop-down list to view the available PreDefined Business Objects (see [Figure 36](#)).

Figure 36 The Configuration Tab



4. Select the item named **Items** from the Predefined Oracle Outbound Business Object drop-down list and click the **Apply** button to create this Business Object.
5. Click the **Items** Business Object button in the design panel as shown in [Figure 37](#).

Figure 37 The Items Configuration Panel



6. Enter appropriate values to configure the Business Object such as Oracle Applications database User Name, Password and so on and click the **Apply** button.

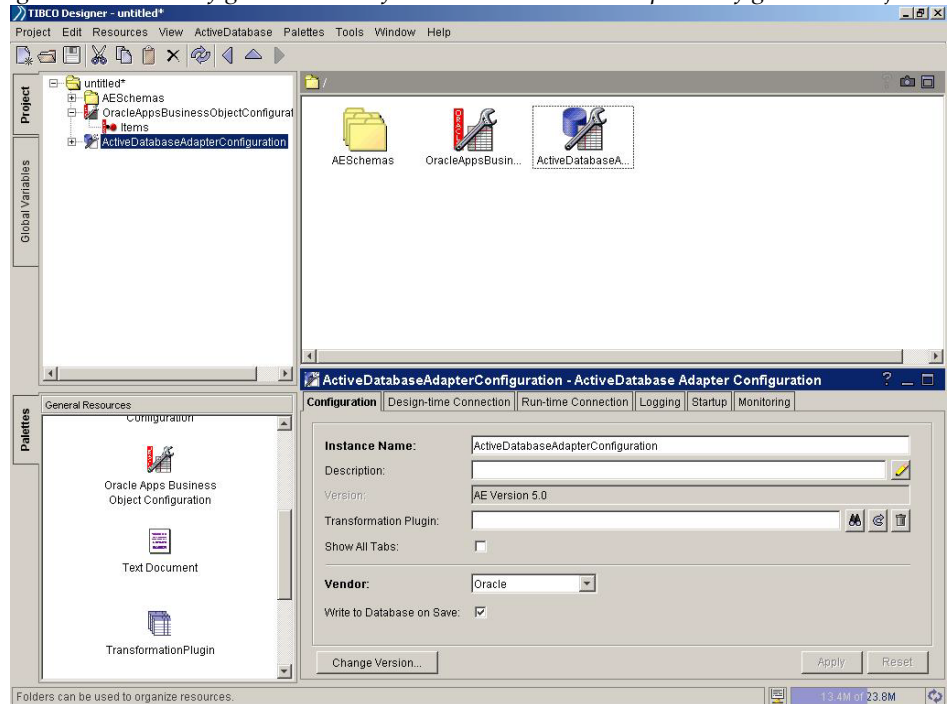
(Use the steps described above to configure other Publisher Business Objects such as Purchase Order, Sales Order, etc.)

7. Click the  button to save the project.

Task D: Configure the corresponding ActiveDatabase Service

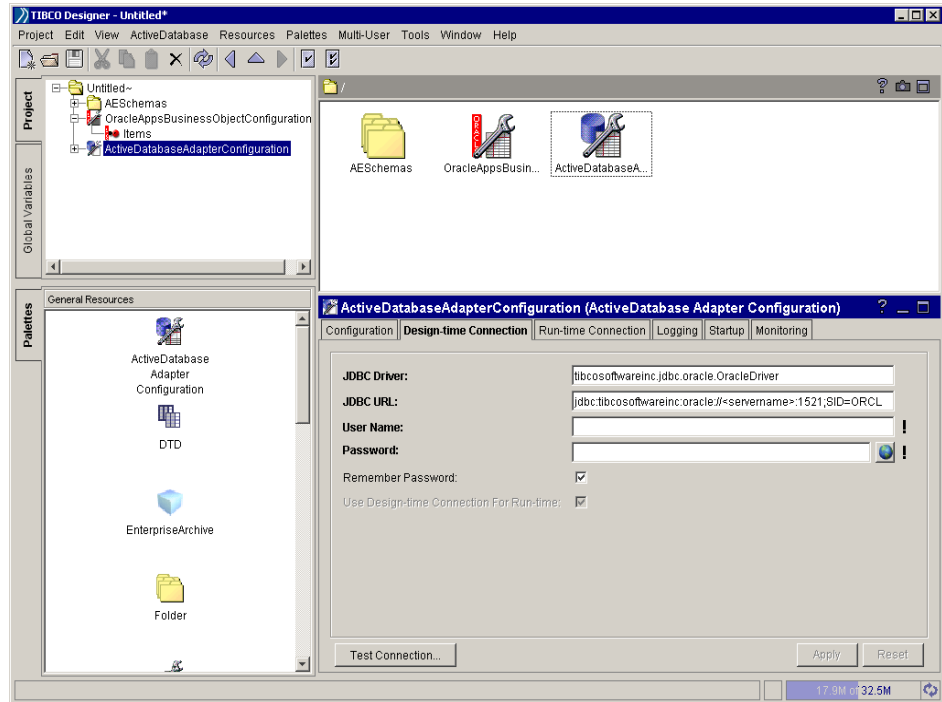
1. Select the root folder in the project panel. Drag the **ActiveDatabaseAdapterConfiguration** button from the palette panel and drop it into the Design Panel (top-right panel).
2. Name the Adapter instance as **ActiveDatabaseAdapterConfiguration** in the Instance Name field of the Configuration tab. Select the **Oracle** item from the vendor drop-down list, clear the **Write to Database on Save** checkbox and click the **Apply** button. Leave the values in other fields as default (see [Figure 38](#)).

Figure 38 The Configuration Tab of the ActiveDatabaseAdapterConfiguration Object



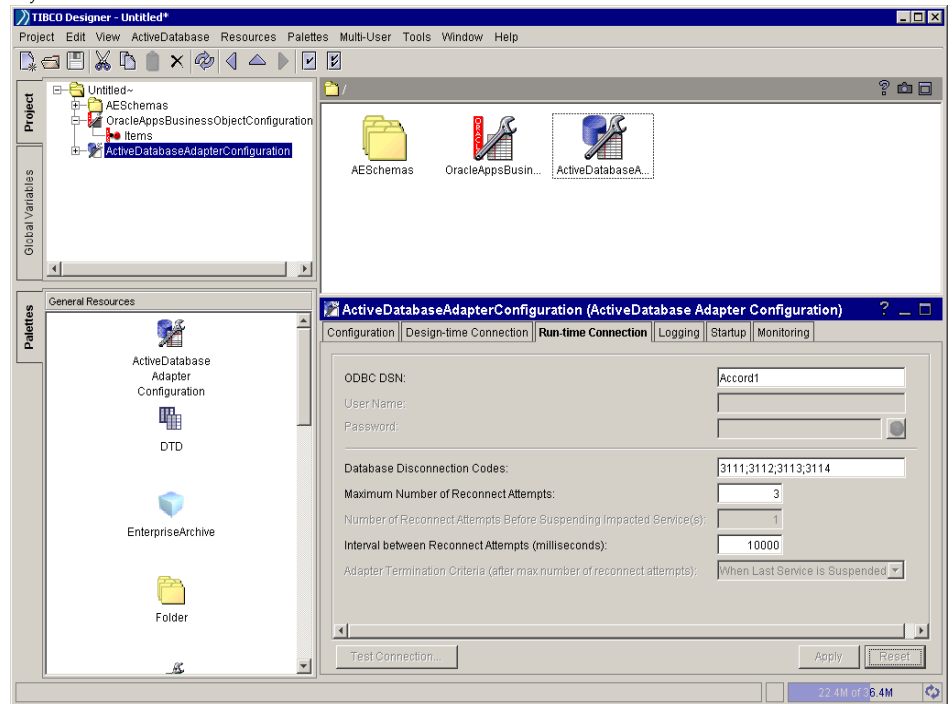
3. Input the appropriate values for JDBC Driver, JDBC URL, User Name, and Password in the Design-time Connection tab. If you want to save the password, Check the **Remember Password** checkbox. Leave the values of other fields as default and click the **Test Connection...** button to check if the connection is successful. Click the **Apply** button after a successful connection (see Figure 39).

Figure 39 The Design-time Connection Tab of the ActiveDatabaseAdapterConfiguration Object



4. Input the appropriate value for ODBC DSN in the Run-time Connection tab and leave the values of other fields as the default. Click the **Apply** button to continue (see [Figure 40](#)).

Figure 40 The Run-time Connection Tab of the ActiveDatabaseAdapterConfiguration Object



5. Select the sub folder Adapter Services under the ActiveDatabaseAdapterConfiguration object in the project panel. You will see three blue clickable Services buttons under the ActiveDatabase Adapter panel (see [Figure 41](#)) and two red clickable Services buttons under the Oracle Applications Adapter panel (see [Figure 42](#)) in the palette panel. You can switch between the ActiveDatabase Adapter panel and the Oracle Applications Adapter panel by selecting the minimized panel at the right bottom corner of the Palettes panel.

Figure 41 The Three Blue Services Buttons in the Palette Panel (Adapter Services Folder)

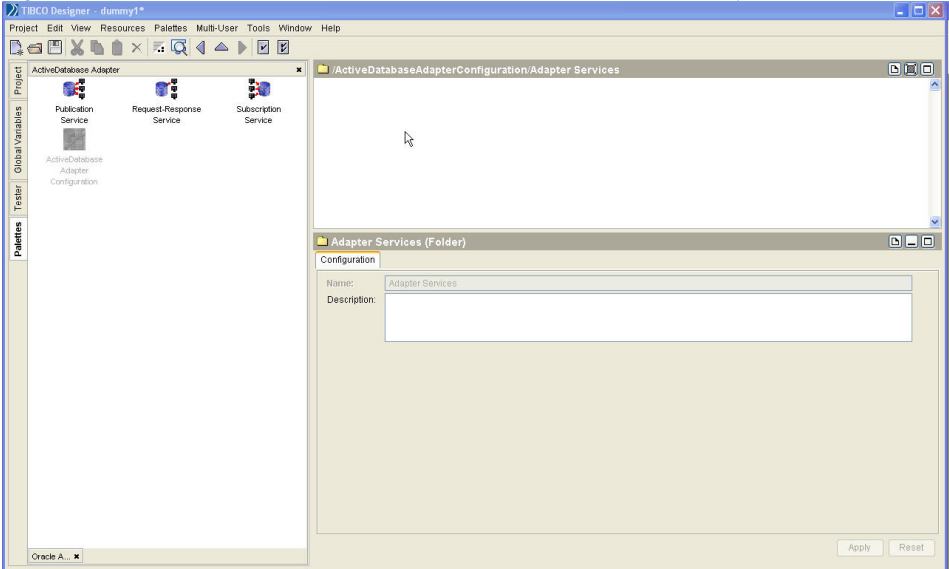
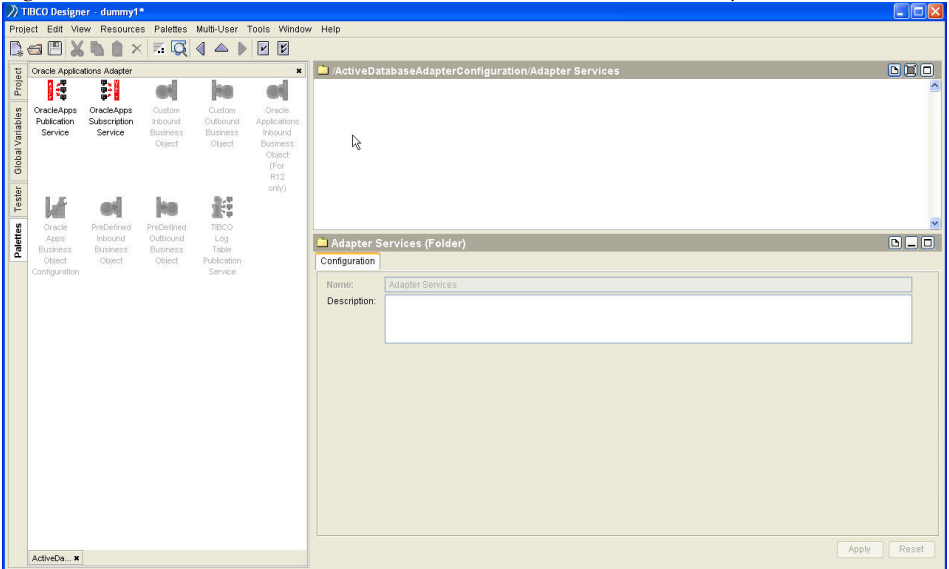

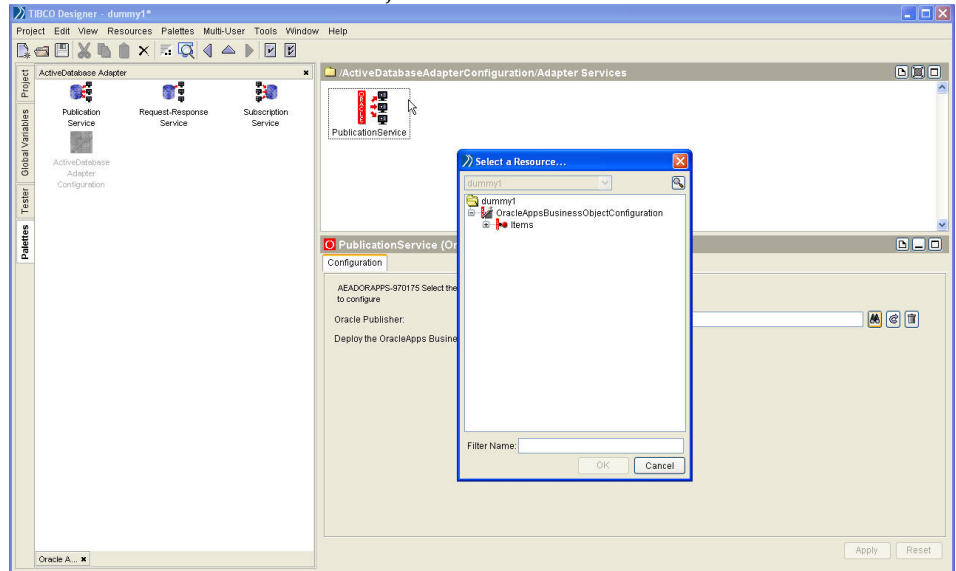


Figure 42 The Two Red Services Buttons in the Palette Panel (Adapter Services Folder)



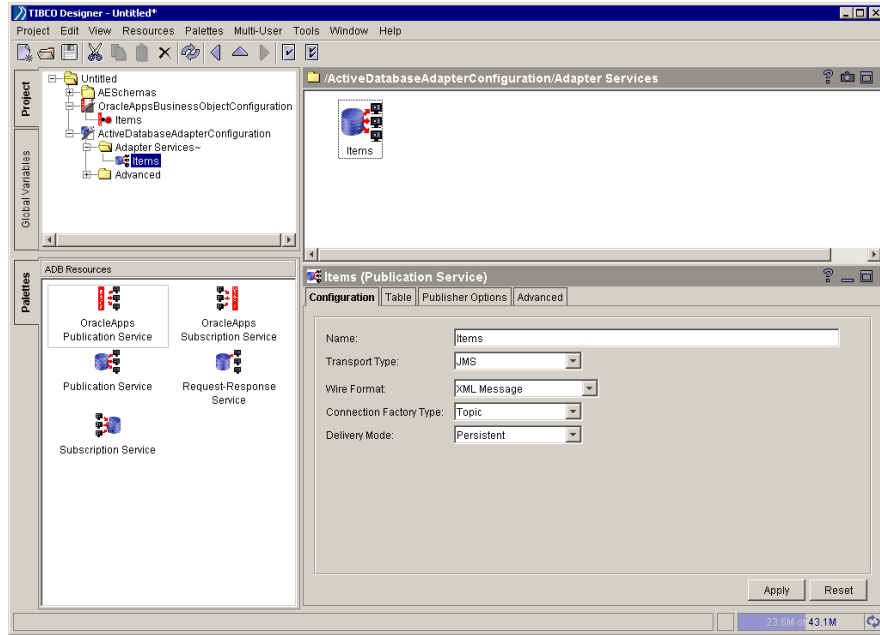
6. Drag the **OracleApps Publication Service** button from the Oracle Applications Adapter panel in the palette panel and drop it into the design panel.

7. Click the  button at the right hand side of the Oracle Publisher field and select the **Items** Business Object created in the Task C.



8. If you need to create the associated Oracle Applications database objects, check the **Deploy the Oracle Applications Business Object** checkbox and click the **Apply** button.
9. Click the **Items** business object in the design panel. Select **JMS** from the Transport Type drop-down list. Leave the values of other fields as default and click the **Apply** button to apply the configuration settings (see Figure 43).

Figure 43 The Configuration of the Items Business Object




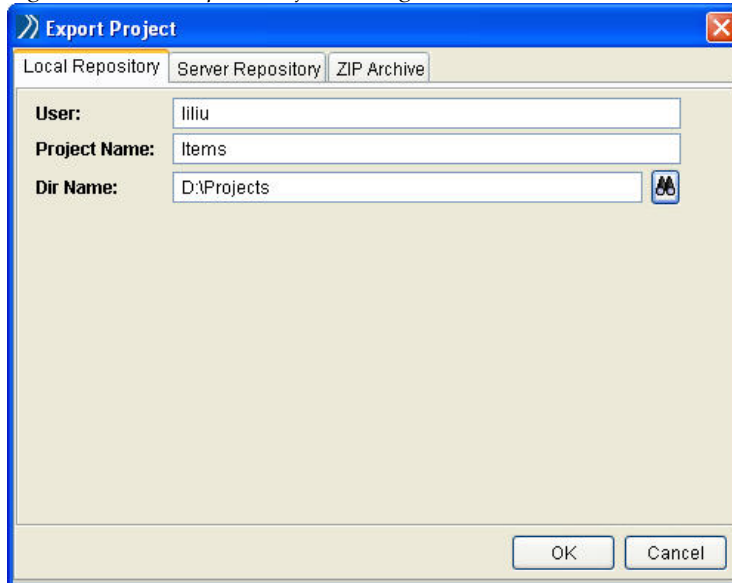
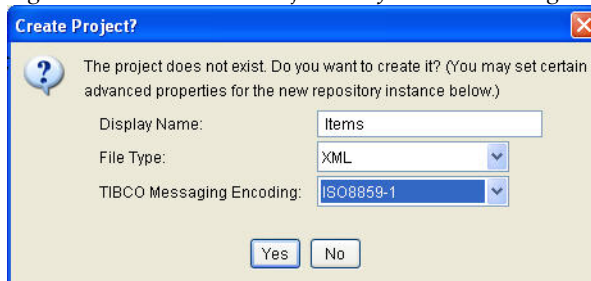
10. Select **Project > Export Full Project** to open the Export Project window. The User field displays the default value.
11. Enter the Project Name in the Project Name field, and click the  button. Specify a location where the local repository.dat file will be saved and click the **OK** button (see Figure 44).

Figure 44 The Export Project Dialog



12. The Create Project Confirmation Dialog will remind you to create a new project or to overwrite an existing project. Leave all the settings in this dialog box as default and click the **Yes** button to confirm the Create Project activity (see Figure 45).

Figure 45 The Create Project Confirmation Dialog



13. Prepare a TRA file for this repository and make sure that properties such as `tibco.repourl` and `tibco.configurl` correspond to the correct values. You can consult the `sample.tra` file provided with the adapter examples.



If you are using the Oracle ODBC driver, make sure that the path `<ORACLE_HOME>/bin` is appended at the end of the variable `tibco.env.CUSTOM_PATH` in the `.tra` file. This is only for Microsoft Windows platforms.

Deploying the Adapter

To deploy the adapter:

1. Check if the SQL script `<OracleAppsInstanceName>_Items_all.sql` is created in the folder `<TIBCO_HOME>/adapter/adorapps/<version_num>/sql`.
2. Open a command window and enter:
`cd <TIBCO_HOME>/adapter/adorapps/<version_num>/sql`.
3. Enter `sqlplus adapter/adapter@vis` in the command line to execute `SQL*Plus`.
4. Execute this script from the `SQL*Plus` prompt on the Oracle Applications Database:

```
SQL>@<OracleAppsInstanceName>_Items_all.sql.
```

This script creates the database objects required to run the adapter.



Enter the username, password, and the connect string to connect to the database before the execution of this script.

Before starting the adapter, edit the properties file required to run the adapter.

Using a text editor, open the file `adbagent.tra` present in `<TIBCO_HOME>/adapter/adorapps/<version_num>/bin` and specify the appropriate values for the repository, configuration URL's and other value as below:

```
— tibco.repourl <Location of .dat file created>
— tibco.configurl /tibco/private/adapter/<ADB Instance Name>
— To customize the .tra file, specify the value for the variable named
  application.args as follows:
  application.args adbagent --propFile <Name of the tra file with path, for
  example, D:/TIBCO_Repos/Sample.tra >
— adb.dsn <Data Source Name for Oracle Applications>
— adb.user <Database schema name where the tibco object has been created>
— tibco.clientVar.adb.password <Corresponding password for the database
  schema mentioned above>
— adb.debug 3
— adb.verbose on
```


Start the Adapter

To start the adapter:

1. Start the JMS Server. Run the `tibjmsd.exe` file from your `../JMS/bin` directory.
2. Certain changes to the environment variable settings are required before running TIBCO Adapter for Oracle Applications on the UNIX platform. Execute the appropriate file to make these changes automatically.



Ensure that you have completed the tasks in [Post Installation Tasks on UNIX on page 68](#) and [Set Environment Variables on page 70](#) before you continue.

For `csh`, execute `<ADORAPPS_HOME>/bin/adorappssetenv.csh`, as shown:

```
% source adorappssetenv.csh
```

For `sh` and `ksh`, execute `<ADORAPPS_HOME>/bin/adorappssetenv.sh`, as shown:

```
$ . ./adorappssetenv.sh
```

The file modifies the following environment variables on UNIX platforms:

- `PATH`
- `SHLIB_PATH` (on HP-UX only)
- `LD_LIBRARY_PATH` (on Solaris)
- `LIBPATH` (on AIX only)

3. In `<TIBCO_HOME>/adapter/adorapps/<VERSION>/bin`:

If you have used `adbagent.tra` as your property file, execute the command **adbagent** in the command prompt.

If you have used your own property file, specify the command as follows:

```
adbagent --run -propFile <Name of tra file with path, for example,  
D:/TIBCO_Repos/Sample.tra >
```

4. Log into Oracle Applications and create a new Item or update an existing Item. A message will be published with the specified subject.

Stop the Adapter

To stop the adapter:

1. Type `adbstop <InstanceName>` from the command line. The `adbstop` executable is located at `$TIBCO_ADORAPPS_HOME/bin` (This option is only for UNIX platform).
2. Broadcast a stop message using the `tibrvsend` command-line tool available in the `TIBCO_HOME/tibrv/bin` directory. The convention of the subject is `<hostname><instanceName>.STOPAdapter` and the convention of the message value is `now`.
3. To stop `adbagent`, click **TaskManager > Processes > adbagent > End Process**.
4. Once the adapter receives the message with a stop subject, it will complete any pending transactions and clean up resources before it terminates.
5. Close the command prompt window in which the TIBCO Adapter for Oracle Applications is running or press **Ctrl-C** in the window.

Chapter 4 **Configuring the Adapter**

This chapter explains how to configure the adapter with adapter services that are used to communicate with Oracle Applications.

Topics

- [Overview, page 98](#)
- [Configure the Adapter Overview, page 99](#)
- [Oracle Applications Business Object Configuration, page 109](#)
- [PreDefined Outbound \(Publisher\) Oracle Applications Business Object Configuration, page 112](#)
- [Custom Outbound \(Publisher\) Oracle Applications Business Object, page 124](#)
- [PreDefined Inbound Current Program \(Subscriber\) Oracle Applications Business Object Configuration, page 135](#)
- [PreDefined Inbound API \(Subscriber\) Oracle Applications Business Object Configuration, page 144](#)
- [Custom Inbound \(Subscriber\) Oracle Application Business Object Configuration, page 149](#)
- [Oracle Applications Inbound Business Object \(For R12 Only\), page 159](#)
- [TIBCO Log Table Publisher Business Object Configuration, page 192](#)
- [Oracle Publisher Configuration, page 200](#)
- [Oracle Subscriber Configuration, page 214](#)
- [Deleting a Publisher or a Subscriber, page 229](#)
- [Customizing a Business Object, page 230](#)
- [Conditional Publishing, page 237](#)
- [Impact of Edit Actions on Database Objects, page 243](#)
- [Setting Encoding Options, page 245](#)

Overview

You can configure the adapter before starting it.

An adapter is configured using TIBCO Designer. Make sure it has been installed correctly before the configuration (see the *TIBCO Designer User's Guide*). Select **Start > All Programs > TIBCO > TIBCO Designer > Designer Documentation** to access the TIBCO Designer User's guide. Select **Start > All Programs > TIBCO > TIBCO Designer > Designer** to start TIBCO Designer.

Configure the Adapter Overview

This chapter gives an overview of how to configure the adapter. It provides instructions on basic operations such as starting and stopping TIBCO Designer, adding resources, saving projects, and TIBCO Designer concepts.

Projects

A *project* is a user-designed information integration. It specifies the way in which the publisher and subscriber handle information or the business logic which will apply to the information. Each TIBCO Designer window contains one project, which is represented as a top-level folder in the window.

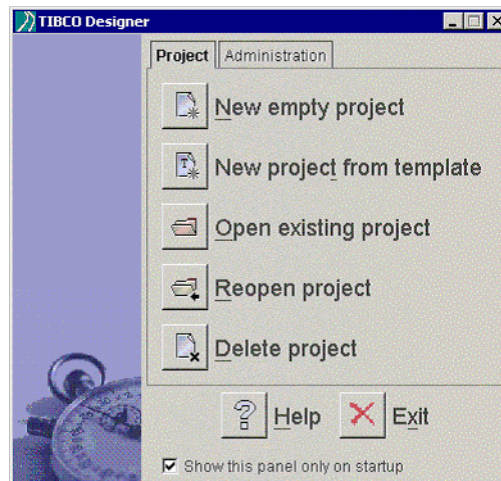
A project is a collection of related *resources*. A TIBCO Designer resource corresponds to an object in a TIBCO application, such as a specific publisher or a specific adapter instance. Resources are displayed in both the project tree and the design panel of the TIBCO Designer window.

Palettes contain the resource templates that you can use to configure your project. The TIBCO Adapter for Oracle Applications palette and the TIBCO ActiveMatrix Adapter for Database palette are loaded into TIBCO Designer during installation.

A typical configuration example is provided as below:

1. Select **Start > All Programs > TIBCO > TIBCO Designer > Designer** to start the TIBCO Designer and select the **New empty project** option (see Figure 46).

Figure 46 The TIBCO Designer Dialog




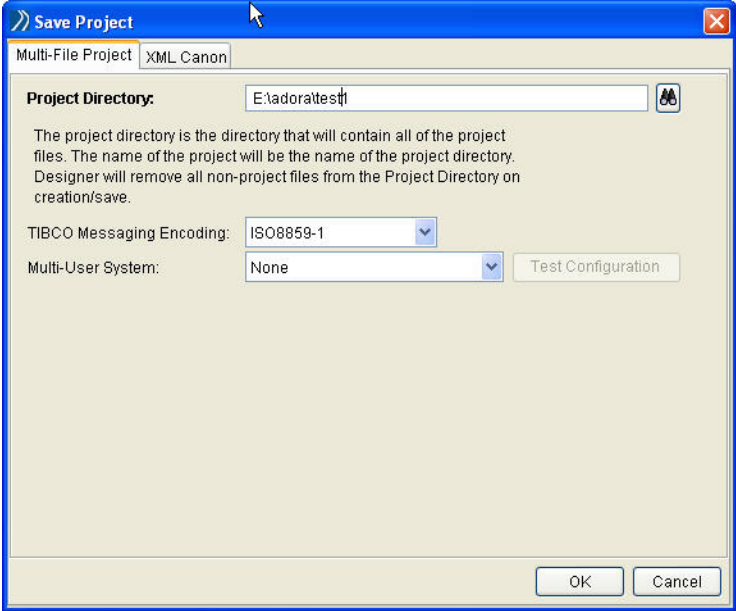
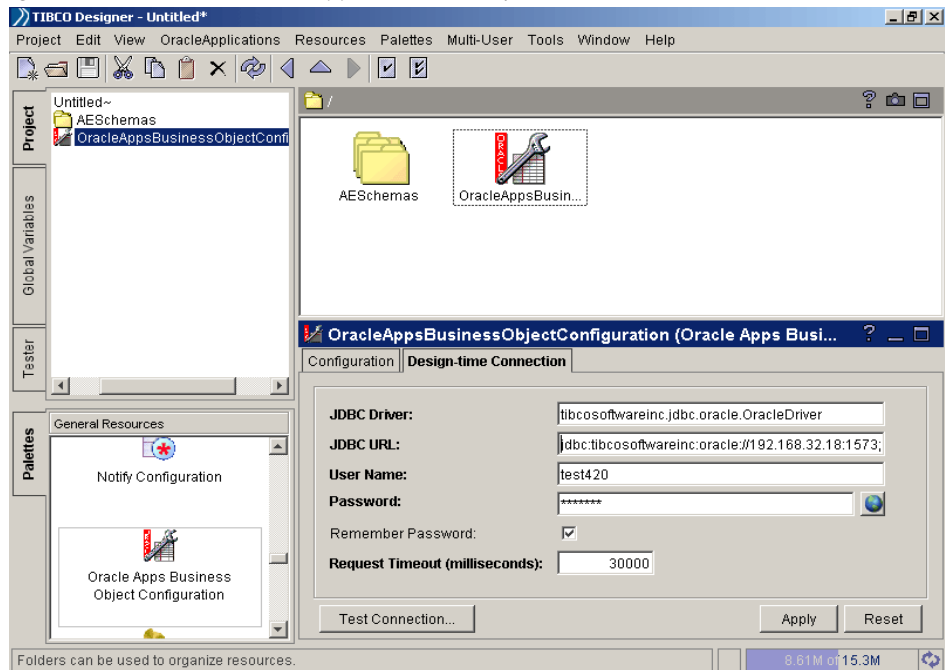
- 2. Select the **Multi-File Project** tab and click the  button to select the location of the project. Click the **OK** button to continue to create a new Multi-File project (see Figure 47).

Figure 47 The Save Project Dialog



3. Drag the **Oracle Apps Business Object Configuration** button from the palette panel and drop it into the design panel (see Figure 48).

Figure 48 Create an Oracle Apps Business Object

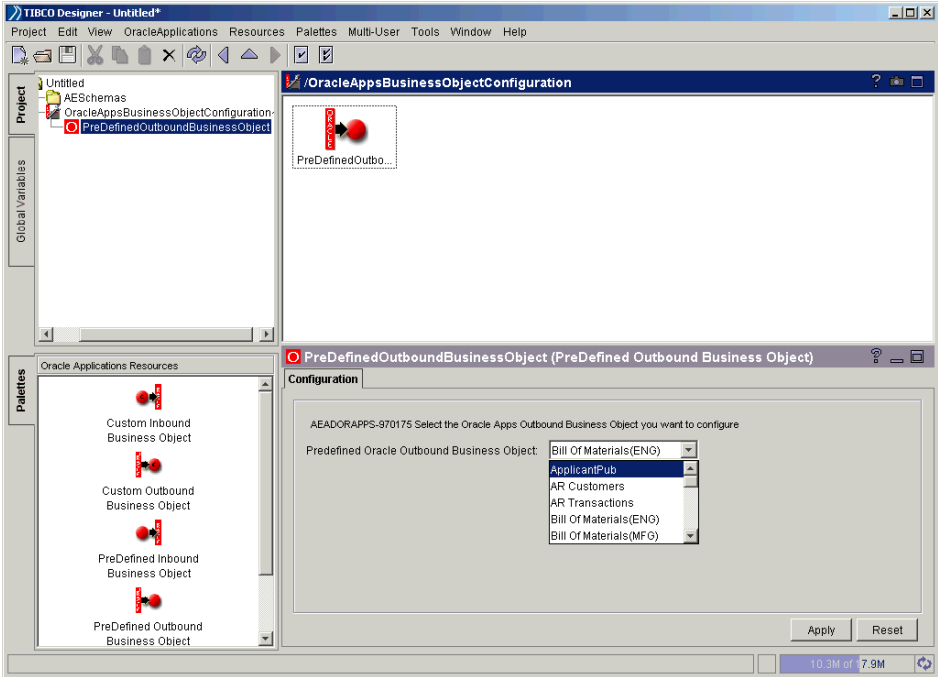


4. Enter the appropriate connection parameters in the **Design-time Connection** tab.
5. Click the sub folder named **OracleAppsBusinessObjectConfiguration** in the project panel. You will see a list of six **Business Object** buttons in the palette panel.

The TIBCO Adapter for Oracle Applications has a set of PreDefined Inbound Business Objects (Subscriber) and PreDefined Outbound Business Objects (Publisher) that you can configure. The adapter also provides the flexibility to create, modify, or delete a **Custom Inbound Business Object** or **Custom Outbound Business Object**.

6. Drag the **Custom (or PreDefined) Outbound Business Object** button or the **Custom (or PreDefined) Inbound Business Object** button from the palette panel and drop it into the design panel. Add the required tables and configure the various parameters. All the parameters required for an adapter service should be given in the Oracle Business Object during configuration. Select a Business Object from the drop-down list in the Configuration tab (see Figure 49).

Figure 49 The Configuration Tab of a Business Object

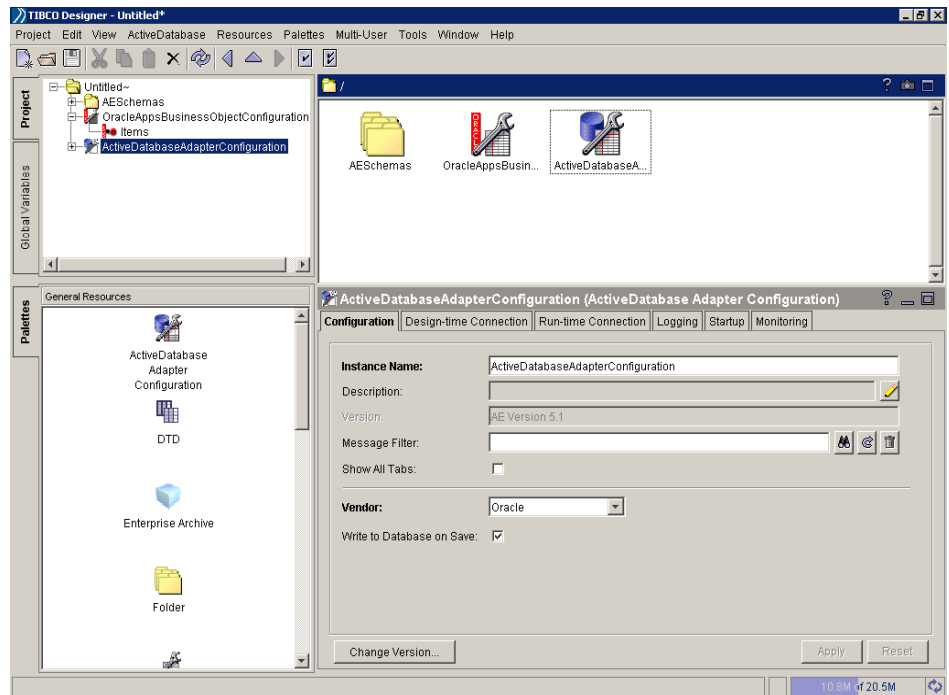


7. Select the root folder in the project tree panel. Drag the **ActiveDatabase Adapter Configuration** button from the palette panel and drop it into the design panel (see Figure 50).



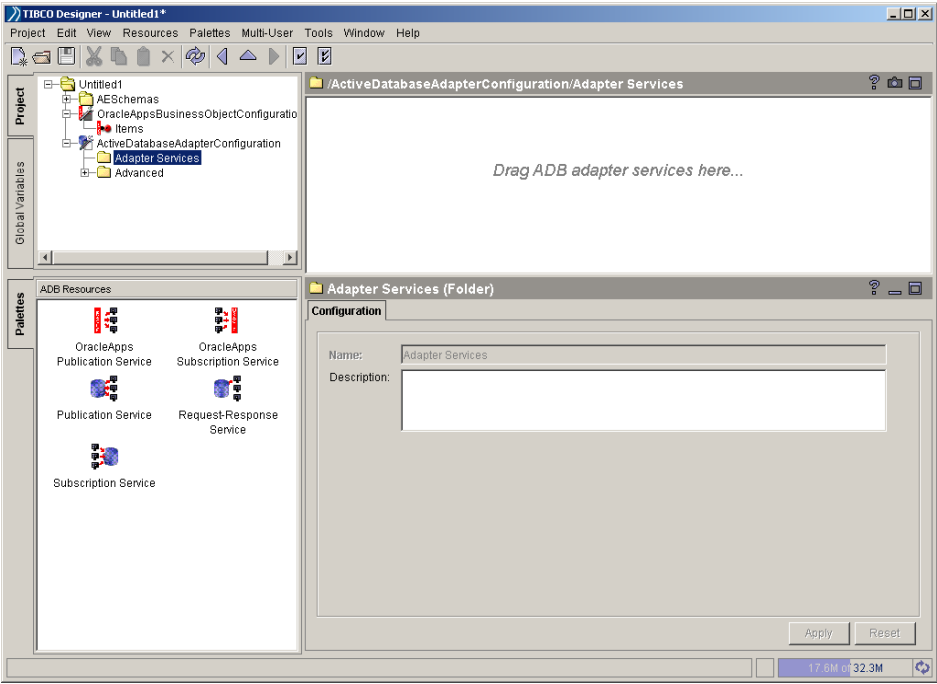
All the required configuration information should be provided with the Oracle Applications Business Objects.

Figure 50 Add the ActiveDatabase Adapter Configuration Button



8. Specify the adapter instance name in the Instance Name field and clear the **Write to Database on Save** checkbox in the Configuration tab.
9. Enter the appropriate connection parameters in the **Design-time Connection** tab and click the **Apply** button.
10. In the project panel expand the ActiveDatabaseAdapterConfiguration resource and double-click the Adapter Service folder or node (see Figure 51).
11. Drag the **OracleApps Publication Service** or **OracleApps Subscription Service** button from the palette panel and drop it into the design panel.

Figure 51 The Adapter Services Node




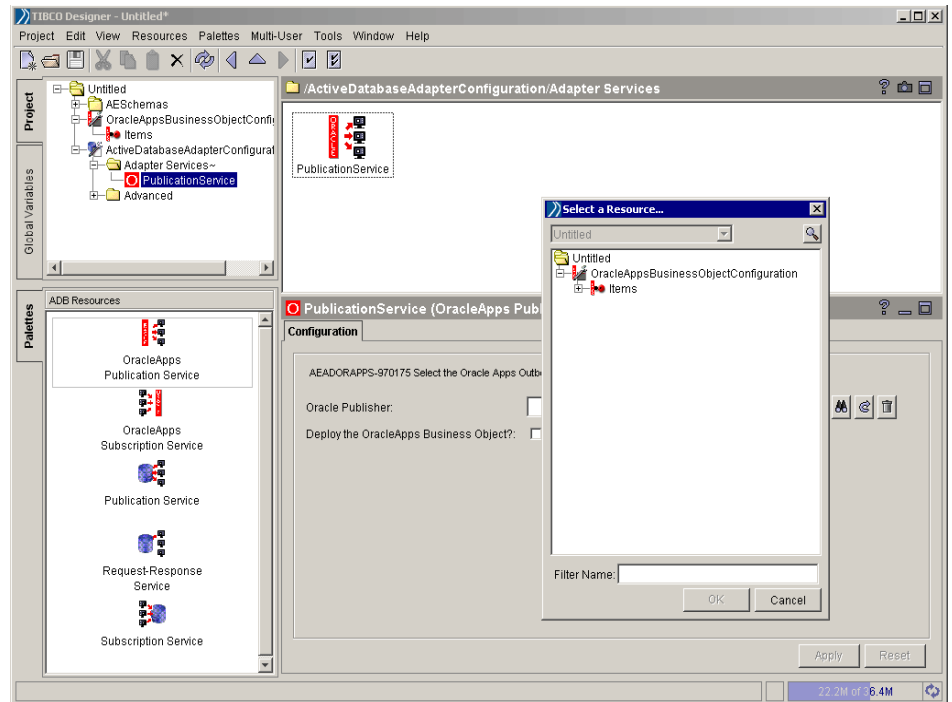

12. Click the  button in the Configuration tab to select the Oracle Inbound or Outbound Business Object configured previously. If you need to deploy the Business Object on the database, check the **Deploy the Oracle Business Object?** checkbox and click the **Apply** button (see Figure 52).

Figure 52 Select a Previously Configured Oracle Inbound or Outbound Business Object



13. Select the adapter service resource in the design panel or project panel to view its configuration values in the configuration panel. DO NOT modify the schema information; that is, the **Class Reference** and the **Endpoint Reference** field in the Advanced tab of the created adapter service. However, you can specify the **Name**, **Transport type**, and **Wire Format** field.
14. Select **Project > Export Full Project** to open an Export Project dialog. You can see the default value in the User field.
15. Name your project and input the Project Name in the Project Name field. Click the  button at the right hand side of the Dir Name field to specify the directory path where the local repository.dat will be saved and click the **OK** button.
16. The Create Project Confirmation Dialog will either remind you to create a new project or to overwrite an existing project. Leave all the settings in this dialog box as default and click the **Yes** button to confirm the Create Project activity.

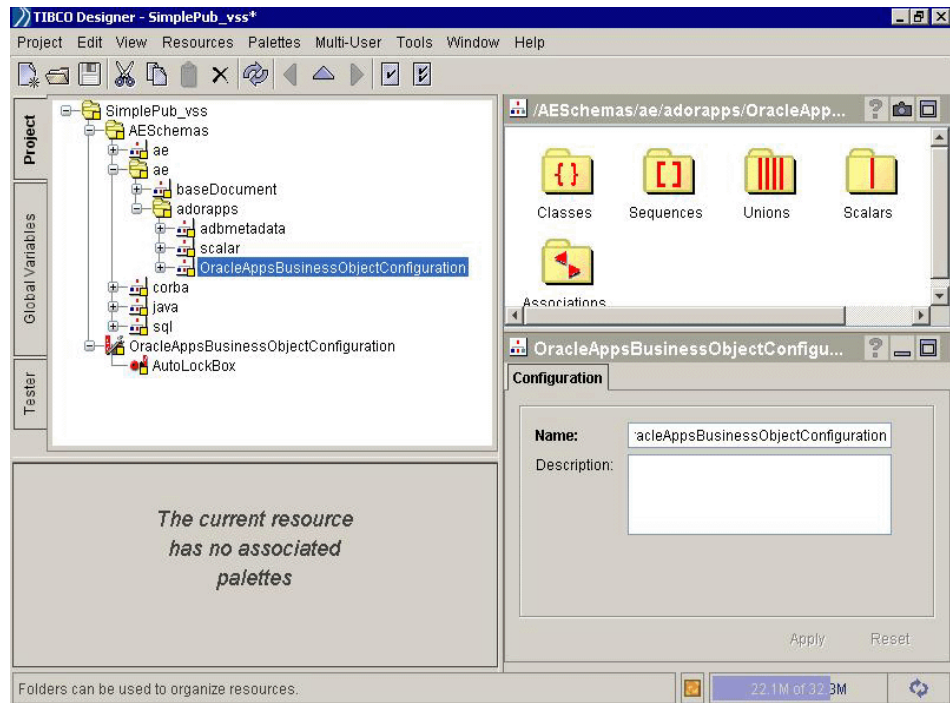
17. Modify the runtime properties file for the adapter with the project name and adapter instance name. Start the adapter instance you have configured. Refer to [Start the Adapter on page 95](#) for more details.

Using the Adapter with a Revision Control System

TIBCO Designer supports revision control systems such as MicroSoft Visual SourceSafe and Perforce. If you are using a revision control system, you must manually add some configured resources to the revision control system and check in the resources after the instance configuration.

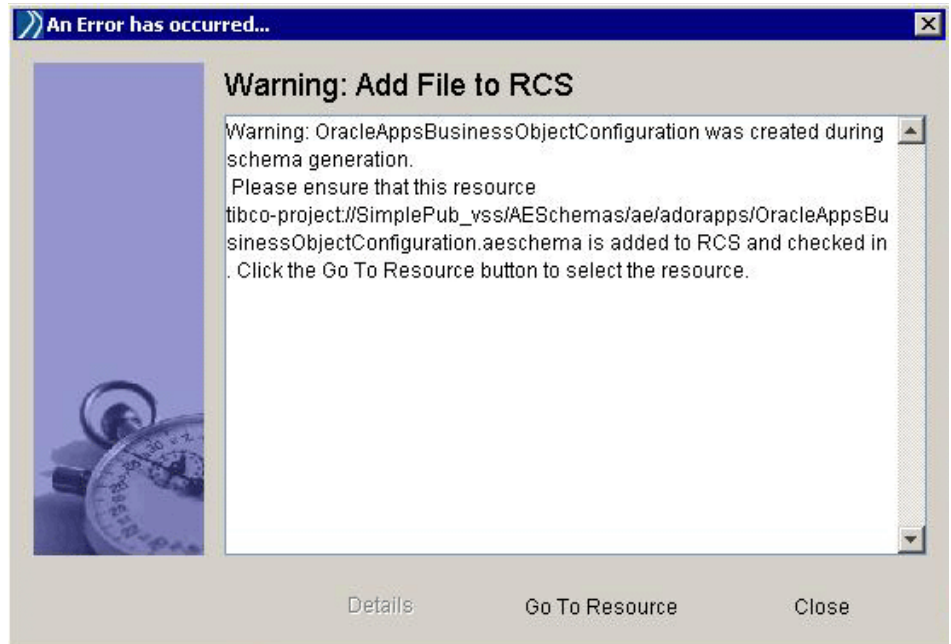
As part of the service configuration, the adapter creates schema files in the root/AESchemas/ae/adorapps directory. For example, if you configure a business object in an adapter configuration named Instance1, the following file is created:

Project_root/AESchemas/ae/adorapps/Instance1.aeschema



When the project is saved and a revision control system has been specified, the adapter displays a warning which indicates that additional files were created and will be added to the revision control system. This warning appears only when the files are created for the first time. The Go To Resource button is located at the bottom of the warning message that allows you to view the resource (see Figure 53). You can click **Multi-User > Add Resources** from the RCS menu command to add these files to the revision control system.

Figure 53 The Additional Files Creation Warning Message



For information on how to use the Multi-User feature in TIBCO Designer, refer to the *TIBCO Designer User's Guide*.

Copy, Cut, Paste and Move Operations

To successfully copy and paste a service from adapter Instance1 to Instance2, the adapter configuration and schema files for the Instance2 must be checked out.

To successfully cut and paste a service from adapter Instance1 to Instance2, the adapter configuration and schema files for both Instance1 and Instance2 must be checked out.

To successfully move a service from adapter Instance1 to Instance2, the adapter configuration and schema files for both Instance1 and Instance2 must be checked out.

Creating Two Adapter Instances in Same Project by Different Users

To create two adapter instances in the same project by two users:

1. Create a new project called **myTestProj**.
2. Save the project using a revision control system, for example MicroSoft Visual Source Safe.
3. Click the root folder and add the files to the revision control system. Click the **yes** button for recursive addition.
4. Check in the project.
5. Sync the project on two machines, Machine A and Machine B.
6. Open the project on Machine A and configure Instance1. Add the Instance1** and the related schema files to the revision control system. Check in the changes.
7. Open the project on Machine B and configure Instance2. Add the Instance2** and the related schema files to the revision control system. Check in the Changes.



The global variables must be populated first otherwise the you cannot configure the instances.

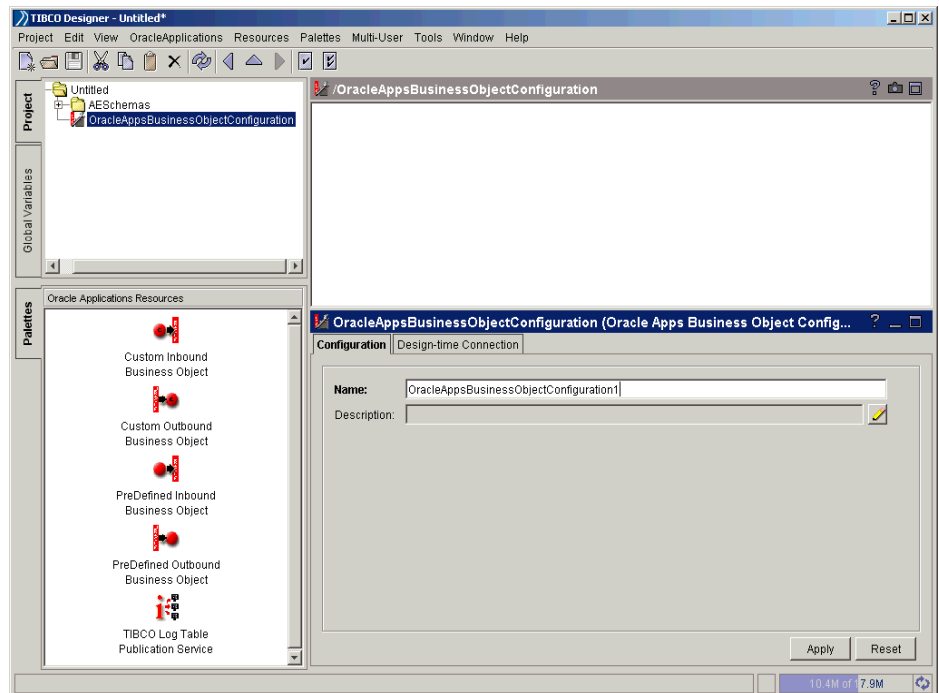
Oracle Applications Business Object Configuration

This instance has two tabs, the Configuration tab and the Design-time connection tab.


Configuration Tab

The Configuration tab consists of Name and Description (see Figure 54). Enter the name of the instance in the Name field. The Description field contains the description of the TIBCO Adapter for Oracle Applications Business Object. This field can be edited.

Figure 54 The Configuration Tab of the OracleAppsBusinessObjectConfiguration Object

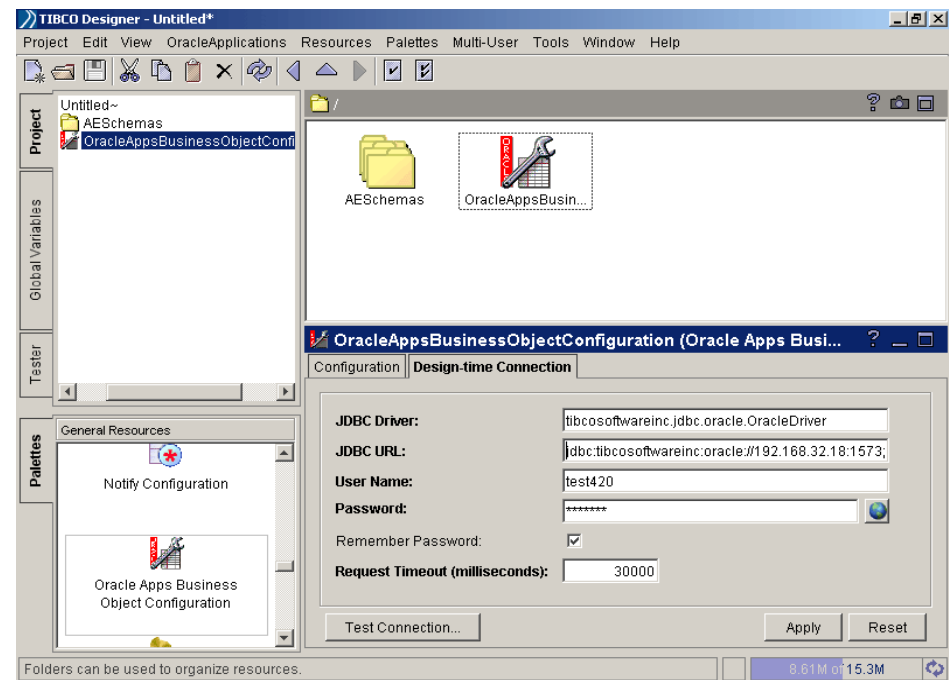


Design-time Connection Tab

Under the **Design-time Connection** tab there are six fields: JDBC Driver, JDBC URL, User Name, Password, Remember Password, and Request Timeout. The  button discards the current value in the field and you can replace its value with a global variable. Enter the appropriate value for all these parameters and click the **Test Connection** button. If the values entered are correct, it will display a Connection Successful message, otherwise, a Connection Failed message is displayed.

These connection parameters are used for establishing the connection to the Oracle Applications Database and fetching the required schemas. Click **Oracle Applications > Save Connection Setting** from the menu bar to save these settings. Saved settings can be found under Oracle Applications > User Connections from the menu bar (see Figure 55).

Figure 55 The Design-time Connection Tab of the OracleAppsBusinessObjectConfiguration Object



The following is a list of Business Objects that can be configured.

- PreDefined Outbound (Publisher) Oracle Applications Business Object Configuration

- Custom Outbound (Publisher) Oracle Applications Business Object Configuration
- PreDefined Inbound (Subscriber) Oracle Applications Business Object Configuration
- Custom Inbound (Subscriber) Oracle Application Business Object Configuration
- Oracle Applications Inbound Business Object (For R12 Only) Configuration
- TIBCO Log Table Publisher Business Object Configuration



- For Custom Inbound transactions, give explicit grants (select) to the Adapter User schema (on the Interface tables that are being used). This has to be done before you can start with the configuration.
- For Custom Outbound transactions, give explicit grants (select) on the Oracle Source tables to the Adapter User schema (which are a part of the Business Object). This has to be done before you start the configuration.
- Confirm that the script `common_all.sql` is run on the TIBCO Adapter for Oracle Applications database before configuring the TIBCO Adapter for Oracle Applications Business Objects. Execute the `common_all.sql` script after the adapter is installed. This step has to be done only once.

PreDefined Outbound (Publisher) Oracle Applications Business Object Configuration

To configure a PreDefined Outbound Oracle Applications Business Object:

1. Drag the **PreDefined Outbound Business Object** button from the palette panel and drop it into the design panel.
2. Select the required Business Object from the PreDefined Outbound Business Object drop-down list under the Configuration tab and click the **Apply** button.
3. The Configuration tab for the Business Object is displayed.
4. Add entries in the following fields and click the **Apply** button.
 - Table – Tablespace
 - Index – Tablespace
 - APPS User Name (Mandatory)
 - APPS User Password (Mandatory)
5. Specify the view names in the Tables and View Mappings tab and click the **Apply** button. You will see a Views tab has been added in the configuration panel.

This tab is used for adding Join tables. Any table added in this tab is considered as a Join table. This table is included in the View definition and the columns selected in the join table is published along with the columns based on the source table.



Do NOT add tables to a Join table. Tables can only be added to views, and NOT to tables added as Join tables.

For details on Customizing a PreDefined Outbound Business Object, See [Customizing a Business Object on page 230](#) in this chapter.

You can set parameters under the following tabs:

- [The Configuration Tab](#)
- [The Tables Tab](#)
- [The Adapter Options Tab](#)
- [The Views Tab](#)

The Configuration Tab

The **Configuration** tab has the following options:

Initially, you have to verify the type of the PreDefined Oracle Outbound (publish) Business Object by selecting a Business Object from the PreDefined Outbound (publish) Business Object drop-down list. After the selection has been applied, you can configure all the parameters for this specific type of Business Object.



The following parameters will only be available after a PreDefined Outbound Business Object has been selected and applied.

- Name

Displays the service name associated with the PreDefined Business Object.

- Table - Tablespace

This specifies the tablespace name in which the tables should be created. If this field is left blank, database tables required for the Business Object are created in the default tablespace (allocated by the database system).

- Index - Tablespace

This specifies the tablespace name in which the indexes should be created. If this field is left blank, database indexes required for the Business Object are created in the default tablespace (allocated by the database system).

- APPS User Name

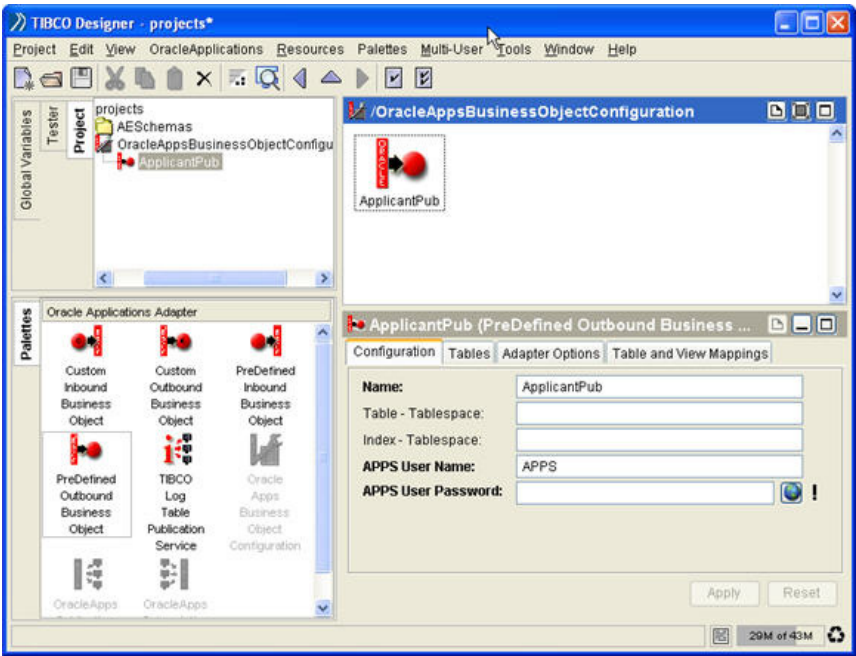
The user name for connecting to the Oracle Applications database user (schema) of the Oracle Applications database. This is a mandatory field.

- APPS User Password

The user password for connecting to the Oracle Applications database user (schema) of the Oracle Applications database. This is a mandatory field.

Sample Screen

Figure 56 PreDefined Outbound Publisher Configuration Tab




The Tables Tab


The **Tables** tab has the following buttons and column headings:





The buttons and column headings mentioned below are disabled for all Predefined Publication transactions.


Buttons

- The Add Table button 

Click to display a dialog box that lists tables available to the database user specified in the adapter Connection tab. Select the table that is to be published when data is inserted into it.
- The Add Details Table button 

Displays a dialog box from which a secondary table can be added to the configuration.
- The Add Other Table button 

Allows you to enter a schema from which available tables are displayed. If the schema password is different from the schema name, a dialog will prompt you to enter the password for that schema.
- The Remove Table button 

Deletes the selected table from the list.
- The Re-find Table from Database button 

Causes TIBCO Designer to refresh stored table schema information by retrieving new information from the database.
- The Allow Key Columns Only Checkbox

A key column or substitute key column is required when publishing by reference, since the publishing table contains only key values. If no column is specified, the publication is not added.

Columns

- Tables and Columns

Lists the table and its columns.
- Type

Lists the primitive type.
- AE Type

Lists the primitive type mapped to an ActiveEnterprise type.
- User Key

Check to define as a user key.

- Use?

Click to publish data in the column after the data insertion.

- Join To

Joins two tables as a parent and a child.

This tab is not available for setting any values. The outbound services are based on views that can be configured using the Views tab. For details on configuring the Views tab see [The Views Tab on page 121](#).



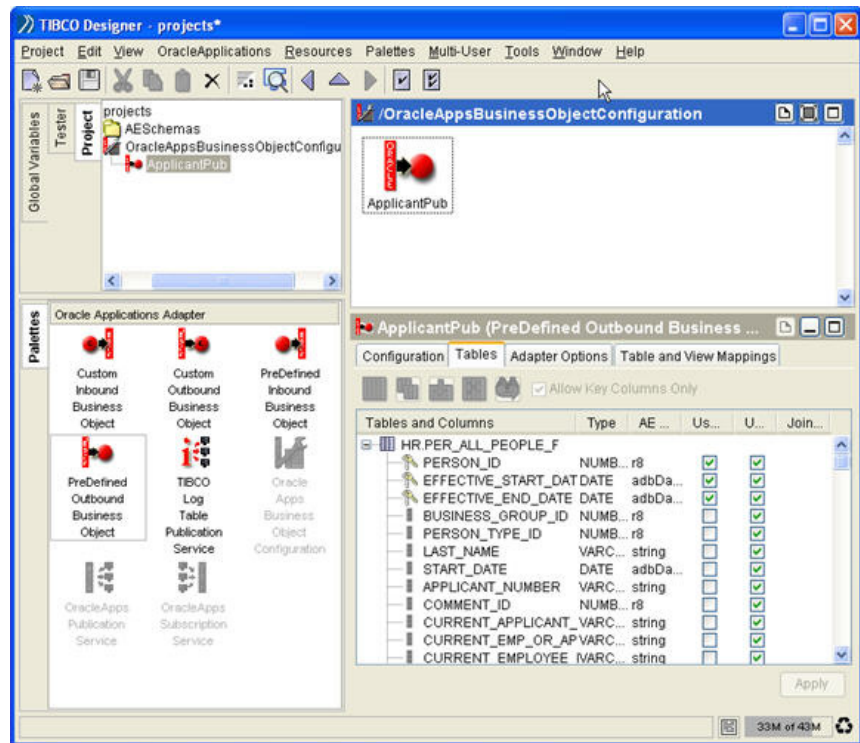
When publishing by reference, a key column or substitute key column is required for populating the publishing table. If the source table has no primary key, it can be set up to publish by reference only if a non-key column is specified when adding the publication. All tables must have at least one column specified as a User Key column. This is required for creating the publisher service database objects based on the source tables.



A child table column that is used as a join Column should be specified as a Key Column.

Sample Screen

Figure 57 PreDefined Outbound Publisher Tables Tab



The Adapter Options Tab



To prevent an error, ensure that the format for both the publisher and subscriber are the same.

The Adapter Options tab has the following options:



The Adapter options mentioned below are disabled for the Predefined Publisher.

- Storage Mode

This field cannot be modified because the Oracle Applications Business Objects use publishing from database views. The default value is Published by Reference.

- Publishing by value copies all specified columns from the source table to the publishing table.
- Publishing by reference copies only key column values to the publishing table.
- If no key column is defined in the database, a substitute non-key column must be defined to Publish by reference.

- Publishing Table

This field is disabled because there are dependencies on database objects associated with the PreDefined Business Object. Possible value is a database table name. This value is set to the one specified in the Business Object properties.

- The table name cannot be qualified by a schema name.
- The publishing table cannot contain any user-created columns where the column name starts with ADB_. These characters are reserved for the adapter.
- By convention, the publishing table should have the prefix P_. For example, if your source table is called MY_ORDER, its publishing table should be named P_MY_ORDER.
- The maximum length of a publishing table name is 30 characters.

- Referred Object

By default, the Business Object will be published from a view. The views can be configured separately in the Views tab after specifying appropriate view names in the Tables and View Mappings tab. This field is populated once the Views are configured (See [The Views Tab, page 121](#)).

Sample Screen

Figure 58 PreDefined Outbound Publisher Adapter Options Tab - Publish by Value

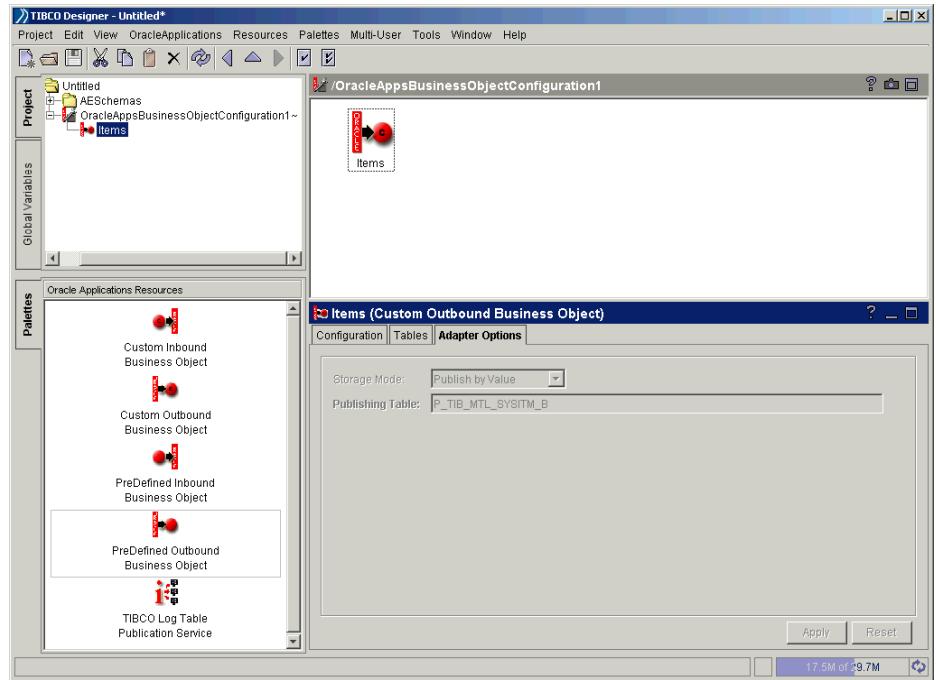
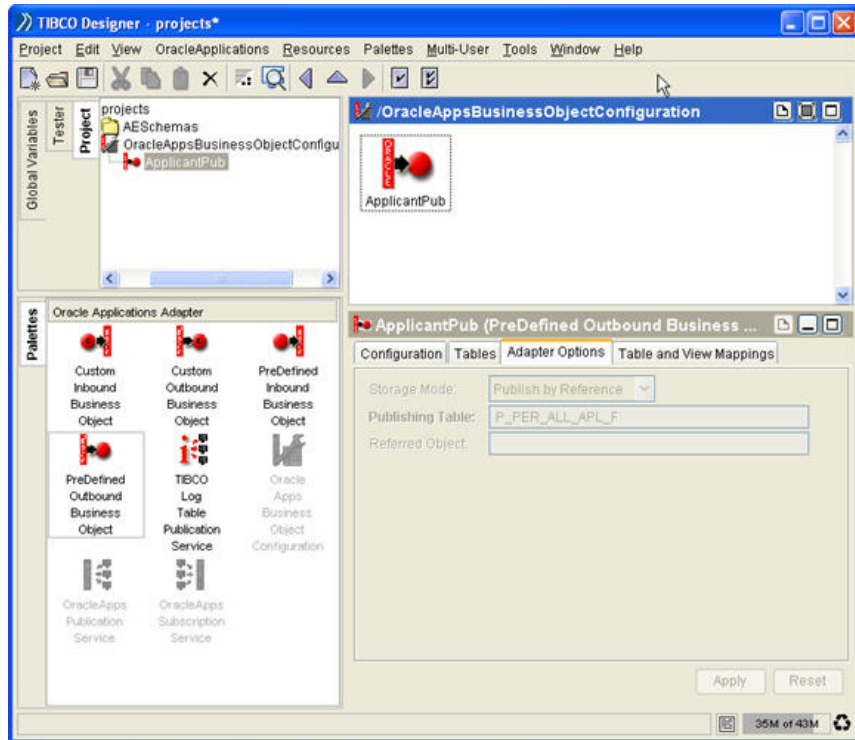


Figure 59 PreDefined Outbound Publisher Adapter Options Tab - Publish by Reference



The Tables and View Mappings Tab

The view names for tables in the Business Object are populated by setting parameters in the Tables and View Mappings tab. This tab is displayed when the value for Storage Mode is specified as Publish by Reference.



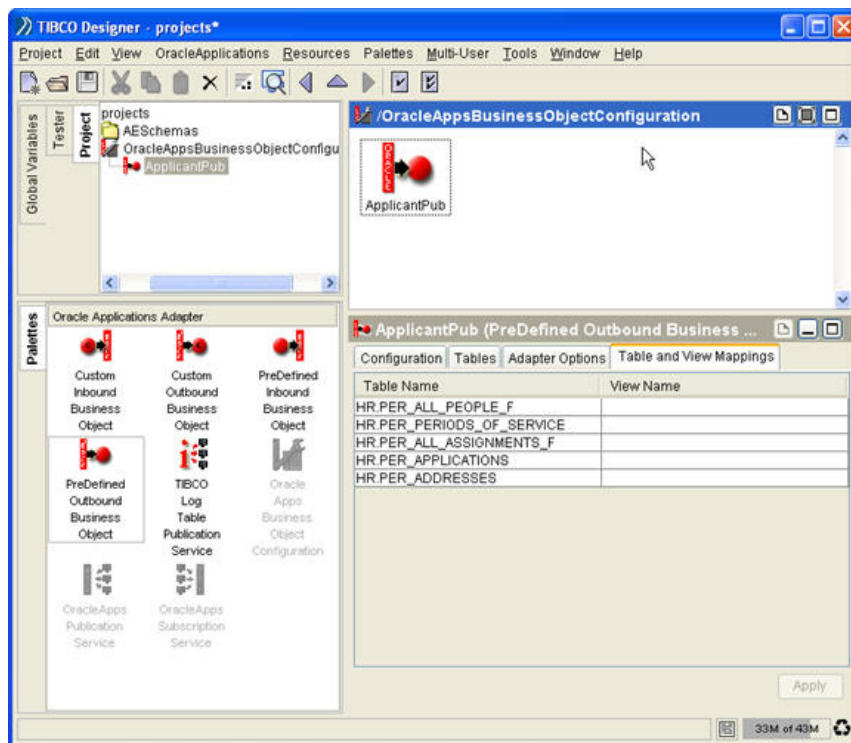
A view is a tailored presentation of the data contained in one or more tables (or other views). A view takes the output of a query and treats it as a table. Hence, a view can be called a *stored query* or a *virtual table*. You can use views in most places where a table can be used.

A valid view name should be entered for each table. The view name is validated for a standard Oracle database view. The maximum length of an Oracle view is 30 characters.

Once all the view names are specified, click the **Apply** button to set the values. If the view names are valid, a **Views** tab will display. For detailed information on how to configure the Views tab, refer to [The Views Tab on page 121](#).

Sample Screen

Figure 60 PreDefined Outbound Publisher Tables and View Mappings Tab



The Views Tab

The view names provided in the Tables and View Mappings tab are used to generate corresponding view objects for the tables present in the Business Object.

By default, the views contain all columns in the tables. These columns can be selected or cleared depending on whether these are to be included for publishing.

The views are based on the source tables. However, any number of JOIN tables can be specified if fields from other tables are to be included in the view for publishing.

Selected columns from the tables added in this tab is included in the view definition for publication.


The **Views** tab has the following buttons and column headings:

- Buttons**
- The Add Details Table button 


Displays a dialog box from which a join table can be added to the configuration from the default user schema provided in the adapter instance Connection tab.

- The Add Other Table button 

Allows you to enter a schema from the displayed available tables. If the schema password is different from the schema name, a dialog will prompt you to enter the password for that schema.

- The Remove Table button 

Deletes the selected JOIN table from the list.

- The Re-find Table from Database button 

Causes the TIBCO Designer to refresh stored table schema information by retrieving new information from the database.

Columns

- Tables and Columns

Lists the all the join tables and their columns.

- Type

Lists the primitive Oracle database type.

- AE Type

Lists the primitive type mapped to an ActiveEnterprise type.

- Use?

Check the box to publish data in the column after insertion.

- Use in Join?

Check the box to add a new column in the view table and use this column to join with other tables.

- Join With

This should be used to specify the join clause during the creation of a view. After checking the Use in Join column, a drop-down box will appear in the Join With column containing a list of attributes used in the Use in Join column. The sub clause `Join` is essential in the `WHERE` condition when you want to link two tables together. The join created is a standard Oracle `AND join`.

The Views tab will refresh itself automatically as the value changes in the Tables and View Mappings tab.

The view specified for the main table is set as the Referred Object and is visible in the Adapter Options tab. See [The Adapter Options Tab on page 117](#) for more details.

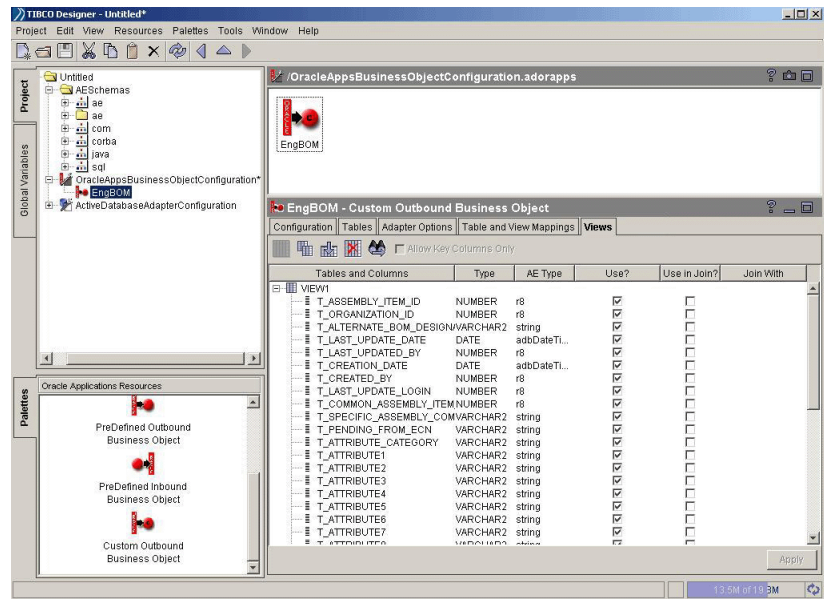


Columns with custom data types in the Oracle Application database are ignored during configuration. These columns will not be displayed and will not be available for configuration.

For example, consider the column GEOMETRY of the complex data type SDO_GEOMETRY in the table AR.HZ_LOCATIONS of the ARCustomers transaction, the column GEOMETRY of data-type SDO_GEOMETRY in the table HR.PER_ADDRESSES of EmployeePub and ApplicantPub transactions. The ODBC driver does not support complex data type. You must create views for all tables associated with the transactions ARCustomers, ApplicantPub and EmployeeSub. The Adapter stops abruptly when views are not configured for these three transactions.

Sample Screen

Figure 61 PreDefined Outbound Publisher View Tab



Custom Outbound (Publisher) Oracle Applications Business Object




Drag a Custom Outbound Business Object resource button from the palette panel and drop it into the design panel. The Configuration tab appears in the configuration panel.

To configure a Custom Outbound Oracle Applications Business Object:

1. Drag the **Custom Outbound Business Object** button from the palette panel and drop it into the design panel.
2. The Configuration tab is displayed. Enter the appropriate values in the following fields and click the **Apply** button.

- Table – Tablespace
- Index – Tablespace
- APPS User Name (Mandatory)
- APPS User Password (Mandatory)

3. The Tables tab is available for adding tables that will constitute the Business Object.

Tables can be added using the Add Table button , Add Details Table button , and Add Other Table button . Relationships between the tables can be specified using the User Key and Join To. properties (specified as key columns) in a parent table that appears in the drop-down list under the Join To column property. The required parent table column can then be used for joining with the child table in a parent-child relationship.

All Classes, Associations and Sequences are based on this relationship. Please refer to the TIBCO Designer Schema Palette Documentation.

4. In the Adapter Options tab, the Storage Mode drop-down menu is disabled when transactions involve multiple tables.
5. Select the **Publish by Reference** option in the Storage Mode drop-down menu and the Referred Object field will appear under the Adapter Options tab. Enter an appropriate table name for the Publishing Table field and click the **Apply** button. The Table and View Mappings tab will be added once the publishing table is successfully applied.
6. In the Tables and View Mappings tab
 - Specify the view names column.
 - Click the **Apply** button.

7. After the configuration of the View, The Referred Object field will populate and the Views tab will be added.

This tab is used for adding Join tables. Any table added in this tab will be considered as a *Join table*. This table is included in the View definition and the columns selected in the join table is published along with the columns based on the source table.

After a column's Use in Join property is set to true (in the source table view), it can be used in joining with a Join table.



Do NOT add tables to a Join table. Tables can only be added to views, but *cannot* be added to Join tables.

For details on Customizing a Custom Outbound Business Object, please refer to the section [Customizing a Business Object on page 230](#) in this chapter.

You can set parameters under the following tabs:

- [The Configuration Tab](#)
- [The Tables Tab](#)
- [The Adapter Options Tab](#)
- [The Tables and View Mappings Tab](#)
- [The Views Tab](#)

The Configuration Tab



The following parameters are available only after the selection and application of a PreDefined outbound business object.

The Configuration tab has the following options:

- Name
Displays the service name associated with this PreDefined service.
- Table - Tablespace
Specifies the tablespace name for newly created tables. If this field is left blank, database tables required for the Business Object is created in the default tablespace (allocated by the database system).
- Index - Tablespace

Specifies the tablespace name for newly created indexes. If this field is left blank, database indexes required for the Business Object is created in the default tablespace (allocated by the database system).

- APPS User Name

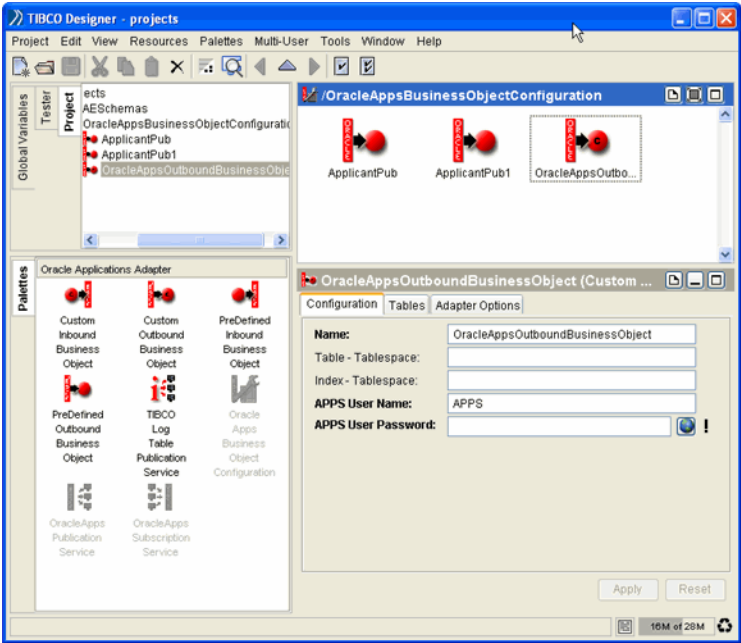
Specify the User Name to connect to the Oracle Applications database user (schema) of the Oracle Applications database.

- APPS User Password

Specify the user password to connect to the Oracle Applications database user (schema) of the Oracle Applications database.

Sample Screen

Figure 62 The Custom Outbound Publisher Configuration Tab



The Tables Tab

The Tables tab has the following buttons and columns:

- Buttons
- The Add Table button


Click to display a dialog box that contains a list of all tables available to the database users. These users are specified in the adapter Connection tab. Select the table to be published when data is inserted into it.

- The Add Details Table button 


Displays a dialog box for you to add a secondary table to the configuration.

- The Add Other Table button 

Allows you to enter a schema from which available tables are displayed. If the schema password is different from the schema name, a dialog will prompt you to enter the password for that schema.

- The Remove Table button 

Deletes the selected table from the list.

- The Re-find Table from Database button 

Causes TIBCO Designer to refresh stored table schema information by retrieving new information from the database.

- The Allow Key Columns Only Checkbox

A key column or substitute key column is required when publishing by reference, because the publishing table contains only key values. The publication will not be added if no column is specified.

Columns

- Tables and Columns

Lists all the tables and their columns.

- Type

Lists the primitive type.

- AE Type

Lists the primitive type mapped to a TIBCO ActiveEnterprise type.

- User Key

Check to define as a user key.

- Use?

Click to publish data in the column after the data insertion.

- Join To

Joins two tables as a parent and a child.

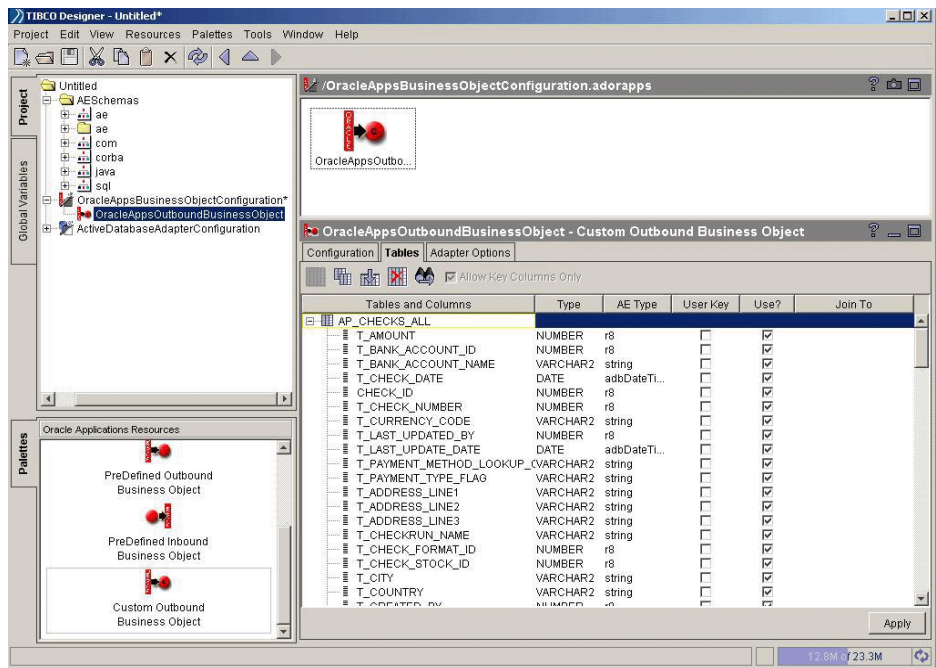
The outbound services can also be based on views that can be configured in the Views tab explained in the subsequent sections. See [The Views Tab on page 132](#) for details.



All tables must have at least one column specified as a Key column. This is required for creating the Materialized View logs on the source tables.

Sample Screen

Figure 63 The Custom Outbound Publisher Tables Tab



The Adapter Options Tab

The **Adapter Options** tab has the following options:

- Storage Mode

Select the **Publish by Value** option or the **Publish by Reference** option from the Storage Mode drop-down menu.

The Publishing by reference option copies only key column values to the publishing table. If no key column is defined in the database, a substitute

non-key column must be defined to satisfy the Publish by Reference option requirement.

The Publish by Value option copies all specified columns from the source table to the publishing table.

If you select this option, the following restrictions apply:

- Publishing tables cannot include columns with the LONG data type.
- If you have a source table that contains a column with the LONG data type, this column cannot be specified for inclusion in the publishing table. This is because the trigger created by the configuration utility cannot refer to the LONG column via the new construct. This is an Oracle restriction. The problem cannot be detected by Oracle during trigger creation. However, when the trigger fires and it attempts to copy the value of the column with the LONG data type to the publishing table, the database connection hangs indefinitely and will eventually break.



For details on the above restriction, refer to the *Oracle SQL Reference manual*.

- If you define parent-child relationships between tables, the publishing table that is created for a parent table should not contain a column with the LONG data type. However, a child table can contain a column with a LONG data type because data on child table rows is not copied using the new construct.
- LONG RAW data is not allowed in the publishing table.



These restrictions do not apply to publishing tables when you publish by reference and LONG or LONG RAW are non-key types.

- During the configuration of a custom outbound business object, which has a table with more than 290 columns and its storage mode has been set to Publish By Value, the compilation of the trigger will fail when the SQL scripts are deployed on the database. There are two workarounds to solve this issue:
 - 1) Split the trigger that contains the logic for Insert, Update, and Delete into three triggers. Recompile the SQL scripts.
 - 2) Configure the storage mode of the custom outbound business object as Publish By Reference.

- Publishing Table

The Database table is used to store a copy of published data. It can take any database table name as its value.

- The table name cannot be qualified by a schema name.
- The publishing table cannot contain any user-created columns where the column name starts with ADB_. These characters are reserved for use by the adapter.
- By convention, the publishing table should have the prefix P_. For example, if your source table is MY_ORDER, its publishing table should be named P_MY_ORDER.
- A publishing table name must be 30 characters maximum.



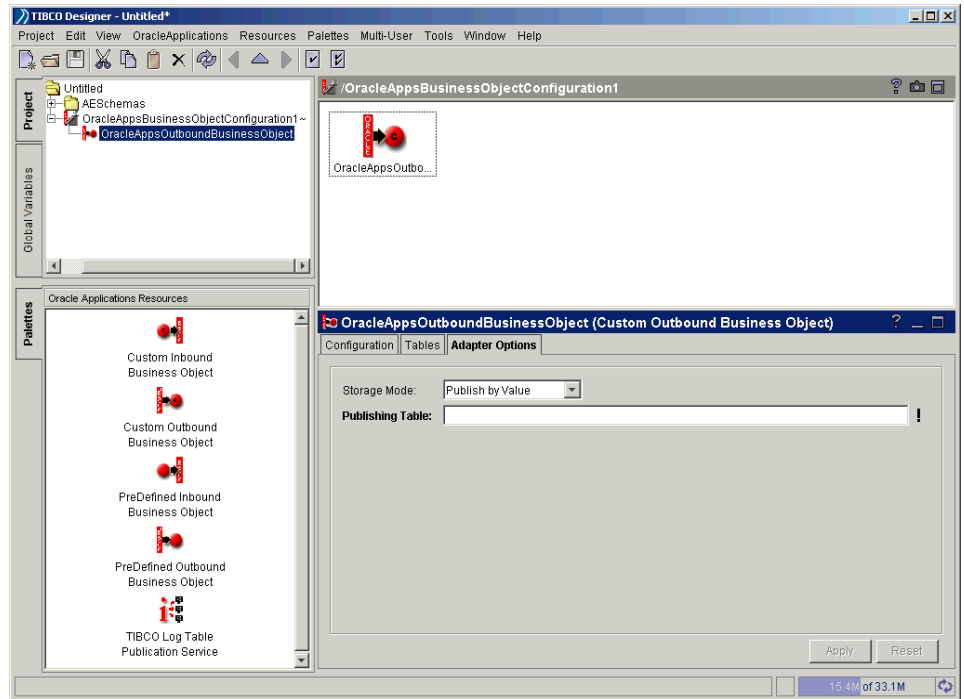
If the transaction involves a single table and if the Oracle Materialized View Log is present for the table, then the Storage Mode drop-down menu is grayed out in the Adapter Options tab and the Publish by Reference option has been selected.

If the transaction involves a single table and if the Oracle Materialized View Log is not present for the table, then you can choose between the Publish by Reference option and the Publish by Value option in the Storage Mode drop-down menu in the Adapter Options tab.

If the transaction involves more than one table, irrespective of the presence of Oracle MV Logs, then only the Publish By Reference option is supported in the storage mode drop-down menu.

Sample Screen

Figure 64 The Custom Outbound Publisher Adapter Options Tab



The Tables and View Mappings Tab

This tab only displays when the Publish by Reference option has been selected in the storage mode drop-down menu in the Adapter Options tab. Publishing from views is only available under the publish by reference mode.

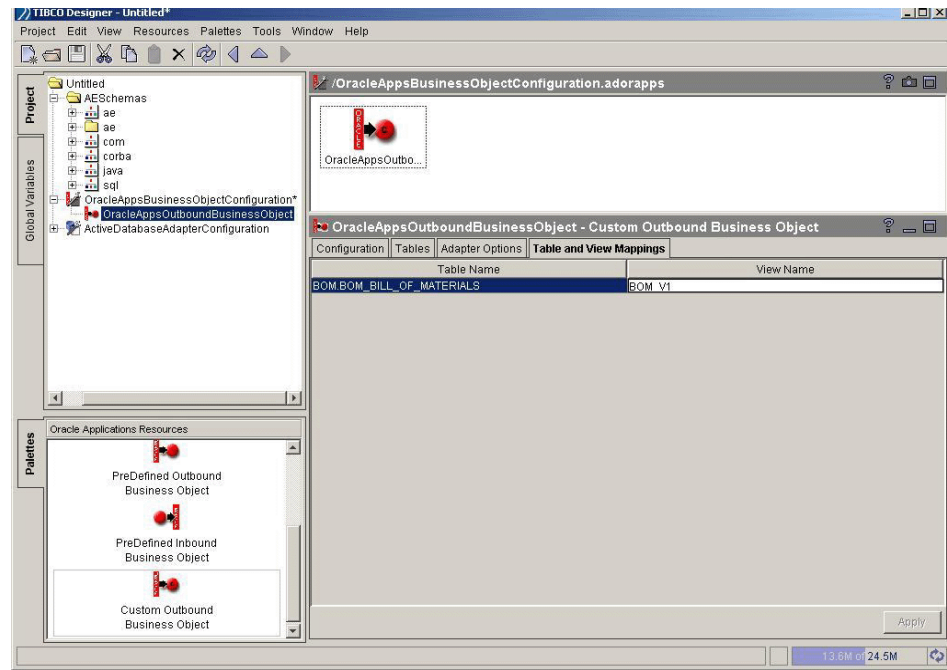
The view names for the tables present in the Business Object can be specified in this tab.

A valid view name should be entered for each table. The view name is validated for a standard Oracle database view. The maximum length of an Oracle view is 30 characters.

Once all the view names are specified, click the **Apply** button to set the values. If the view names are specified correctly, the Views tab is displayed. See [The Views Tab on page 132](#) for details on configuring the views tab.

Sample Screen

Figure 65 The Custom Outbound Publisher Table and View Mappings Tab



The Views Tab



The view names provided in the Tables and View Mappings tab is used to generate corresponding view objects for the tables present in the Business Object.

By default, the views contain all columns in the tables. Select or deselect these columns depending on whether they need to be included for publishing.


The selected table columns in this tab are included in the view definition.

The views are based on the source tables. However, any number of join tables can be specified if fields from other tables need to be included in the view for publishing.


Buttons

- The Add Details Table button 
Displays a dialog box where a join table can be added to the configuration from the default user schema provided in the adapter instance Connection tab.
- The Add Other Table button 

Allows you to enter a schema from a list of the available tables. If the schema password is different from the schema name, a dialog will prompt you to enter the password for that schema.

- The Remove Table button 

Deletes the selected JOIN table from the list.

- The Re-find Table from Database button 

Causes TIBCO Designer to refresh stored table schema information by retrieving new information from the database.

Columns

- Tables and Columns

Lists all the join tables and their columns.

- Type

Lists the primitive Oracle database type.

- AE Type

Lists the primitive type mapped to an ActiveEnterprise type.

- Use?

Check the box to publish data in the column after the insertion.

- Use in Join?

Check the box to add a new column in the view table and use this column to join with other tables. Only these columns are made available for use in the join information; that is, the WHERE clause of the SQL statement for the view

- Join With

This should be used to specify the Join clause during the creation of a view. After checking the Use in Join column, a drop-down list will appear in the Join With column. This drop-down list contains a list of attributes used in the Use in Join column (from the corresponding view to which this join table has been added). The sub clause Join is essential in the WHERE condition when you want to link two tables together. The join created is a standard Oracle AND join.

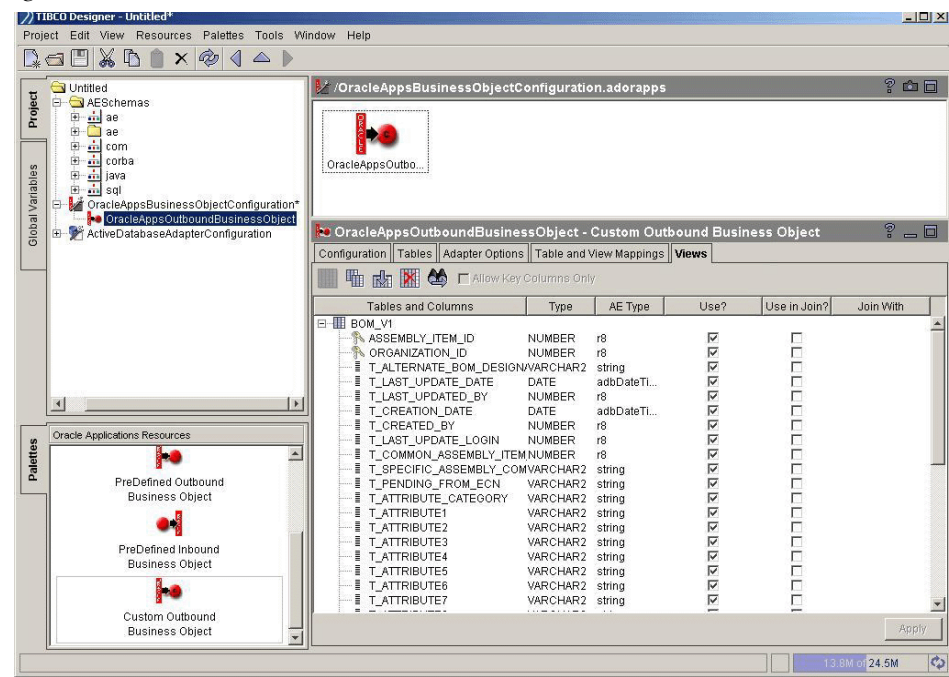
The view specified for the main table is set as the Referred Object and is visible in the Adapter Options tab. See [The Adapter Options Tab on page 128](#) for details.



If a table needs to be added to the Business Object is not listed in the tables drop-down list, execute the query `GRANT SELECT ON <TableName> to <TibcoUser>`, where <TibcoUser> is the User Name provided in the Connection tab of the adapter instance. Before executing this query, connect to the apps schema.

Sample Screen

Figure 66 The Custom Outbound Publisher Views Tab



PreDefined Inbound Current Program (Subscriber) Oracle Applications Business Object Configuration

To configure a PreDefined Inbound Oracle Applications Business Object:

1. Drag the **PreDefined Inbound Business Object** button from the palette panel and drop it into the design panel.
2. Select the required Business Object option from the Predefined Oracle Inbound Business Object drop-down list and click the **Apply** button.
3. The Configuration tab is displayed. Add entries in the following fields and click the **Apply** button.
 - Table – Tablespace
 - Index – Tablespace
 - APPS User Name (Mandatory)
 - APPS User Password (Mandatory)
4. The Tables tab displays a list of tables and their relationships in the Business Object. You can alter any properties manually under this tab.
5. In the Subscriber Options tab, specify the Exceptions Table name if required.

The PreCommit Stored Procedure field is populated with the name of the TIBCO PL/SQL Package. The procedure name is in the format of
`<APPUserName>.<PackageName>.main_adb`.

A value in this field signifies that the Subscribe with Reply feature of the adapter is used. For more information on this feature refer to *TIBCO ActiveMatrix Adapter for Database User's Guide*.

To enable End-to-End Traceability, the user should expose the columns in the Column1 For End-To-End Traceability field and the Column2 For End-To-End Traceability field.

For Non API transactions, if the Enter the Pre Commit Stored Procedure field is populated (the Subscription With Reply Feature option is selected), you need to populate the User Name field, the Language field, the Application Name field, the Responsibility Name field and the Concurrent Program Name field respectively. Once you select the appropriate Oracle Applications User and the Language options from the drop-down list, the Application Name field becomes active. The value of this field depends on the Language chosen. Once you have put the appropriate value in the Application name field, the Responsibility Name field and the Concurrent Program name field become active. The value of these two fields depends on the Application User and the

Language chosen. After you enter the appropriate values for these fields, do not forget to click the **Fetch** button at the right hand side of the field to confirm the selection. Otherwise, the application will not recognize the name entered.



This step is mandatory for the Subscription With Reply feature for Non-API transactions.

If the normal subscription feature is to be used, remove the value in this field to make it blank. In this case, the adapter will NOT invoke the PreCommit Stored Procedure. You will have to register the `<PackageName>.main` procedure in Oracle Applications and submit the registered program as a concurrent request in Oracle Applications.



It is not required to populate the User field, the Language field, the Application field, and the Responsibility field if the subscription is simple.

For details on customizing the PL/SQL Package generated by the Oracle Applications palette (only for Non-API transactions), see [Registering a Concurrent Program on page 429](#), [Oracle Application Transactions on page 277](#) (Structure of PL/SQL Packages, Mandatory Parameters for All Transactions) and [Trace Messages on page 379](#).

6. Set the message subject on which the service will subscribe to incoming messages in the Advanced tab.

Do NOT modify the `Class Reference` property.

For details on Customizing a PreDefined Inbound Business Object, see [Customizing a Business Object on page 230](#) in this chapter.

You can set parameters under the following tabs:

- [The Configuration Tab](#)
- [PreDefined Inbound API \(Subscriber\) Oracle Applications Business Object Configuration](#)
- [The Subscriber Options Tab](#) (see Figure 69)

The Configuration Tab

The Configuration tab has the following options:

- **PreDefined Oracle Inbound Business Object** — Initially, this is the only field in the Configuration tab. The drop-down box lists all available inbound (Subscriber) Business Objects. Once you select a Business Object and click

Apply, the Configuration tab displays the parameters that can be configured for that transaction.

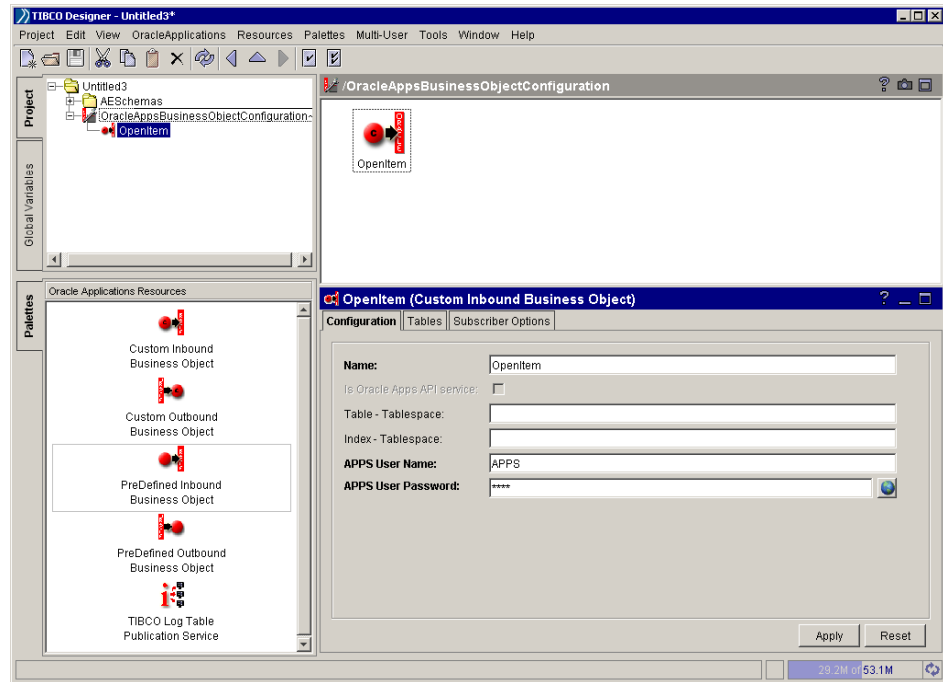


The following parameters are available only after a PreDefined inbound business object has been selected and applied.

- **Name**
Displays the service name associated with this PreDefined Business Object.
- **Table Tablespace**
This specifies the tablespace name of the newly created tables. If this field is left blank, database tables required for the Business Object are created in the default tablespace (allocated by the database system).
- **Index Tablespace**
This specifies the tablespace name of the newly created tables. If this field is left blank, database indexes required for the Business Object are created in the default tablespace (allocated by the database system).
- **APPS User Name**
The User Name used for connecting to the Oracle Applications database user (schema) of the Oracle Applications database.
- **APPS User Password**
The user password used for connecting to the Oracle Applications database user (schema) of the Oracle Applications database.
- **Is API Service**
This field is checked or becomes *inactive* depending on whether the selected Business Object is an Oracle Application API Transaction.

Sample Screen

Figure 67 The PreDefined Inbound Subscriber Configuration Tab



The Tables Tab

This tab displays the fixed set of tables constituting the Business Object. The Tables tab has the following column headings. You cannot modify any of values because the PreDefined Business Objects have a fixed set of tables that represent a specific transaction in Oracle Applications.

- Columns**
- **Tables and Columns**
Lists the table and its columns.
 - **Type**
Lists the primitive type.
 - **AE Type**
Lists the primitive type mapped to a TIBCO ActiveEnterprise type.
 - **User Key**
Defines the column as a user key if it is checked.

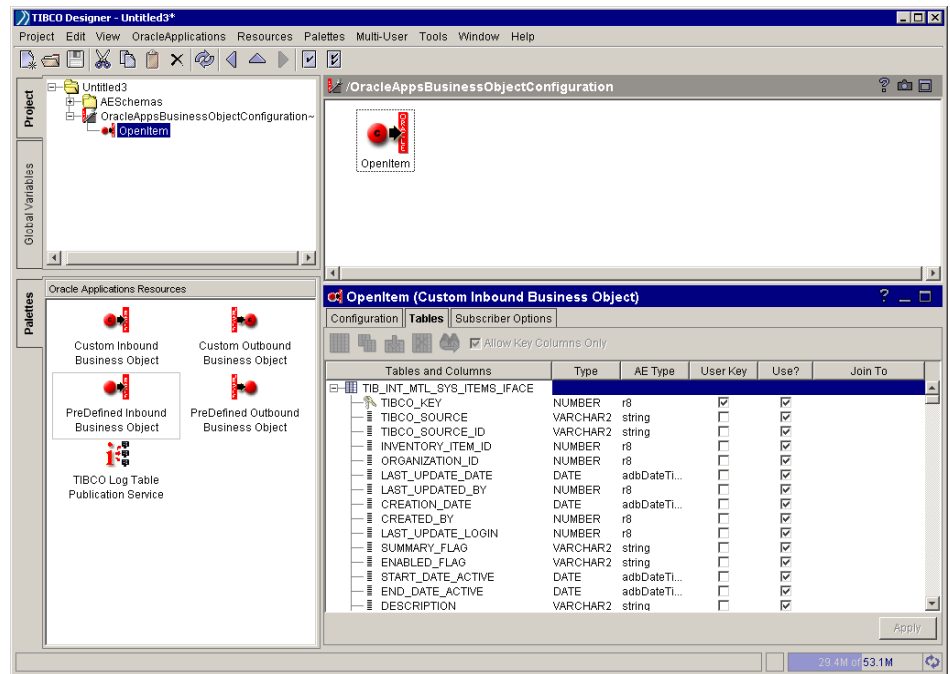
- Use?
Enables the column to be updated when a message arrives if it is checked.
- Join To
Joins two tables together.



Subscribe to the columns with update request only. A NULL value is populated for columns that are configured to but have not received data yet. Columns will retain their default values before such configuration.

Sample Screen

Figure 68 The PreDefined Inbound Subscriber Tables Tab



The Subscriber Options Tab (see Figure 69)

- Column1 For End-To-End Traceability
The Column entered is used for End-to-End Traceability. Click the **Fetch** button to choose the desired column from the available column list if you require the End-to-End traceability feature. See [End-to-End Traceability on page 486](#) for more details.

- Column2 For End-To-End Traceability

This Column entered is used for End-to-End Traceability. Click the **Fetch** button to choose the desired column from the available column list if you require the End-to-End traceability feature. See [End-to-End Traceability on page 486](#) for more details.

- Exceptions Table

Name of exception table. If the adapter cannot write to the subscriber table, it will write the error messages in this exception table. If the table does not exist, the subscriber adapter will create it by default.

The exception table *cannot* contain any user-created columns which contains the prefix ADB_. This prefix is reserved for the adapter use. (See the *TIBCO ActiveMatrix Adapter for Database User's Guide*.)

- Pre Commit Stored Procedure

The value entered here represents the stored procedure name the subscriber will call after the insertion, update, or deletion database operations and prior to the commit. You can use this stored procedure as an intermediate to do further processing inside the database and return the results to the adapter. The stored procedure will be called under all circumstances.

- User Name

The User Name is used to derive the User ID, which is required to run concurrent programs without registering it or submitting import programs.

For subscription with reply for Non-API transactions, specify the User Name in the User Name text box and click the **Fetch** button. It is recommended that you select the same from the drop-down list and not type it manually. For example, specify OP% in the User Name text box and click the **Fetch** button. All User Names preceded with OP display in a drop-down.

Since the concurrent program is not being registered this value of the User ID derived from User Name is required to set up the client information and initialize the profile. For simple subscription, the User ID is required to submit import programs. In which case, it can be entered from the Oracle Applications front end.

For each transaction,

- Go to the form/request screen where the Oracle Concurrent Request is submitted.
- Go to **Help > Diagnostics > Examine > Block**, select the \$PROFILES\$ option.
- Select Field - USER_ID and specify the appropriate value.

- Language

This field sets the language used by the Application and the Responsibility.



This is required so that the Application Name can be derived.

- Application Name

The Application Name is used to derive the Application ID. The Application ID is required to run the concurrent programs without registering it or to submit any import programs.

For Subscription With Reply for Non-API transactions, specify the Application Name in the Application Name text box and click the **Fetch** button. It is recommended that you select the same from the drop-down and not type it manually. For example, specify AM% in the User Name text box and click the **Fetch** button. All Application Names preceded with AM, display in a drop-down.

This can only be done once you select a Language. This is because an Application is dependant on the Language. Since the concurrent program is not being registered, the value Application Id derived from Application Name is required to set up client information and initialize the profile.

For simple subscription the Application Id is required to submit import programs. In which case, it can be entered from the Oracle Applications front end.

For each transaction,

- Go to the form/request screen where the Oracle Concurrent Request is submitted,
- Go to **Help > Diagnostics > Examine > Block** and select the \$PROFILES\$ option.
- Select RESP_APPL_ID and specify the appropriate value.

- Responsibility Name

The Responsibility Name is used to derive the Responsibility ID, which is required to run the concurrent programs without registering it or submitting any import programs.

For subscription with reply for Non API transactions, specify the Responsibility Name in the Responsibility Name text box and click the **Fetch** button. It is recommended that you select the same from the drop-down and not type it manually. For example, specify Oracle% in the User Name text box

and click the **Fetch** button. All Responsibility Names preceded with Oracle, display in a drop-down menu.

The Responsibility Name should be selected only after the Language and the Application have been selected. This is because a Responsibility is dependant on the Language and the Application. Since the concurrent program is not being registered this value Responsibility ID derived from Responsibility Name is required to set up client information and initialize the profile.

For simple subscription the Resp ID is required to submit import programs. In which case, it can be entered from the Oracle Applications front end.

For each transaction,

- Go to the form/request screen where the Oracle Concurrent Request is submitted.
- Click **Help > Diagnostics > Examine > Block**.
- Select the \$PROFILES\$ option.
- Select RESP_ID and specify the appropriate value.

- Concurrent Program Name

The Concurrent Program Name is used to derive the Concurrent Program, which is required to run the concurrent programs without registering it or submitting any import programs.

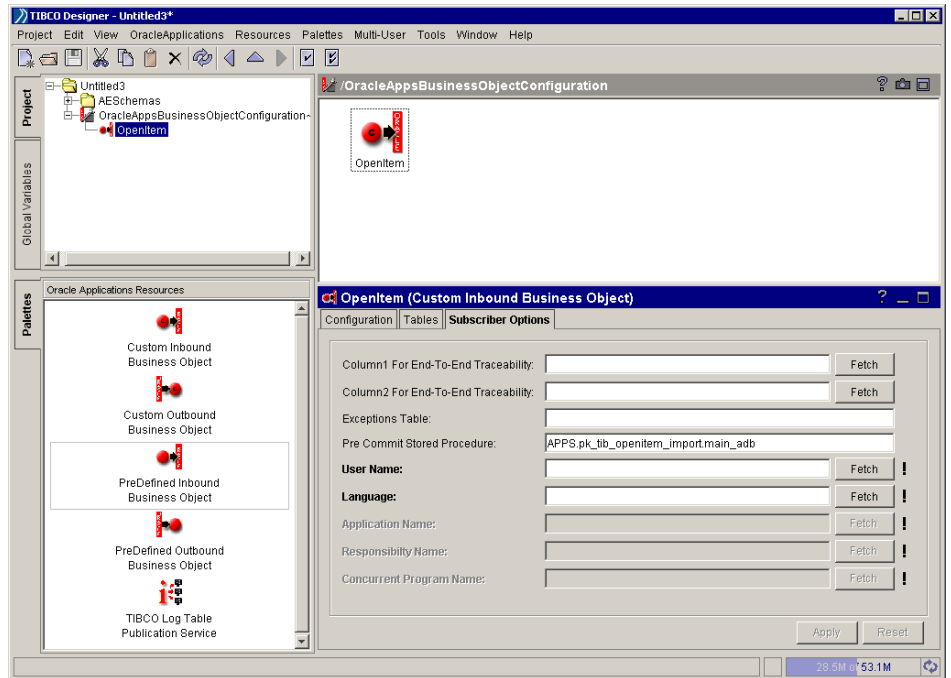
For subscription with reply for Non API transactions, specify the Concurrent Program Name in the Concurrent Program Name text box and click the **Fetch** button. It is recommended that you select the same from the drop-down and not type it manually. For example, specify Import% in the Concurrent Program Name text box and click the **Fetch** button. All Concurrent Program Names preceded with Import, display in a drop-down.

The Concurrent Program Name should be selected only after the Language and the Application have been selected. This is because a Concurrent Program Name is dependant on the Language and the Application.

Based on the Concurrent Program Name selected, the Concurrent Program parameters prefixed with C1 are appended to the Header Intermediate Table. The prefixed parameters are seen in the Tables tab. For example, Open Item has seven Concurrent Program parameters in the Header Intermediate table TIB_INT_MTL_SYS_ITEMS_IFACE. One of the parameters is C1_ORGANIZATION_ID.

For simple subscription the Concurrent Program Name is required to submit the Import programs. In this case, it can be entered from the Oracle Applications front end. See [Dynamic Population of Concurrent Program Parameters on page 498](#) for more information.

Figure 69 The PreDefined Inbound Subscriber Options Tab



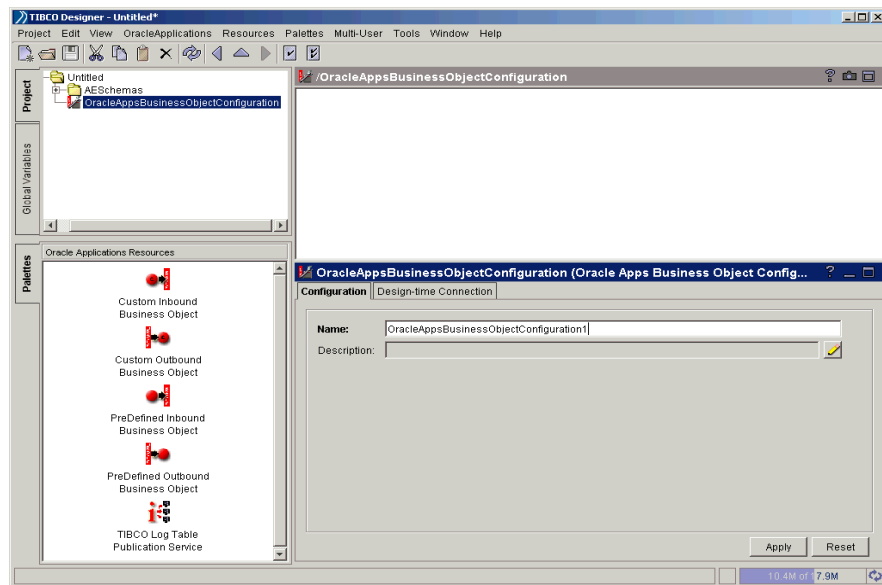
PreDefined Inbound API (Subscriber) Oracle Applications Business Object Configuration

To configure a PreDefined Inbound API Oracle Applications Business Object:

1. Drag the **Oracle Apps Business Object Configuration** button from the palette panel and drop it into the design panel.

Sample Screen

Figure 70 The PreDefined Inbound API Subscriber Configuration Tab



2. Enter the appropriate connection parameters in all fields of the Design-time Connection tab.
3. Click the **OracleAppsBusinessObjectConfiguration** button in the project panel. A list of Business Object buttons will display in the palette panel.
4. Drag the **PreDefined Inbound Business Object** button from the palette panel and drop it into the design panel.
5. Select the required Business Object from the drop-down list and click the **Apply** button. An Error Deploying Scripts error message sometimes pops up. If this happens, undo the SQL scripts in the SQL folder, delete the Business Object and reconfigure it to avoid this error.

6. You will see the Configuration tab in the configuration panel. Add entries in the following fields and click the **Apply** button.
 - Table: Tablespace
 - Index: Tablespace
 - APPS User Name (Mandatory)
 - APPS User Password (Mandatory)
7. The Tables tab displays the tables and their relationships as existing in the Business Object. This tab will not be available for setting any properties.
8. In the Subscriber Options tab, specify the Exceptions Table if required. The PreCommit Stored Procedure field is populated with the name of the TIBCO PL/SQL Package. The procedure name is in the format of:
`<APPSUserName>.<PackageName>.main_adb.`

A value in this field shows that the adapter's Subscribe with Reply feature is being used. For more information on this feature, refer to the *TIBCO ActiveMatrix Adapter for Database User's Guide*.

You can set parameters under the following tabs:

- [The Configuration Tab](#)
- [The Tables Tab](#)
- [The Subscriber Options Tab](#)

The Configuration Tab

The Configuration tab has the following options:

- PreDefined Oracle Inbound Business Object

This is the only field in the Configuration tab immediately after the creation of the business object. You can see a drop-down list of all available inbound (subscribe) Business Objects. Once you select a Business Object and click the Apply button, a set of configuration options will display under the Configuration tab for that transaction.

- Name

The service name associated with this PreDefined Business Object.

- Table Tablespace

This specifies the tablespace name of the newly created table. If this field is left blank, the database tables required for the Business Object is created in the default tablespace that is allocated by the database system.

The following parameters are available only after a PreDefined Inbound Business Object has been selected and applied.

- Index Tablespace

Specifies the tablespace name of the newly created indexes. If this field is left blank, database indexes required for the Business Object is created in the default tablespace (allocated by the database system).

- APPS User Name

The User Name for connecting to the Oracle Applications database user (schema) of the Oracle Applications database.

- APPS User Password

The user password for connecting to the Oracle Applications database user (schema) of the Oracle Applications database.

- Is API Service

This field is checked because this process involves API configuration. You cannot modify this setting.

The Tables Tab

This tab displays the fixed set of tables constituting the Business Object. The Tables tab has the following column headings. You cannot modify any of values because the PreDefined Business Objects have a fixed set of tables that represent a specific transaction in Oracle Applications.

Columns

- Tables and Columns

Lists the table and its columns.

- Type

Lists the primitive type.

- AE Type

Lists the primitive type mapped to a TIBCO ActiveEnterprise type.

- User Key

Defines the column as a user key if it is checked.

- Use?

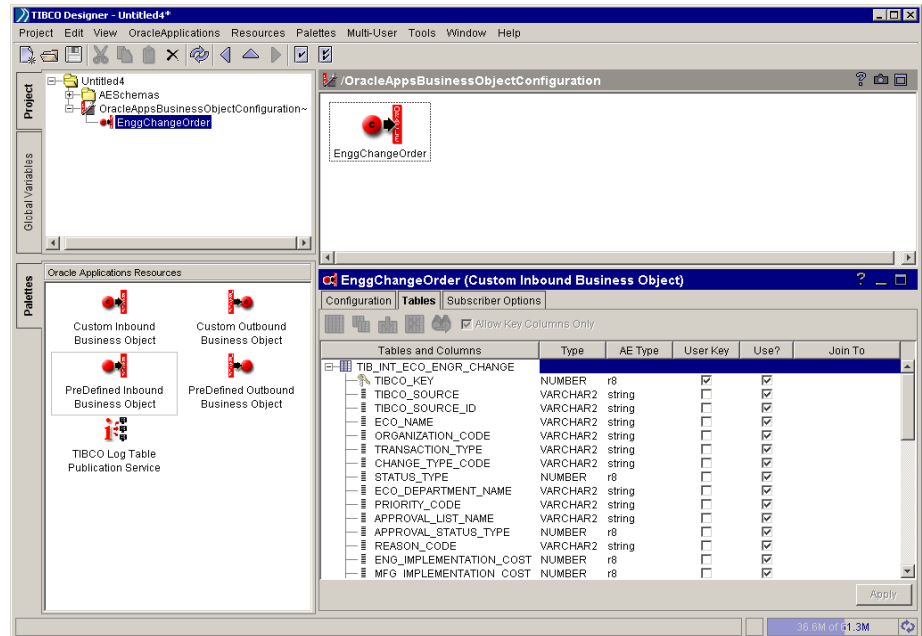
Enables the column to be updated when a message arrives if it is checked.

- Join To

Joins two tables together. Subscribe to the columns with update request only. A NULL value is populated for columns that are configured to receive data but have not yet received it. The columns will retain their default values before configuration.

Sample Screen

Figure 71 The PreDefined Inbound API Subscriber Tables Tab



The Subscriber Options Tab

- Exceptions Table

Name of exception table. If the adapter cannot write to the subscriber table, it will write the error messages in this exception table. If the table does not exist, the subscriber adapter will create it by default.

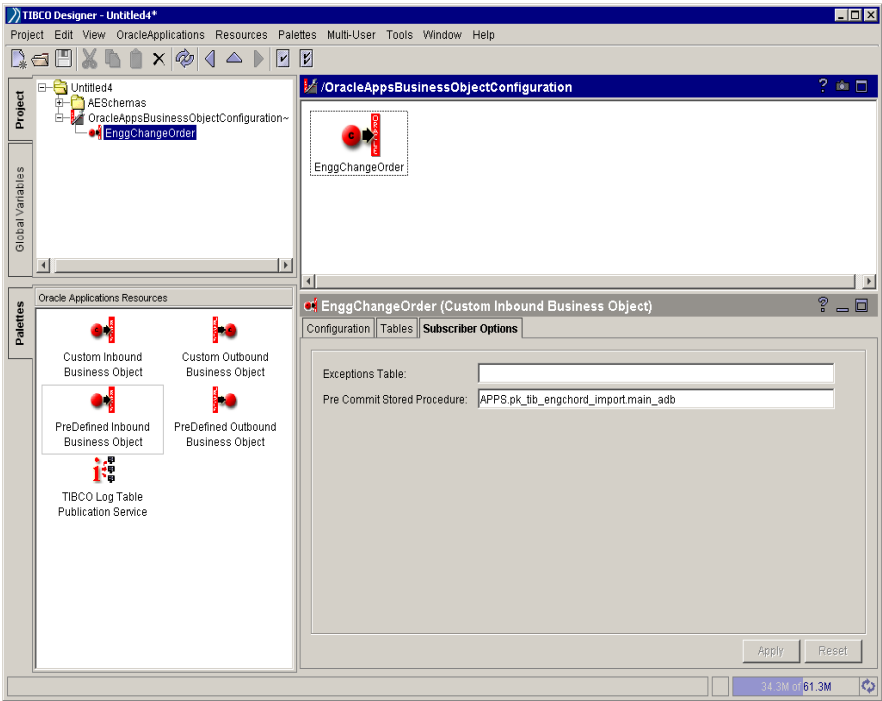
The exception table *cannot* contain any user-created columns which contain the prefix ADB_. This prefix is reserved for adapter use. See *TIBCO ActiveMatrix Adapter for Database User's Guide* for more details.

- Pre Commit Stored Procedure

The value entered represents the stored procedure name the subscriber will call after the insertion, update, or deletion database operations and prior to

the commit. You can use this stored procedure as an intermediate to accomplish further processing inside the database and return the results to the adapter. The stored procedure will be called under all circumstances.

Figure 72 The PreDefined Inbound API Subscriber Options Tab



The adapter always uses the Oracle Applications API signature with maximum number of input parameters.




Custom Inbound (Subscriber) Oracle Application Business Object Configuration

To configure a Custom Inbound Oracle Applications Business Object:

1. Drag the **Custom Inbound Business Object** button to the design panel.
2. You can see the **Configuration** tab displayed in the configuration panel. Enter values in the fields below and click the **Apply** button.

- Table – Tablespace
- Index – Tablespace
- APPS User Name (Mandatory)
- APPS User Password (Mandatory)

3. The Tables tab is available for adding tables that will constitute the Business Object.

Tables can be added using the Add Table button , Add Details Table button , and Add Other Table button . Relationships between the tables can be specified using the User Key and Join To properties (specified as key columns) in a parent table that appears in the drop-down list under the Join To column property. The required parent table column can then be used to join with the child table in a parent-child relationship.

All Classes, Associations and Sequences are based on this relationship. Please refer to the TIBCO Designer Schema Palette Documentation.

4. In the Subscriber Options tab, specify the Exceptions Table if required.

Specify a value for the PreCommit Stored Procedure field. This represents the TIBCO PL/SQL Package. The procedure name should be in the format of `<APPSUserName>.<PackageName>.main_adb`.

A value in this field shows that the adapter's Subscribe with Reply feature is used. For more information on this feature, see *TIBCO ActiveMatrix Adapter for Database User's Guide*.

5. In this case of Subscription with Reply for Non API transactions, the User Name, Language, Application, Responsibility and Concurrent Program Name fields need to be entered. In case of simple subscription, these need not be entered from TIBCO Designer.

If the normal subscription feature is to be used, leave this field blank. In this case, the adapter will NOT invoke the PreCommit Stored Procedure. You will have to register the `<PackageName>.main` procedure in Oracle Applications

and submit the registered program as a concurrent request in Oracle Applications.

For details on customizing the PL/SQL Package generated by the Oracle Applications palette (only for Non-API transactions), see [Oracle Application Transactions on page 277](#) (Structure of PL/SQL Packages, Mandatory Parameters for All Transactions) and [Trace Messages on page 379](#).

6. Set the message subject on which the service will subscribe to incoming messages in the Advanced tab.

Do NOT modify the Class Reference property.

For details on Customizing a Custom Inbound Business Object, please refer to the section [Customizing a Business Object on page 230](#) in this chapter.

You can set parameters under the following tabs:

- [The Configuration Tab](#)
- [Oracle Publisher Configuration](#)
- [The Subscriber Options Tab](#) (see Figure 75)

The Configuration Tab

The Configuration tab has the following options:.



The following parameters are available only after a PreDefined Inbound Business Object has been selected and applied.

- Name

The service name associated with this PreDefined Business Object.

- Is API Service

Check this field depending on whether the required Business Object is an Oracle Application API Transaction.

- Table Tablespace

This specifies the tablespace name of the newly created table. If this field is left blank, the database tables required for the Business Object are created in the default tablespace that is allocated by the database system.

- Index Tablespace

Specifies the tablespace name of the newly created indexes. If this field is left blank, database indexes required for the Business Object are created in the default tablespace which is allocated by the database system.

- APPS User Name

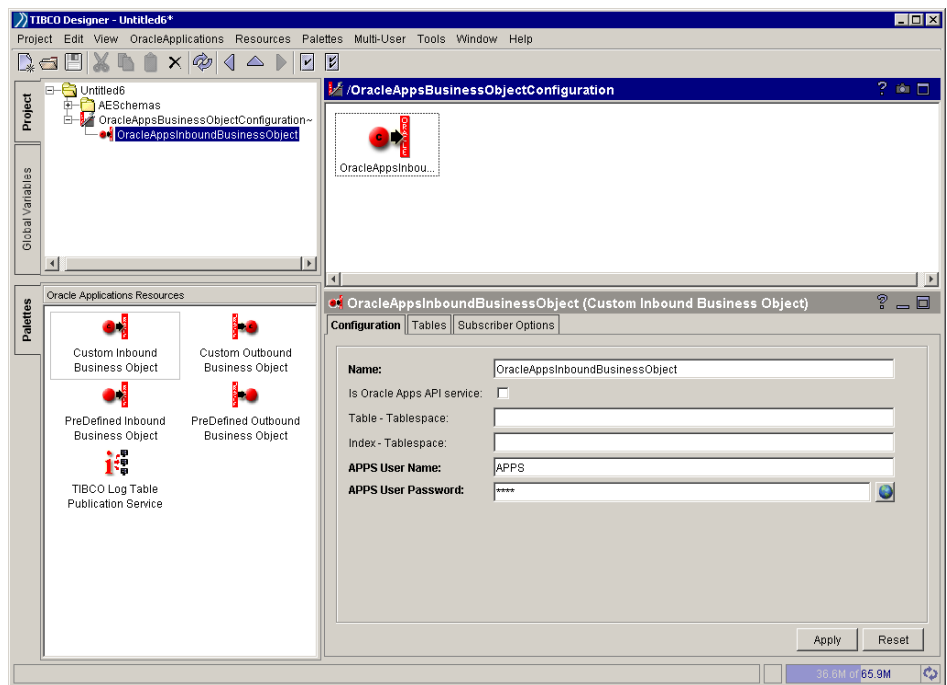
The User Name for connecting to the Oracle Applications database user (schema) of the Oracle Applications database. This field is mandatory.

- APPS User Password

The user password for connecting to the Oracle Applications database user (schema) of the Oracle Applications database. This field is mandatory.

Sample Screen

Figure 73 The Custom Inbound Subscriber Configuration Tab



The Tables Tab

This tab is used for adding tables to the Business Object. The Tables tab has the following buttons and column headings:

Buttons

- The Add Table button 


Click to display a dialog box that lists tables available to the database user specified in the adapter Connection tab. Select the table to be used as the subscription table. It will insert data after receiving it.

- The Add Details Table button 

Displays a dialog box from which a secondary table can be added to the configuration.

- The Add Other Table button 

Allows you to enter a schema from the list of available tables. If the schema password is different from the schema name, a dialog will prompt you to enter the password for that schema.

- The Remove Table button 

Deletes the selected table from the configuration.

- The Allow Key Columns Only Checkbox

A key column or substitute key column is required when publishing by reference, since the subscription table contains only key values. If no column is specified, the subscription table will not be added.

Columns

- Tables and Columns

Lists the table and its columns.

- Type

Lists the primitive type.

- AE Type

Lists the primitive type mapped to an ActiveEnterprise type.

- User Key

Check to define as a user key.

- Use?

Click to enable a column to be updated when receiving a message.



Subscribe to the columns with update request only. A NULL value is populated to columns that are configured to but have not received data yet. Columns will retain their default values before such configuration.

- Join To

Joins a child table to its parent table. The child table shows a drop-down list containing columns which have been marked as User Keys of the parent table.



The Inbound services are based on a set of defined tables for a particular transaction. See *TIBCO ActiveMatrix Adapter for Database User's Guide* for details.

A NULL value is populated to columns that are configured to but have not received data yet. Columns will retain their default values before such configuration.

Adding tables for an API Business Object

Create the tables that need to be added to the Business Object before configuration. A Sample Table structure for an Employee API is given under [Creating TIBCO Intermediate Tables for Custom Inbound API Business Objects on page 448](#).

Create these tables under the Adapter User account (where the TIBCO database objects will be created).

After the required tables have been created in the database, add them to the account using the  and  buttons.


Columns that need to be used for joining their child tables should each be marked as a User Key for each table. Mark the TIBCO_KEY column as the User Key. For all the child tables, choose the appropriate parent columns from the drop-down list to provide join information.

The column TIBCO_KEY should also be used along with other keys to join with its parent table's TIBCO_KEY column.



- The TIBCO_KEY column is used as one of the Key columns for all Subscription services for both PreDefined and Custom. The TIBCO_KEY field links the header record to the child record in a Subscription service. To create the relationship between the children and their grandchildren, the TIBCO_KEY is one of the join keys along with the normal Oracle-specific Business Keys.
- The user specified in the Connection tab of the adapter instance should be TIBCO user.

Adding Tables for a Non-API Business Object

Add the required source tables (Oracle Interface Tables) using the  button. After the required source table is selected from the list of choices, you are prompted to enter the corresponding Schema Name (TIBCO Intermediate Table name). The Intermediate tables are used as the destination tables for creating the subscriber service. See [Subscription Service Overview on page 18](#).

The Intermediate tables are created under the Adapter user account (where the TIBCO database objects are created).

For each table, columns used for joining its child tables will be marked as User Key. All columns named TIBCO_KEY will be marked as User Key. You should provide join information for each child table by selecting the appropriate parent columns from the drop-down list.

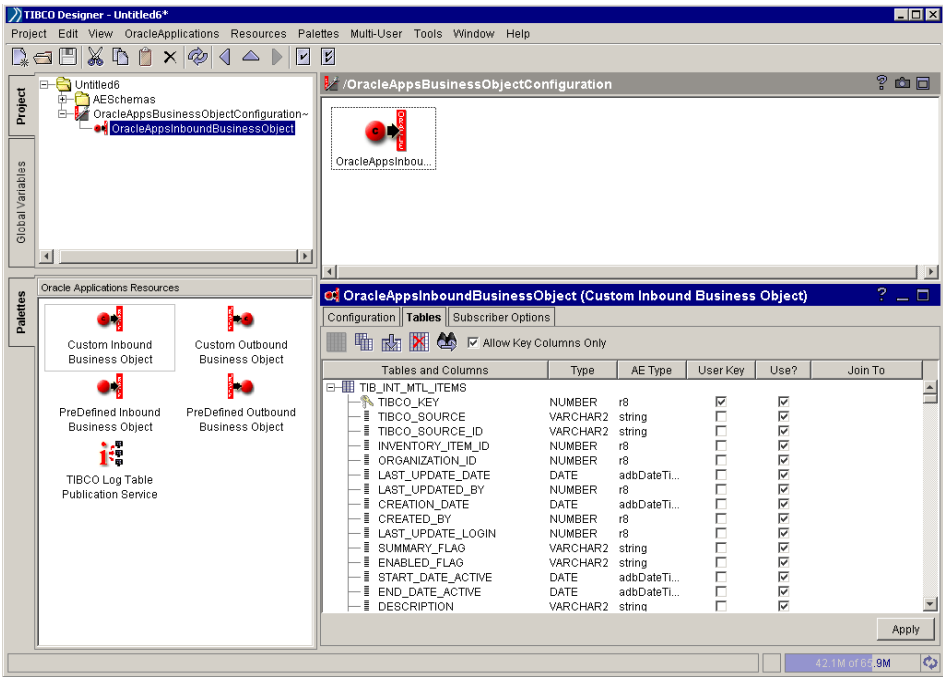
Along with other keys, use the column named TIBCO_KEY to join the TIBCO_KEY column with its parent table.



- The TIBCO_KEY column is used as one of the Key columns for all Subscription services (PreDefined and Custom). The TIBCO_KEY field relates the header record to the child record in a Subscription service. To create a relationship between children and their grandchildren, the TIBCO_KEY is one of the join keys along with the normal Oracle-specific Business Keys.
- Specify the user in the Connection tab of the adapter instance as TIBCO user.

Sample Screen

Figure 74 The Custom Inbound Subscriber Tables Tab



The Subscriber Options Tab (see Figure 75)

- Exceptions Table

The name of the exception table. If the adapter cannot write to the subscriber table, it will write the error messages in this exception table. If the table does not exist, the subscriber adapter will create it by default.

The exception table *cannot* contain any user-created columns which contain the prefix ADB_. This prefix is reserved for adapter use. See *TIBCO ActiveMatrix Adapter for Database User's Guide* for more details.

- Pre Commit Stored Procedure

The value entered represents the PL/SQL stored procedure name the subscriber will call after the insertion, update, or deletion database operations and prior to the commit. You can use this stored procedure as an intermediate media to accomplish further processing inside the database and return the results to the adapter. The adapter will not call a stored procedure if you leave this field blank. The value should be in the form of `<APPSUserName>.<PackageName>.<main_adb>`. The Oracle Application Palette will return an error if this value is not specified in the required format.

- Pre Commit Stored Procedure for an API Business Object

The Stored Procedure will NOT be created during configuration. You have to manually create the required procedure in the Oracle Applications Adapter user account with the package name similar to the one provided in this field.

- Pre Commit Stored Procedure for a Non-API Business Object

The SQL file `<AdapterInstance>_<BusinessObjectName>_all.sql` generated in the `<ADORAPPS_HOME>/sql` folder contains the PL/SQL package. This package will not contain the error-logging portion. The error-logging portion must be manually added to the generated SQL file (`<AdapterInstance>_<BusinessObjectName>_all.sql`). This file should be compiled if the Business Object has been deployed in the Oracle Applications database.

- User Name

This field is used to derive the User ID. This ID is required when running the concurrent programs without registering it or submitting import programs.

For subscription with reply for Non-API transactions, specify the User Name in the User Name field and click the **Fetch** button. It is recommended that you select the user name from the drop-down list instead of typing it manually to avoid typographical errors. For example, enter `OP%` in the User Name field and click the **Fetch** button. All User Names starting with the string `OP` will display in the drop-down list.

Since the concurrent program is not being registered, the User ID value is required to set up the client information and initialize the profile. The user ID

is required to submit import programs for simple subscriptions, but in this case, it can be entered from the Oracle Applications front end. This can be achieved from the ERP applications.

For each transaction, you have to log in to ERP, and

- Access the Request Form where the Oracle Concurrent Request is submitted
- Click **Help > Diagnostics > Examine > Block** and select the \$PROFILES\$ option
- Select the **USER_ID** option from the drop-down list in the pop-up message box and assign it with the appropriate value.

- Language

The language field is required in order to choose the required Application and Responsibility based on the chosen language.



This field is compulsory as the Application Name is dependent on it.

- Application Name

This field is used to derive the Application ID, which is required when running the concurrent programs without registering it or submitting import programs.

For Subscription With Reply for Non API transactions, specify the Application Name in the Application Name field and click the **Fetch** button. It is recommended that you select the application name from the drop-down list instead of typing it manually to avoid typographical errors. For example, enter AM% in the Application Name field and click the **Fetch** button. All Application Names starting with the string AM will display in the drop-down list.

The Application Name field will only become active after you have entered a valid Language value because an Application is dependant on the Language chosen. Since the concurrent program is not being registered, the value Application ID derived from Application Name is required to set up client information and initialize the profile.

The Application ID is required to submit import programs for simple subscriptions, but in this case, it can be entered from the Oracle Applications front end. This can be achieved from the ERP applications.

For each transaction, you have to log in to ERP, and

- Access the Request Form where the Oracle Concurrent Request is submitted
- Click **Help > Diagnostics > Examine > Block** and select the \$PROFILES\$ option
- Select the **RESP_APPL_ID** option from the drop-down list in the pop-up message box and assign it with the appropriate value.

- Responsibility Name

This field is used to derive the Responsibility ID, which is required when running the concurrent programs without registering it or submitting import programs.

For subscription with reply for Non-API transactions, specify the Responsibility Name in the Responsibility Name field and click the **Fetch** button. It is recommended that you select the responsibility name from the drop-down list instead of typing it manually to avoid typographical errors. For example, specify `Oracle%` in the Responsibility Name field and click the **Fetch** button. All Responsibility Names starting with the string Oracle will display in the drop-down list.

The Responsibility Name field will only become active after you have entered a valid value for Language and Application as a Responsibility is dependant on the Language and the Application used. Since the concurrent program is not being registered, this value Responsibility ID derived from Responsibility Name is required to set up client information and initialize the profile.

The Responsibility ID is required to submit import programs for simple subscriptions, but in this case, it can be entered from the Oracle Applications front end. This can be achieved from the ERP applications.

For each transaction, you have to log in to ERP, and

- Access the Request Form where the Oracle Concurrent Request is submitted
- Click **Help > Diagnostics > Examine > Block** and select the \$PROFILES\$ option
- Select the **RESP_ID** option from the drop-down list in the pop-up message box and assign it with the appropriate value.

- Concurrent Program Name

This field is used to derive the Concurrent Program, which is required when running the concurrent programs without registering it or submitting import programs.

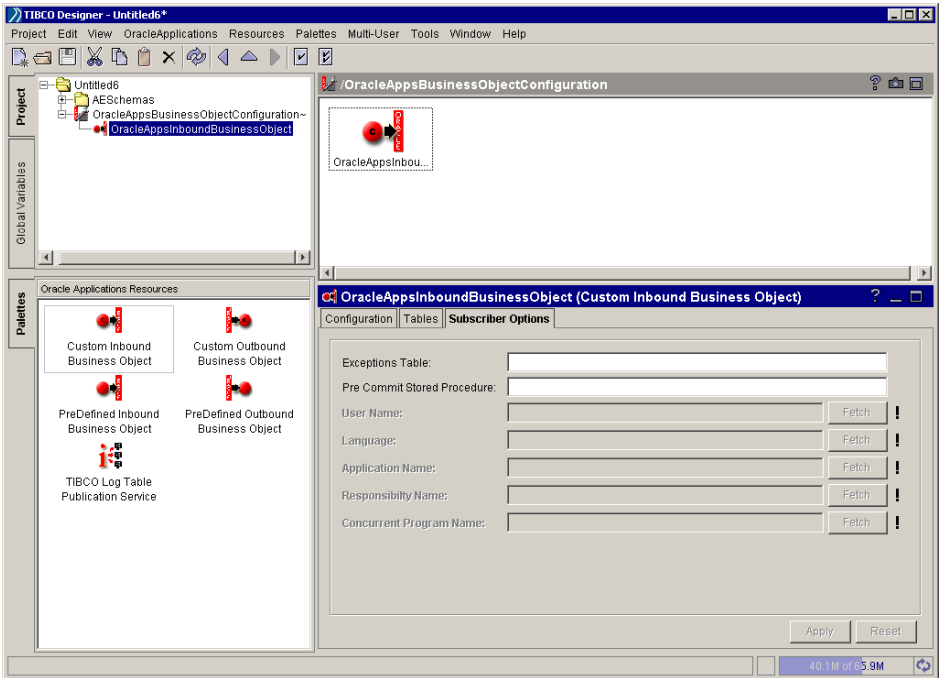
For subscription with reply for Non API transactions, specify the Concurrent Program Name in the Concurrent Program Name field and click the **Fetch** button. It is recommended that you select the Concurrent Program Name from the drop-down list instead of typing it manually to avoid typographical errors. For example, specify `Import%` in the Concurrent Program Name field and click the **Fetch** button. All Concurrent Program Names starting with the string `Import` will display in the drop-down list.

The Concurrent Program Name field will only become active after you entered a valid value for Language and Application as a Concurrent Program is dependant on the Language and the Application used.

Based on the Concurrent Program Name selected, the Concurrent Program parameters prefixed with `CONC` are appended to the Header Intermediate Table.

The Concurrent Program Name is required to submit import programs for simple subscriptions. However, in this case, it can be entered from the Oracle Applications front end.

Figure 75 The Custom Inbound Subscriber Options Tab



Oracle Applications Inbound Business Object (For R12 Only)

Conventionally, a business object is provided by the Oracle Integration Repository. It is either an Oracle Application API or a Current Program. Every time Oracle develops a new API, TIBCO Adapter for Oracle Applications has to develop a new business object to support it. This action can be time consuming.

The Oracle Applications Inbound Business Object (For R12 Only) can automatically generate PL/SQL scripts using the information provided by the Interface. These SQL scripts are dynamic according to the information provided. Therefore, you can generate your own object for every Oracle API.

The Oracle Applications Inbound Business Object (For R12 Only) requires the use of Oracle JPublisher to generate the database information for the API or Concurrent Program selected for integration. Oracle JPublisher inspects the database and generates a SQL wrapper file for a specific API or Concurrent Program. This SQL wrapper contains information about how many different methods are contained within an API package, how many parameters are required for each method, and the data types of those parameters. In addition and most importantly, Oracle JPublisher exposes the complex datatypes that may be contained within a package to be accessible outside of the package.

To configure an Oracle Applications Inbound Business Object (For R12 Only) service, the wrapper generated for the desired API must be made available to the machine that the adapter is installed on. During configuration, the adapter will make use of this wrapper and generate the proper script based on this wrapper.

The Oracle Applications Inbound Business Object (For R12 Only) allows for three types of services: API Simple services, API Wrapper services, and Concurrent Program services.

An API Simple service enables you to provide API services dynamically without wrapper; an API wrapper service enables you to provide API services dynamically with wrapper; and a Concurrent Program service enables you to run concurrent programs dynamically.

For information on how to configure these services, refer to the related sections.

Prerequisites for Configuring an Oracle Applications Inbound Business Object (For R12 Only) Service



- The corresponding API or Concurrent Program is selected. To select an API or Concurrent Program, you need to log into Oracle E-Business Suite's Integration Repository. If you have trouble accessing Oracle E-Business Suite's Integration Repository, contact your Oracle e-Business Suite administrator. Refer to [Appendix P, Selecting APIs and Concurrent Programs from Oracle](#)

[EBS R12 Integration Repository, on page 511](#) for instructions on selecting APIs and Concurrent Programs from Oracle EBS Integration Repository.

- Generate the PL/SQL wrapper for the API you want to use.
- Execute the PL/SQL wrapper against the Oracle e-Business Suite database using the database user specified during the creation of the script. For more information about creating the PL/SQL wrapper, please refer to the Appendix to [Appendix O on page 503](#).
- Make sure the wrapper file is accessible to the machine where the adapter is installed and is accessible at design time configuration.

Generic Procedures for Creating an Oracle Applications Inbound Business Object (For R12 Only)

To configure an Oracle Applications Inbound Business Object (For R12 Only) service:

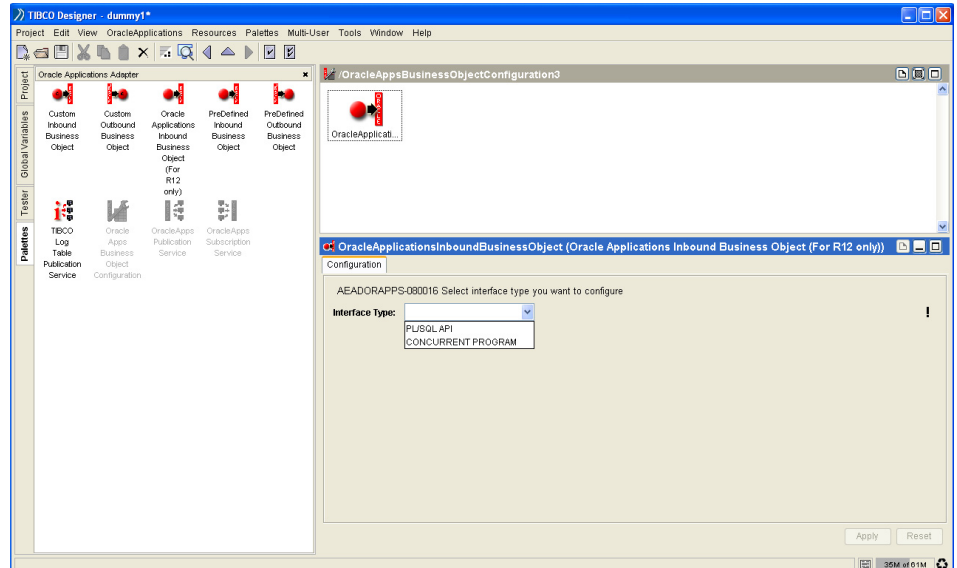
1. Open TIBCO Designer and create a new project.
2. Restore the Oracle Applications Adapter window within the Palettes tab. Drag the **Oracle Apps Business Object Configuration** button and drop it into the Design panel.
3. Under the Configuration tab, you can input a name for the Object in the Name field or leave it as default. You can also write a description for the object by clicking the  button.
4. Enter the appropriate values for JDBC Driver, JDBC URL, Username, Password and Request Timeout (milliseconds) under the Design-time Connection tab or select a pre-saved user connection by clicking **OracleApplications > User Connections > <Connection_Name>**. Click the **Test Connection...** button to ensure the connection is set up successfully. You will see a message box with a Connection Successful message. If not, re-check the connection information you provided in the Design-time Connection tab.
5. Click the  button from the menu bar to save your project.
6. Select the Oracle Apps Business Object you have just created from the project panel. Drag the **Oracle Applications Inbound Business Object (For R12 Only)** button from the Palettes panel and drop it into the design panel.



Ensure that you complete the configuration setting for the business object in once or delete the previous incomplete configuration and reconfigure the business in once. Separate configuration process will cause object creation failure.

7. You will see a Configuration tab in the configuration panel. There are two options in the Interface Type drop-down list, the PL/SQL API option and the CONCURRENT PROGRAM option (see Figure 76). Each option will be described in detail below.

Figure 76 The Inbound Business Object Configuration Options

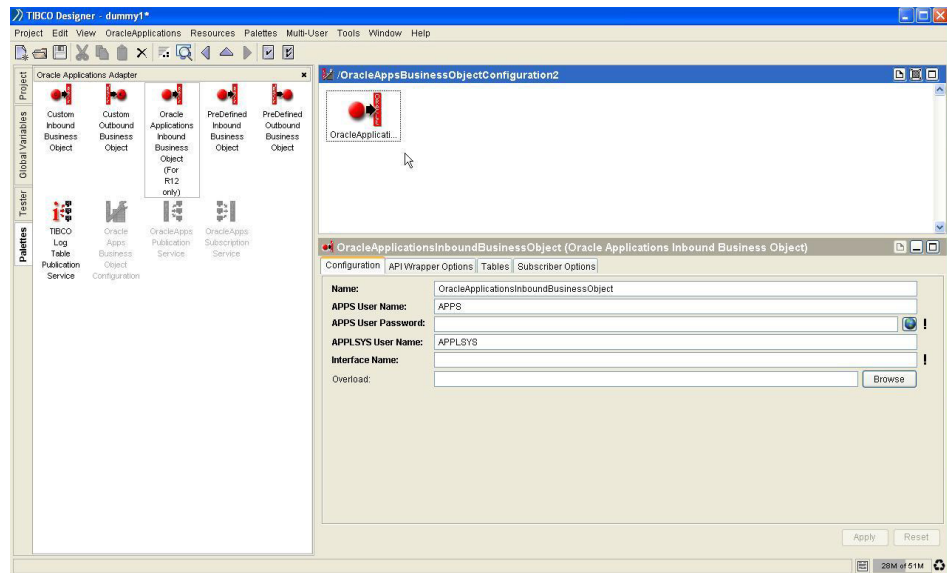


The PL/SQL API Option

If you choose the PL/SQL API option:

1. Select the **PL/SQL API** option from the Interface Type drop-down list and click the **Apply** button.
2. After applying the settings, you will see four tabs in the configuration panel. They are the Configuration tab, the API Wrapper Options tab, the Tables tab, and the Subscriber Options tab (see Figure 77).

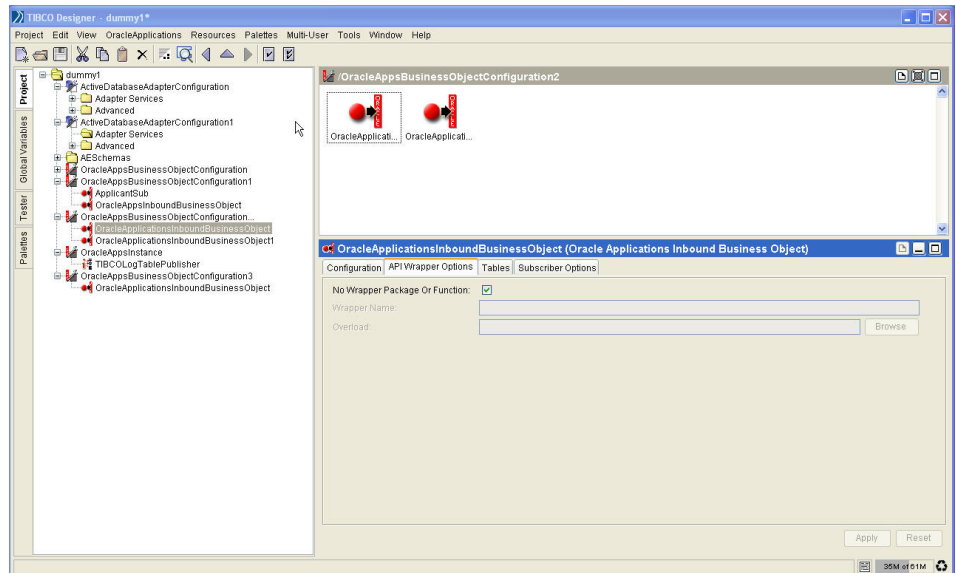
Figure 77 The Inbound Business Object Configuration Panel



3. Enter the appropriate values for the Name, APPS User Name, APPS User Password, APPLSYS User Name, Interface Name, and Overload fields in the Configuration tab. Note that the Interface Name field must be filled in the format of `<packageName>.<methodName>`.
4. Click the **Browse** button at right hand side of the Overload field and you will see an Input Overload message box. Select the correct option from the Please Select an Existing Overload Value drop-down list and click the **OK** button.
5. Click the **Apply** button to apply the object configuration.

There are two kinds of PL/SQL APIs, one with simple types and one with wrappers. When the adapter deals with the boolean type or special types, the process can be complicated. A special type means a type defined within the package instead of the Database. Therefore, a wrapper is needed to simplify this process. JPublisher can generate PL/SQL scripts automatically and encapsulate related procedures and variables, hence, it can act as a wrapper in the adapter to convert complex APIs into simple APIs.

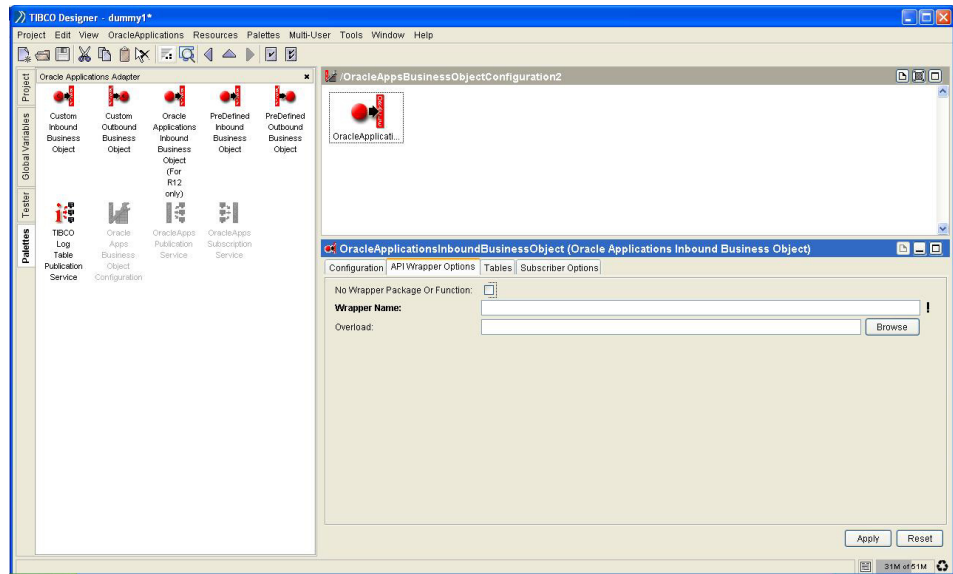
Figure 78 The Inbound Business Object Without Wrapper



If you choose the one with simple types (see Figure 78):

- Ensure that the No Wrapper Package Or Function checkbox is checked in the API Wrapper Options tab and click the **Apply** button.
- Expand the table in the Tables tab to view all the table columns. Ensure that the TIBCO_KEY column is set as the User Key. In other words, the User key column must be checked for the TIBCO_KEY column. All table columns must be set as Used.
- Enter an exception table name in the Exception Table field and click the **Apply** button. A procedure will be invoked automatically and the name of the procedure will populate automatically in the Pre Commit Procedure Name field in the format of <PackageName>.main_adb. This field is grayed out and therefore cannot be changed manually.

Figure 79 The Inbound Business Object With Wrapper



If you choose the one with wrappers (see Figure 79):

- Ensure that the No Wrapper Package Or Function checkbox is unchecked in the API Wrapper Options tab and enter a name of the wrapper in the Wrapper Name field.
- The wrapper name must be represented in the format of `<plsqlPackageName>.<ProcedureName>`. You can get the wrapper name by following the process below:
 - a. Install the JPublisher utility.
 - b. Set the appropriate JPublisher class paths and run the JPublisher utility following the format below:

```
java oracle.jspub.Main -user=<apps_username>/<apps_password>
-url=jdbc:oracle:thin:@<servername>:1521:<SID>
-plsqlmap=always -plsqlfile=<plsqlfilename>
-plsqlpackage=<packagename> -sql=<API_Package_Name>
```

```
-plsqlonly=<ture_or_false>
-dir=<base_Dir_For_Generated_Files>
```

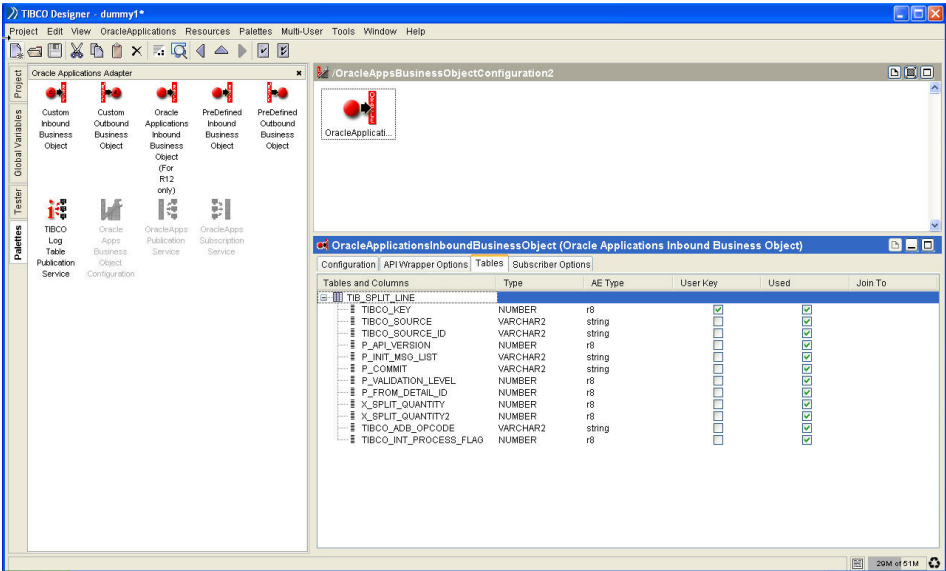
- c. This command will generate two SQL files that are used to get the procedure name for the wrapper.
 - d. Log in to the database as APPS, grant the Execute privilege to the user to access the API that the wrapper is assigned to. Execute the generated SQL scripts under the current adapter user account.
 - e. Log in to the database as APPS, grant the EXECUTE privilege to the user to access the API that the wrapper appointed to by execute the following:

```
GRANT EXECUTE ON BOM_BO_PUB TO <username>;
```
 - f. Log in to the database as the adapter user which you grant the privilege in the step above, then execute the followings:

```
CREATE SYNONYM BOM_BO_PUB FOR APPS.BOM_BO_PUB;
```

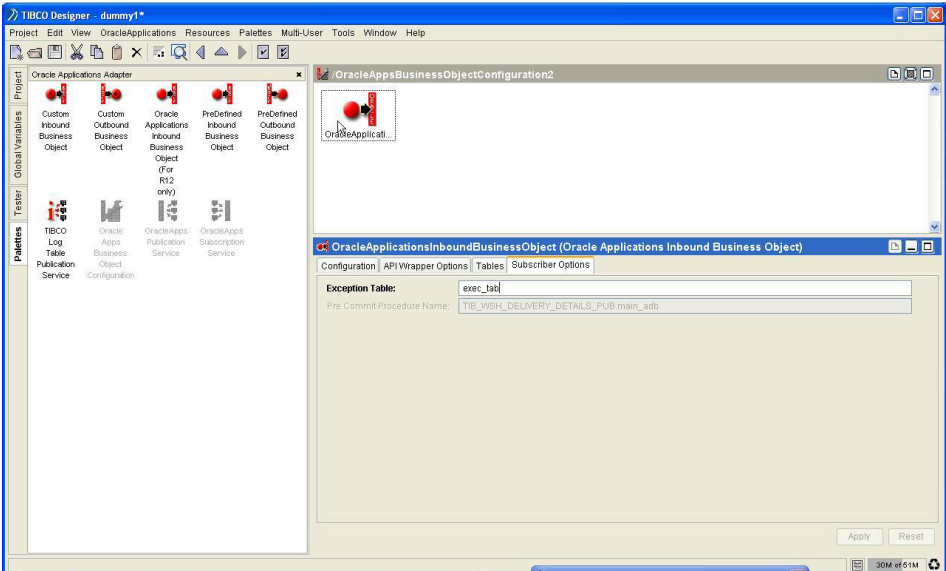
 where BOM_BO_PUB is a package name.
 - g. Execute the generated SQL scripts under the current adapter user account.
 - h. Copy and paste the PL/SQL Package Name into the Wrapper Name field as the first part of its name.
 - i. Open the generated SQL and find the package name that you just provided in the Wrapper Name field. Select a desired procedure from the package and paste its name into the Wrapper Name field as the second part of its name. Do not forget to place a dot to separate these two parts.
- Click the **Browse** button at right hand side of the Overload field and you will see an Input Overload message box. Select the correct option from the Please Select an Existing Overload Value drop-down list and click the **OK** button. Click the **Apply** button to apply the API Wrapper Options configuration.
 - Expand the table in the Tables tab to view all the table columns. Ensure that the TIBCO_KEY column is set as the User Key. In other words, the User Key column must be checked for the TIBCO_KEY column. All table columns must be set as Used (see Figure 80).


Figure 80 The Inbound Business Object Tables Tab



- Enter an exception table name in the Exception Table field and click the **Apply** button. A procedure will be invoked automatically and the name of the procedure will populate automatically in the Pre Commit Procedure Name field in the format of <PackageName>.main_adb. This field is grayed out and therefore cannot be changed manually (see Figure 81).

Figure 81 The Inbound Business Object Subscriber Options Tab



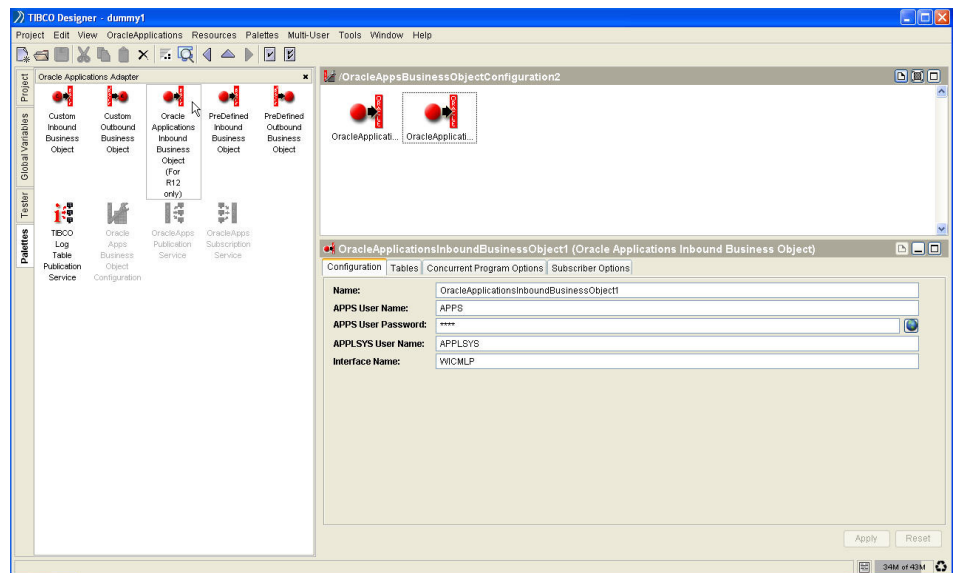
6. Click the  button from the menu bar to save your project.

The Concurrent Program Option

If you choose the CONCURRENT PROGRAM option:

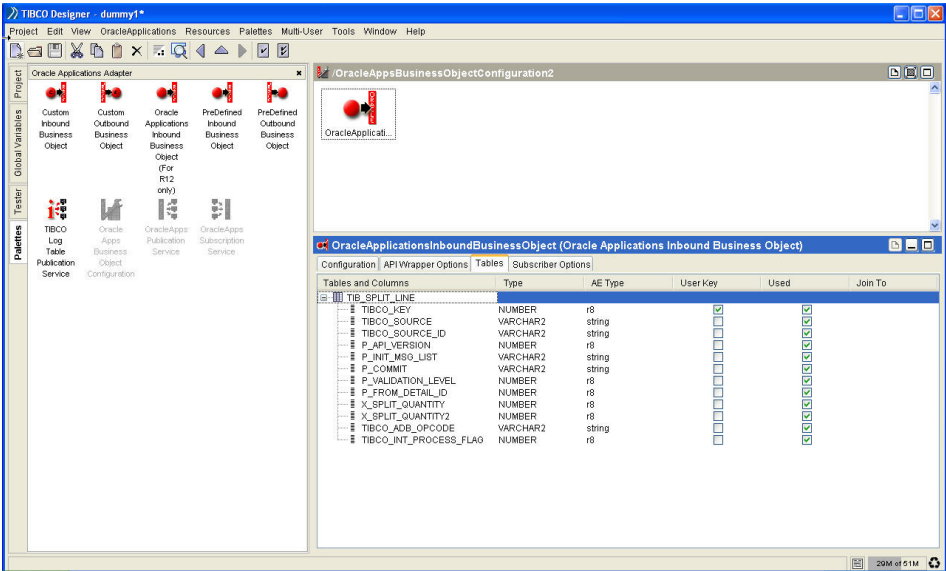
1. Select the **CONCURRENT PROGRAM** option from the Interface Type drop-down list and click the **Apply** button.
2. After applying the settings, you will see four tabs in the configuration panel. They are the Configuration tab, the Tables tab, the Concurrent Program Options tab, and the Subscriber Options tab (see Figure 82).

Figure 82 The Concurrent Programs Configuration



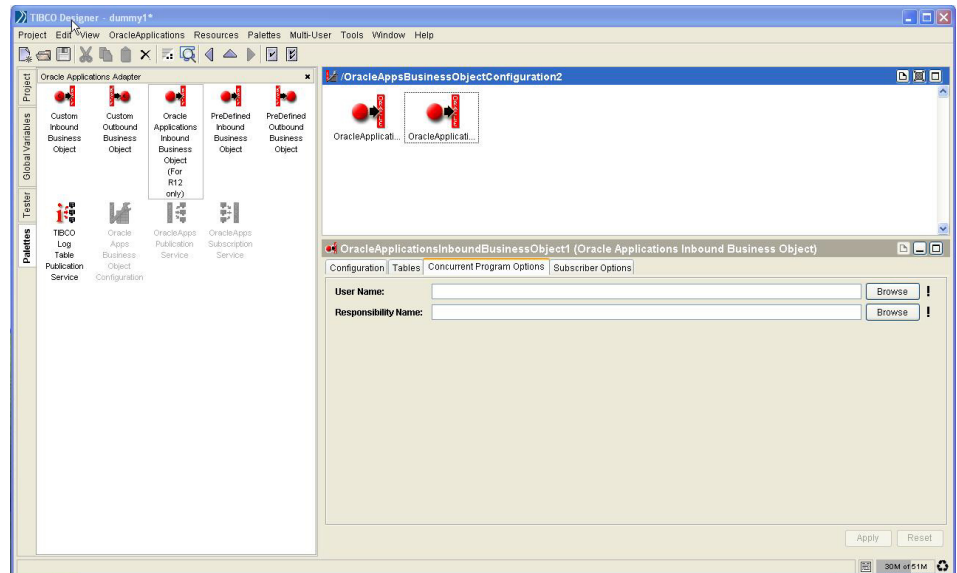
3. Enter the appropriate values for the Name, APPS User Name, APPS User Password, APPLSYS User Name, and Interface Name fields in the Configuration tab.
4. Click the **Apply** button to apply the object configuration.
5. Expand the table in the Tables tab to view all the table columns. Ensure that the TIBCO_KEY column is set as the User Key. In another word, the User Key column must be checked for the TIBCO_KEY column. All table columns must be set as Used (see Figure 83).

Figure 83 The Inbound Business Object Tables Tab



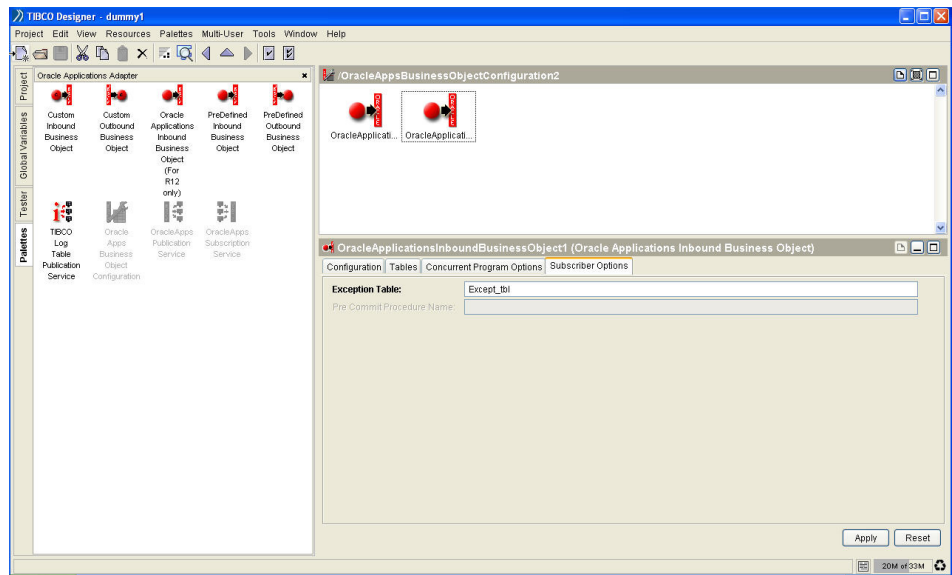
6. Click the **Browse** button at the right hand side of the User Name field in the Concurrent Program Options tab. A message box will pop up with a list of user names. Select the appropriate option from the list and click the **OK** button (see Figure 84).
7. Click the **Browse** button at the right hand side of the Responsibility Name field in the Concurrent Program Options tab. A message box will pop up with a list of responsibility names. Select the desired responsibility from the list and click the **OK** button (see Figure 84).


Figure 84 The Concurrent Program Options Tab



8. Click the **Apply** button to apply the Concurrent Program Options configuration.
9. Enter an exception table name in the Exception Table field and click the **Apply** button. A procedure will be invoked automatically and the name of the procedure will populate automatically in the Pre Commit Procedure Name field in the format of `<PackageName>.main_adb`. This field is grayed out and therefore cannot be changed manually (see Figure 85).

Figure 85 The Object Subscriber Options Tab



10. Click the  button from the menu bar to save your project.

Functionalities of Tabs

You can set parameters under the following tabs:

- [The Configuration Tab](#)
- [The API Wrapper Options Tab](#)
- [The Concurrent Program Options Tab](#)
- [The Tables Tab](#)
- [The Subscriber Options Tab](#)

The Configuration Tab

The Configuration tab has the following options (see Figure 86):

- Name
The service name associated with this Oracle Applications Inbound Business Object.
- APPS User Name
The User Name used to connect to the Oracle Applications database user (schema) of the Oracle Applications database. This field is mandatory.

- APPS User Password

The user password used to connect to the Oracle Applications database user (schema) of the Oracle Applications database. This field is mandatory.

- APPLSYS User Name

Another user name used to connect to the Oracle Applications database user (schema) of the Oracle Applications database. This field is mandatory.

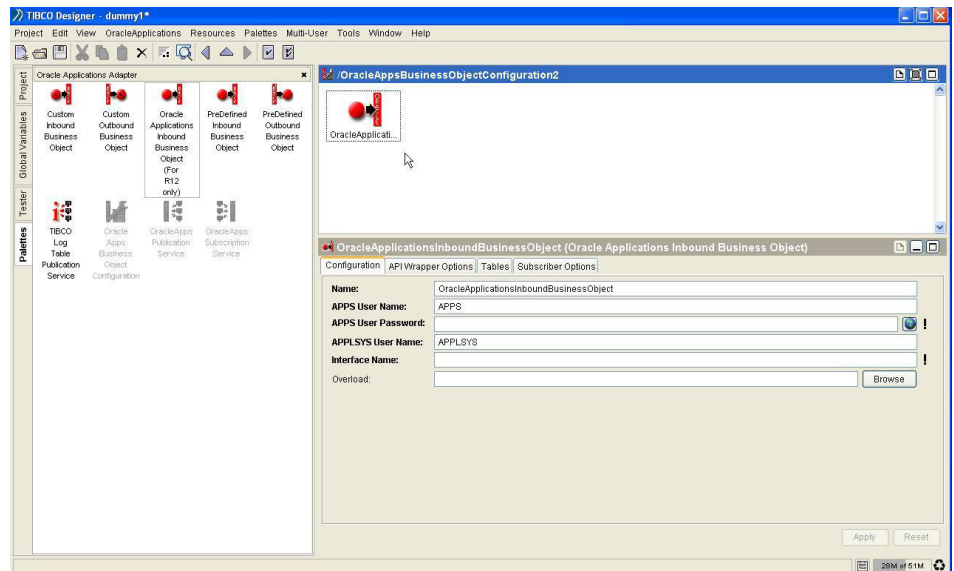
- Interface Name

This name is used to dedicate a concurrent program name or dedicate the corresponding PL/SQL API.

- Overload

A numerical attribute which contains a unique number to differentiate interfaces that share the same name, but takes a different number of parameters.

Figure 86 The Business Object Configuration Tab



The API Wrapper Options Tab

The API Wrapper Options tab has the following options (see Figure 87):

- The No Wrapper Package Or Function Checkbox

Decides whether this API is a simple type or a type with wrapper.

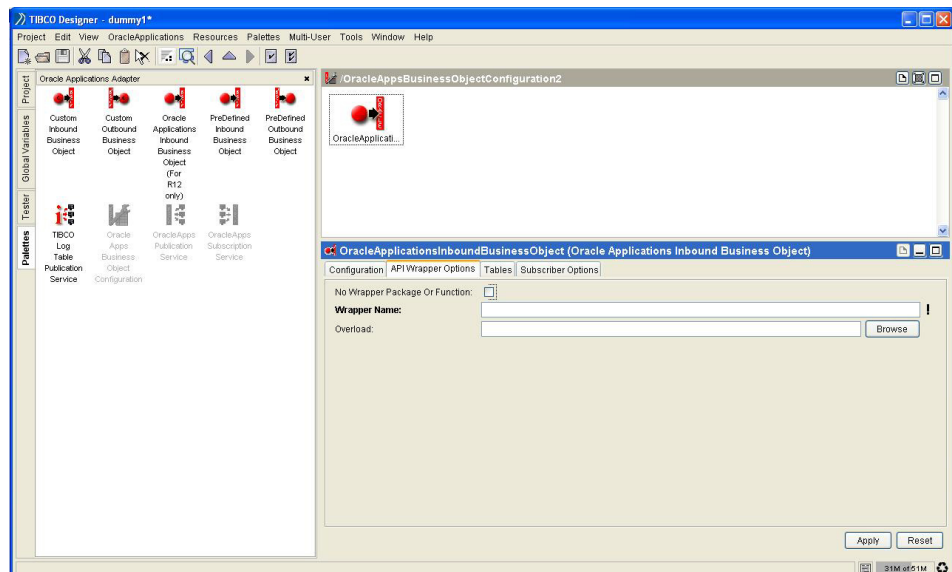
- Wrapper Name

This name is used by JPublisher for the wrapping process or to process complicated data types.

- **Overload**

A numerical attribute which contains a unique number to differentiate interfaces that share the same name, but takes a different number of parameters. If the wrapper name is unique, then the overload value will be NULL.

Figure 87 The Business Object API Wrapper Options Tab



The Concurrent Program Options Tab

The Concurrent Program Options tab has the following options (see Figure 88):

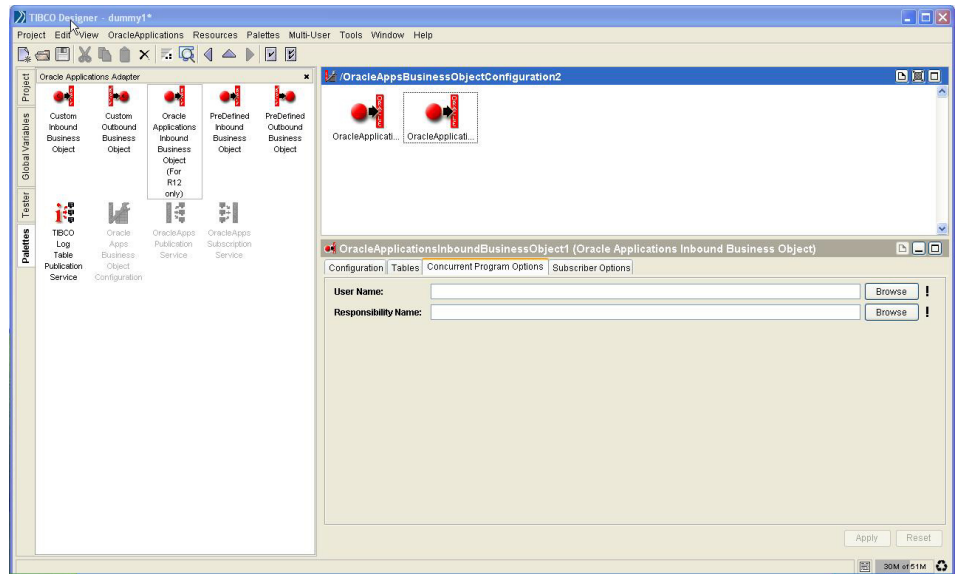
- **User Name**

This field is used to derive the User ID. This ID is required when running the concurrent programs without registering it or submitting import programs.

- **Responsibility Name**

This field is used to derive the Responsibility ID, which is required when running the concurrent programs without registering it or submitting import programs.

Figure 88 The Concurrent Program Options Tab



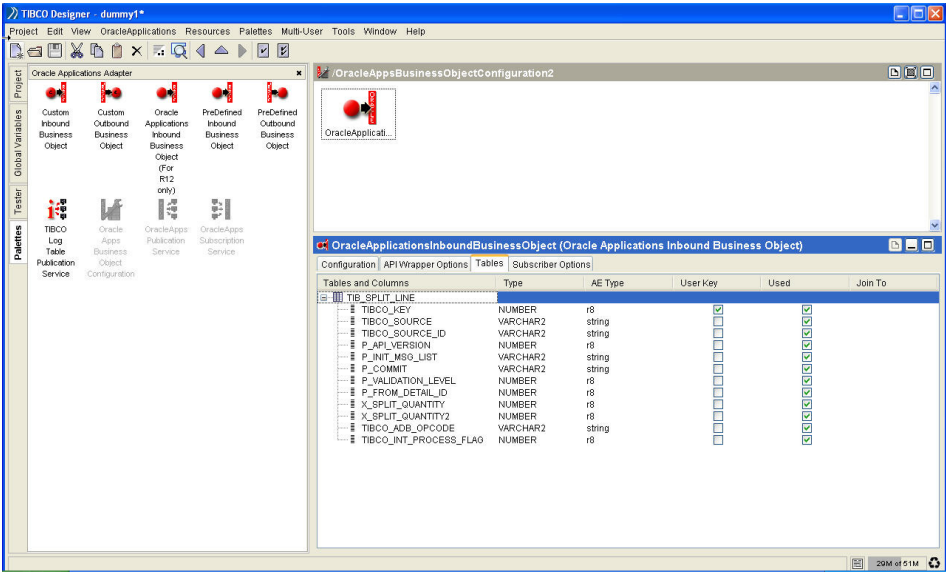
The Tables Tab

This tab is used to configure the table settings for the Business Object. The Tables tab has the following column headings (see Figure 89):

Columns

- **Tables and Columns**
Lists the table and its columns.
- **Type**
Lists the primitive type.
- **AE Type**
Lists the primitive type mapped to an ActiveEnterprise type.
- **User Key**
Defines the column as user key if it is checked.
- **Used**
Enables the column to be updated when a message arrives if it is checked.
- **Join To**
Joins a child table to its parent table. The child table presents a drop-down list containing columns which have been marked as User Keys of the parent table.

Figure 89 The Inbound Business Object Tables Tab



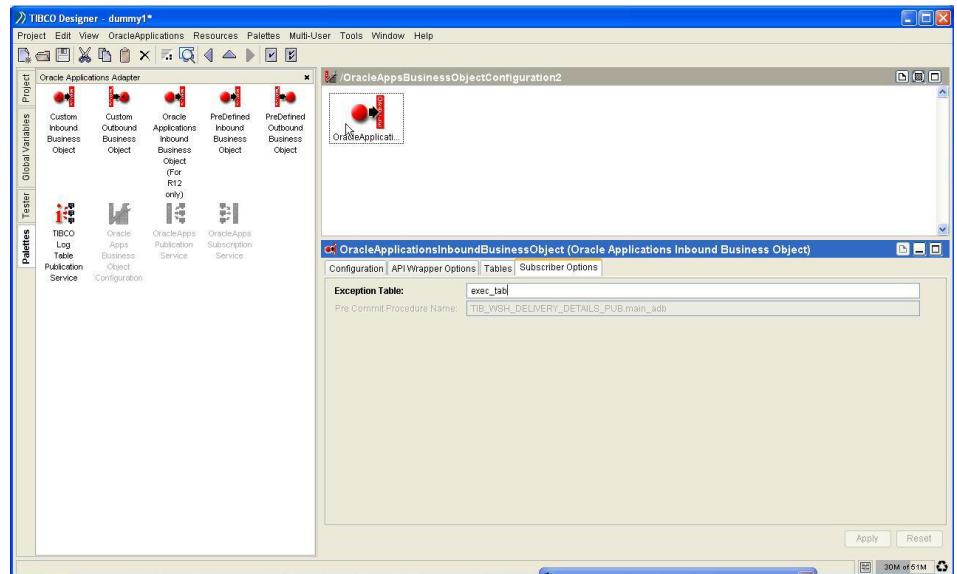
The Subscriber Options Tab

The Configuration tab has the following options (see Figure 90):

- Exceptions Table
The name of the exception table. If the adapter cannot write to the subscriber table, it will write the error messages in this exception table. If the table does not exist, the subscriber adapter will create it by default.

The exception table *cannot* contain any user-created columns which contain the prefix ADB_. This prefix is reserved for use with the adapter. See *TIBCO ActiveMatrix Adapter for Database User's Guide* for more details.
- Pre Commit Stored Procedure Name
The value entered represents the PL/SQL stored procedure name the subscriber will call after the insertion, update, or deletion database operations and prior to the commit. You can use this stored procedure as an intermediate to accomplish further processing inside the database and return the results to the adapter. The adapter will not call a stored procedure if you leave this field blank. The value should be in the form of <PackageName>.main_adb. The Oracle Application Palette will return an error if this value is not specified in the required format.

Figure 90 The Object Subscriber Options Tab



Oracle Applications Inbound Business Object (for R12 Only) Services Configuration Examples

Oracle Applications Inbound Business Object (for R12 Only) allows for three types of services: API Simple services, API Wrapper services, and Concurrent Program services. To configure any of these services, you need first to select the corresponding APIs or concurrent programs from Oracle E-Business Suite (Oracle EBS) Integration Repository. Refer to [Appendix P, Selecting APIs and Concurrent Programs from Oracle EBS R12 Integration Repository, on page 511](#) for instructions on selecting APIs or concurrent programs from Oracle E-Business Suite (Oracle EBS) Integration Repository.

API Simple Service Configuration

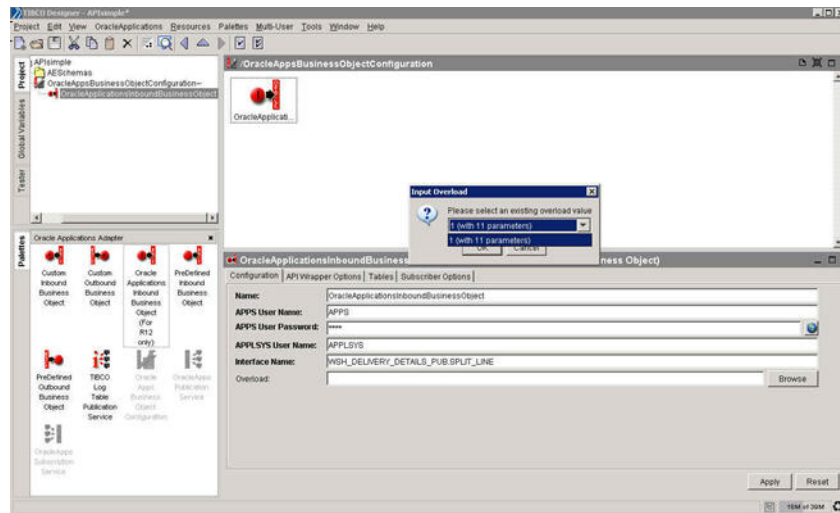
To configure an API Simple service:

1. Open TIBCO Designer and create a new project, naming the project as APISimple.
2. Restore the Oracle Applications Adapter window within the Palettes tab. Drag the **Oracle Apps Business Object Configuration** button and drop it into the Design panel.
3. Enter the appropriate values for JDBC Driver, JDBC URL, User name, Password and Request Timeout (milliseconds) under the Design-time

Connection tab. Click the **Test Connection...** button to ensure the connection is set up successfully.

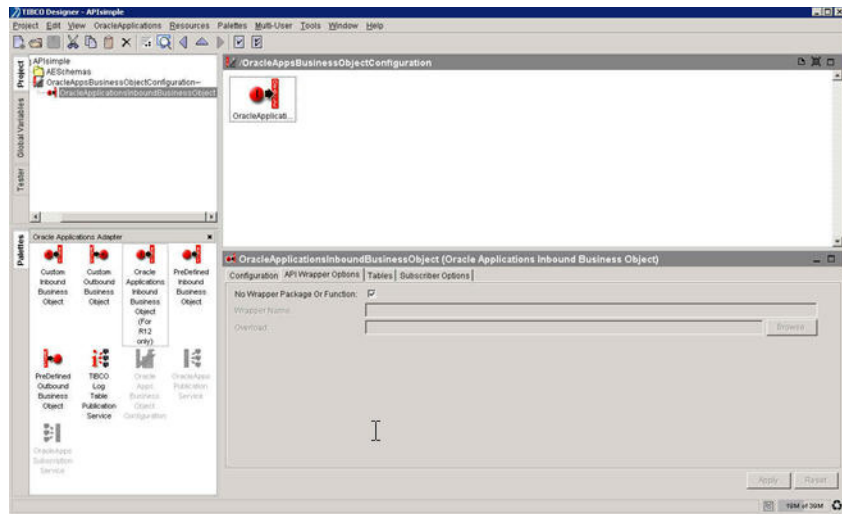
4. Drag and drop the **Oracle Applications Inbound Business Object (For R12 Only)** button from Palettes to the design panel. In the Configuration tab, select **PL/SQL API** from the Interface Type drop-down list. Click the **Apply** button.
5. Select the Configuration tab and enter APPS username, APPS password, APPLSYS username, interface name, and overload in the corresponding fields as prompted, as shown in [Figure 91, API Simple Service Configuration \(1\)](#). The interface name needs to be in the format of *<package name>.<method name>*. You can obtain the interface name from Oracle EBS R12 Integration Repository. Refer to [Appendix P, Selecting APIs and Concurrent Programs from Oracle EBS R12 Integration Repository](#), on page 511 for more.

Figure 91 API Simple Service Configuration (1)



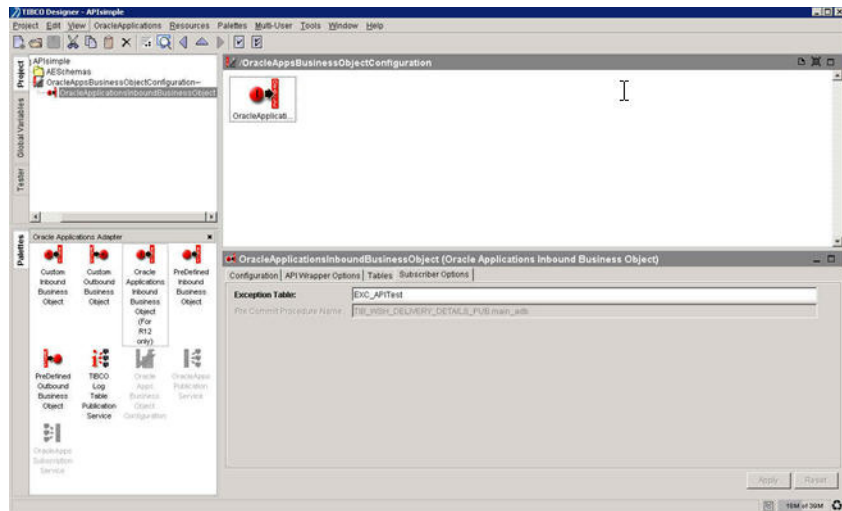
6. Select the **API Wrapper Options** tab, make sure the **No Wrapper Package Or Function** check box is checked, and click the **Apply** button to proceed, as shown in [Figure 92, API Simple Service Configuration \(2\)](#).

Figure 92 API Simple Service Configuration (2)



7. Select the Subscriber Options tab, enter the name of the exception table in the Exception Table text box, and click the **Apply** button, as shown in Figure 93, API Simple Service Configuration (3).

Figure 93 API Simple Service Configuration (3)



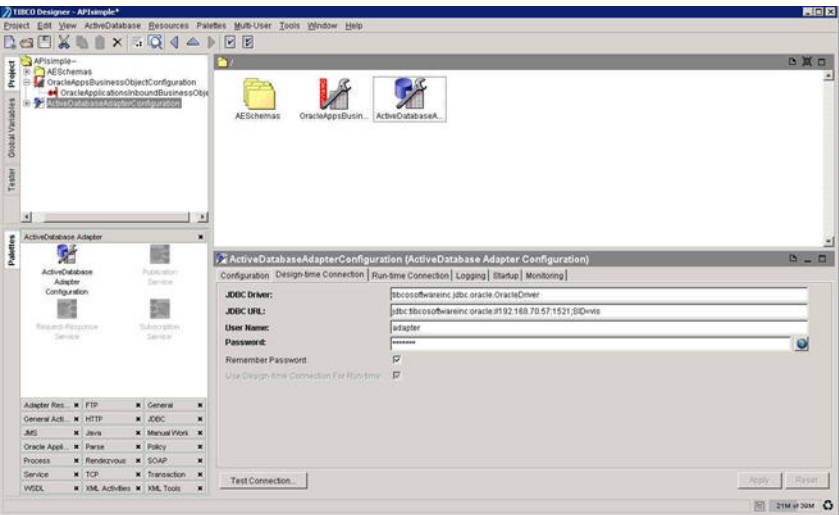


The pre commit procedure name will be generated by the adapter after you provide the exception table name and click the **Apply** button.

Also, the adapter generates two SQL scripts under `<adoraapp_home>/<version>/sql` for test. You need to execute the two scripts before run the adapter.

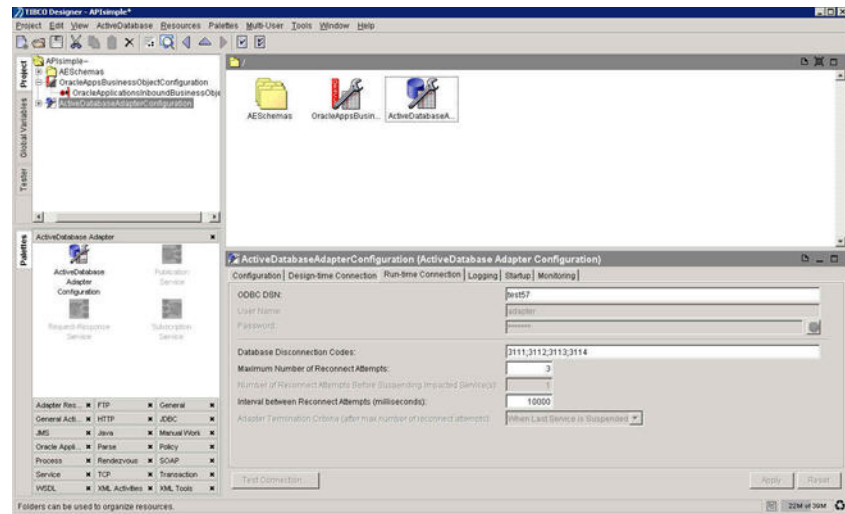
- 8. Drag and drop the **ActiveDatabase Adapter Configuration** button to the design panel. Set the JDBC settings in the Design-time Connection tab, and test the connection by click the **Test Connection...** button, as shown in [Figure 94, API Simple Service Configuration \(4\)](#).

Figure 94 API Simple Service Configuration (4)



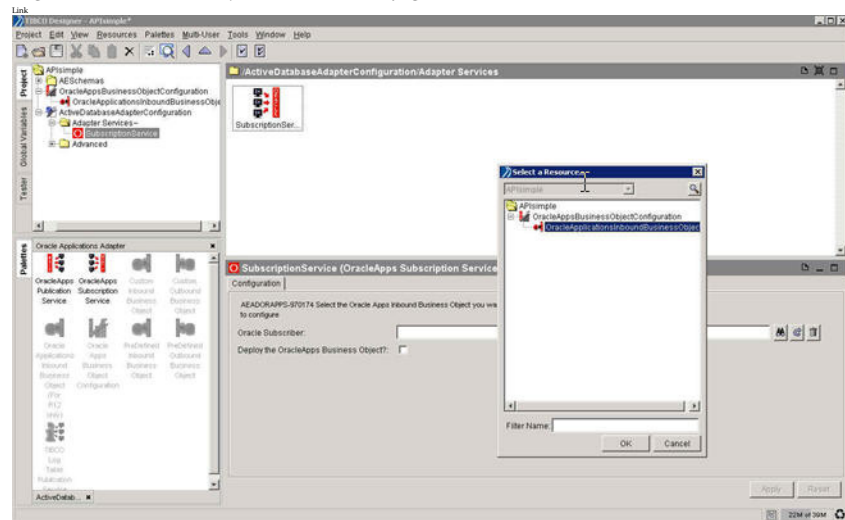
- 9. Select the Run-time Connection tab and set ODBC DSN as prompted, as shown in [Figure 95, API Simple Service Configuration \(5\)](#).

Figure 95 API Simple Service Configuration (5)



10. Drag and drop the **OracleApps Subscription Service** button to the design panel. In the Configuration tab, click the **Browse Resources** button and select the Oracle Apps Business Object from the dialog box that appears, as shown in Figure 96, API Simple Service Configuration (6).

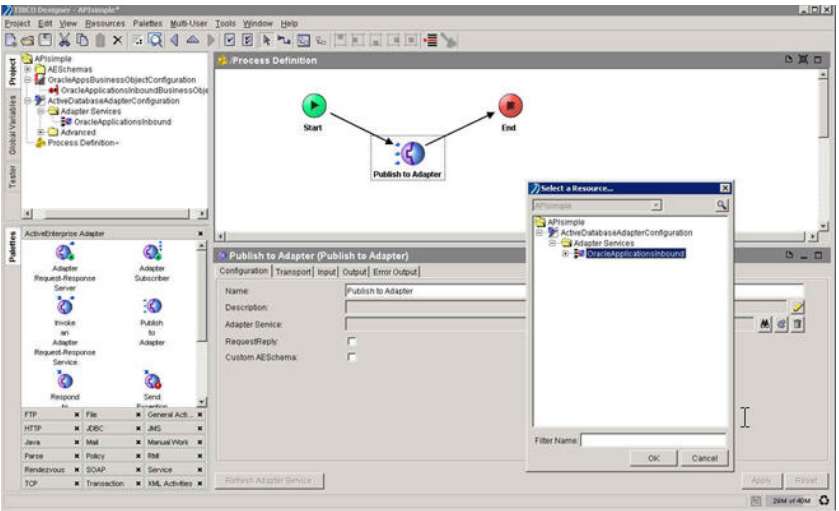
Figure 96 API Simple Service Configuration (6)



11. Link an ADB instance to the Oracle Application Business Object and create a new process in TIBCO Designer.

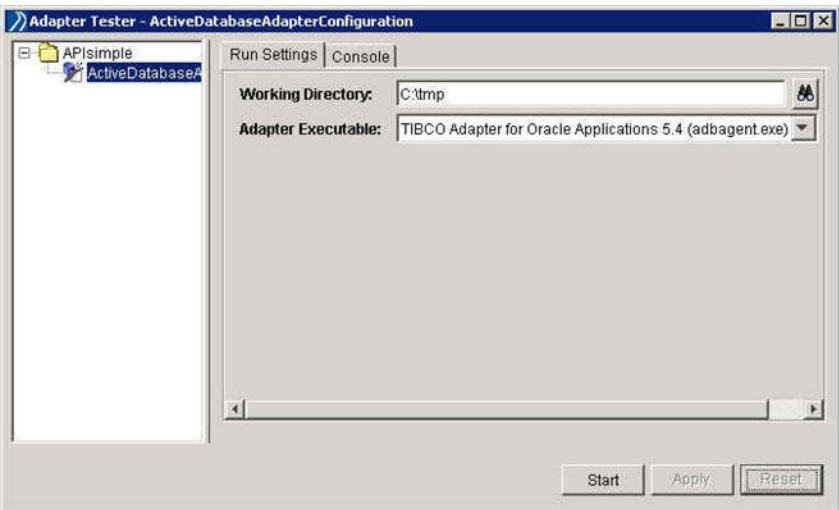
- 12. Add the Publish to Adapter activity to the process and select the ADB instance as an adapter service, as shown in [Figure 97, API Simple Service Configuration \(7\)](#).

Figure 97 API Simple Service Configuration (7)



- 13. Switch to the **Input** tab and set activity input. Execute the SQL scripts to create database objects.
- 14. Configure the adapter named Adapter Tester (as shown in [Figure 98, Configuring an Adapter](#)) and start the adapter.

Figure 98 Configuring an Adapter



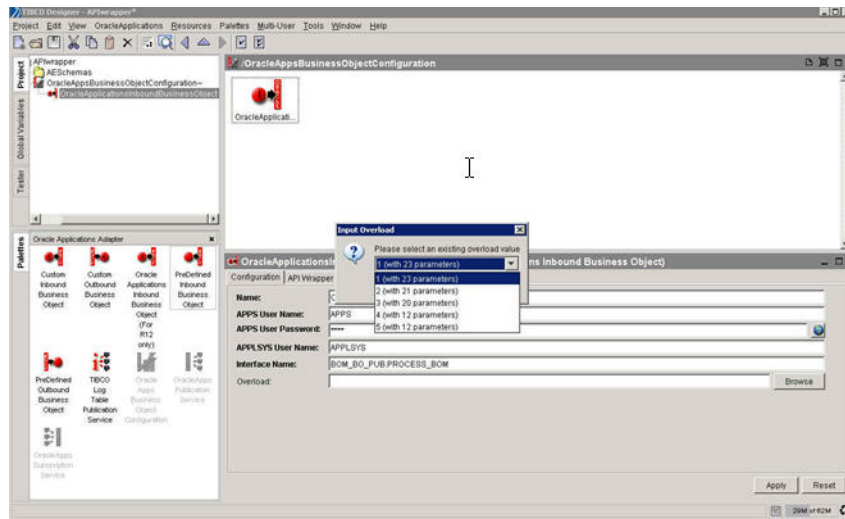
15. Start the process.

API Wrapper Service Configuration

To configure an API Wrapper service:

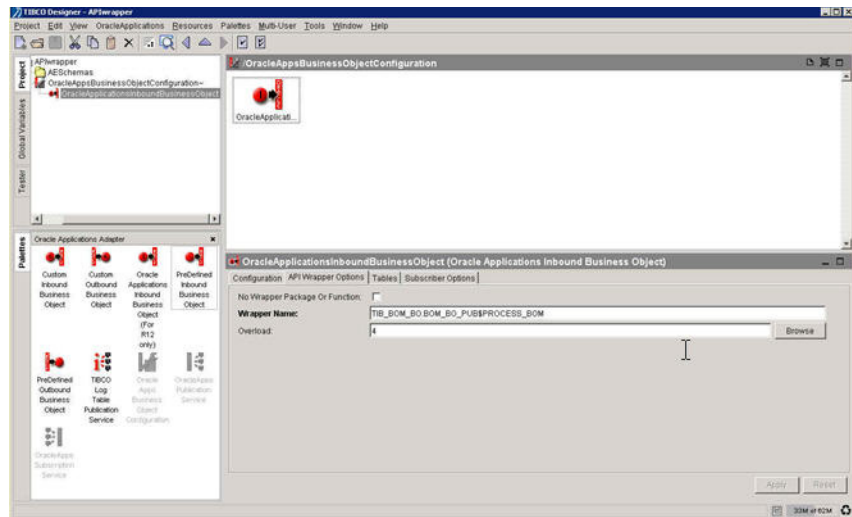
1. Open TIBCO Designer and create a new project, naming the project as **APIWrapper**.
2. Restore the Oracle Applications Adapter window within the Palettes tab. Drag the **Oracle Apps Business Object Configuration** button and drop it into the Design panel.
3. Enter the appropriate values for JDBC Driver, JDBC URL, User name, Password and Request Timeout (milliseconds) under the Design-time Connection tab. Click the **Test Connection...** button to ensure the connection is set up successfully.
4. Drag and drop the **Oracle Applications Inbound Business Object (For R12 Only)** button from Palettes to the design panel. In the Configuration tab, select **PL/SQL API** from the Interface Type drop-down list. Click the **Apply** button.
5. Select the Configuration tab and enter APPS username, APPS password, APPLSYS username, interface name, and overload in the corresponding fields as prompted, as shown in [Figure 99, API Wrapper Service Configuration \(1\)](#). The interface name needs to be in the format of *<package name>.<method name>*. You can obtain the interface name from Oracle EBS R12 Integration Repository. Refer to [Appendix P, Selecting APIs and Concurrent Programs from Oracle EBS R12 Integration Repository, on page 511](#) for more.

Figure 99 API Wrapper Service Configuration (1)



6. Launch JPublisher, execute the wrapper script to generate the required wrapper database object in the database. For more information on generating wrapper database objects using JPublisher, refer to [Appendix O, Tips on Using Oracle JPublisher, on page 503](#).
7. Select the **API Wrapper Options** tab. Make sure the **No Wrapper Package Or Function** check box is not checked, enter the name of the wrapper object generated in step 6 in the Wrapper Name text box, and have the Overload setting be the same as that in the Configuration tab. Click the **Apply** button to proceed, as shown in [Figure 100, API Wrapper Service Configuration \(2\)](#).

Figure 100 API Wrapper Service Configuration (2)



8. Select the **Subscriber Options** tab, enter the name of the exception table in the Exception Table text box, and click the **Apply** button, as shown in [Figure 101, API Wrapper Service Configuration \(3\)](#).



The pre commit procedure name will be generated by the adapter after you provide the exception table name and click the **Apply** button.

Also, the adapter generates two SQL scripts under `<adoraapp_home>/<version>/sql` for test. You need to execute the two scripts before run the adapter.

9. Drag and drop the **ActiveDatabase Adapter Configuration** button to the design panel. Set the JDBC settings in the Design-time Connection tab and test the connection by click the **Test Connection...** button, as shown in [Figure 102, API Wrapper Service Configuration \(4\)](#).

Figure 101 API Wrapper Service Configuration (3)

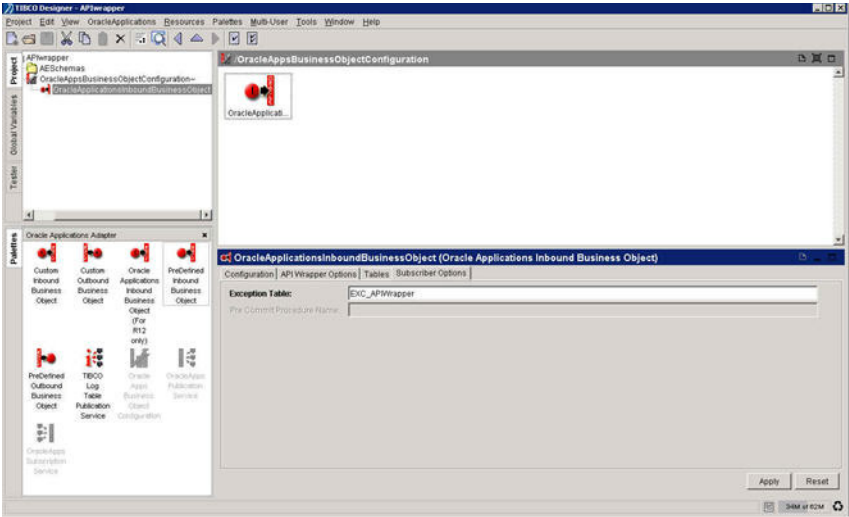
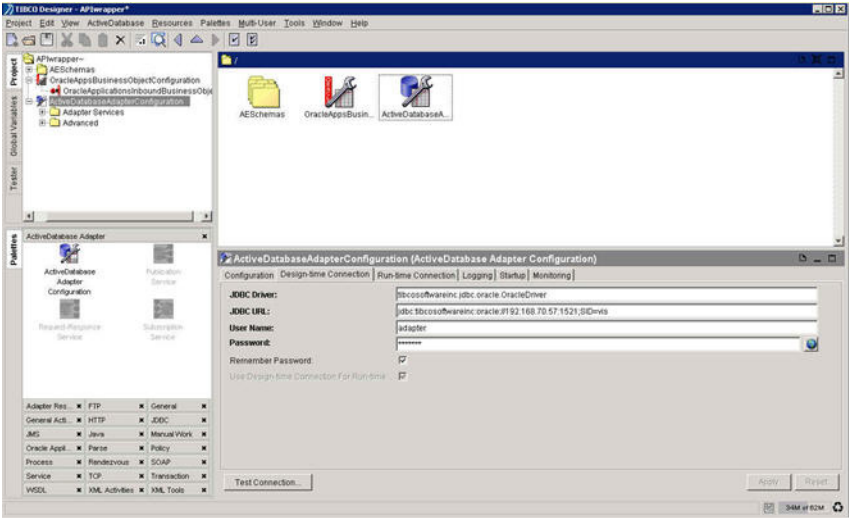
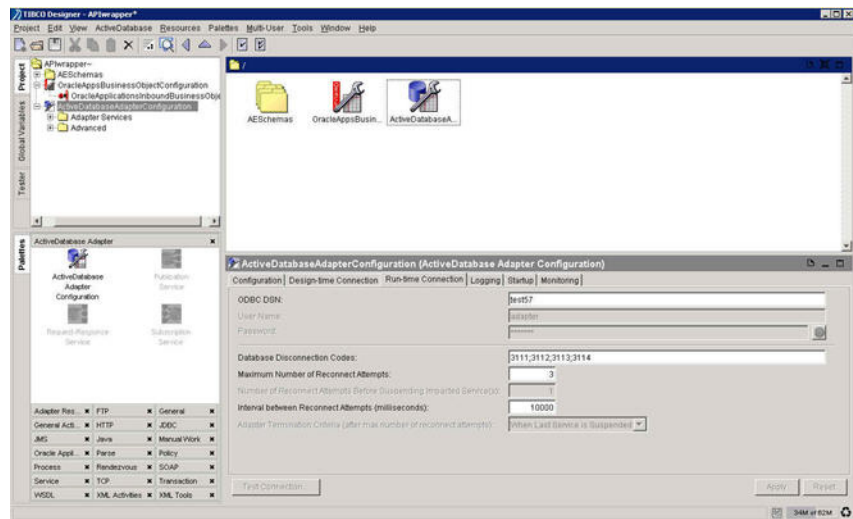


Figure 102 API Wrapper Service Configuration (4)



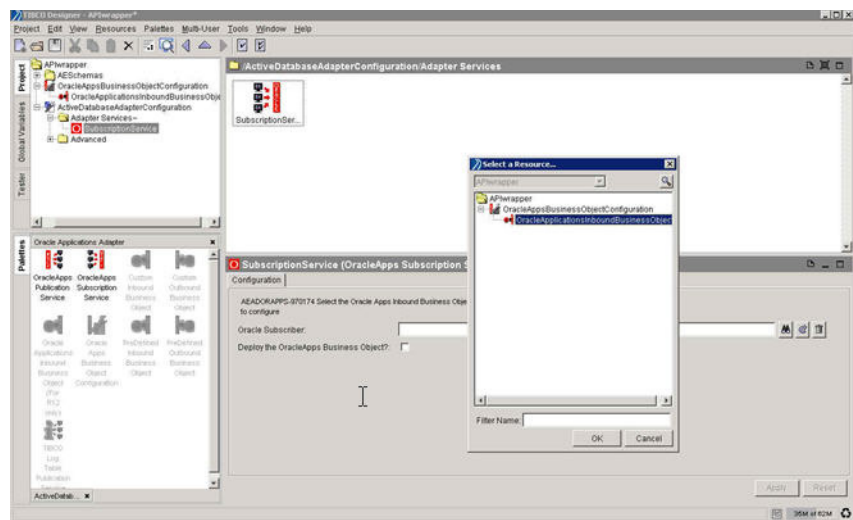
10. Select the Run-time Connection tab and set ODBC DSN as prompted, as shown in Figure 103, API Wrapper Service Configuration (5).

Figure 103 API Wrapper Service Configuration (5)



11. Drag and drop the **OracleApps Subscription Service** button to the design panel. In the Configuration tab, click the **Browse Resources** button and select the Oracle Apps Business Object from the dialog box that appears, as shown in [Figure 104, API Wrapper Service Configuration \(6\)](#).

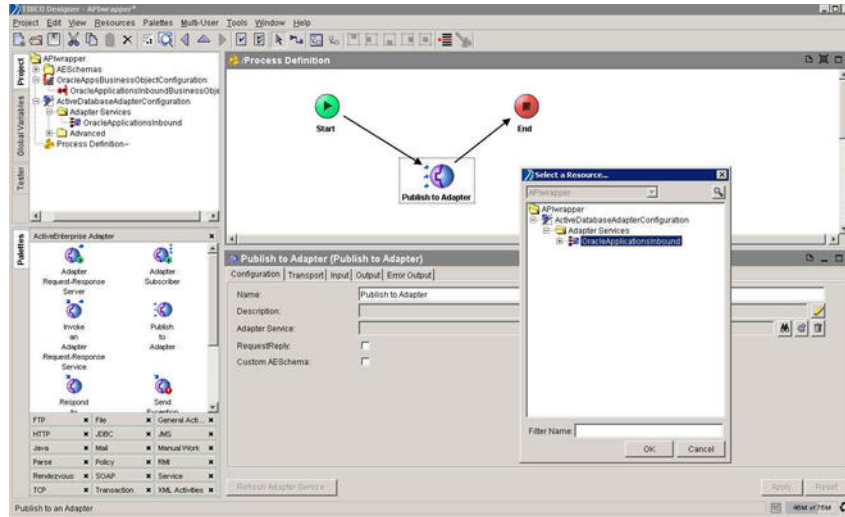
Figure 104 API Wrapper Service Configuration (6)



12. Link an ADB instance to the Oracle Application Business Object and create a new process in TIBCO Designer.

13. Add the Publish to Adapter activity to the process and select the ADB instance as an adapter service, as shown in [Figure 105, API Wrapper Service Configuration \(7\)](#).

Figure 105 API Wrapper Service Configuration (7)



14. Switch to the **Input** tab and set activity input. Execute the SQL scripts to create database objects.
15. Configure the adapter named Adapter Tester (as shown in [Figure 98, Configuring an Adapter](#)) and start the adapter.
16. Start the process.

Concurrent Program Service Configuration

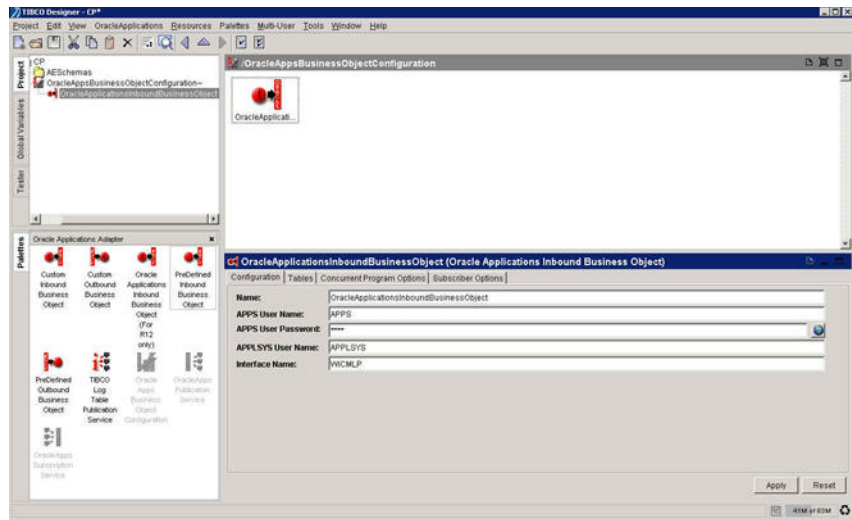
To configure a Concurrent Program service:

1. Open TIBCO Designer and create a new project, naming the project as CP.
2. Restore the Oracle Applications Adapter window within the Palettes tab. Drag the **Oracle Apps Business Object Configuration** button and drop it into the Design panel.
3. Enter the appropriate values for JDBC Driver, JDBC URL, User name, Password and Request Timeout (milliseconds) under the Design-time Connection tab. Click the **Test Connection...** button to ensure the connection is set up successfully.
4. Drag and drop the **Oracle Applications Inbound Business Object (For R12 Only)** button from Palettes to the design panel. In the Configuration tab,

select **CONCURRENT PROGRAM** from the Interface Type drop-down list. Click the **Apply** button.

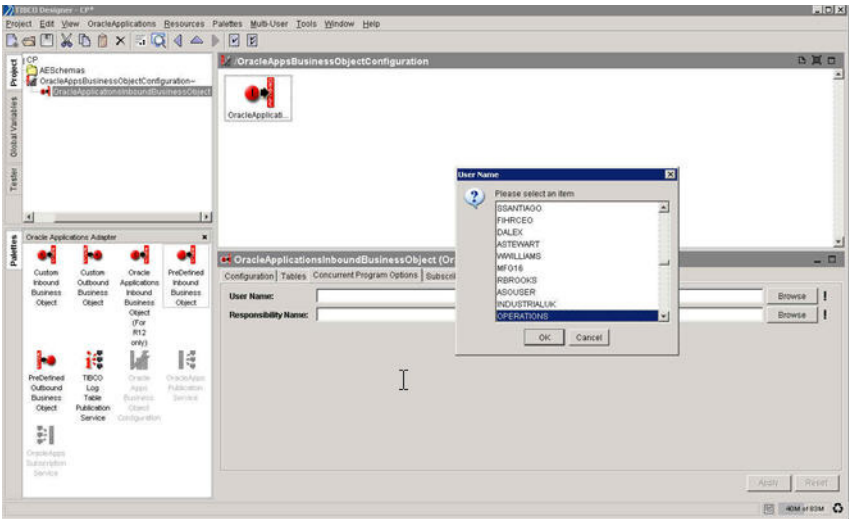
5. Select the Configuration tab and enter APPS username, APPS password, APPLSYS username, and interface name in the corresponding fields as prompted, as shown in [Figure 106, Concurrent Program Service Configuration \(1\)](#). You can obtain the interface name from Oracle EBS R12 Integration Repository. Refer to [Appendix P, Selecting APIs and Concurrent Programs from Oracle EBS R12 Integration Repository](#), on page 511 for more.

Figure 106 Concurrent Program Service Configuration (1)



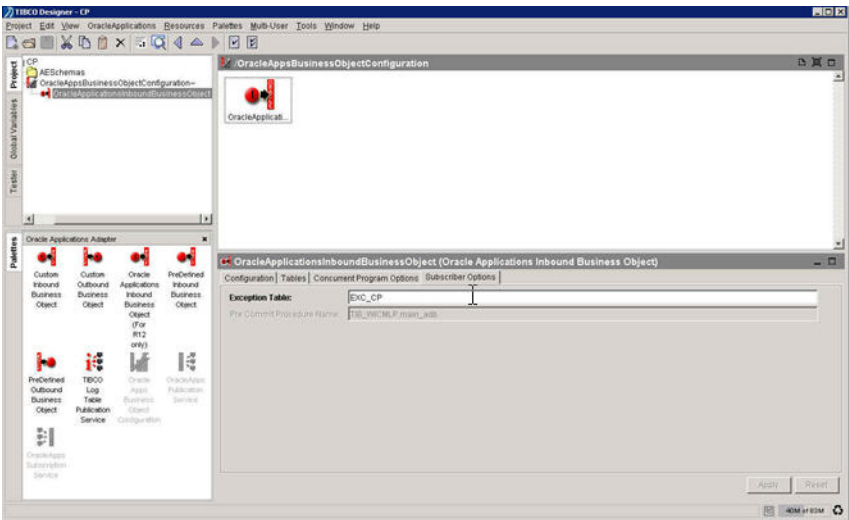
6. Switch to the Concurrent Program Options tab, set the username and responsibility name as prompted, as shown in [Figure 107, Concurrent Program Service Configuration \(2\)](#).

Figure 107 Concurrent Program Service Configuration (2)



7. Select the **Subscriber Options** tab, enter the name of the exception table in the Exception Table text box, and click the **Apply** button, as shown in [Figure 108, Concurrent Program Service Configuration \(3\)](#).

Figure 108 Concurrent Program Service Configuration (3)



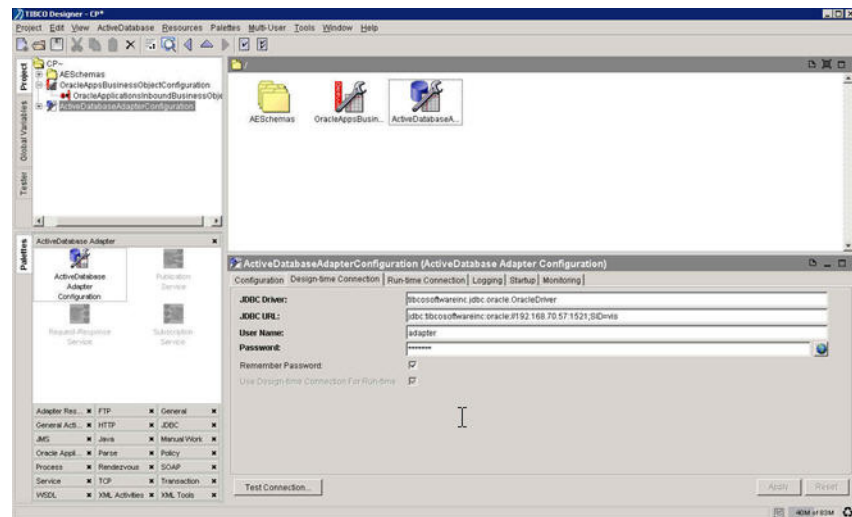


The pre commit procedure name will be generated by the adapter after you provide the exception table name and click the **Apply** button.

Also, the adapter generates two SQL scripts under `<adoraapp_home>/<version>/sql` for test. You need to execute the two scripts before run the adapter.

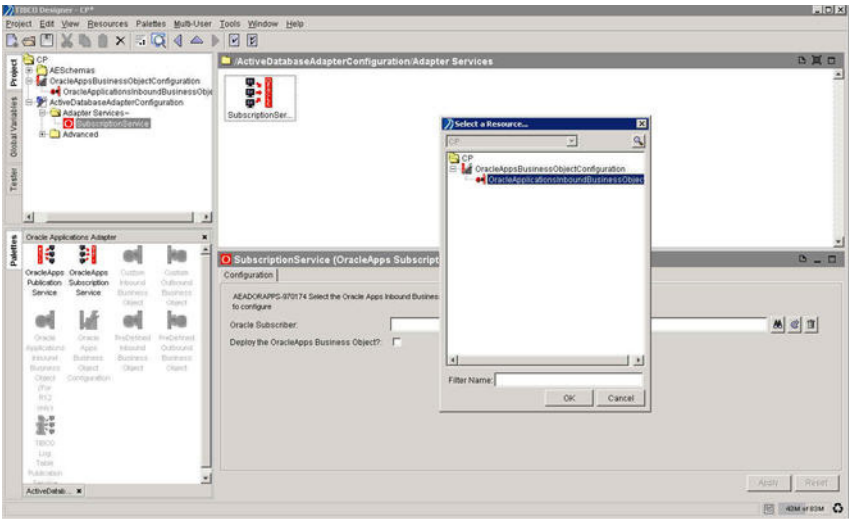
8. Drag and drop the **ActiveDatabase Adapter Configuration** button to the design panel. Set the JDBC settings in the Design-time Connection tab and test the connection by click the **Test Connection...** button, as shown in [Figure 109, Concurrent Program Service Configuration \(4\)](#).

Figure 109 Concurrent Program Service Configuration (4)



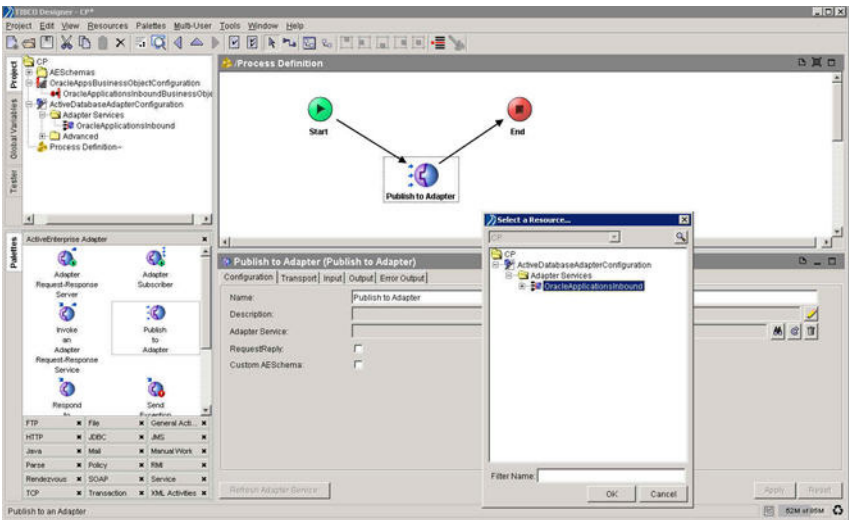
9. Select the Run-time Connection tab and set ODBC DSN as prompted.
10. Drag and drop the **OracleApps Subscription Service** button to the design panel. In the Configuration tab, click the **Browse Resources** button and select the **Oracle Apps Business Object** from the dialog box that appears, as shown in [Figure 110, Concurrent Program Service configuration \(5\)](#).

Figure 110 Concurrent Program Service configuration (5)



11. Link an ADB instance to the Oracle Application Business Object and create a new process in TIBCO Designer.
12. Add the Publish to Adapter activity to the process and select the ADB instance as an adapter service, as shown in [Figure 111, Concurrent Program Service Configuration \(6\)](#).

Figure 111 Concurrent Program Service Configuration (6)



13. Switch to the **Input** tab and set activity input. Execute the SQL scripts to create database objects.

14. Configure the adapter named Adapter Tester (as shown in [Figure 98, Configuring an Adapter](#)) and start the adapter.
15. Start the process.

TIBCO Log Table Publisher Business Object Configuration

Drag the **TIBCO Log Table Publication Service** button from the palette panel and drop it into the design panel. A connection is established in the database that is used to fetch the TIBCO Log Table schema.

The name of the table is `TIB_INT_LOG_SUB` and this table's user key is `ERROR_SEQUENCE_ID`. This value cannot be changed.

To configure a TIBCO Log Table Publisher Business Object:

- In the Configuration tab, specify a name for the Log Publisher Business Object. Enter the appropriate value for Table - TableSpace and Index - TableSpace fields.
- You will see the Log Table `TIB_INT_LOG_SUB` in the Tables tab. You *cannot* change any properties in this tab.
- The TIBCO Log Table Message Subjects tab presents a list of all the available PreDefined Inbound Business Objects (transactions). Each of the Inbound Business Objects have a Short Transaction Name that is populated when a record is inserted into the TIBCO Log Table by the corresponding PL/SQL Procedure. The Log Publisher Business Object can be configured to publish data on select transactions when the relevant records are inserted into the TIBCO Log Table.

The transactions can be selected and message subjects can be specified for the selected transactions. Messages from the TIBCO Log Table is published on these subjects when a record corresponding to a selected transaction (Business Object) is inserted into the TIBCO Log Table. The mode of publishing is always `Publish By Value`, so that incremental changes in the TIBCO Log Table can be published.

- The Adapter Options tab displays the Storage Mode and Publishing Table properties. You cannot alter the setting of these fields. The Publishing Table is fixed for the TIBCO Log Table Publisher Business Object.



Do not change any properties in the Advanced tab.

For details on Customizing a TIBCO Log Table Business Object, see [Customizing a Business Object on page 230](#) in this chapter.

You can set parameters under the following tabs:

- [The Configuration Tab](#)
- [The Tables Tab](#)

- [The Adapter Options Tab](#)
- [The TIBCO Log Table Message Subjects Tab](#)
- [Database Components](#)

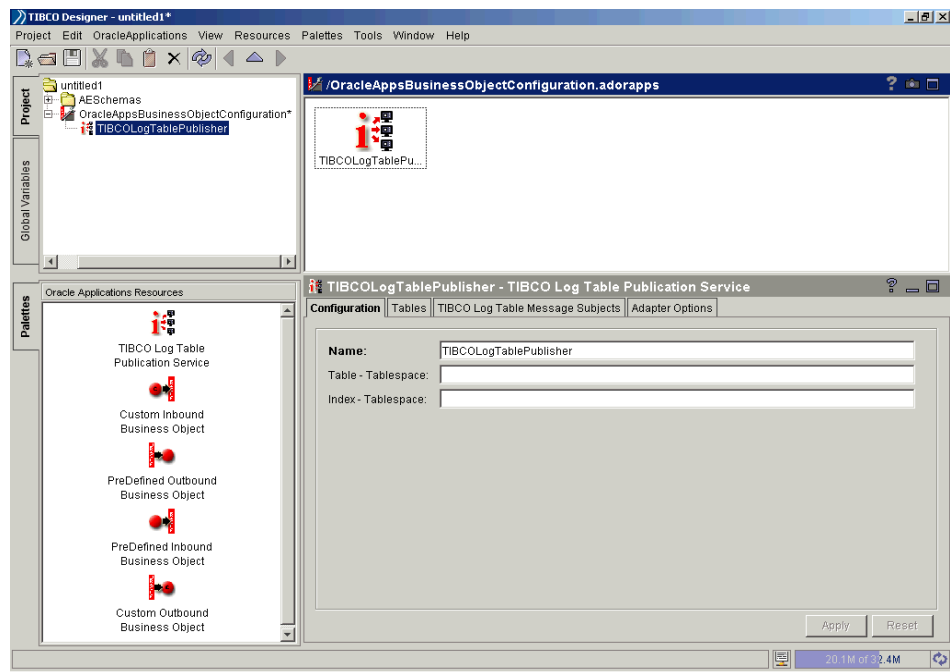
The Configuration Tab

The following parameters are available after a TIBCO Log Table Publisher has been dragged from the palette panel and dropped to the design panel:

- **Name**
Displays the service name associated with this PreDefined service.
- **Table Tablespace**
Specifies the tablespace name of the newly created table. If this field is left blank, a database table required for the Business Object is created in the default tablespace (allocated by the database system).
- **Index Tablespace**
Specifies the tablespace name of the newly created indexes. If this field is left blank, a database index required for the Business Object is created in the default tablespace (allocated by the database system).




Sample Screen

Figure 112 The TIBCO Log Table Publisher Configuration Tab




The Tables Tab


The Tables tab is *not* available for editing because the TIBCO Log Table is unchangeable in this adapter version. It has the following buttons and column headings:

- Buttons**
- The Add Table button 
Click to display a dialog box that lists tables available to the database user specified in the adapter Connection tab. Select the table to publish after the data insertion.
 - The Add Details Table button 
Displays a dialog box from which a secondary table can be added to the configuration.
 - The Add Other Table button 

Allows you to enter a schema from the list of available tables. If the schema password is different from the schema name, a dialog will prompt you to enter the password for that schema.

- The Remove Table button 

Deletes the selected table from the list.

- The Re-find Table from Database button 

Refreshes stored table schema information by retrieving new information from the database.

- The Allow Key Columns Only checkbox

A key column or substitute key column is required when publishing by reference, since the subscription table contains only key values. If no column is specified, the publication table will not be added.

Columns

- Tables and Columns

Lists the table and its columns.

- Type

Lists the primitive type.

- AE Type

Lists the primitive type mapped to an ActiveEnterprise type.

- User Key

Defines the column as a user key if it is checked.

- Use?

Enables the column to be updated when a message arrives if it is checked.

- Join To

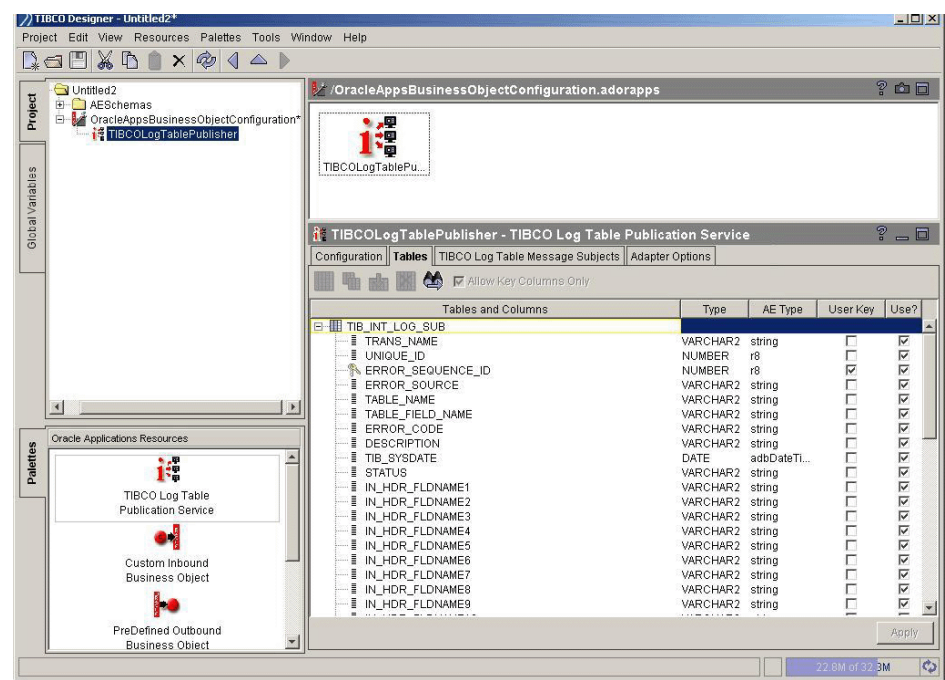
Joins a child table to its parent table.



All tables must have at least one column defined as a Key column. This is required for creating the Materialized View logs in the source tables.

Sample Screen

Figure 113 The TIBCO Log Table Publisher Tables Tab



The TIBCO Log Table Message Subjects Tab

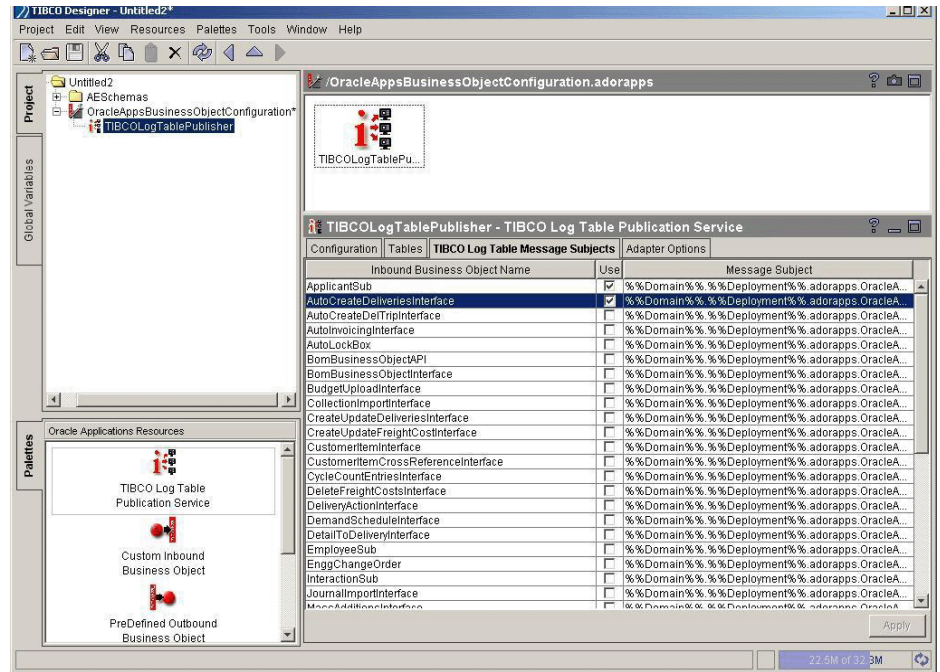
This tab provides a list of the available PreDefined Inbound Business Objects for which the Log Table Publisher can be configured.

You can select a particular transaction if any changes based on that transaction are to be published.

A message subject can only be specified if you check the checkbox that corresponds to that transaction.

The Log Publisher publishes any insertions occurring in the TIB_INT_LOG_SUB table corresponding to the specified transactions on the subject name entered. A default message subject is generated. This message can be modified.

Figure 114 TIBCO Log Table Publisher - the TIBCO Log Table Message Tab



The Adapter Options Tab

The Adapter Options tab has the following options:

The format configuration for both the publisher and subscriber must be the same. This means the publisher and subscriber endpoints should be configured with the same wire format; otherwise an error will occur.

- **Storage Mode**

You cannot alter the value of this field because it is inactive. The default value for this field is `Publish by Value` and cannot be changed.

This mode copies all the specified columns in the source table to the publishing table.

For details on the above restriction, see Oracle SQL Reference manual.



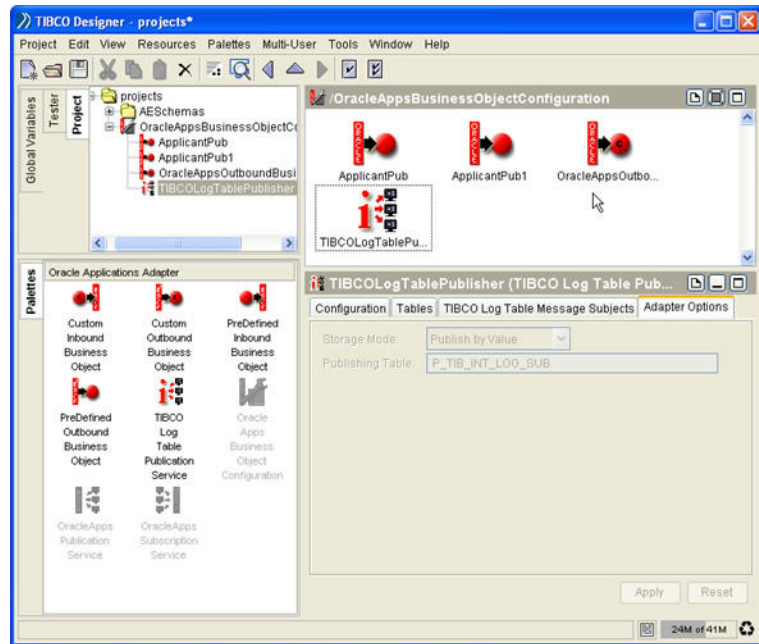
These restrictions do not apply to publishing tables when you publish by reference and when `LONG` or `LONG RAW` are non-key types.

- **Publishing Table**

You cannot alter the value of this field because it is inactive. The value for this field is based on the TIBCO Log Table and cannot be changed. For further details, see *TIBCO ActiveMatrix Adapter for Database User's Guide*.

Sample Screen

Figure 115 The TIBCO Log Table Publisher Adapter Options Tab



Database Components

Deploying a TIBCO Log Table Publisher entails creation of the following database components:

- A Publishing Table named P_TIB_INT_LOG_SUB
- A sequence named P_TIB_INT_LOG_SUB_SEQ on the publishing table
- Indexes named IDX1_P_TIB_INT_LOG_SUB
- IDX2_P_TIB_INT_LOG_SUB and IDX3_P_TIB_INT_LOG_SUB on the publishing table
- A trigger on the publishing table named TRI_P_TIB_INT_LOG_SUB

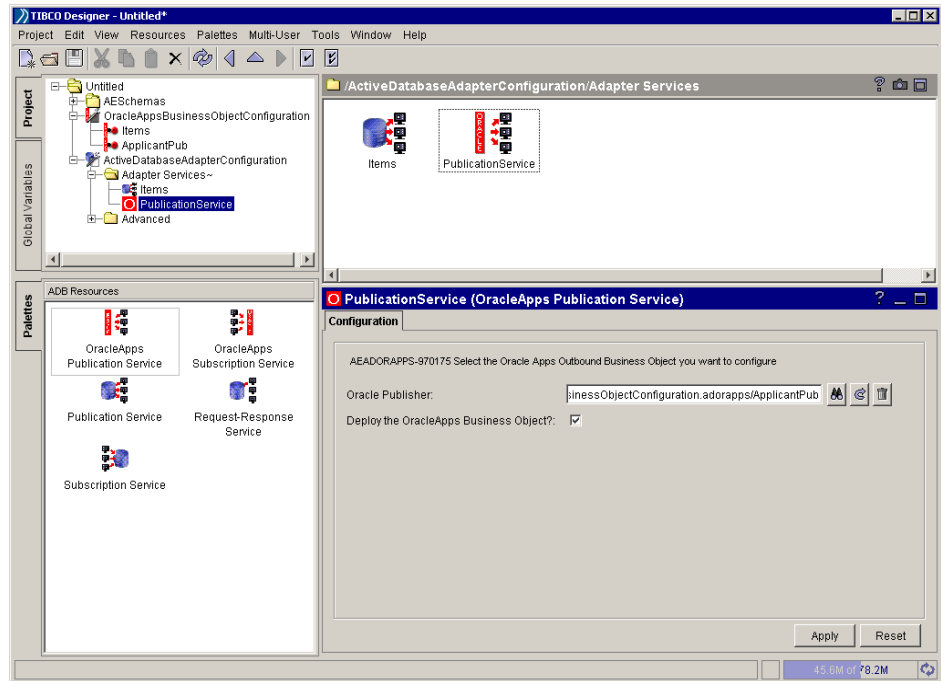
Oracle Publisher Configuration

1. Select the root folder. Drag the **ActiveDatabase Adapter Configuration** button from the palette panel and drop it into the design panel.
2. Specify the adapter instance name and clear the **Write to Database on Save** checkbox in the Configuration tab.
3. Enter the connection parameters in the Design-time Connection tab and click the **Apply** button.
4. In the project panel, expand the node of this resource and double-click the **Adapter Service** folder.
5. Drag the **OracleApps Publication Service** button and drop it into the design panel.

You will see a Configuration tab in the configuration panel.

The Reference field shown on the form can be used to select any available Outbound Business Objects (including TIBCO Log Table Publisher Business Object) configured in the project. Select the desired business object and check the **Deploy Business Object on Database?** checkbox if the associated database objects need to be created. Click the **Apply** button to apply the settings.

Figure 116 Oracle Applications Publisher



The Oracle Publisher service will generate the SQL scripts in the `<ADORAPPS_HOME>/sql` folder. This is required for creating the database objects constituting the corresponding Business Object. These scripts will be deployed if the **Deploy OracleApps Business Object?** checkbox is checked.

This will also include the TIBCO Adapter creation for ActiveDatabase objects.

At this point, the database objects associated with the Business Object are automatically deployed if the **Deploy OraApps Business Object?** checkbox is checked.

The following Oracle database components are associated with a TIBCO Oracle Applications Publisher Service:

- Materialized View Logs (MV Logs) on each source table. This is not a database component for the TIBCO Log Publisher.
- Publishing Table.
- Sequence on the publishing table.
- Indexes on the publishing table.
- Views (if configured). This is not a database component for the TIBCO Log Publisher.
- Triggers on the MV Logs created above (for a TIBCO Log Publisher, the trigger is created directly on the TIBCO Log Table).



An Outbound Business Object once successfully used in creating this service cannot be used for creating another service. Do not select a Business Object that has already been used in creating a Publication service.

A TIBCO ActiveMatrix Adapter for Database Publication service is created with the properties configured in the corresponding Business Object. An input box appears where the name for this service is specified. This name must be unique.

For details on this service, please consult the *TIBCO ActiveMatrix Adapter for Database User's Guide*.

You can configure parameters under the following tabs that are part of a standard TIBCO ActiveMatrix Adapter for Database Publisher service:

- [The Configuration Tab](#)
- [The Tables Tab](#)
- [The Publisher Options Tab](#)
- [The Advanced Tab](#)

The Configuration Tab



For details on configuring properties in each of these tabs, see *TIBCO ActiveMatrix Adapter for Database User's Guide* accompanying this document.

The following parameters are available after a PreDefined Publish Transaction has been selected and applied.

- Name

Specify a unique name for the adapter service. This field must not be modified.

- Transport Type

Select the message transport type the service applies from the drop-down list.

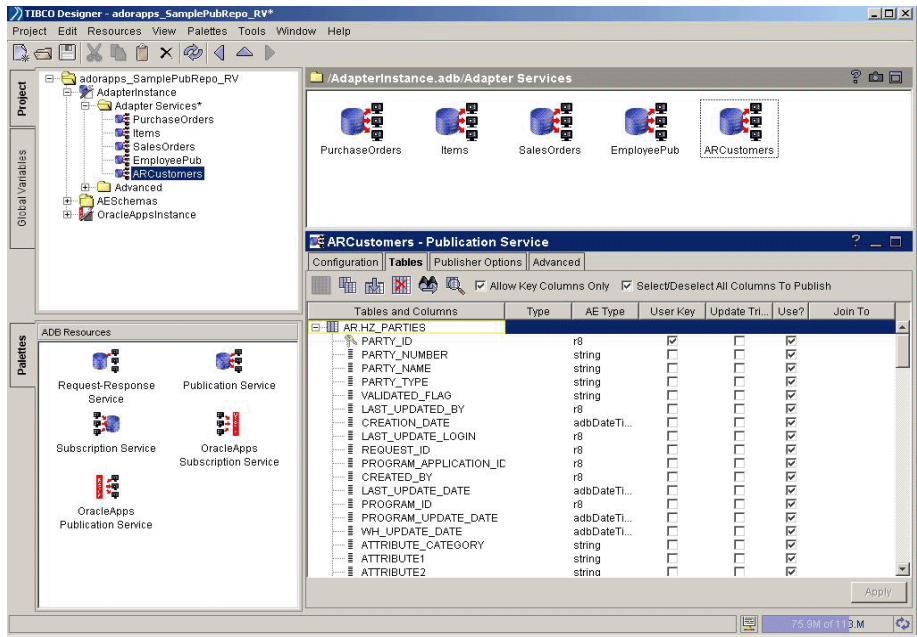
- JMS

Indicates that the service uses the JMS (Java Messaging Service) transportation mode is used by the service. An adapter service that uses the JMS transportation mode can only communicate with the TIBCO Enterprise for JMS server.

- Rendezvous

Indicates that TIBCO Rendezvous is used to transport messages.

Figure 117 Oracle Publisher Configuration- Rendezvous



If you select Rendezvous in the Transport Type field, the following options need to be specified:

- Wire Format

The format of the published messages. The drop-down list will contain different options depending on the Transport Type selected.

One of the following wire formats can be selected for sending messages. Publishers and subscribers can only send and receive data after they agreed on a specific wire format. You can select one of the followings:

— Active Enterprise Message

A descriptive message that is written externally in XML format. This is supported by the TIBCO Adapter SDK. Control information for validation is sent in the message. This format allows ActiveEnterprise components to perform extra validation on messages that are sent or received.

— Rendezvous Message

A self-descriptive message format used by TIBCO Rendezvous applications. Control information for validation is not sent in the message. For Rendezvous wire format, a message which has the type RVMSG_RVMSG

(Rendezvous 5.x) or TIBRVMSG_MSG (Rendezvous 6.x) is sent when this format is chosen. If a publisher sends a simple RVMSG_STRING or TIBRVMSG_STRING, an exception is passed to the subscribing application.

- Quality of Service

The service type for message publications. You can select one of the followings:

- Reliable

Ensures reliable message delivery as long as the physical network is working. This choice is appropriate when some message loss can be tolerated.

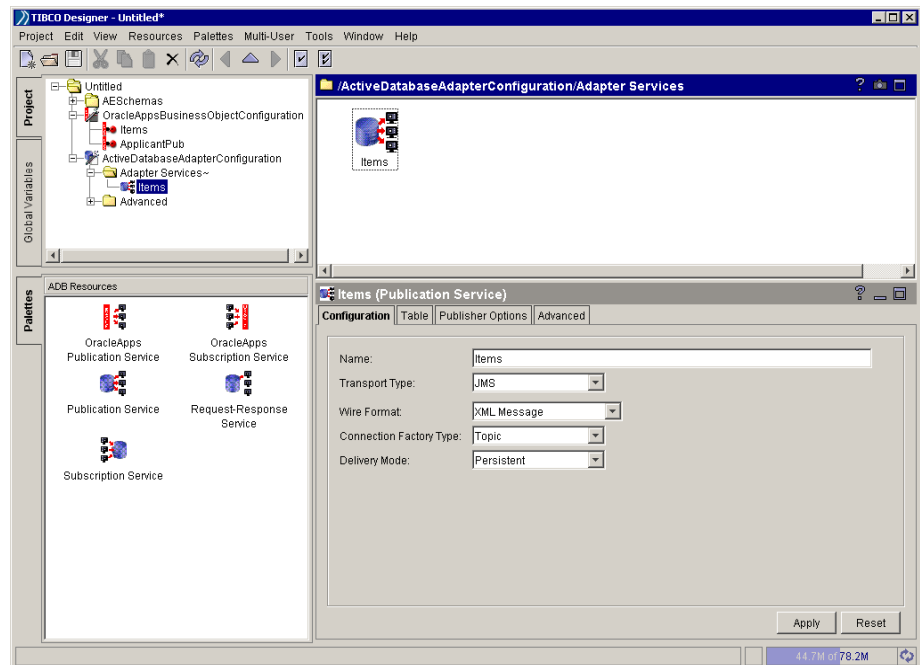
- Certified (default)

This option guarantees that messages will reach their intended recipient despite network failure.



The Transactional option is no longer supported. However, this option is included in the user interface for backward compatibility with version 5.0 and 4.x adapters.

Figure 118 Oracle Publisher Configuration - JMS



If you have selected JMS in the Transport Type field, specify the following options:

- Wire Format

The publication format for the messages. Different options are provided in this drop-down list depending on the selected Transport Type.

Only one wire format is available for the JMS transport type. Publishers and subscribers can only send and receive data if they agree on a specific wire format.

- ActiveEnterprise XML allows you to retrieve data as XML documents and metadata as XML schemas (XSD).

- Delivery Mode

The message delivery mode. There are two options in this drop-down list.

- The Persistent option indicates that the message is available to a JMS client even if the JMS server goes down.
- The Non-persistent option indicates that the message will not be available to a JMS client if the JMS server goes down.

See the *TIBCO Enterprise for JMS User's Guide* for more information.

- Connection Factory Type

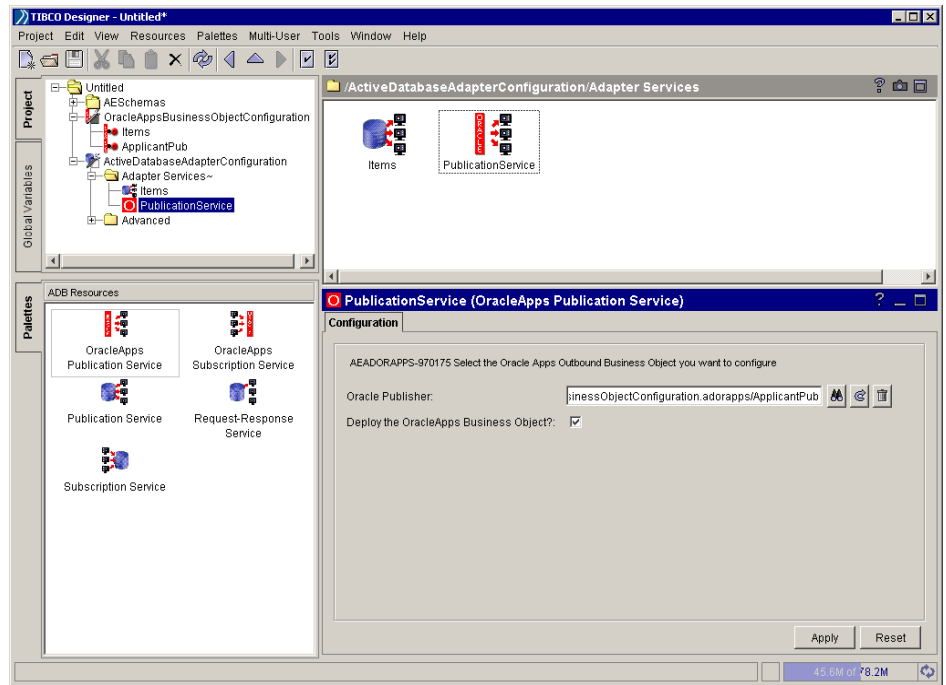
A message can be published to a topic or sent to a queue.

- The Queue option indicates that a message sent to a queue is consumed by one receiver. Each message has only one receiver though multiple receivers can connect to the queue. Only the first receiver that accesses the queue can get the message. This messaging model is known as point-to-point.
- The Topic option indicates that a message published to a topic is broadcast to one or more subscribers. All messages published to the topic are received by all services that have subscribed to the topic. This messaging model is known as publish-subscribe.

For details on JMS concepts refer to the *TIBCO Enterprise for JMS User's Guide*.

Sample Screen

Figure 119 Oracle Publisher Configuration Tab



The Tables Tab

The Tables tab *cannot* be edited because it contains the tables configured in the Business Object. The Tables tab has the following column headings.

Columns

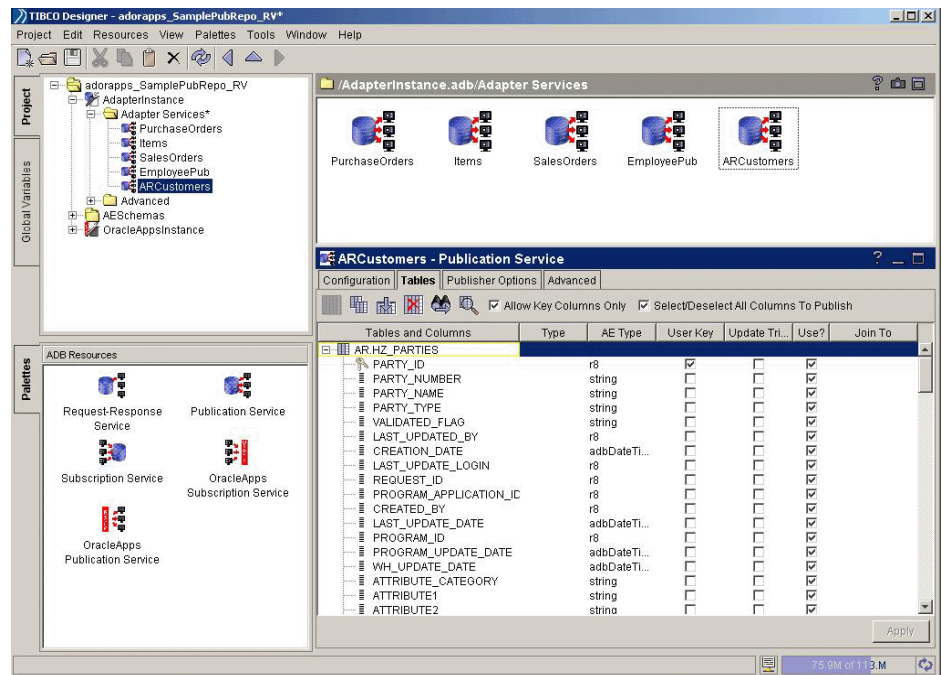
- **Tables and Columns**
Lists the table and its columns.
- **Type**
Lists the primitive type.
- **AE Type**
Lists the primitive type mapped to an ActiveEnterprise type.
- **User Key**
Defines the column as a user key if it is checked.
- **Use?**
Enables the column to be updated when a message arrives if it is checked.
- **Join To**
Joins two tables together.



DO NOT delete any tables because they contain the tables configured in the Business Object.

Sample Screen

Figure 120 The Oracle Publisher Tables Tab



The Publisher Options Tab

The Publisher Options tab has the following options:

- Storage Mode

Do not modify this field. The value can either be set to Publish by Value or Publish by Reference. It must contain the same value as the Storage Mode value in the Oracle Business Object configuration.

Publishing by value copies all specified columns from the source table to the publishing table. Publishing by reference copies only the key column values to the publishing table. If no key column is defined in the database, a substitute non-key column must be set to publish by reference. (See *TIBCO ActiveMatrix Adapter for Database User's Guide*)

- Publishing Table

Do not modify this field. Its value is set in the Business Object.

- Referred Object

Do not modify this field. Its value is set in the Business Object.

- Update Mode

Select the method for updating tables.

- The Update option is used to update a row in the destination table only if the row exists.
- The Upsert option is used to update or insert a row in the destination table if the row exists. This is the default option.

- The Enable Loop Detection Checkbox

Check to enable loop detection and prevent infinite loops when the publishing and destination table refer to the same table.



While configuring a Publication service, the Loop Detection feature is not enabled even if the Enable Loop Detection checkbox is checked in the Publisher Options tab.

- The Do Not Generate Triggers Checkbox

Do not modify this checkbox. It is set in the Business Object.

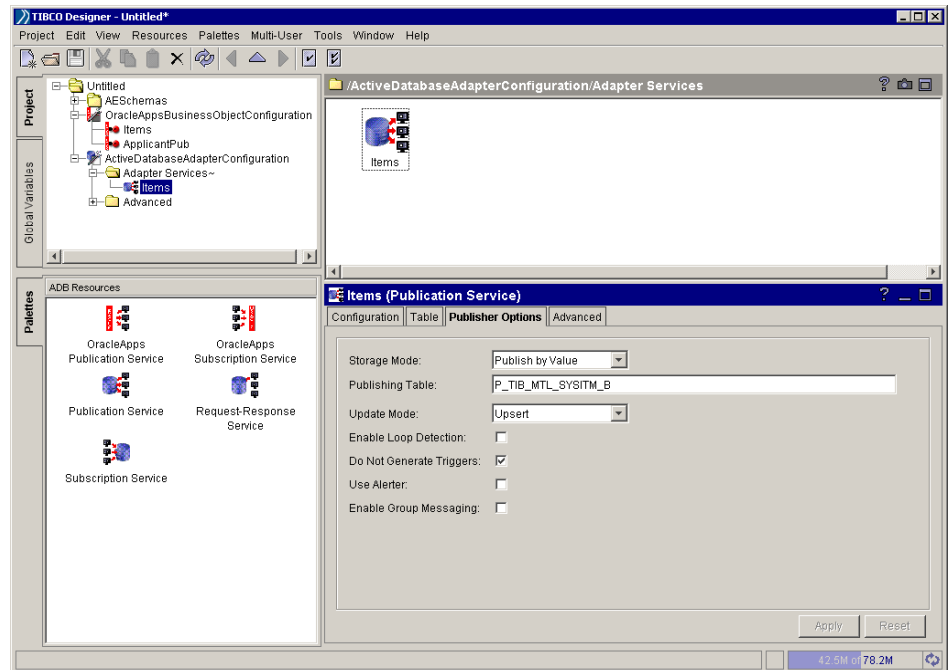
- The Use Alerter Checkbox

The alerter process is used to asynchronously notify running publication service instances of database changes. The adapter does not need to poll its publishing table for the existence of new rows. Use the alerter only when database changes are not frequent.

- The Enable Group Messaging Checkbox
Check to enable the messages to be sent in the package.

Sample Screen

Figure 121 Oracle Publisher - The Publisher Options Tab



The Advanced Tab

The Advanced tab has the following options:

- Message Subject

Do not modify this field. Its value is set in the Business Object.



- If the JMS option has been chosen as the Transport Type in the Configuration tab, then the Advanced tab will be renamed as the *JMS Advanced* tab and you will see a Destination field displayed in the JMS Advanced tab instead of the Message Subject field.
- If you need to override the subject name for a specific adapter configuration, enter the required value in the Message Subject field instead of changing the Global Variables. The required update operation must be carried out manually after you have overridden the subject name manually because the subject name will not generate automatically. If you override the subject name, the endpoint referenced by this service is automatically updated with the modified subject name. You cannot modify the subject at the endpoint level.

- Endpoint Reference

Do not modify this field. Its value is set in the Business Object.

- Class Reference

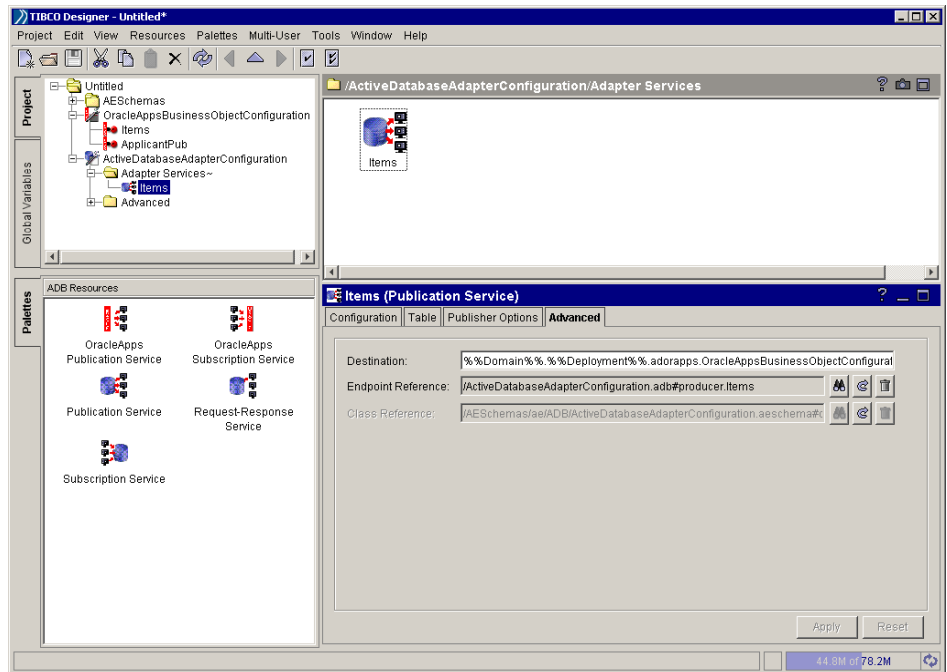
Do not modify this field. Its value is set in the Business Object.



If JMS has been chosen as the Transport Type in the Configuration tab, you will see a Destination field displayed in the JMS Advanced tab instead of the Message Subject field.


Sample Screen

Figure 122 The Oracle Publisher Advanced Tab



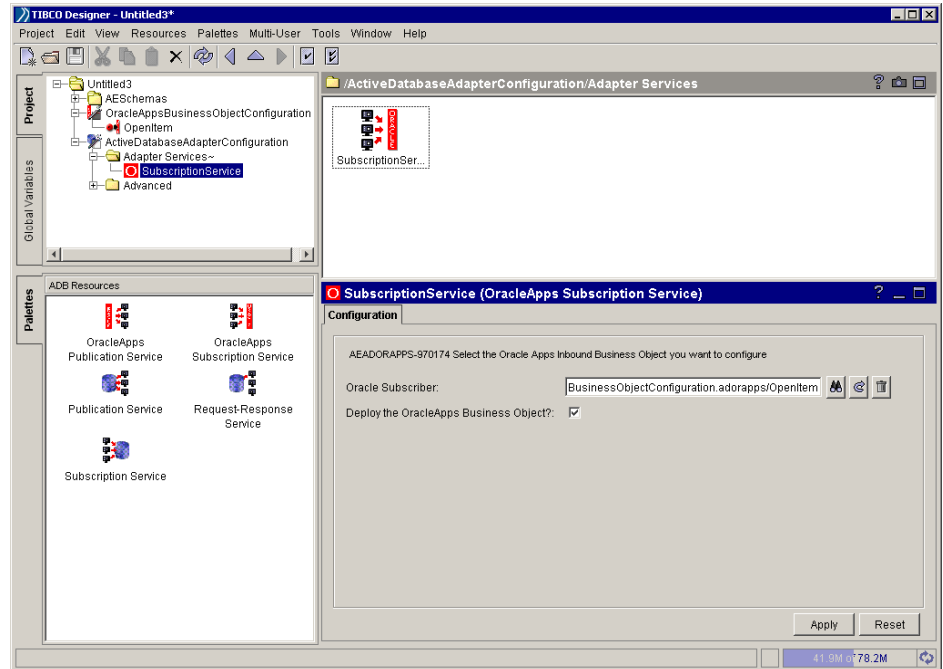
Oracle Subscriber Configuration

1. Drag the **ActiveDatabase Adapter Configuration** button from the palette panel and drop it into the design panel.
2. In the Configuration tab, specify the adapter's Instance Name and clear the **Write to Database on Save** checkbox.
3. Enter the Connection parameters in the Design-time Connection tab and click the **Apply** button.
4. Expand the ActiveDatabase Adapter Configuration folder in the project panel and double-click the **Adapter Service** folder to open it.
5. Drag the **OracleApps Subscription Service** button and drop it into the design panel.

You will see a Configuration tab in the configuration panel. The  button which is at the right hand side of the Oracle Subscriber field can be used to select any object among all the available Inbound Business Objects configured in the project.

Select the desired business object and click the **Apply** button. An input box will display and a unique name for this service must be specified.

Figure 123 Oracle Applications Subscriber



Select the required business object and check the **Deploy Business Object on Database?** checkbox if the associated database objects need to be created. Click the **Apply** button. The Oracle Subscriber service will generate the scripts required for creating the database objects constituting the corresponding Business Object.

The database objects will also be created if the **Deploy Business Object on Database?** option has been checked. This will also include the creation of TIBCO ActiveMatrix Adapter for Database objects.

The following Oracle database components are associated with TIBCO Adapter for Oracle Applications Subscriber Service:

- TIBCO Intermediate Tables
- Indexes on the Intermediate Tables
- Sequence on the Header Intermediate Table
- Synonyms for each Intermediate Table in the <APPSUser>
- PL/SQL Package



An Inbound Business Object used successfully in creating this service cannot be used in creating other services. Do not select a Business Object that has already been used in creating a Subscription service.

A TIBCO ActiveMatrix Adapter for Database Subscriber service is created with the properties configured in the corresponding Business Object. For details on this service, see *TIBCO ActiveMatrix Adapter for Database User's Guide*.

You can set parameters under the following tabs that are part of a standard TIBCO ActiveMatrix Adapter for Database Subscriber service.

You can configure parameters under the following tabs:

- [The Configuration Tab](#)
- [The Tables Tab](#)
- [The Child Table Mappings Tab](#)
- [The Child Exception Table Mappings Tab](#)
- [The Subscriber Options Tab](#)
- [The Advanced Tab](#)



For details on configuring properties in each of these tabs, see the *TIBCO ActiveMatrix Adapter for Database User's Guide*.

The Configuration Tab



The following parameters are available only after a subscribe transaction has been selected and applied.

The Configuration tab has the following options:

- Name

Specify a unique name for the adapter service.

- Transport Type

Select the message transport type used by the service.

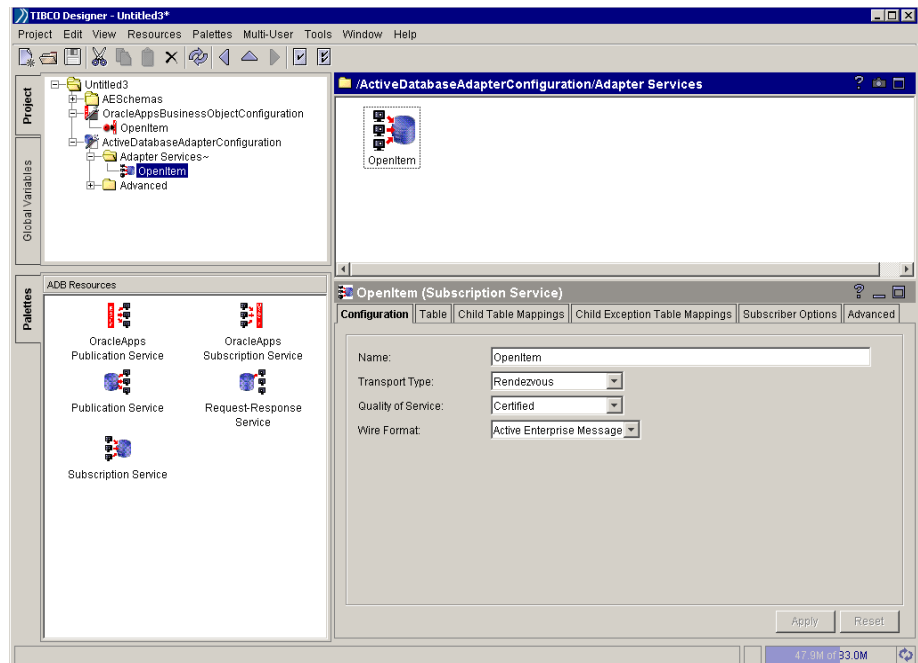
— JMS

Indicates that the service uses JMS (Java Messaging Service) transportation mode. An adapter service that uses the JMS transportation mode can only communicate with the TIBCO Enterprise for JMS server.

— Rendezvous

Indicates that TIBCO Rendezvous is used to transport messages.

Figure 124 Oracle Subscriber Configuration- Rendezvous



If you have selected the Rendezvous option in the Transport Type field, specify the following options:

- Wire Format

The format for the published messages. The drop-down list will contain different options depending on the Transport Type selected.

One of the following wire formats can be selected for sending messages. Publishers and subscribers can only send and receive data after they agree on a specific wire format. You can select one of the following:

— Active Enterprise Message

A descriptive message that is written externally in the XML format. This is supported by the TIBCO Adapter SDK. Control information for validation is sent in the message. This format allows ActiveEnterprise components to perform extra validation on messages that are sent or received.

— Rendezvous Message

A self-descriptive message format used by TIBCO Rendezvous applications. Control information for validation is not sent in the message. For Rendezvous wire format, a message which has the type RVMSG_RVMSG (Rendezvous 5.x) or TIBRVMSG_MSG (Rendezvous 6.x) is sent when this format is chosen. If a publisher sends a simple RVMSG_STRING or TIBRVMSG_STRING, an exception is passed to the subscribing application.



The wire format for both the publisher and subscriber must be the same, otherwise an error will occur.

- Quality of Service

The service type for message Subscription. You can select one of the followings:

- Reliable

Ensures reliable message delivery as long as the physical network is working. This choice is appropriate when some message loss can be tolerated.

- Certified (default)

This option guarantees that messages will reach their intended recipient despite network failure.

- Distributed Queue

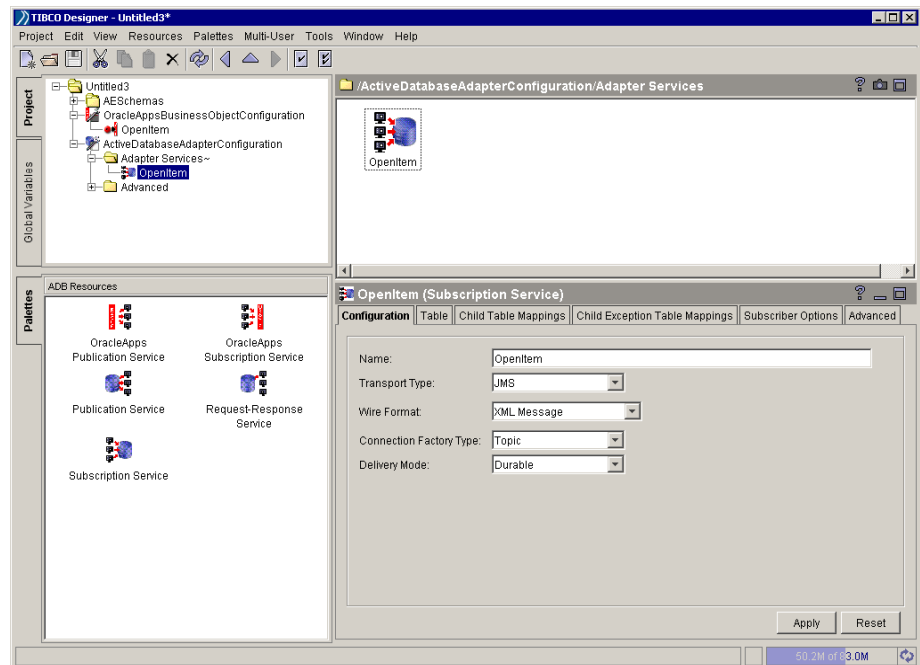
This option is used for load balancing.



The Transactional option is no longer supported. However, this option is included in the user interface for backward compatibility with version 5.0 and 4.x adapters.

See *TIBCO Rendezvous Concepts* for details in quality of service.

Figure 125 Oracle Subscriber Configuration- JMS



If you select the JMS option in the Transport Type field, specify the following options:

- Wire Format

The publication format for the messages. Different options are provided in this drop-down list depending on the selected Transport Type.

Only one wire format is available for the JMS transport type. Publishers and subscribers can only send and receive data if they agree on a specific wire format.

- The XML Message option allows you to retrieve data as XML documents and retrieve metadata as XML schemas (XSD).

- Delivery Mode

An adapter subscription service can be durable or non-durable.

- The Durable option indicates that the service is registered with a JMS server. Messages sent to a durable subscription service are held by the JMS server until they are consumed by the service. The service will receive its messages when it restarts if messages were sent when the service was down.
- The Non-durable option indicates that the service is not registered with the JMS server. Messages sent to a non-durable subscription service are not held by the JMS server. It will not receive the messages from the JMS server if the service was down.

For details, refer to the *TIBCO Enterprise for JMS User's Guide*.

- Connection Factory Type

A message can be published to a topic or sent to a queue.

- The Queue option indicates that a message sent to a queue is consumed by one receiver. Each message has only one receiver though multiple receivers can connect to the queue. Only the first receiver that accesses the queue can get the message. This messaging model is known as point-to-point.
- The Topic option indicates that a message published to a topic is broadcast to one or more subscribers. All messages published to the topic are received by all services that have subscribed to the topic. This messaging model is known as publish-subscribe.

For details on JMS concepts refer to the *TIBCO Enterprise for JMS User's Guide*.





Opaque (Serialized to a single column on the table) messages are stored in a database table and not parsed by the adapter. Messages can be retrieved by a database program that puts the message back into its original format. For more information, refer to the section Message Logging Feature in chapter 7, in *TIBCO ActiveMatrix Adapter for Database User's Guide*.


The Tables Tab


The Tables tab *cannot* be edited because it contains the tables configured in the Business Object. The Tables tab has the following buttons and column headings:

Buttons

- The Add Table button 

Click to display a dialog box that list tables available to the database user specified in the adapter Connection tab. Select the table to be used as the subscription table. It will insert data after receiving it.
- The Add Details Table button 

Displays a dialog box from which a secondary table can be added to the configuration.
- The Add Other Table button 

Allows you to enter a schema from the list of available tables. If the schema password is different from the schema name, a dialog will prompt you to enter the password for that schema.
- The Remove Table button 

Deletes the selected table from the configuration.
- The Allow Key Columns Only Checkbox
- A key column or substitute key column is required when receiving by reference, since the subscription table contains only key values. If no column is specified, the subscription table will not be added.

Columns

- Tables and Columns

Lists the table and its columns.
- Type

Lists the primitive type.
- AE Type

Lists the primitive type mapped to an ActiveEnterprise type.

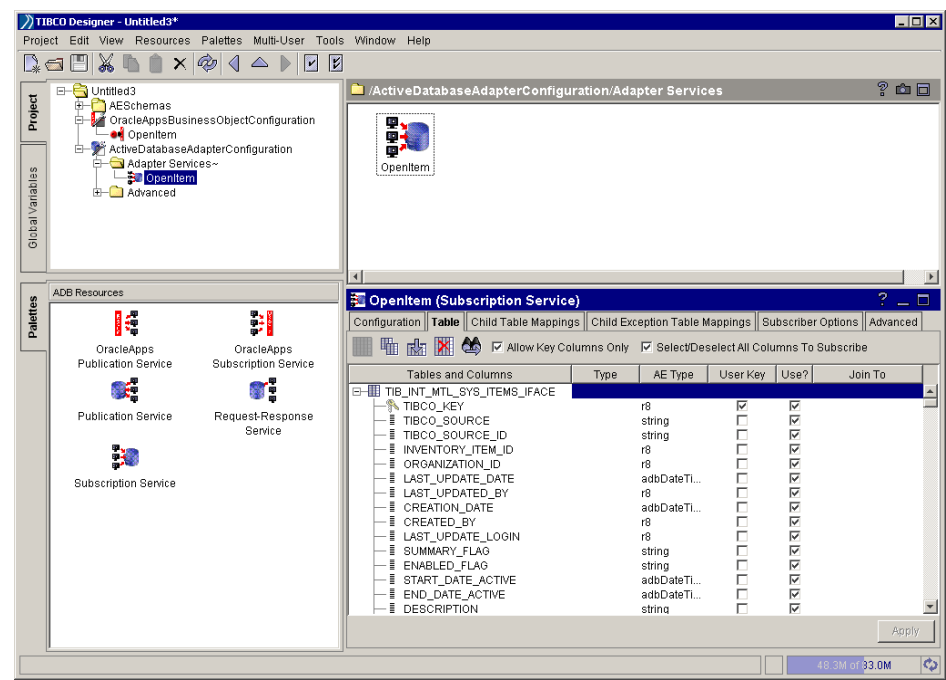
- User Key
Defines the column as a user key if it is checked.
- Use?
Enables the column to be updated when a message arrives if it is checked. Only subscribe columns will be updated. A NULL value is written to columns that are configured to receive but have not received data yet. Columns will retain their default values before such configuration (if any).
- Join To
Joins two tables together.



DO NOT delete any tables because it contains the tables configured in the Business Object.

Sample Screen

Figure 126 The Oracle Subscriber Tables Tab



The Child Table Mappings Tab

The options in this tab are only active when child tables have been specified under the Tables tab.

- Subscriber Child Table Name

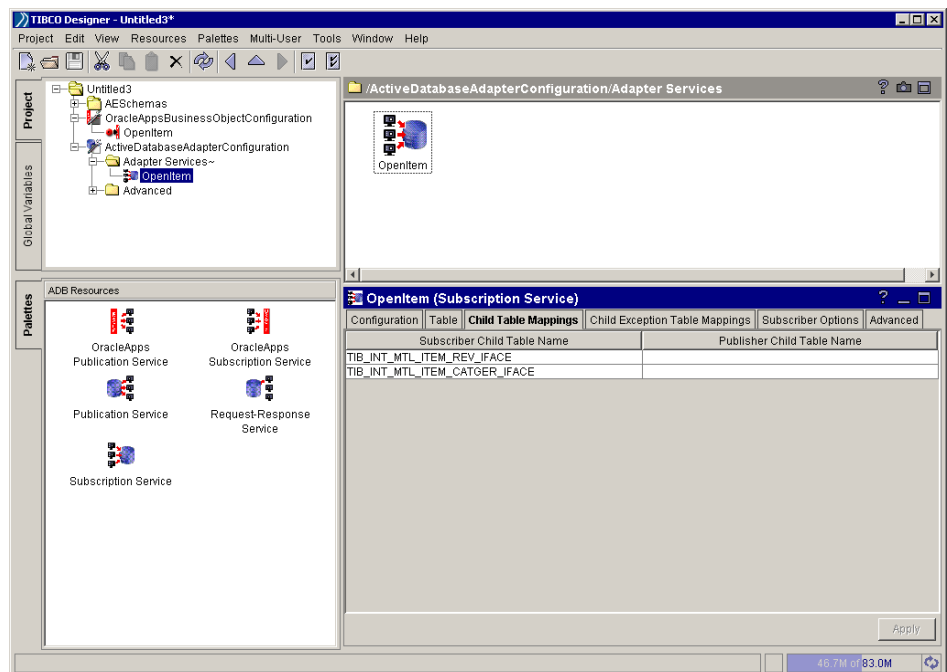
Lists the child tables that can be mapped. Entries are created automatically when you add child tables for subscription service.

- Publisher Child Table Name

Enter the corresponding publisher child table name in the Publisher Child Table Name column for each entry in the Subscriber Table name column. Under the database user account, give a prefix to the table name if your vendor uses this convention.

Sample Screen

Figure 127 The Oracle Subscriber Child Table Mappings Tab



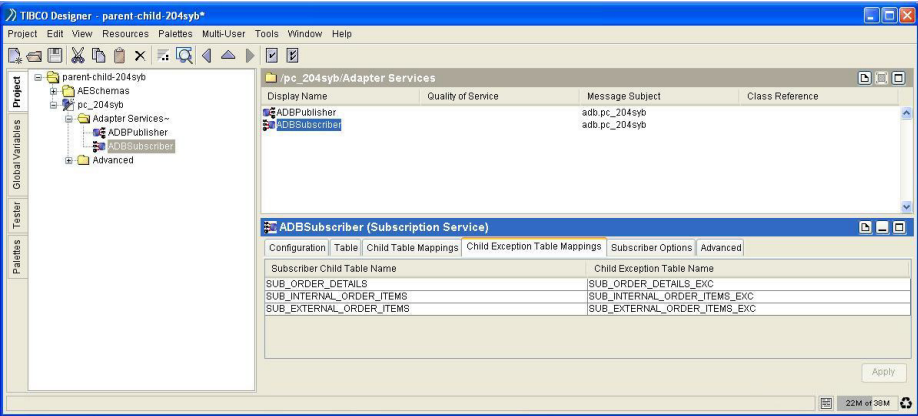
The Child Exception Table Mappings Tab

This tab contains the following columns:

- Subscriber Child Table Name
Lists the child tables that can be mapped. Entries are created automatically when you add child tables for subscription service.
- Child Exception Table Name
Enter the Exception table name for the child table. Under the database user account, give a prefix to the exception table name if your vendor uses this convention.

Sample Screen

Figure 128 The Oracle Subscriber Child Exception Table Mappings Tab



The Subscriber Options Tab

This tab has the following options:

- Exceptions Table

The exception table *cannot* contain any user-created columns which contain the prefix ADB_. This prefix is reserved for the adapter use.

- Pre Commit Stored Procedure

This field must *not* be modified because this value is specified in the Business Object.

- Reply Sender Quality of Service

If the subscriber must send a reply to the sender, this value identifies the quality of service used when sending the reply. Possible values are:

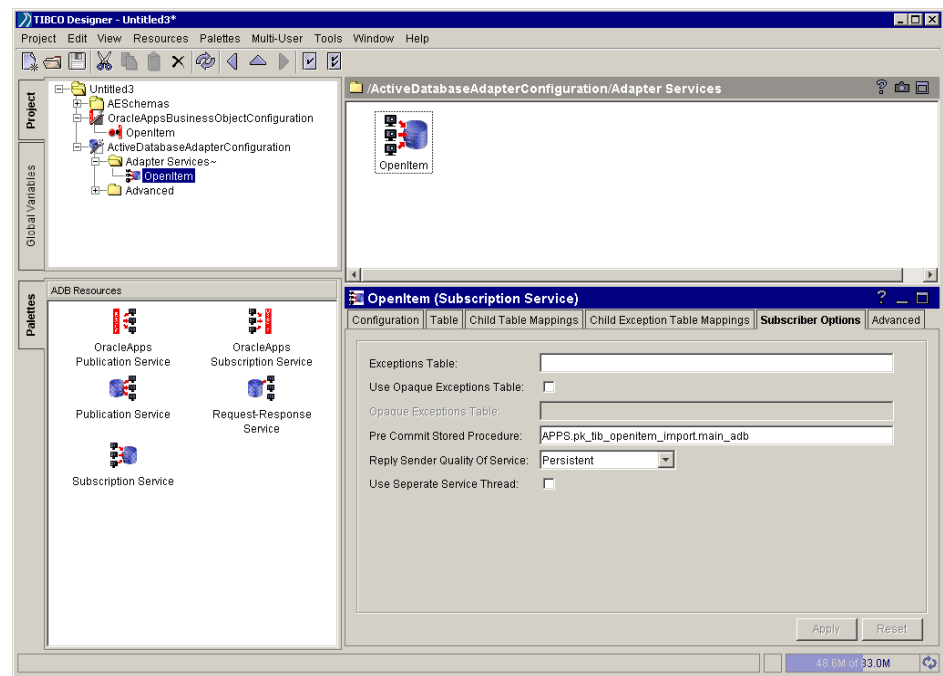
- Reliable

Ensures reliable message delivery as long as the physical network is working. This choice is appropriate when some message loss can be tolerated.

- Certified (default)

This option guarantees that the message will reach its intended recipient despite the network failure.

Figure 129 The Oracle Subscriber -Subscriber Options Tab



The Advanced Tab

The available options in this tab must *not* be modified because they correspond to the values in the corresponding Business Object. See *TIBCO ActiveMatrix Adapter for Database User's Guide* for further details.

The Advanced tab has the following options:



- Message Subject (For TIBCO Rendezvous transport type only)
By default, a service uses a message subject which is generated using the Domain and Deployment global variables, the adapter acronym, the adapter instance name, and the service name. If you use this default subject, make sure that the values for the Domain and Deployment variables are not empty. You can type a TIBCO Rendezvous subject name that is different from the default in this field. See *TIBCO Rendezvous Concepts* for information about specifying subject names.
- Destination (For JMS transport type only)
The subscriber destination. A service uses a default destination which is generated using the Domain and Deployment global variables, the adapter

acronym, the adapter instance name and the service name. If you use this default destination, make sure that the values for the `Domain` and `Deployment` variables are not empty.


Alternatively, you can manually enter a destination in this field. The destination does not have to be predefined in the TIBCO Enterprise Message Service server. It can be static or dynamic.

See the *TIBCO Enterprise Message Service User's Guide* for more information about destinations.

- Endpoint Reference

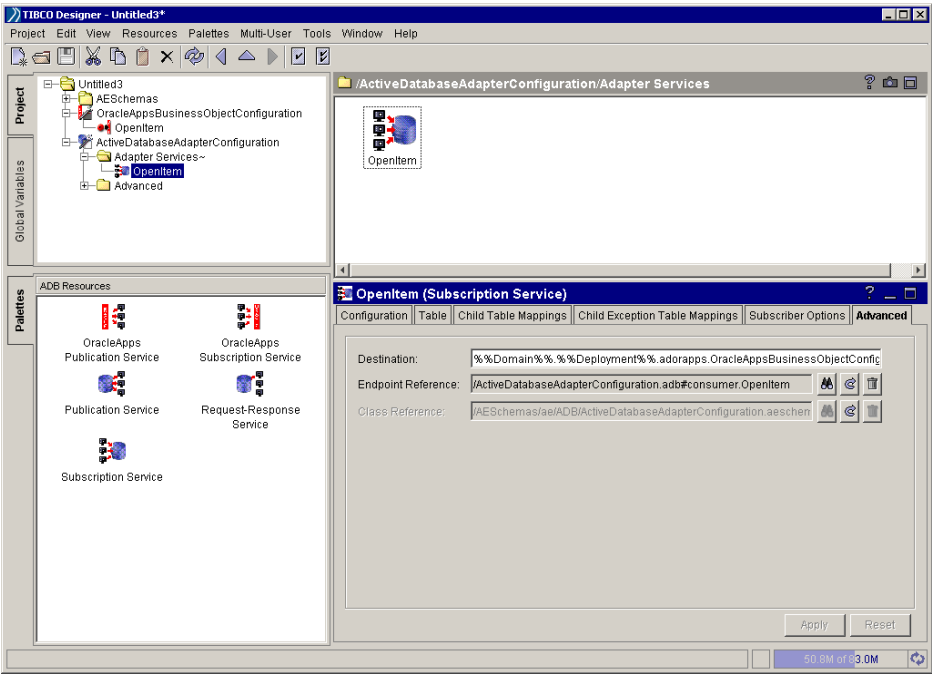
Displays the endpoint reference. Click the  button to change the endpoint reference or the  button to reconfigure the existing reference. You can also click the  button to remove the reference. Endpoint reference objects are explained in detail in *TIBCO Designer Palette Reference*.

- Class Reference

Click the  button to reconfigure the existing reference. Class reference objects are explained in detail in *TIBCO Designer Palette Reference*.

Sample Screen

Figure 130 The Oracle Subscriber Advanced Tab



Deleting a Publisher or a Subscriber

Publisher service

- The deletion of a Publisher service involves two resources - the Service created from the Business Object and the Business Object itself.
- If a Business Object has been successfully used in creating and deploying an Oracle Publisher service, the database objects for that Business Object is removed from the database upon its deletion from the project.

Subscription Service

- The deletion of a Subscriber service involves two resources - the Subscriber Service created from the Business Object and the Business Object itself.
- If a Business Object has been successfully used in creating and deploying an Oracle Subscriber service, the database objects for that Business Object is removed from the database upon its deletion from the project.



Specify the Oracle Applications database user ID and password for each Business Object that is configured.

If a Business Object has been successfully used in creating and deploying an Oracle Subscriber service, the corresponding objects created in the database are deleted when the Business Object is removed from the project. It is therefore recommended that a Business Object only be removed after deleting the corresponding Publication/Subscription Service.

Testing the Adapter

You can test the adapter by starting the adapter and checking for the correct messages. If your preparations and configuration are successful, the adapter will connect to TIBCO Adapter for Oracle Applications and the messages will show no errors.

Customizing a Business Object

The sections below refer to the scenarios when a Business Object needs to be customized after it has been associated with a service (used for configuring a service).

PreDefined Publisher Business Object

Modifying properties in the Views tab.

For each view, join tables can be added or removed. If the Business Object has been deployed on the database, any changes made here are saved to the database after clicking the **Apply** button. If the Business Object has not been deployed, only the scripts are generated in the `sql` folder.



Do not run the run-time agent on the project during the project configuration. The database objects are created during the configuration time, but the schema metadata is not available to the run-time agent.

Custom Publisher Business Object

Modifying properties in the Tables, Adapter Options, and Table and View Mappings tabs.

For each view, join tables can be added or removed. If the Business Object has been deployed on the database, any changes made here are saved to the database when you click the **Apply** button. If the Business Object has not been deployed, only the scripts are generated in the `sql` folder.



Do not run the runtime agent on the project during the project configuration. The database objects are created during the configuration time, but the schema metadata is not available to the runtime agent.

The user can modify properties in the Tables, Adapter Options, and Table and View Mappings tabs depending on whether the Business Object has been deployed on the database or not:

Case #1: The Business Object was *not* deployed on the database.

- The Tables tab
 - Child tables can be added or removed. If any property in the main table needs to be changed (including the main tables), delete the corresponding service (created using the Business Object), click the **Yes** button when prompted with the deleting the schemas option, then modify the Business

Object. After applying all the other changes, drag the **OracleApps Publication Service** button from the palette panel and drop it into the design panel and select this Business Object. The publisher service is created with changes.

- The Adapter Options tab
 - The Storage Mode *cannot* be changed in this tab. If this needs to be changed, delete the corresponding adapter service created using the Business Object. Click the **No** button when prompted with the deleting the schemas option. Modify the Business Object that is changing the Storage Mode, and create a new object using the Oracle Publisher.
 - The Publishing Table *cannot* be changed in this tab. If the publishing table needs to be changed, then delete the corresponding service that was created using the Business Object. Click the **Yes** button when prompted with the deleting the schemas option, and then make modifications to the Business Object. After applying all the other changes, drag the **OracleApps Publication Service** button from the palette panel and drop it into the design panel, then select this Business Object. The publisher service is created with changes.
- The Table and View Mappings tab
 - The child view names can be changed.
 - The main view name *cannot* be changed in this tab. If this needs to be changed, delete the corresponding adapter service that was created using the Business Object. Click the **No** button when prompted with the deleting the schemas option. Modify the Business Object (changing the Storage Mode) and create a new one using the Oracle Publisher.

Case #2: The Business Object was deployed on the database

- The Tables tab
 - Do *not* modify any properties in this tab.
- The Adapter Options tab
 - Do *not* modify any properties in this tab.
- The Table and View Mappings tab
 - Do *not* modify any properties in this tab.

PreDefined Subscriber Business Object

You can modify properties in the following tabs depending on whether the Business Object has been deployed on the database or not.

Case #1: The Business Object was *not* deployed on database

- The Configuration tab
 - All properties in this tab can be modified. If the Business Object name needs to be changed, delete the corresponding adapter service created using the Business Object. Click the **No** button when prompted with the deleting the schemas option, modify the Business Object and associate the Business Object with a new subscriber service (by dragging the **OracleApps Subscription Service** button from the palette panel and dropping it into the design panel).
- The Tables tab
 - The table column property Use? can be set to true (default) or false. The classes are appropriately regenerated without affecting any functionalities. In this case, you can select the columns you need to subscribe to.



Do *not* uncheck the Use Property on a Key column.

- The Subscriber Options tab
 - The value for Reply Sender Quality of Service can be modified
- The Advanced tab
 - The value for Message Subject can be modified.

Case #2: The Business Object was deployed on database

- The Configuration tab
 - To change a property, you need first to delete the corresponding adapter service created using the Business Object. When prompted with the deleting the schemas option, click the **No** button, modify the Business Object, and then associate the Business Object with a new subscriber service (by dragging the **OracleApps Subscription Service** button from the palette panel and dropping it into the design panel).
- The Tables tab
 - The table column property Use? can be set to true (default) or false. The classes are appropriately regenerated without affecting any functionalities.



Do *not* uncheck the Use Property on a Key column.

- The Subscriber Options tab
 - Properties in this tab cannot be modified.
- The Advanced tab
 - Properties in this tab cannot be modified.

Custom Subscriber Business Object

The user can modify properties in the following tabs depending on whether the Business Object has been deployed in the database or not:

CASE #1: The Business Object was *not* deployed in the database

- The Configuration tab
 - Any property in this tab can be modified.
 - To change the Business Object name, you need first to delete the corresponding adapter service created using the Business Object. When prompted with the deleting the schemas option, click the **No** button, modify the Business Object, and then associate the Business Object with a new Subscriber service (by dragging the **OracleApps Subscription Service** button from the palette panel and dropping it into the design panel).
- The Tables tab
 - Tables can be added or removed from the Business Object. To make changes in this tab, you need first to delete the corresponding adapter service created using the Business Object. When prompted with the deleting the schemas option, click the **Yes** button, modify the Business Object, and then associate the Business Object with a new Subscriber service (by dragging the **OracleApps Subscription Service** button from the palette panel and dropping it into the design panel).
- The Subscriber Options tab
 - All properties in this tab can be modified. To make changes in this tab, you need first to delete the corresponding adapter service created using the Business Object. When prompted with the deleting the schemas option, click the **Yes** button, modify the Business Object, and then associate the Business Object with a new Subscriber service (by dragging the **OracleApps Subscription Service** button from the palette panel and dropping it into the design panel).

- The Advanced tab
 - Message Subject

This property can be modified. To make changes in this tab, you need first to delete the corresponding adapter service created using the Business Object, click the **Yes** button when prompted with the deleting the schemas option, modify the Business Object and associate the Business Object with a new Subscriber service (by dragging the **OracleApps Subscription Service** button from the palette panel and dropping it into the design panel).

- Class Reference

This field is used for display purposes only and must *not* be modified.

CASE #2: The Business Object was deployed in the database

- The Configuration tab
 - To change a property in this tab, you need first to delete the corresponding adapter service created using the Business Object. When prompted with the deleting the schemas option, click the **No** button, modify the Business Object, and then associate the Business Object with a new Subscriber service (by dragging the **OracleApps Subscription** button from the palette panel and dropping it into the design panel).
- The Tables tab
 - The table column property Use? can be set to true (default) or false. The classes are appropriately regenerated without affecting any functionalities. In this case, you can select the columns you need to subscribe to.



Do *not* uncheck the Use Property on a Key column. Tables must *not* be added or removed from this tab.

- The Subscriber Options tab
 - Properties in this tab cannot be modified.
- The Advanced tab
 - Properties in this tab cannot be modified.

TIBCO Log Table Publisher

You can modify properties in the following tabs depending on whether the Log Publisher Business Object has been deployed on the database or not.

CASE #1: The Business Object was *not* deployed in the database

- The Configuration tab
 - Any property in this tab can be modified.
 - To change the Business Object name, you need first to delete the corresponding adapter service created using the Business Object. When prompted with the deleting the schemas option, click the **No** button, modify the Business Object, and then associate the Business Object with a new Publication service (by dragging the **OracleApps Publication Service** button from the palette panel and dropping it into the design panel).
- The Tables tab
 - Properties in this tab cannot be modified.
- The TIBCO Log Table Message Subjects tab
 - The required PreDefined Inbound Business Objects can be selected and corresponding message subjects (on which data from the TIBCO Log Table has to be published) can be specified. Once the properties have been applied by clicking the **Apply** button, the changes are written to the following file:


```
<TIBCO_HOME>/adapter/adorapps/<version_num>/sql/<AdapterInstance>_<BusinessObjectName>_all.sql.
```
- The Advanced tab
 - Message Subject
 - This property must *not* be modified.
 - Class Reference
 - This field is used for display purpose only and must *not* be modified.

CASE #2: The Business Object was deployed on database

- The Configuration tab
 - Properties in this tab cannot be modified.
- The Tables tab
 - Properties in this tab cannot be modified.

- The TIBCO Log Table Message Subjects tab
 - The required PreDefined Inbound Business Objects can be selected and corresponding message subjects (on which data from the TIBCO Log Table has to be published) can be specified. Once the properties have been applied by clicking the **Apply** button, the changes are written to the following file:


```
<TIBCO_HOME>/adapter/adorapps/5.3/sql/<AdapterInstance>_<BusinessObjectName>_all.sql.
```

Changes made in this tab are reflected in the trigger on the TIBCO Log Table. Note that only the trigger on the TIBCO Log Table is modified - the publishing table and other database components are not altered.
- The Advanced tab
 - Message Subject

Properties in this tab cannot be modified.



Specify the Oracle Applications database user ID and password for each Business Object that is configured if they are available for configuration.

Schemas (classes, associations, sequences, and so on) are shared between a Business Object and the corresponding service. Therefore, deleting a Business Object or the corresponding service will remove the schemas involved with both. Keep this in mind when performing the delete operation on a Business Object or Service.

If a Business Object has been successfully used in creating and deploying an Oracle Subscriber service, the corresponding objects created in the database are deleted when the Business Object is removed from the project.

Conditional Publishing

This section explains how to publish data based on specific business conditions. A publishing service can be customized to publish data only when certain fields in the Oracle Application source tables (the tables configured in a Business Object) are updated to have the required values.

The Purchase Orders Outbound Business Object is used here as an example. In this example, the field named `AUTHORIZATION_STATUS` in the table `PO_HEADERS_ALL` can have the value of `Authorized` or `Pending`.

The adapter configured to run on this business object is required to publish data only when the value of the `AUTHORIZATION_STATUS` field changes. Otherwise, the insert and delete operations on this table are published in the normal way.

This can be achieved by the following:

1. After configuring the project in TIBCO Designer (using the TIBCO Adapter for Oracle Applications palette), alter the materialized view log on the table `PO_HEADERS_ALL` to include the `AUTHORIZATION_STATUS` field. This can be done by executing the SQL -> `ALTER TABLE MLOG$PO_HEADERS_ALL ADD(AUTHORIZATION_STATUS VARCHAR2(25));` query in the `<PO_User>` within the Oracle Applications Database.
2. Alter the publishing table (for example, `P_PO_HEADERS_ALL`) to include the `AUTHORIZATION_STATUS` field by executing the SQL -> `ALTER TABLE P_PO_HEADERS_ALL ADD AUTHORIZATION_STATUS VARCHAR2(25);` query.
3. Modify the trigger created on the MV Log of the header table. The SQL script `<InstanceName>_<BusinessObjectName>.all.sql` containing the trigger is generated by the Oracle Applications palette and placed in the `<TIBCO_HOME>/adapter/adorapps/5.3/sql` folder.

You need to add the following to the trigger body:

```
CREATE OR REPLACE TRIGGER <TriggerName> AFTER INSERT ON
PO.MLOG$_PO_HEADERS_ALL
FOR EACH ROW
DECLARE
    W_COUNT NUMBER(10);
    W_COUNT1 NUMBER(10);
    W_COUNT2 NUMBER(10);
    W_COUNT3 NUMBER(10);
    W_AUTHORIZATION_STATUS VARCHAR2(25);
BEGIN
    IF INSERTING THEN
        IF :NEW.DMLTYPE$$ = 'I' AND :NEW.OLD_NEW$$ = 'N' THEN
            SELECT COUNT(ADB_SEQUENCE) INTO W_COUNT
            FROM P_PO_HEADERS_ALL
            WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
            AND ADB_OPCODE = 1
            AND ADB_L_DELIVERY_STATUS = 'N'
            AND ROWNUM = 1;
            IF (W_COUNT = 0) THEN
                INSERT INTO P_PO_HEADERS_ALL
                (
                    PO_HEADER_ID,
                    AUTHORIZATION_STATUS,
                    ADB_SUBJECT,
                    ADB_SEQUENCE,
                    ADB_SET_SEQUENCE,
                    ADB_TIMESTAMP,
                    ADB_OPCODE,
                    ADB_UPDATE_ALL,
                    ADB_REF_OBJECT,
                    ADB_L_DELIVERY_STATUS,
                    ADB_L_CMSEQUENCE
                )
                VALUES (
                    :NEW.PO_HEADER_ID,
                    :NEW.AUTHORIZATION_STATUS,
                    NULL,
                    P_PO_HEADERS_ALL_SEQ.NEXTVAL,
                    0,
```



```

        SYSDATE,
        1,
        NULL,
        'TIB_PUB_PURORDER_HDR_V',
        'N',
        -1 );
    END IF;
END IF;
IF :NEW.DMLTYPE$$ = 'U' AND :NEW.OLD_NEW$$ = 'U' THEN
    INSERT INTO P_PO_HEADERS_ALL
    (
    PO_HEADER_ID,
    AUTHORIZATION_STATUS,
    ADB_SUBJECT,
    ADB_SEQUENCE,
    ADB_SET_SEQUENCE,
    ADB_TIMESTAMP,
    ADB_OPCODE,
    ADB_UPDATE_ALL,
    ADB_REF_OBJECT,
    ADB_L_DELIVERY_STATUS,
    ADB_L_CMSEQUENCE
    )
VALUES (
    :NEW.PO_HEADER_ID,
    :NEW.AUTHORIZATION_STATUS,
    NULL,
    P_PO_HEADERS_ALL_SEQ.NEXTVAL,
    0,
    SYSDATE,
    2,
    NULL,
    'TIB_PUB_PURORDER_HDR_V',
    'T',
    -1 );
END IF;

IF :NEW.DMLTYPE$$ = 'U' AND :NEW.OLD_NEW$$ = 'N' THEN
    SELECT AUTHORIZATION_STATUS INTO W_AUTHORIZATION_STATUS
    FROM P_PO_HEADERS_ALL

```

```

WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
AND ADB_OPCODE = 2
AND ADB_L_DELIVERY_STATUS = 'T';

IF (:NEW.AUTHORIZATION_STATUS <> W_AUTHORIZATION_STATUS)
THEN
  UPDATE P_PO_HEADERS_ALL
  SET ADB_L_DELIVERY_STATUS = 'N',
      AUTHORIZATION_STATUS = :NEW.AUTHORIZATION_STATUS
  WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
  AND ADB_OPCODE = 2
  AND ADB_L_DELIVERY_STATUS = 'T';
ELSE
  DELETE FROM P_PO_HEADERS_ALL
  WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
  AND ADB_OPCODE = 2
  AND ADB_L_DELIVERY_STATUS = 'T';
END IF;
END IF;

IF :NEW.DMLTYPE$$ = 'D' AND :NEW.OLD_NEW$$ = 'O' THEN
  SELECT COUNT(ADB_SEQUENCE) INTO W_COUNT2
  FROM P_PO_HEADERS_ALL
  WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
  AND (ADB_OPCODE = 3)
  AND ADB_L_DELIVERY_STATUS = 'N'
  AND ROWNUM = 1;
  IF (W_COUNT2 = 0) THEN
  SELECT COUNT(ADB_SEQUENCE) INTO W_COUNT3
  FROM P_PO_HEADERS_ALL
  WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
  AND ADB_OPCODE = 2
  AND ADB_L_DELIVERY_STATUS = 'N'
  AND ROWNUM = 1;
  IF (W_COUNT3 > 0) THEN
    DELETE FROM P_PO_HEADERS_ALL
    WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
    AND ADB_OPCODE = 2
    AND ADB_L_DELIVERY_STATUS = 'N';
  END IF;
END IF;

```

```

INSERT INTO P_PO_HEADERS_ALL
(
  PO_HEADER_ID,
  AUTHORIZATION_STATUS,
  ADB_SUBJECT,
  ADB_SEQUENCE,
  ADB_SET_SEQUENCE,
  ADB_TIMESTAMP,
  ADB_OPCODE,
  ADB_UPDATE_ALL,
  ADB_REF_OBJECT,
  ADB_L_DELIVERY_STATUS,
  ADB_L_CMSEQUENCE
)
VALUES (
  :NEW.PO_HEADER_ID,
  :NEW.AUTHORIZATION_STATUS,
  NULL,
  P_PO_HEADERS_ALL_SEQ.NEXTVAL,
  0,
  SYSDATE,
  3,
  NULL,
  'TIB_PUB_PURORDER_HDR_V',
  'N',
  -1 );
END IF;
END IF;
END IF;
END <TriggerName>;"

```

Observe the following code fragment used in the trigger above:

```

"IF :NEW.DMLTYPE$$ = 'U' AND :NEW.OLD_NEW$$ = 'N' THEN
  SELECT AUTHORIZATION_STATUS INTO W_AUTHORIZATION_STATUS
  FROM P_PO_HEADERS_ALL
  WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
  AND ADB_OPCODE = 2
  AND ADB_L_DELIVERY_STATUS = 'T';

IF (:NEW.AUTHORIZATION_STATUS <> W_AUTHORIZATION_STATUS)

```

```

THEN
    UPDATE P_PO_HEADERS_ALL
    SET ADB_L_DELIVERY_STATUS = 'N',
        AUTHORIZATION_STATUS = :NEW.AUTHORIZATION_STATUS
    WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
    AND ADB_OPCODE = 2
    AND ADB_L_DELIVERY_STATUS = 'T';
ELSE
    DELETE FROM P_PO_HEADERS_ALL
    WHERE PO_HEADER_ID = :NEW.PO_HEADER_ID
    AND ADB_OPCODE = 2
    AND ADB_L_DELIVERY_STATUS = 'T';
END IF;
END IF;"

```

Whenever records in the PO_HEADERS_ALL table are updated, the updated data is published only after the value of AUTHORIZATION_STATUS is changed (from Pending to Authorized, or vice versa).

In this manner, data can be configured to be published based on specific conditions that represent the required business functionalities.

This customization can be done for both PreDefined Business Objects and Custom Oracle Applications Outbound Business Objects.

Impact of Edit Actions on Database Objects

This section explains the impact of various edit actions on the database objects in TIBCO Adapter for Oracle Applications:

1. Copy or paste a Business Object Instance resource:
 - The instance retains all its properties.
 - The associated schemas (if any) are copied. All the business objects contained within the instance are copied.
 - The associated database objects are not created regardless of whether the Business Object was or was not deployed.
2. Copy or paste an Outbound or Inbound Business Object resource:
 - The business object retains all its properties.
 - The associated schemas (if any) are copied. All the business objects contained within the instance are copied.
 - The associated database objects are not created regardless of whether the Business Object was or was not deployed.
3. Cut or paste a Business Object Instance resource:
 - The instance retains all its properties.
 - After the **Cut** operation, all the business objects contained within the instance are deleted and created at the new location after the **Paste** operation. For each business object, the associated database objects are deleted if the business object had been deployed on the database.
 - If you do not delete the schemas associated with each business object, the schemas will be left intact and will be copied under the new instance schema VFile after the **Paste** operation.
 - When pasting the business object, the associated database objects are not created in the Oracle Applications Database regardless of whether the Business Object was or was not deployed.
4. Cut or paste an Outbound or Inbound Business Object resource:
 - The business object retains all its properties.
 - After the **Cut** operation, the business object is deleted from the project and created at the new location after the **Paste** operation. For each business

object, the associated database objects is deleted if the business object had been deployed on the database.

- If you do not delete the schemas associated with the business object, the schemas will be left intact and will be copied under the new instance schema VFile after the **Paste** operation.
 - When pasting the business object, the associated database objects are not created in the Oracle Applications Database regardless of whether the Business Object was or was not deployed.
5. To undo operations on a Business Object Instance or an Outbound or Inbound Business Object:
- The database objects cannot be restored to their original state after an **Undo** operation on any of the operations listed below are performed on a Business Object Instance or an Outbound or Inbound Business Object:
 - Create
 - Delete
 - Copy and Paste
 - Cut and Paste
 - If any of these operations result in the generation of SQL scripts, these scripts can then be executed on the database for restoring the Business Object-related database objects.

Setting Encoding Options

Refer to Chapter 9 Setting Encoding Options in the *TIBCO ActiveMatrix Adapter for Database Users Guide*.

Chapter 5

Deploying and Starting an Adapter Using TIBCO Administrator



This chapter provides an overview about deploying, starting, stopping, and monitoring adapter services using the TIBCO Administrator web interface.

Topics

- [Create an EAR File in TIBCO Designer, page 248](#)
- [Deploy the Project, page 249](#)
- [Start or Stop the Adapter, page 252](#)
- [Monitor the Adapter, page 253](#)

Create an EAR File in TIBCO Designer

Generate an Enterprise Archive file (EAR) that contains information about the adapter services to deploy.

The EAR file contains information on what you wish to deploy. This could be one or more adapter services, one or more TIBCO BusinessWorks process engines, or both.



Building an archive creates the EAR file, which you can then deploy from TIBCO Administrator. If you make changes to the business processes or adapter services included in the archive, you need to rebuild the archive. Saving the project does not affect the archive.

In TIBCO Designer, follow these steps to create an EAR:

1. Configure the adapter services.
2. Drag and drop the **Enterprise Archive** resource from the palette panel to the design panel. If there are any configured adapter services in your project, an Adapter Archive resource becomes available in the palette panel.
3. Drag the **Adapter Archive** into the design panel and specify information in the Configuration tab, then click **Apply**.
4. Go to the Enterprise Archive and click **Build Archive** to create the archive file.

See Also

See the *TIBCO Designer User's Guide* for more information about this procedure. The guide is available from the Designer Help menu.

Deploy the Project

Before deploying a project, the machine on which the adapter is installed must be part of a TIBCO administration domain. After you have installed the TIBCO Administration Server, any machine on which you install TIBCO Runtime Agent (required by an adapter) can be added to the administration domain. The TIBCO software installed on the machine is then visible and accessible via the TIBCO Administrator GUI.

When you deploy a project, startup scripts and other information about the different components are sent to the machines to which the components were assigned. The project data store and TIBCO Administration Server are updated with the deployed components.

To deploy a project:

1. Import the EAR file into TIBCO Administrator Enterprise Edition.
2. Assign adapter archives in the EAR file to adapters installed in the administration domain and likewise assign process archives to process engines.
3. Specify startup options for each adapter service.

Password Handling

At design time, the adapter uses a password to connect to the background application and fetch metadata. At run-time, the adapter uses a password to connect to the back-end application and interact with it. If you create a 4.x configuration using TIBCO Designer 5.1, and use the configuration against a 4.x adapter version, some special considerations are required for security.

When deploying the adapter check the `Service` property of the global variable in the global variables section, then go to the `Advanced` tab of the adapter archive and set the password value under the `Run-Time Variables` section.



Do not set the password to type `Password` in the global variables section for adapter configurations that are set to `AE Version 4.0` or `AE Version 5.0` (in the `Configuration` tab `Version` field) or any intermediate version.

Deploying Oracle Application Adapter Projects Using TIBCO Administrator

Using Migration Tool on Microsoft Windows

The migration tool is used to deploy Oracle Application Adapter projects using TIBCO Administrator. The Migration Tool gets installed while installing the TIBCO Adapter for Oracle Applications. This Tool is installed in the following location:

<TIBCO_HOME>/adapter/adorapps/<version_num>/bin/Migrationtool directory. This Migration Tool folder contains a Migrator.bat batch file.

To run the Migration Tool:

1. Open the Migrator.bat batch file and edit the REPO_URL. Specify the path of the Oracle Application Adapter repository, for example set
REPO_URL=D:\logTablePub.
2. Edit the CONFIG_URL specifying the Configuration URL of the Oracle Application Adapter repository. For example, set
CONFIG_URL=/tibco/private/adapter/logTablePub.
3. From the command prompt go to
<TIBCO_HOME>/adapter/adorapps/<version_num>/bin/Migrationtool and run the Migrator in the command prompt. Now the Oracle Application Adapter repository is converted and is ready for deployment using the TIBCO Administrator.
4. Convert the repository to a multfile format and configure the EAR file and deploy the EAR file using TIBCO Administrator.

Using Migration Tool on UNIX

The Migration Tool is used to deploy Oracle Application Adapter projects using TIBCO Administrator. This is installed while installing the TIBCO Adapter for Oracle Applications in the following location:

<TIBCO_HOME>/adapter/adorapps/<version_num>/Migrationtool directory. This Migration Tool folder contains a Migrator shell file.

To run the Migration Tool:

1. Open the Migrator.csh file and edit the REPO_URL. Specify the repository path of the 4x or 5.0.0 repository. For example, set
REPO_URL=/opt/logTablePub.
2. Edit the CONFIG_URL specifying the Configuration URL of the Oracle Application Adapter repository. For example set
CONFIG_URL=/tibco/private/adapter/logTablePub.

3. Log on to
`<TIBCO_HOME>/adapter/adorapps/<version_num>/bin/Migrationtool`
and change the shell to C shell and run the Migrator CSH. Now the Oracle Application Adapter repository is converted and is ready for deployment using TIBCO Administrator.
4. Convert the repository to a multifile format and configure the EAR file and deploy the EAR file using TIBCO Administrator.

Password Handling

At design time, the adapter uses a password to connect to the background application and fetch metadata. At run-time, the adapter uses a password to connect to the back-end application and interact with it. If you create a 4.x configuration using TIBCO Designer 5.1.3, and use the configuration against a 4.x adapter version, some special considerations are required for security.

When deploying the adapter check the Service property of the global variable in the global variables section, then go to the Advanced tab of the adapter archive and set the password value under the Run-Time Variables section.



Do not set the password to type Password in the global variables section for adapter configurations that are set to AE Version 4.0 or AE Version 5.0 (in the Configuration tab Version field) or any intermediate version.

See Also

See the *TIBCO Administrator User's Guide* for an introduction to the TIBCO administration domain and detailed information about the above steps.

See the *TIBCO Administrator Server Configuration Guide* for fault tolerance information.

Start or Stop the Adapter

The TIBCO Administrator `Application Management` module allows you to start, and stop deployed applications.

To start an adapter service from the module:

1. In the Administrator GUI left pane, expand **Application Management** > *Application-Name* > **Service Instances**.
2. In the `Service Instance` panel, select the checkbox next to the adapter service.
3. Click the **Start Selected** button.
4. The status changes from `Stopped` to `Starting` up to `Started`.
5. To stop the adapter service, click the **Stop Selected** button.

See Also

See the *TIBCO Administrator User's Guide* for more information.

Monitor the Adapter

TIBCO Administrator offers a number of monitoring options.

- Specify alerts and TIBCO Hawk rulebases for each machine in the domain.
- Specify alerts and Hawk rulebases for each adapter service.
- View the log for each adapter service.

See Also

See the *TIBCO Administrator User's Guide* for information about configuring the above monitoring options.

Chapter 6

Monitoring the Adapter Using TIBCO Hawk

This chapter explains how to use TIBCO Hawk microagents to monitor and manage the adapter.

Topics

- [Overview, page 256](#)
- [Starting TIBCO Hawk Software, page 257](#)
- [The Auto-Discovery Process, page 258](#)
- [Invoking Microagent Methods, page 259](#)
- [Available Microagents, page 260](#)

Overview

TIBCO Hawk is a sophisticated tool for enterprise-wide monitoring and managing of all distributed applications and systems. System administrators can use it to monitor adapters in a wide area network of any size. TIBCO Hawk can be configured to monitor system and adapter parameters and to take actions when PreDefined conditions occur. These actions include: sending alarms that are graphically displayed in the TIBCO Hawk display, sending email, paging, running executables, or modifying the behavior of a managed adapter.

Unlike other monitoring applications, TIBCO Hawk relies on a purely distributed intelligent agent architecture using publish or subscribe to distribute alerts. TIBCO Hawk uses TIBCO Rendezvous for all messaging and thus gains the benefits and scalability from the TIBCO Rendezvous features of publish/subscribe, subject name addressing, interest-based routing, and reliable multicast.

TIBCO Hawk is a purely event-based system that uses alerts. The agents are configured with rules that instruct them on everything from what and how to monitor to what actions to take when problems are discovered. Thus the workload is fully distributed throughout the enterprise. Every agent is autonomous in that it does not depend on other components to perform its functions.

The TIBCO Hawk Enterprise Monitor consists of these components:

- **Display**—GUI front end that displays alarms and provides editors to create rule bases, create tests, view messages, and invoke microagents to request information or initiate an action.
- **Agents**—Intelligent processes that perform monitoring and take actions as defined in rules.
- **Rulebases**—Rules that are loaded by agents to determine agent behavior.
- **Application Management Interface (AMI)**—Manages network applications via TIBCO Rendezvous and supports communication between a network application and monitoring TIBCO Hawk agents, including the ability to examine application variables, invoke methods, and monitor system performance.
- **Microagents**—Feed information back to TIBCO Hawk and expose action methods to rulebases.

For more information, see the TIBCO Hawk documentation.

Starting TIBCO Hawk Software

The TIBCO Hawk agent can be configured to start automatically during the system boot cycle. See the *TIBCO Hawk Installation and Configuration* guide for information about starting TIBCO Hawk.

The *TIBCO Hawk Administrator's Guide* explains how to start the TIBCO Hawk Display.

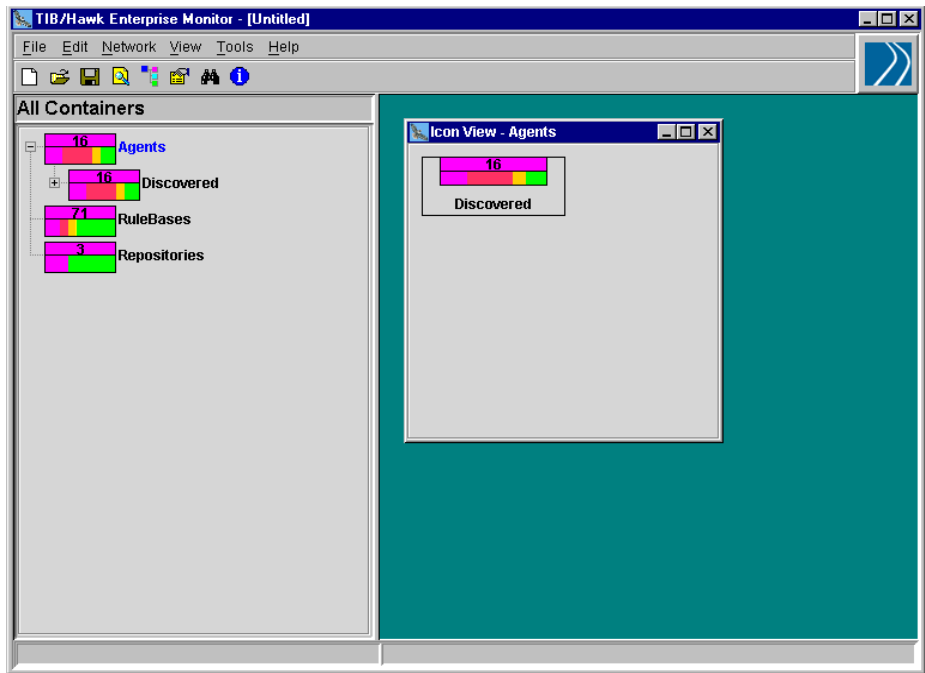
The guides are included in your TIBCO Hawk software installation area.

The Auto-Discovery Process

After you start an instance of TIBCO Hawk Display, it continually discovers machines running TIBCO Hawk agents on your network. Container icons are created for each agent, and arranged hierarchically in clusters. By default, agent icons are clustered according to subnets.

At first, the Agents container is empty. Its counter displays a value of zero and, on the right, the Discovered counter is also at zero. Both icons are initially green in color to show that no alerts or warning messages, are in effect. As agents are discovered, the counters increment to reflect the current number of discovered agents:

Figure 131 TIBCO Hawk Enterprise Monitor



Monitored network nodes are arranged in a hierarchical tree of containers. Clicking a container in the left panel displays nested items on the right.

Icon colors change to reflect the highest level of alert found on discovered agents. For an explanation of icon elements and characteristics, see your *TIBCO Hawk Administrator's Guide*.

Invoking Microagent Methods

Refer to **Invoking Microagent Methods** in Appendix C Monitoring the Adapter in TIBCO ActiveEnterprise from *TIBCO ActiveMatrix Adapter for Database Users Guide*.

Available Microagents

Refer to **Available Microagents** in Chapter 10 Monitoring the Adapter from *TIBCO ActiveMatrix Adapter for Database Users Guide*.

Chapter 7 **Setting Encoding Options**

This chapter describes how to use TIBCO Adapter for Oracle Applications in an international environment using non-ASCII languages.

Topics

- [Overview, page 262](#)
- [Setting TIBCO Messaging Encoding, page 264](#)
- [Configuring the Adapter to Communicate with the Database, page 266](#)
- [Relevant Environment Settings, page 270](#)

Overview

TIBCO Adapter for Oracle Applications provides internationalization support by taking advantage of Unicode, which is supported by the TIBCO Adapter SDK. Currently the international data support is provided for text data only. Database name, database schema, metadata and error messages do not currently support international character sets, but these features will be available in a future release.

The following illustration shows an example of TIBCO Adapter for Oracle Applications in a Japanese language environment serving databases of different encodings. The adapter converts the encoding between Unicode and the character sets of the databases.

Figure 132 Example of Unicode Conversion

In this example, Shift-JIS encoded Japanese data is retrieved from an Oracle database by the adapter's publication service, which converts the data into UTF-8 and publishes the UTF-8 data to the TIBCO messaging environment. An adapter subscription service receives this message and converts it from UTF-8 to EUC-JP, and inserts the data into an Oracle EUC-JP database to which it is connected.

By using UTF-8 as the TIBCO Messaging encoding between TIBCO ActiveEnterprise components, the two adapter services can serve databases of different encodings by exchanging data without data garbled.



In certain cases, intermediate entities (such as an ODBC driver, ODBC driver manager, or the database vendor's proprietary database client) may also perform some encoding conversion. This is discussed later in this chapter.

In circumstances where only ASCII and Latin-1 data (including English and Western European languages) are exchanged in the system, you can use Latin-1 encoding as the TIBCO Messaging encoding between TIBCO ActiveEnterprise components. It is not necessary to use UTF-8 as the TIBCO Messaging encoding in this case. UTF-8 is only required when the data to be exchanged is *not* included in the Latin-1 character set, as is the case with Asian languages and other European languages.

To configure the adapter to work correctly in transmitting non-ASCII data, you must set the TIBCO messaging encoding property. The following sections explain how to do this.

Setting TIBCO Messaging Encoding

TIBCO ActiveEnterprise components (TIBCO applications and adapters) use TIBCO Messaging encoding when communicating with each other. In the case of TIBCO Adapter for Oracle Applications, publication services use this encoding to publish data into TIBCO messaging system, and subscription services use it to receive data from the TIBCO messaging system.



The services do not use TIBCO Messaging encoding to connect to the background database. See [Configuring the Adapter to Communicate with the Database on page 266](#) for more information.

There are currently two choices for TIBCO Messaging encoding: Latin-1 (ISO8859-1) for transmitting ASCII and Latin-1 data, and UTF-8 for transmitting data of other languages.

The encoding property is set on the project itself at design time, and in the TIBCO administration server's property file when creating a TIBCO administration domain.

Encoding in a Server-based Project

The TIBCO administration server setting is used when the project is exported to a server repository or deployed using TIBCO Administrator Enterprise Edition.

For a server based project, the TIBCO messaging encoding is set by the `repo.encoding` property in the server's `tibcoadmin<domain-name>.tra` configuration file (located in `<install-path>/tibco/administrator/n.n/bin/`).

The encoding is set when using the TIBCO Domain Utility to create the domain or by editing the `repo.encoding` property in the `.tra` configuration file.

Each adapter or TIBCO application that uses the same server for storing and retrieving configuration data uses this encoding setting when communicating to each other. This assures that all TIBCO components (including adapters and other TIBCO applications) that belong to the same project use the same encoding value to communicate.

Encoding in a Local Project File

If the project's configurations are saved in a local project file; which is the normal case when the project is in design stage before deployment, the TIBCO Messaging encoding is determined by the encoding property saved in the local project file.

The encoding value is set on the root project folder in TIBCO Designer. By default, the value is set to ISO8859-1. You can change the value by selecting the folder and under the Project Settings tab, changing the value for the TIBCO Message Encoding field.

In order for different TIBCO components (each of which have their own local file repository) to communicate with each other without data garbled, they must use the same TIBCO Messaging encoding; that is, all these components must set their local project encoding to the same value.



The encoding property of a local project file is only for determining the TIBCO Messaging encoding, not the encoding used for the persistent storage of the project files. The project files are always saved as UTF-8 encoded files.

After a project is deployed as a server-based project, the encoding property set at design time is superseded by the encoding property set for the TIBCO administration server.

Configuring the Adapter to Communicate with the Database

The TIBCO message encoding is a project setting used for data conversion among different components. However, the adapter uses a different encoding value for exchanging data with the database. Because there are many intermediate entities between the adapter and the database (such as the ODBC driver manager, ODBC driver, and possibly the database native client), and each of these entities may conduct encoding conversions to the data stream flow between the service and the database, the data encoding from the database might not be known, while the data encoding to the database must be compatible to that of the database.

To eliminate the complexity introduced by these intermediate entities, you can make some configuration changes that ensure the service's encoding settings are compatible with the database instance's encoding. This also boosts the performance by eliminating unnecessary intermediate data conversions.

The configuration changes differ depending on the operating system that the service is running on, the database vendor, the ODBC driver used, and other factors. The following sections explain how to configure the adapter to communicate with the database.

Configuring the Adapter at Design Time

Use the following steps to set the adapter's encoding.

1. Determine your database server encoding. This is the encoding you want to use for the adapter's encoding. For example, for the Oracle database, the encoding can be found by querying the database with:

```
select * from nls_database_parameters;
```
2. Set the NLS_LANG environment variable for the client system.

Set the client machine's NLS_LANG environment variable to be consistent to the database instance's character set (NLS_CHARACTERSET) value. The client machine is where the adapter service is going to run. Consult your database administrator for the NLS_CHARACTERSET value of the database instance to connect to, and how to choose the right NLS_LANG value.

As an example, the system may use an Oracle database instance with the NLS_CHARACTERSET value set to JA16SJIS. In this case, the client machine's NLS_LANG environment variable must be set to JAPANESE_JAPAN.JA16SJIS.

The correct setting of the NLS_LANG environment variable eliminates the encoding conversion possibly conducted by the Oracle proprietary client.

3. Set the Adapter encoding in TIBCO Designer.

You must select or enter a valid encoding that the adapter understands. The adapter assumes the data coming from the database is in this encoding and ensures that data sent to the database is in this encoding. The encoding given here should match the encoding set for the database to which the adapter connects.

There are two ways to do this configuration:

- In TIBCO Designer, set this in the adapter resource. In the Configuration tab, click **Show All Tabs**. In the General tab, choose the encoding that is compatible with the encoding of the database instance to be connected to. Refer to [Table 11, TIBCO Adapter for Oracle Applications Adapter and Oracle NLS_Lang Values](#).

The adapter may support more encoding values than those shown in Designer. You can type in these encoding values. For more information about these, refer to Appendix B in *TIBCO Adapter Concepts*.

- You can modify the adapter encoding property, `adb.encoding`, in the adapter's `.tra` file. This setting supersedes the encoding option chosen in Designer.

4. Set the ODBC DSN in TIBCO Designer

This field is set under the adapter's Run-time Connection tab. You must also define the ODBC DSN on the machine on which the adapter is running.

- On UNIX, modify the `odbc.ini` file
`/tibco/adapter/adorapps/<version_num>/adb/odbc`. In addition, you must set the correct `IANAAppCodePage` to match the database encoding, if you are using a wire protocol driver. See [IANAAppCodePage on page 267](#) for a list of `IANAAppCodePage` values. The complete list is available in `<install-path>/tibco/adapter/adorapps/<version_num>/odbc/books/odbceref/odbceref.pdf`
- On Microsoft Windows, use the Windows ODBC GUI to define a system DSN. Note that if you are using a TIBCO wire protocol driver, the machine on which the adapter is running must be on the locale that matches the database encoding.

Table 10 *IANAAppCodePage*

Value	Encoding (Language)
3	ASCII (English)
4	ISO_8859-1 (Western European)
17	Shift_JIS (Japanese)

Table 10 IANAAppCodePage

Value	Encoding (Language)
18	EUC_JP (Japanese)
38	EUC_KR (Korean)
113	GBK (Simplified Chinese)
2026	BIG5 (Traditional Chinese)

Special Configurations for Supporting a UTF-8 Database

This section applies only to TIBCO ODBC drivers.

Companies doing business globally may require their database to be multilingual; that is, a single database instance supporting multiple languages so the data from the company's global business can be stored in it. Unicode is a natural choice for the encoding of this kind of database instance.

This requires the adapter to be able to use Unicode, or specifically, UTF-8 encoding, to communicate with the database. While this is already a supported feature, an exception exists when using TIBCO ODBC drivers.

TIBCO Adapter for Oracle Applications contains a binary mode for use with an embedded TIBCO driver when UTF-8 is the encoding to communicate with the background database instance. The configurations described below are applicable to both Microsoft Windows and UNIX platforms, and both non-wire and wire protocol drivers.

1. Set the adapter's encoding option to UTF-8.
2. In the adapter's .tra file, uncomment the following line:
#ADB.WCHAR = SQL_C_BINARY

This forces the adapter to use binary mode to communicate with the database.



These are for the internationalization configuration, which forces the adapter to use binary mode to communicate to the database instance. No further encoding related configurations are necessary. For example, on UNIX there is no need to set the AppCodePage attribute, and when using a non-wire driver there is no need to set the proprietary database client.

Run-time Environment Variables

In general, after deploying the adapter, all information should be in the correct place; however, you should verify the deployed `.tra` file and platform settings. The encoding conversion is dependent on the `TIB_ICU_DATA` environment variable. The deployed adapter `.tra` file should have a property called `tibco.env.TIB_ICU_DATA`. Make sure it points to the directory containing the `tibicudata.dat` file.

For the Oracle database, make sure the installed `NLS_LANG` attribute's value matches the database character set. If not, set `NLS_LANG` to the correct value before starting the adapter. See [TIBCO Adapter for Oracle Applications Adapter and Oracle NLS_Lang Values on page 270](#) for a list of values.

Relevant Environment Settings

In order to support international languages, the adapter uses the `tibicudata.dat` file to convert between the original database encoding and the TIBCO Messaging encoding (usually UTF-8). The environment variable `TIB_ICU_DATA` is already configured to point to the directory containing this file. Do not change this setting.

This environment variable setting is required on both Microsoft Windows and UNIX platforms.

The following table lists the TIBCO Adapter for Oracle Applications encoding values as shown in Designer.

Table 11 TIBCO Adapter for Oracle Applications Adapter and Oracle NLS_Lang Values

ActiveDatabase Adapter Encoding	Oracle NLS_LANG (containing NLS_CHARACTERSET)	Description
ASCII	US7ASCII	7-bit ASCII
ISO8859-1	language_territory.WE8ISO8859P1	ISO8859-1 (Latin-1), West European
Shift_JIS (CP943)	JAPANESE_JAPAN.JA16SJIS	Japanese Shift-JIS, CP943
Shift_JIS (TIBCO)	JAPANESE_JAPAN.JA16SJIS	Variant of IBM-943 (created by TIBCO for flavoring some MS-932 conversions)
KSC-5601	KOREAN_KOREA.KO16KSC5601	Korean KSC-5601
Big5	TRADITIONAL CHINESE_TAIWAN.ZHT16BIG5	Traditional Chinese Big5 with Euro
GBK	SIMPLIFIED_CHINESE_CHINA.ZHS16CGB231280	Simplified Chinese GBK (superset of GB2312-80)
EUC-JP	JAPANESE_JAPAN.JA16EUC	Japanese EUC
UTF8	AMERICAN_AMERICA.UTF8	Unicode Transformation Format-8

Appendix A **Available Transactions**

This appendix describes the available PreDefined Publish and Subscribe transactions.

Topics

- [Publish Transactions \(PreDefined Outbound\), page 272](#)
- [Subscribe Transactions \(PreDefined Inbound\), page 274](#)

Publish Transactions (PreDefined Outbound)

The following publish transactions are available.

Table 12 Publish Transactions

Module Name	Transaction Name
Engineering	Engineering BOM
General Ledger	GL balances
General Ledger	Journals
Inventory	Item Category
Inventory	Items
Inventory	On-hand quantity
Manufacturing	BOM Revisions
Manufacturing	BOM Routings
Manufacturing	Manufacturing BOM
Oracle Purchasing	Purchase Orders
Oracle Purchasing	Suppliers
Order Entry	Sales Orders
Payables	Invoices
Payables	Payments/Checks
Receivables	AR Transactions
Receivables	AR Customers
Shipping	Pick Details
Manufacturing	EnggChangeNotification
HRMS	Applicant
HRMS	Employee

Table 12 Publish Transactions

Module Name	Transaction Name
CRM	Interaction

Subscribe Transactions (PreDefined Inbound)

The following subscribe transactions are available.

Table 13 *Subscribe Transactions (Non-API)*

Module Name	Transaction Name
BOM	BOM Business Object Interface
Cost Management	Periodic Cost Open Interface
General Ledger	Budget Upload Interface
General Ledger	Journal Import Interface
Inventory	Customer Item Cross Reference Interface
Inventory	Customer Item Interface
Inventory	Cycle Count Entries Interface
Inventory	Open Item Interface
Inventory	Open Replenishment Interface
Inventory	Open Transaction Interface
Oracle Assets	Mass Additions Interface
Oracle Assets	Open Budget Interface
Oracle Master Scheduling	Open Forecast Interface
Oracle Master Scheduling	Open Master Schedule Interface
Oracle Purchasing	Open Purchasing Interface
Oracle Purchasing	Open Receiving Interface
Oracle Purchasing	Open Requisition Interface
Order Management	Integrate Order Mgmt using Order Import Interface
Order Management/ Oracle Receivables	Auto Invoicing Interface

Table 13 *Subscribe Transactions (Non-API)*

Module Name (Cont'd)	Transaction Name (Cont'd)
Payables	Open Payables Interface
Quality	Collection Import Interface
Receivables	Auto Lock Box
Receivables	Customer Interface
Release Management	Demand Schedule Interface
Work in Progress	Open Move Interface
Work in Progress	Open Resource Interface
Work in Progress	Work Order Interface

Table 14 *Subscribe Transactions (API)*

Module Name	Transaction Name
Order Management	Pick Confirm
Order Management	AutoCreate Deliveries
Order Management	AutoCreate DeliveriesTrip
Order Management	Update Shipping Attributes
Order Management	Detail To Delivery
Order Management	Split Line
Order Management	Create Update Delivery
Order Management	Delivery Action
Order Management	Create Update Freight Costs
Order Management	Validate Freight Cost Types
Order Management	Delete Freight Costs
Manufacturing	EnggChangeOrder

Table 14 *Subscribe Transactions (API)*

Module Name	Transaction Name
Manufacturing	BillsOfMtlAPI
HRMS	Applicant
HRMS	Employee
CRM	Interaction

Appendix B Oracle Application Transactions

This appendix gives you an overview of the structure of the PL/SQL packages and lists the various Oracle Application transactions that are provided with the adapter. It also provides details about the parameters that are required by the PL/SQL validations procedures and error logging procedures.

Topics

- [Structure of PL/SQL Packages, page 279](#)
- [Mandatory Parameters for All Transactions, page 281](#)
- [Customizing TIBCO Procedure for Normal Subscription, page 283](#)
- [Auto Invoicing, page 299](#)
- [AutoLockBox, page 301](#)
- [Bill of Materials, page 304](#)
- [BudgetUpload, page 306](#)
- [CollectionImport, page 308](#)
- [CustomerItem, page 310](#)
- [CustomerItemXRef, page 312](#)
- [CycleCountEntries, page 314](#)
- [DemandSchedules, page 316](#)
- [JournalImport, page 319](#)
- [MassAdditions, page 321](#)
- [MasterSchedule, page 323](#)
- [Open Replenishment, page 325](#)
- [OpenBudget, page 327](#)
- [OpenForecast, page 329](#)
- [OpenItem, page 331](#)

- [OpenMove](#), page 333
- [OpenResource](#), page 334
- [OpenTransaction](#), page 336
- [OrderEntry](#), page 338
- [PayablesOpen](#), page 340
- [PeriodicCost](#), page 343
- [PurchasingOpenDocs](#), page 345
- [RecCustomer](#), page 348
- [Receiving](#), page 349
- [Requisition](#), page 352
- [WorkOrder](#), page 354
- [Oracle Application APIs](#), page 356

Structure of PL/SQL Packages

This section gives you an overview of the different types of PL/SQL packages used with the adapter.

Procedure **main_adb**

This is a wrapper procedure called by the adapter in the subscription with reply scenario. This procedure passes three OUT parameters which are required as the reply parameters by the adapter.

In case of subscription, this procedure is not required.

Called From None

Calling Program This procedure calls the main program.

Procedure **main**

This is the main procedure called by the adapter in the subscription with reply scenario. In this case, this is called from the procedure `main_adb`.

In the simple subscription scenario, this is the starting procedure. This procedure passes two OUT parameters and five IN parameters which are required as the reply parameters by the TIBCO Adapter for Oracle Applications.

Called From In a subscription with reply scenario, `main_adb`.

In a simple subscription scenario, none.

Calling Program This procedure calls the `<TransactionName>_Validation` and `submit_<transName>` procedures.

Procedure **submit_<transName>**

This procedure is used to submit concurrent programs.

Called From The `main` procedure.

Calling Program None

Procedure **<TransactionName>_Validation**

This procedure is called by the main procedure. It inserts records from the TIBCO intermediate tables into the interface table.

- Called From** The main procedure.
- Calling Program** This procedure calls `Pre_<transName>` and `PostPre_<transName>` procedures.

Procedure `pro_pre_<TransactionName>`

This procedure is called by the `<TransactionName>_Validation` procedure. This procedure allows you to carry out your own modifications to records.

- Called From** The `<TransactionName>_Validation` procedure.
- Calling Program** None

Procedure `pro_post_<TransactionName>`

This procedure is called by the `<TransactionName>_Validation` procedure. This procedure allows you the flexibility of carrying out referential integrity checks. However, you should not modify record values here.

- Called From** The `<TransactionName>_Validation` procedure.
- Calling Program** None

Mandatory Parameters for All Transactions

There are three possible scenarios in subscription:

- **Subscription** — Here the TIBCO import procedure is scheduled.



For required parameters, you must select a value from the List Of Values (list is obtained from the value set that has been associated). For other parameters, you can select a value from the List of values or default the value to some value which has been explicitly set while defining the value set for that parameter.

- **Subscription with Reply**— Here the TIBCO import procedure is invoked by the adapter and the results of processing are sent back if a reply subject is specified in the incoming message.



For parameters that are being assigned to NULL, you can enter values. If values are specified, only those records with matching values are processed. You can also change values for other parameters in the `main_adb` procedure.

Mandatory Parameters

For a subscription service certain parameters must be set by the user for all transactions and similarly for a subscription service with reply certain parameters must be set in the `main_adb` procedure for all transactions. The parameters are as follows:

- `concurrent_flag` — This flag indicates whether the PL/SQL procedure are registered as a Concurrent program.

For subscription where no reply is expected, set to Y, the TIBCO PL/SQL procedures are registered as concurrent programs.

For subscription with reply, set to N inside the `main_adb` procedure for each package for each transaction. The TIBCO PL/SQL procedures are executed without registering them as concurrent programs. Only profiles are set and initialization is done.

- `skip_validn_flag`—This default is N for both subscription and subscription with reply. Retain the default setting.
- `user_id`—The User Id required to run concurrent programs without registering it or to submit import programs.

For subscription with reply, specify the `user_id` in the `main_adb` procedure. Since the concurrent program is not being registered, this value is required to set up client info and initialize the profile.

For simple subscription the `user_id` is required to submit import programs. For each transaction, go to the form/request screen where the Oracle

Concurrent Request is submitted, go to **Help>Diagnostics>Examine>Block**, select the \$PROFILES\$ option, then select Field - USER_ID and specify the appropriate value.

- **resp_id**—The resp_id required to run concurrent programs without registering it or to submit import programs.

For subscription with reply, specify the resp_id in the main_adb procedure. Since the concurrent program is not being registered, this value is required to set up client info and initialize the profile.

For simple subscription the resp_id is required to submit import programs. For each transaction, go to the form/request screen where the Oracle Concurrent Request is submitted, then go to **Help>Diagnostics>Examine>Block**, select the \$PROFILES\$ option, then select Field - RESP_ID and specify the appropriate value.

- **resp_appl_id**—The resp_appl_id required to run concurrent programs without registering it or to submit import programs.

For subscription with reply, specify the resp_appl_id in the main_adb procedure. Since the concurrent program is not being registered, this value is required to set up client info and initialize the profile.

For simple subscription the resp_appl_id is required to submit import programs. For each transaction, go to the form/request screen where the Oracle Concurrent Request is submitted, go to **Help>Diagnostics>Examine>Block**, select the \$PROFILES\$ option, then select Field - RESP_APPL_ID and specify the appropriate value.



The Header and the Detail tables for all the Transactions are related with the help of the TIBCO_KEY column.

Consider three tables A,B,C in which A is in the Header table,B is the child of A and C is the child of B. All the three tables will have the column TIBCO_KEY. Table A and Table B will relate to each other with the TIBCO_KEY field that is, $A.TIBCO_KEY = B.TIBCO_KEY$. Similarly B and C should relate to each other with the TIBCO_KEY field and the normal Oracle Business Key. For example, if ORG_ID is a Oracle Business Key relating B and C then $B.TIBCO_KEY=C.TIBCO_KEY$ and $B.ORG_ID=C.ORG_ID$.

Customizing TIBCO Procedure for Normal Subscription

For Normal Subscription the signature of the main procedure in all the Stored Procedures has to be modified to comment out the out_var1, out_var2 variables and locally declare them within the main procedure.

Sample procedure of PreDefined Inbound EmployeeSub Business Object

```

CREATE OR REPLACE PACKAGE pk_tib_hr_employee
-- $Header: $
/*-----
PACKAGE: pk_tib_hr_employee
PURPOSE: To create employee by calling
hr_applicant_api.create_employee API.

AUTHOR DATEVERDESCRIPTION
Infosys 28-Aug-20021.00 Created
-----*/
AS
  PROCEDURE pro_pre_employee(
-----
    -- This is the procedure which gives Users the flexibility to
    perform
    -- their own modifications to the records.
    -----
    out_errbuf          OUT          VARCHAR2,
    out_retcode          OUT          VARCHAR2,
    p_emp_rec            IN OUT      TIB_INT_HRMS_EMP%ROWTYPE,
    p_unique_id          IN          NUMBER,
    p_return_status      OUT          VARCHAR2
  );
  PROCEDURE pro_post_employee(
-----
    -- This is the procedure which gives Users the flexibility to
    perform
    -- validations.
    -----
    out_errbuf          OUT          VARCHAR2,
    out_retcode          OUT          VARCHAR2,
    p_emp_rec            IN OUT      TIB_INT_HRMS_EMP%ROWTYPE,
    p_unique_id          IN          NUMBER,
    p_return_status      OUT          VARCHAR2
  );
  PROCEDURE main_emp (
-----
    --PROCEDURE: main_emp
    -- PURPOSE: the main procedure which in turn calls validation
    -- procedure and the API
    -----
    out_errbuf          OUT          VARCHAR2,
    out_retcode          OUT          VARCHAR2,

```

```

        in_skip_validn_flag    IN          VARCHAR2,
        out_var1               IN OUT      NUMBER,
        out_var2               IN OUT      VARCHAR2
    );
-----
    --PROCEDURE:main_adb
    -- PURPOSE:  this is an overload version of main_emp
    --           to be used by adb
-----

    PROCEDURE  main_adb
    -- (
    --out_var1 OUT NUMBER,
    --out_var2 OUT VARCHAR2,
    --out_var3 OUT NUMBER
    -- )
    ;
END pk_tib_hr_employee;
/
--ENDSTMT
CREATE OR REPLACE PACKAGE BODY pk_tib_hr_employee
AS
    PROCEDURE pro_pre_employee (
        out_errbuf          OUT          VARCHAR2,
        out_retcode          OUT          VARCHAR2,
        p_emp_rec            IN OUT      TIB_INT_HRMS_EMP%ROWTYPE,
        p_unique_id          IN          NUMBER,
        p_return_status      OUT          VARCHAR2 )
    IS
    BEGIN
out_errbuf := NULL;
out_retcode := 0;
-----
    -- If required,put the Custom validation and
    -- conversion code here
    -----

        NULL;
    END pro_pre_employee;
    PROCEDURE pro_post_employee (
        out_errbuf          OUT          VARCHAR2,
        out_retcode          OUT          VARCHAR2,
        p_emp_rec            IN OUT      TIB_INT_HRMS_EMP%ROWTYPE,
        p_unique_id          IN          NUMBER,
        p_return_status      OUT          VARCHAR2
    )
    IS
    BEGIN
out_errbuf := NULL;
out_retcode := 0;
-----
    -- If required,put the Custom validation and
    -- conversion code here
    -----

        NULL;
    END pro_post_employee;
    PROCEDURE main_emp (
        out_errbuf          OUT          VARCHAR2,
        out_retcode          OUT          VARCHAR2,

```

```

        in_skip_validn_flag    IN          VARCHAR2,
        out_var1               IN OUT      NUMBER,
        out_var2               IN OUT      VARCHAR2
    )
IS
    c_transaction_name          CONSTANT VARCHAR2 (10)      :=
'HRMSEmpl';
    c_import_failed              CONSTANT NUMBER              :=
1;
    c_waiting_to_be_processed    CONSTANT NUMBER              :=
2;
    c_validation_failed          CONSTANT NUMBER              :=
3;
    c_validation_success         CONSTANT NUMBER              :=
4;
    c_import_success             CONSTANT NUMBER              :=
5;
    l_num_process_status         NUMBER;
    l_num_seq_id                 NUMBER;
    l_chr_errbuf                 VARCHAR2 (240);
    l_chr_retcode                VARCHAR2 (150);
    l_chr_return_status          VARCHAR2 (150);
    l_chr_employee_number        VARCHAR2 (30);
    l_num_person_id              NUMBER;
    l_num_assignment_id          NUMBER;
    l_num_per_object_version     NUMBER;
    l_num_asg_object_version     NUMBER;
    l_dte_per_effective_start_date DATE;
    l_dte_per_effective_end_date DATE;
    l_chr_full_name              VARCHAR2 (240);
    l_num_per_comment_id         NUMBER;
    l_num_assignment_sequence    NUMBER;
    l_chr_assignment_number      VARCHAR2 (30);
    l_bol_name_combination_warning BOOLEAN;
    l_bol_assign_payroll_warning BOOLEAN;
    w_error_source               CONSTANT VARCHAR2 (10)      :=
'ORACLE';
    w_status_log_rec             pk_tib_log.log_rec_type;
    l_skip_rec_exception         EXCEPTION;
    l_chr_error_description      VARCHAR2(240);
    l_object_version_number      NUMBER;
    l_old_object_version_number  NUMBER;
    l_dte_effective_start_date   date;
    l_dte_effective_end_date     date;
    l_chr_full_name1             varchar2(240);
    l_chr_comment_id             number;
    l_b_name_combination_warning boolean;
    l_b_assign_payroll_warning   boolean;
    l_b_orig_hire_warning        boolean;
    w_success_msg                VARCHAR2(2000);
    --Cursor for selecting records from the interface table
    CURSOR cur_employee_iface (
        p_process_flag
TIB_INT_HRMS_EMP.tibco_int_process_flag%TYPE )
    IS
        SELECT      *
        FROM TIB_INT_HRMS_EMP

```

```

        WHERE tibco_int_process_flag = p_process_flag
    FOR UPDATE;
BEGIN
    out_retcode := '0';
    out_errbuf := NULL;
    fnd_file.put_line (
fnd_file.LOG,
'-----Started Processing-----'
);
--delete previously processed records
DELETE TIB_INT_HRMS_EMP
where tibco_int_process_flag = c_import_success;
--Update the Intermediate Process Flag to Waiting to be
Processed
UPDATE TIB_INT_HRMS_EMP
    SET tibco_int_process_flag = c_waiting_to_be_processed
    WHERE tibco_int_process_flag IS NULL;
--process the records one by one
FOR cur_employee_iface_rec IN
    cur_employee_iface (c_waiting_to_be_processed)
LOOP
    --Update the Intermediate Table with unique id
    --SELECT tib_int_hrms_emp_s.NEXTVAL
    -- INTO l_num_seq_id
    -- FROM DUAL;
    --UPDATE TIB_INT_HRMS_EMP
    -- SET tibco_unique_id = l_num_seq_id
    -- WHERE CURRENT OF cur_employee_iface;
    l_num_process_status := 0;
    BEGIN
        SAVEPOINT process_employee;
        IF in_skip_validn_flag = 'N'
        THEN
            -- call the custom procedure for validation
            -- or conversion
            pro_pre_employee (
                l_chr_errbuf,
                l_chr_retcode,
                cur_employee_iface_rec,
                cur_employee_iface_rec.TIBCO_KEY,
                l_chr_return_status
            );
            IF l_chr_retcode <> '0'
            THEN
                --failed during validation
                l_num_process_status := c_validation_failed;
                RAISE l_skip_rec_exception;
            END IF;
            -- call the custom procedure for validation
            -- or conversion
            pro_post_employee (
                l_chr_errbuf,
                l_chr_retcode,
                cur_employee_iface_rec,
                cur_employee_iface_rec.TIBCO_KEY,
                l_chr_return_status
            );
        );
    );

```



```

IF l_chr_retcode <> '0'
THEN
  --failed during validation
  l_num_process_status := c_validation_failed;
  RAISE l_skip_rec_exception;
END IF;
END IF;
IF cur_employee_iface_rec.create_update_flag = 'INSERT'
THEN
  BEGIN
    l_chr_employee_number :=
cur_employee_iface_rec.employee_number;
    hr_employee_api.create_employee (
      FALSE ,
      cur_employee_iface_rec.hire_date,
      cur_employee_iface_rec.business_group_id,
      cur_employee_iface_rec.last_name,
      cur_employee_iface_rec.sex,
      cur_employee_iface_rec.person_type_id,
      cur_employee_iface_rec.per_comments,

cur_employee_iface_rec.date_employee_data_verified,
      cur_employee_iface_rec.date_of_birth,
      cur_employee_iface_rec.email_address,
      l_chr_employee_number,

cur_employee_iface_rec.expense_check_send_to_addres,
      cur_employee_iface_rec.first_name,
      cur_employee_iface_rec.known_as,
      cur_employee_iface_rec.marital_status,
      cur_employee_iface_rec.middle_names,
      cur_employee_iface_rec.nationality,
      cur_employee_iface_rec.national_identifier,
      cur_employee_iface_rec.previous_last_name,
      cur_employee_iface_rec.registered_disabled_flag,
      cur_employee_iface_rec.title,
      cur_employee_iface_rec.vendor_id,
      cur_employee_iface_rec.work_telephone,
      cur_employee_iface_rec.attribute_category,
      cur_employee_iface_rec.attribute1,
      cur_employee_iface_rec.attribute2,
      cur_employee_iface_rec.attribute3,
      cur_employee_iface_rec.attribute4,
      cur_employee_iface_rec.attribute5,
      cur_employee_iface_rec.attribute6,
      cur_employee_iface_rec.attribute7,
      cur_employee_iface_rec.attribute8,
      cur_employee_iface_rec.attribute9,
      cur_employee_iface_rec.attribute10,
      cur_employee_iface_rec.attribute11,
      cur_employee_iface_rec.attribute12,
      cur_employee_iface_rec.attribute13,
      cur_employee_iface_rec.attribute14,
      cur_employee_iface_rec.attribute15,
      cur_employee_iface_rec.attribute16,
      cur_employee_iface_rec.attribute17,
      cur_employee_iface_rec.attribute18,

```

```

cur_employee_iface_rec.attribute19,
cur_employee_iface_rec.attribute20,
cur_employee_iface_rec.attribute21,
cur_employee_iface_rec.attribute22,
cur_employee_iface_rec.attribute23,
cur_employee_iface_rec.attribute24,
cur_employee_iface_rec.attribute25,
cur_employee_iface_rec.attribute26,
cur_employee_iface_rec.attribute27,
cur_employee_iface_rec.attribute28,
cur_employee_iface_rec.attribute29,
cur_employee_iface_rec.attribute30,
cur_employee_iface_rec.per_information_category,
cur_employee_iface_rec.per_information1,
cur_employee_iface_rec.per_information2,
cur_employee_iface_rec.per_information3,
cur_employee_iface_rec.per_information4,
cur_employee_iface_rec.per_information5,
cur_employee_iface_rec.per_information6,
cur_employee_iface_rec.per_information7,
cur_employee_iface_rec.per_information8,
cur_employee_iface_rec.per_information9,
cur_employee_iface_rec.per_information10,
cur_employee_iface_rec.per_information11,
cur_employee_iface_rec.per_information12,
cur_employee_iface_rec.per_information13,
cur_employee_iface_rec.per_information14,
cur_employee_iface_rec.per_information15,
cur_employee_iface_rec.per_information16,
cur_employee_iface_rec.per_information17,
cur_employee_iface_rec.per_information18,
cur_employee_iface_rec.per_information19,
cur_employee_iface_rec.per_information20,
cur_employee_iface_rec.per_information21,
cur_employee_iface_rec.per_information22,
cur_employee_iface_rec.per_information23,
cur_employee_iface_rec.per_information24,
cur_employee_iface_rec.per_information25,
cur_employee_iface_rec.per_information26,
cur_employee_iface_rec.per_information27,
cur_employee_iface_rec.per_information28,
cur_employee_iface_rec.per_information29,
cur_employee_iface_rec.per_information30,
cur_employee_iface_rec.date_of_death,
cur_employee_iface_rec.background_check_status,
cur_employee_iface_rec.background_date_check,
cur_employee_iface_rec.blood_type,
cur_employee_iface_rec.correspondence_language,
cur_employee_iface_rec.fast_path_employee,
cur_employee_iface_rec.fte_capacity,
cur_employee_iface_rec.honors,
cur_employee_iface_rec.internal_location,
cur_employee_iface_rec.last_medical_test_by,
cur_employee_iface_rec.last_medical_test_date,
cur_employee_iface_rec.mailstop,
cur_employee_iface_rec.office_number,
cur_employee_iface_rec.on_military_service,

```

```

        cur_employee_iface_rec.pre_name_adjunct,
        cur_employee_iface_rec.projected_start_date,
        cur_employee_iface_rec.resume_exists,
        cur_employee_iface_rec.resume_last_updated,
        cur_employee_iface_rec.second_passport_exists,
        cur_employee_iface_rec.student_status,
        cur_employee_iface_rec.work_schedule,
        cur_employee_iface_rec.suffix,
        cur_employee_iface_rec.benefit_group_id,
        cur_employee_iface_rec.receipt_of_death_cert_date,
        cur_employee_iface_rec.coord_ben_med_pln_no,
        cur_employee_iface_rec.coord_ben_no_cvg_flag,
        cur_employee_iface_rec.coord_ben_med_ext_er,
        cur_employee_iface_rec.coord_ben_med_pl_name,

cur_employee_iface_rec.coord_ben_med_insr_crr_name,

cur_employee_iface_rec.coord_ben_med_insr_crr_ident,
        cur_employee_iface_rec.coord_ben_med_cvg_strt_dt,
        cur_employee_iface_rec.coord_ben_med_cvg_end_dt,
        cur_employee_iface_rec.uses_tobacco_flag,
        cur_employee_iface_rec.dpdnt_adoption_date,
        cur_employee_iface_rec.dpdnt_vlntry_svce_flag,
        cur_employee_iface_rec.original_date_of_hire,
        cur_employee_iface_rec.adjusted_svc_date,
        cur_employee_iface_rec.town_of_birth,
        cur_employee_iface_rec.region_of_birth,
        cur_employee_iface_rec.country_of_birth,
        cur_employee_iface_rec.global_person_id,
        l_num_person_id,
        l_num_assignment_id,
        l_num_per_object_version,
        l_num_asg_object_version,
        l_dte_per_effective_start_date,
        l_dte_per_effective_end_date,
        l_chr_full_name,
        l_num_per_comment_id,
        l_num_assignment_sequence,
        l_chr_assignment_number,
        l_bol_name_combination_warning,
        l_bol_assign_payroll_warning,
        l_b_orig_hire_warning
    );
    l_num_process_status := c_import_success;
    fnd_file.put_line (
fnd_file.LOG,
        '-----Created employee-----'||
        'Employee No : ' || l_chr_employee_number
    );
EXCEPTION
    WHEN OTHERS
    THEN
        DBMS_OUTPUT.put_line ('Error' || SUBSTR (SQLERRM,
1, 150));
        l_num_process_status := c_import_failed;
        l_chr_error_description := SUBSTR (SQLERRM, 1, 240);
        fnd_file.put_line (

```

```

        fnd_file.LOG,
        'Unexpected error occurred '
        || 'while calling the API'
        || ' The oracle error message is : '
        || TO_CHAR (SQLCODE)
        || SQLERRM
    );
    RAISE l_skip_rec_exception;
END;
ELSIF cur_employee_iface_rec.create_update_flag = 'UPDATE'
THEN
    BEGIN
        select max(object_version_number)
        into l_object_version_number
        from PER_ALL_PEOPLE_F
        where person_id = cur_employee_iface_rec.person_id ;
    l_chr_employee_number := cur_employee_iface_rec.employee_number;
    l_old_object_version_number :=
    l_object_version_number;
    hr_person_api.update_person(false,
    cur_employee_iface_rec.effective_date,
    cur_employee_iface_rec.datetrack_update_mode,
    cur_employee_iface_rec.person_id,
    l_object_version_number
    ,cur_employee_iface_rec.person_type_id,
    cur_employee_iface_rec.last_name,
    cur_employee_iface_rec.applicant_number,
    cur_employee_iface_rec.pER_comments,
    cur_employee_iface_rec.date_employee_data_verified,
    cur_employee_iface_rec.date_of_birth           --in          date
    default hr_api.g_date
    ,cur_employee_iface_rec.email_address           --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.employee_number         --in
    --outvarchar2
    ,cur_employee_iface_rec.expense_check_send_to_addres --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.first_name               --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.known_as                 --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.marital_status           --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.middle_names             --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.nationality              --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.national_identifier       --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.previous_last_name       --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.registered_disabled_flag --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.sex                      --in
    varchar2 default hr_api.g_varchar2
    ,cur_employee_iface_rec.title                    --in
    varchar2 default hr_api.g_varchar2

```

```

,cur_employee_iface_rec.vendor_id          --in
number default hr_api.g_number
,cur_employee_iface_rec.work_telephone     --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute_category --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute1         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute2         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute3         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute4         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute5         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute6         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute7         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute8         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute9         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute10        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute11        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute12        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute13        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute14        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute15        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute16        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute17        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute18        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute19        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute20        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute21        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute22        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute23        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute24        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute25        --in
varchar2 default hr_api.g_varchar2

```

```

,cur_employee_iface_rec.attribute26           --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute27           --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute28           --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute29           --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.attribute30           --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information_category --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information1       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information2       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information3       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information4       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information5       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information6       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information7       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information8       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information9       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information10      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information11      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information12      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information13      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information14      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information15      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information16      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information17      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information18      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information19      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information20      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information21      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information22      --in
varchar2 default hr_api.g_varchar2

```

```

,cur_employee_iface_rec.per_information23      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information24      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information25      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information26      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information27      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information28      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information29      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.per_information30      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.date_of_death          --in
date      default hr_api.g_date
,cur_employee_iface_rec.background_check_status --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.background_date_check --in
date      default hr_api.g_date
,cur_employee_iface_rec.blood_type             --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.correspondence_language --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.fast_path_employee      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.fte_capacity            --in
number    default hr_api.g_number
,cur_employee_iface_rec.hold_applicant_date_until --in
date      default hr_api.g_date
,cur_employee_iface_rec.honors                 --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.internal_location       --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.last_medical_test_by    --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.last_medical_test_date --in
date      default hr_api.g_date
,cur_employee_iface_rec.mailstop               --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.office_number          --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.on_military_service     --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.pre_name_adjunct        --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.projected_start_date    --in
date      default hr_api.g_date
,cur_employee_iface_rec.rehire_authorizor      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.rehire_recommendation  --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.resume_exists          --in
varchar2 default hr_api.g_varchar2

```

```

,cur_employee_iface_rec.resume_last_updated      --in
date      default hr_api.g_date
,cur_employee_iface_rec.second_passport_exists    --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.student_status            --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.work_schedule             --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.rehire_reason             --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.suffix                   --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.benefit_group_id          --in
number    default hr_api.g_number
,cur_employee_iface_rec.receipt_of_death_cert_date --in
date      default hr_api.g_date
,cur_employee_iface_rec.coord_ben_med_pln_no      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.coord_ben_no_cvg_flag     --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.coord_ben_med_ext_er      --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.coord_ben_med_pl_name     --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.coord_ben_med_insr_crr_name --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.coord_ben_med_insr_crr_ident --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.coord_ben_med_cvg_strt_dt --in
date      default hr_api.g_date
,cur_employee_iface_rec.coord_ben_med_cvg_end_dt  --in
date      default hr_api.g_date
,cur_employee_iface_rec.uses_tobacco_flag         --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.dpdnt_adoption_date       --in
date      default hr_api.g_date
,cur_employee_iface_rec.dpdnt_vlntry_svce_flag    --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.original_date_of_hire     --in
date      default hr_api.g_date
,cur_employee_iface_rec.adjusted_svc_date         --in
date      default hr_api.g_date
,cur_employee_iface_rec.town_of_birth             --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.region_of_birth           --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.country_of_birth          --in
varchar2 default hr_api.g_varchar2
,cur_employee_iface_rec.global_person_id          --in
varchar2 default hr_api.g_varchar2
,l_dte_effective_start_date                      --out   date
,l_dte_effective_end_date                      --out   date
,l_chr_full_name                                --out   varchar2
,l_chr_comment_id                               --out   number
,l_b_name_combination_warning                   --out   boolean
,l_b_assign_payroll_warning                     --out   boolean
,l_b_orig_hire_warning                          --out   boolean

```



```

);
        l_num_process_status := c_import_success;
        fnd_file.put_line (
fnd_file.LOG,
'-----Created employee-----' ||
'Employee No : ' || l_chr_employee_number
);
EXCEPTION
    WHEN OTHERS
    THEN
        DBMS_OUTPUT.put_line ('Error' || SUBSTR (SQLERRM,
1, 150));
        l_num_process_status := c_import_failed;
        l_chr_error_description := SUBSTR (SQLERRM, 1, 240);
        fnd_file.put_line (
            fnd_file.LOG,
                'Unexpected error occurred '
                || 'while calling the API'
                || ' The oracle error message is : '
                || TO_CHAR (SQLCODE)
                || SQLERRM
            );
        RAISE l_skip_rec_exception;
    END;
END IF;
EXCEPTION
    WHEN l_skip_rec_exception
    THEN
        ROLLBACK TO process_employee;
END;
UPDATE TIB_INT_HRMS_EMP
    SET tibco_int_process_flag = l_num_process_status
    WHERE CURRENT OF cur_employee_iface;
w_status_log_rec := NULL;
IF l_num_process_status = c_import_success
THEN
    w_status_log_rec.trans_name := c_transaction_name;
    w_status_log_rec.table_name := 'PER_ALL_PEOPLE_F';
    w_status_log_rec.unique_id := l_chr_employee_number;
    w_status_log_rec.error_source := w_error_source;
    w_status_log_rec.status := 'S';
    w_status_log_rec.in_hdr_fldname1 := 'EMPLOYEE_NUMBER';
    w_status_log_rec.in_hdr_fldvalue1 :=
l_chr_employee_number;
    w_status_log_rec.in_hdr_fldname2 := 'LAST_NAME';
    w_status_log_rec.in_hdr_fldvalue2 :=
cur_employee_iface_rec.last_name;
    w_status_log_rec.in_hdr_fldname3 := 'FIRST_NAME';
    w_status_log_rec.in_hdr_fldvalue3 :=
cur_employee_iface_rec.first_name;
    w_status_log_rec.out_hdr_fldname1 := 'PERSON_ID';
    w_status_log_rec.out_hdr_fldvalue1 := l_num_person_id;
    w_status_log_rec.out_hdr_fldname2 := 'ASSIGNMENT_ID';
    w_status_log_rec.out_hdr_fldvalue2 :=
l_num_assignment_id;
    IF cur_employee_iface_rec.create_update_flag = 'UPDATE'
    THEN

```

```

        w_status_log_rec.in_hdr_fldname3 := 'Old Object
Version Number';
        w_status_log_rec.in_hdr_fldvalue3 :=
l_old_object_version_number;
        w_status_log_rec.out_hdr_fldname3 := 'New Object Version
Number';
        w_status_log_rec.out_hdr_fldvalue3 := l_object_version_number ;

/***** added to log the success messages 31-12-2002 *****/
        w_success_msg:=(w_status_log_rec.out_hdr_fldname1)||':'||
(w_status_log_rec.out_hdr_fldvalue1) ||', '||
(w_status_log_rec.out_hdr_fldname2) ||': '
|| (w_status_log_rec.out_hdr_fldvalue2) ||', '||
(w_status_log_rec.out_hdr_fldname3) ||': '
|| (w_status_log_rec.out_hdr_fldvalue3);
        out_var2:= CONCAT(out_var2,CONCAT(w_success_msg,', '));
        out_var1:=0;
        ELSE
w_success_msg:=(w_status_log_rec.out_hdr_fldname1)||':'||
(w_status_log_rec.out_hdr_fldvalue1) ||', '||
(w_status_log_rec.out_hdr_fldname2) ||': '
|| (w_status_log_rec.out_hdr_fldvalue2);
        out_var2:=
CONCAT(out_var2,CONCAT(w_success_msg,', '));
        out_var1:=0;
/*****
        END IF;
    ELSE
        w_status_log_rec.trans_name := c_transaction_name;
        w_status_log_rec.table_name := 'PER_ALL_PEOPLE_F';
        w_status_log_rec.unique_id := l_chr_employee_number;
        w_status_log_rec.error_source := w_error_source;
        w_status_log_rec.status := 'E';
        w_status_log_rec.description := l_chr_error_description;
        w_status_log_rec.in_hdr_fldname1 := 'EMPLOYEE_NUMBER';
        w_status_log_rec.in_hdr_fldvalue1 :=
l_chr_employee_number;
        w_status_log_rec.in_hdr_fldname2 := 'LAST_NAME';
        w_status_log_rec.in_hdr_fldvalue2 :=
cur_employee_iface_rec.last_name;
        w_status_log_rec.in_hdr_fldname3 := 'FIRST_NAME';
        w_status_log_rec.in_hdr_fldvalue3 :=
cur_employee_iface_rec.first_name;
        out_var1:=1;
        --out_var2:=w_status_log_rec.description;
out_var2:=CONCAT(out_var2,CONCAT(w_status_log_rec.description,', '))
);
        IF l_num_process_status = c_validation_failed
        THEN
            w_status_log_rec.ERROR_CODE := 'VALIDATION FAILED';
        ELSIF l_num_process_status = c_import_failed
        THEN
            w_status_log_rec.ERROR_CODE := 'IMPORT FAILED';
        END IF;
        fnd_file.put_line (
fnd_file.LOG,
'-----Error-----' ||

```

```

'Last Name  : ' || cur_employee_iface_rec.last_name ||
'First Name : ' || cur_employee_iface_rec.first_name
);
    END IF;
    pk_tib_log.log_interface_status (w_status_log_rec);
END LOOP;
COMMIT;
fnd_file.put_line (
fnd_file.LOG,
'-----End of Processing-----'
);
EXCEPTION
WHEN OTHERS
THEN
    fnd_file.put_line (
        fnd_file.LOG,
        'Unexpected error occurred ' ||
        ' The oracle error message is : ' ||
        TO_CHAR (SQLCODE) ||
        SQLERRM
    );
    DBMS_OUTPUT.put_line (
        'Unexpected error occurred '
        || ' The oracle error message is : '
        || TO_CHAR (SQLCODE)
        || SQLERRM
    );
    out_errbuf := 'Unexpected error occurred '
        || ' The oracle error message is : '
        || TO_CHAR (SQLCODE)
        || SQLERRM;
    out_retcode := '2';

END main_emp;

PROCEDURE main_adb
--(
--out_var1 OUT NUMBER,
--out_var2 OUT VARCHAR2,
--out_var3 OUT NUMBER
--)
IS
l_chr_errbuf VARCHAR2(240);
l_chr_retcode VARCHAR2(240);
l_chr_skip_validn_flag VARCHAR2(1) := 'N';
--out_var1 NUMBER;
--out_var2 VARCHAR2(2000);
BEGIN
/*main_emp(
l_chr_errbuf,
l_chr_retcode,
l_chr_skip_validn_flag,
out_var1,
out_var2
); */
DBMS_OUTPUT.put_line (
'Execution Starts....');

```

```
        END main_adb;  
    END pk_tib_hr_employee;  
/
```



The `main_emp` procedure needs to be registered as an Oracle Concurrent Request. You will have to run the Oracle Concurrent Request from the Oracle Applications front end.

Auto Invoicing

Application	Oracle Receivables (AR)
Oracle Import	RAXTMR
Concurrent Program Name	Autoinvoice Master Program
Parameters	<p>Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.</p> <ul style="list-style-type: none">• Number of Instances (Required)• Batch Source• Default Date• Base Due Date on Trx Date• Organization Id <p>Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.</p> <ul style="list-style-type: none">• Number of Instances• Batch Source• Default Date• Base Due Date on Trx Date• Organization Id



The optional parameters (which are required by the Oracle Application concurrent program) are default to NULL in the call to the concurrent program. If needed you can modify the optional values.

In the case of Subscription Reply, the program defaults to NULL for Batch Source and Default Date, 1 for number of instances and Y for Base Due Date on Trx Date.

The key fields relating the header (RA_INTERFACE_LINES_ALL) and the detail tables (RA_INTERFACE_SALESCREDITS_ALL) are the INTERFACE_LINE_ATTRIBUTE(1-15). However, the adapter requires that the relationship keys between the incoming header record and the line records (for the TIBCO Intermediate tables) be populated in the INTERFACE_LINE_ID field.

Table 15 Data Types for Subscription - Auto Invoicing

Parameter	Data Type
Number Of Instances	NUMBER
Batch Source	VARCHAR2 (50)
Default Date	DATE
Base Due Date On Trx Date	VARCHAR2 (1)
Organization Id	NUMBER



Ensure that the Oracle Interface tables are purged before running the transaction for the first time.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- BATCH_SOURCE_NAME
- LINE_TYPE
- SET_OF_BOOKS_ID

The following key fields are logged in the TIBCO Log Table indicating success:

- CUSTOMER_TRX_ID
- LAST_UPDATE_DATE

Error Logging

Errors are logged into AR.RA_INTERFACE_ERRORS_ALL table.

AutoLockBox

Application: Oracle Receivables (AR)

Oracle Import: ARLPLB

**Concurrent
Program Name** Process Lockboxes

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- New Transmission
- Transmission Name (Required)
- Submit Validation
- Pay Unrelated Invoices
- GL Date
- Report Format (Required)
- Complete Batches Only
- Submit Postbatch (Required)
- Alternate Name Search Option
- Post Partial Amount or Reject Entire Receipt



In the case of a simple subscription, the concurrent program is launched using a form at the front end where the parameters are specified. Since data is accessed from TIBCO intermediate Tables the submit import, data file and control file option is not available inside the TIBCO program while setting the TIBCO program parameters, although this option is available from the front end. Data is directly imported in to the Oracle source tables, it is not staged in the Interim Cash Receipt tables, though this feature is supported by ORACLE AutoLockBox Import Interface.

Subscription with Reply—Specify the following parameters in the TIBCO Adapter for Oracle Application's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- New Transmission
- Transmission Name

- Submit Validation
- Pay Unrelated Invoices
- GL Date
- Report Format
- Complete Batches Only
- Submit Postbatch
- Alternate Name Search Option
- Post Partial Amount or Reject Entire Receipt.

In case of Subscription Reply, the defaults are NULL for Transmission Name and, 'A' for Report Format and 'Y' for Submit Postbatch.



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

Table 16 Data Types for Subscription - AutoLockBox

Parameter	Data Type
New Transmission	VARCHAR2(40)
Transmission Name	VARCHAR2(30)
Submit Validation	VARCHAR2(40)
Pay Unrelated Invoices	VARCHAR2(40)
GL Date	DATE
Report Format	VARCHAR2(40)
Complete Batches Only	VARCHAR2(40)
Submit Postbatch	VARCHAR2(40)
Alternate Name Search Option	VARCHAR2(40)
Post Partial Amount or Reject Entire Receipt	VARCHAR2(40)



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed to the TIBCO program.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- RECEIPT_NUMBER
- DEPOSIT_DATE
- ORG_ID

The following key fields are logged in the TIBCO Log Table indicating success:

- CASH_RECEIPT_ID
- LAST_UPDATE_DATE

Error Logging:

Errors get logged into the AR_PAYMENTS_INTERFACE_ALL table for each processed record in the STATUS column.

Bill of Materials

- Application:

Oracle Bills Of Material
- Oracle Import:

BMCOIN
- Concurrent Program Name

Bill and Routing Interface
- Parameters:

Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

 - All Org Flag (Required)
 - Import Bill Of Materials (Required)
 - Delete Processed Rows (Required)

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

 - All Org Flag
 - Import Bill Of Materials
 - Delete Processed Rows
 - Specify all the above values as 1.



The Import Routings parameter that the Oracle Import program has is not provided in the TIBCO PL/SQL procedure.



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

In case of Subscription Reply, enter 1 for both All Org Flag,1 for Import Bill Of Materials and 1 for Delete Processed Rows

Table 17 Data Types for Subscription - Bill of Materials

Parameter	Data Type
All Org Flag	NUMBER
Import Bill Of Materials	NUMBER

Table 17 Data Types for Subscription - Bill of Materials

Parameter	Data Type
Delete Processed Rows	NUMBER



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- ASSEMBLY_ITEM_ID
- ORGANIZATION_ID
- ALTERNATE_BOM_DESIGNATOR

The following key fields are logged in the TIBCO Log Table indicating success:

- BILL_SEQUENCE_ID
- LAST_UPDATE_DATE

Error Logging:

Errors are logged into MTL_INTERFACE_ERRORS and TRANSACTION_ID contains the key relating the interface records to the error table entries.

BudgetUpload

Application: Oracle General Ledger

Oracle Import: GLBBSU

Concurrent Program Name Upload Budget Amounts

- Parameters:** Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.
- Budget Name (Required)
 - Budget Entity Name (Required)
- Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.
- Budget Name
 - Budget Entity Name
 - Enter NULL for both Budget Entry Name and Budget Entity Name



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

In case of Subscription Reply, enter NULL for both Budget Name and Budget Entity Name.

Table 18 Data Types for Subscription - BudgetUpload

Parameter	Data Type
Budget Name	VARCHAR2(15)
Budget Entity Name	VARCHAR2(25)

- Key Fields** The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:
- BUDGET_NAME

- BUDGET_ENTITY_NAME

The following key fields are logged in the TIBCO Log Table indicating success:

- PERIOD_TYPE
- PERIOD_YEAR
- CODE_COMBINATION_ID
- BUDGET_VERSION_ID
- LAST_UPDATE_DATE

Error Logging: Errors are logged in the log file generated by the Oracle Application as a result of running the concurrent program.

CollectionImport

Application: Oracle Quality

Oracle Import: QLTTTRAMB

Concurrent Program Name Collection Import Manager

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Transaction Type (Required)
- Worker Rows
- Gather Statistics

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Transaction Type
- Worker Rows
- Gather Statistics

In case of Subscription Reply, enter NULL for Transaction Type and a default of 200 for Worker Rows and YES for Gather Statistics.

Table 19 Data Types for Subscription - CollectionImport

Parameter	Data Type
Worker Rows	NUMBER
Transaction Type	VARCHAR2(80)
Gather Statistics	VARCHAR2(80)

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- TRANSACTION_INTERFACE_ID



This parameter cannot be tracked in case of success.

The following key fields are logged in the TIBCO Log Table indicating success:

- PLAN_ID
- COLLECTION_ID
- OCCURRENCE
- LAST_UPDATE_DATE

Error Logging: Errors are logged into the QA_INTERFACE_ERRORS table.

CustomerItem

Application: Oracle Inventory

Oracle Import: INVCIINT

Concurrent Program Name Import Customer Items

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Abort On Error
- Delete Successful Rows

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Abort On Error
- Delete Successful Rows



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

In case of Subscription Reply, enter N for Abort On Error and Delete Successful Rows.

Table 20 Data Types for Subscription - CustomerItem

Parameter	Data Type
Abort On Error	VARCHAR2(1)
Delete Successful Records	VARCHAR2(1)



Ensure that the Oracle Interface tables are purged before running the transaction for the first time.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- CUSTOMER_ITEM_NUMBER

The following key fields are logged in the TIBCO Log Table indicating success:

- CUSTOMER_ITEM_ID
- LAST_UPDATE_DATE

Error Logging: Errors are logged into the Interface table MTL_CI_INTERFACE with records where the PROCESS_MODE = 2.

CustomerItemXRef

- Application:

Oracle Inventory
- Oracle Import:

INVCIIINTX
- Concurrent Program Name


Import Customer Item Cross References
- Parameters:

Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

 - Abort on Error (Required)
 - Delete Record (Required)

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (`main_adb`). Ensure that you associate valid values through appropriate value sets with each of these parameters.


 - Abort on Error
 - Delete Record
 - Specify `N` for `Abort On Error` and `Y` for `Delete Successful Rows`.



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

Table 21 Data Types for Subscription - CustomerItemXRef

Parameter	Data Type
Abort on Error	VARCHAR2(30)
Delete Record	VARCHAR2(30)



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.



Ensure that the Oracle Interface tables are purged before running the transaction for the first time.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- CUSTOMER_ITEM_NUMBER
- INVENTORY_ITEM_ID

The following key fields are logged in the TIBCO Log Table indicating success:

- CUSTOMER_ITEM_ID
- LAST_UPDATE_DATE

Error Logging:

Errors are logged into the Interface table MTL_CI_XREFS_INTERFACE with records where the PROCESS_MODE = 2.

CycleCountEntries

Application:	Oracle Inventory
Oracle Import:	MTL_CCEOI_IMPORT
Concurrent Program Name	Import cycle count entries from open interface
Parameters:	<p>Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.</p> <ul style="list-style-type: none">• Cycle Count Header Id (Required)• Number of Worker• Commit Point• Error Report Level• Delete Processed Records <p>Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.</p> <ul style="list-style-type: none">• Cycle Count Header Id• Number of Worker• Commit Point• Error Report Level• Delete Processed Records <p>Enter NULL for Cycle Count Header Id.</p>



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

Table 22 Data Types for Subscription - CycleCountEntries

Parameter	Data Type
Cycle Count Header Id	NUMBER
Number of Worker	NUMBER

Table 22 Data Types for Subscription - CycleCountEntries

Parameter	Data Type
Commit Point	NUMBER
Error Report Level	NUMBER
Delete Processed Records	NUMBER



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- CC_ENTRY_INTERFACE_ID

The following key fields are logged in the TIBCO Log Table indicating success:

- CYCLE_COUNT_ENTRY_ID
- LAST_UPDATE_DATE

Error Logging:

Errors are logged into the MTL_CC_INTERFACE_ERRORS table. For each processed record in the Interface Table the errors in the MTL_CC_INTERFACE_ERRORS are related through the REQUEST_ID column.

DemandSchedules

Application:	Oracle Release Management
Oracle Import:	RLMDSP
Concurrent Program Name	Process replenishment counts
Parameters:	<p>Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.</p> <ul style="list-style-type: none">• Schedule Purpose Code• Issue Date From• Issue Date To• Trading Partner From• Trading Partner To• ShipTo Address From• ShipTo Address To• Schedule Reference Number• Issue Warnings for Dropped Parts. <p>Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (<code>main_adb</code>). Ensure that you associate valid values through appropriate value sets with each of these parameters.</p> <ul style="list-style-type: none">• Schedule Purpose Code• Issue Date From• Issue Date To• Trading Partner From• Trading Partner To• ShipTo Address From• ShipTo Address To• Schedule Reference Number

- Issue Warnings for Dropped Parts



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

In case of Subscription with reply the user has to specify those parameters in the TIBCO program's outermost wrapper procedure.

Table 23 Data Types for Subscription - DemandSchedules

Parameter	Data Type
Schedule Purpose Code	NUMBER
Issue Date From	DATE
Issue Date To	DATE
Trading Partner From	VARCHAR2(30)
Trading Partner To	VARCHAR2(30)
ShipTo Address From	VARCHAR2(40)
ShipTo Address To	VARCHAR2(40)
Schedule Reference Number	NUMBER
Issue Warnings for Dropped Parts	VARCHAR2(40)



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- INTERFACE_HEADER_ID
- SCHEDULE_REFERENCE_NUM

The following key fields are logged in the TIBCO Log Table indicating success:

- HEADER_ID
- LAST_UPDATE_DATE

Error Logging: Errors are logged in the RLM_DEMAND_EXCEPTIONS table. For each processed record in the Interface Table, the errors in the RLM_DEMAND_EXCEPTIONS are related through the REQUEST_ID column.

JournalImport

Application: Oracle General Ledger

Oracle Import: GLLEZL

**Concurrent
Program Name** Journal Import

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Source Name (Required)
- Group Id (Required)
- Post Errors To Sequence
- Start Date
- End Date
- Create Summary Journal
- Import Descriptive Flexifields

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (`main_adb`). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Source Name
- Group Id
- Post Errors To Sequence
- Start Date
- End Date
- Create Summary Journal
- Import Descriptive Flexifields



The optional parameters, `Start_Date` and `End_Date` required by the Oracle concurrent program are set as NULL in the call to the concurrent program.

In case of Subscription Reply, enter NULL for both Source Name and Group Id, NULL for Post Errors To Sequence, Create Summary Journal and Import Descriptive Flexifields, NULL for Start Date and End Date.

Table 24 Data Types for Subscription - JournalImport

Parameter	Data Type
Source Name	VARCHAR2(25)
Group Id	NUMBER
Post Errors To Sequence	VARCHAR2(1)
Start Date	VARCHAR2(30)
End Date	VARCHAR2(30)
Create Summary Journal	VARCHAR2(1)
Import Descriptive Flex	VARCHAR2(1)

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- USER_JE_CATEGORY_NAME
- USER_JE_SOURCE_NAME
- SET_OF_BOOKS_ID



In case of success, JE_SOURCE_NAME , JE_CATEGORY_NAME and SET_OF_BOOKS_ID are tracked.

The following key fields are logged in the TIBCO Log Table indicating success:

- JE_HEADER_ID
- NAME
- JE_BATCH_ID
- LAST_UPDATE_DATE

Error Logging: Errors are logged into the GL_INTERFACE table itself. Records which are in error are retained in the table GL_INTERFACE table. Successful records are imported into the Success Table.

MassAdditions

Application: Oracle Assets

Oracle Import: FAMAPT

Concurrent Program Name Mass Additions Post

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Book (Required)
- Mode (Required)

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Book
- Mode

Specify Book as NULL and Mode as NORMAL.



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

In case of Subscription Reply, enter NULL for Book and NORMAL for Mode.

Table 25 Data Types for Subscription - MassAdditions

Parameter	Data Type
Book	VARCHAR2(30)
Mode	VARCHAR2(30)



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- MASS_ADDITION_ID
- BOOK_TYPE_CODE

The following key fields are logged in the TIBCO Log Table indicating success:

- ASSET_NUMBER
- LAST_UPDATE_DATE

Error Logging: Errors are Logged in the Oracle Log File.

MasterSchedule

Application: Oracle Master Scheduling/MRP

Oracle Import: MRCRLF (Planning Manager)

**Concurrent
Program Name** Planning Manager

Parameters: No specific parameters are required.

For a subscription service add the required parameters during configuration of the concurrent program and ensure that appropriate values are populated.

Subscription with reply is not supported.



For transactions involving the Planning and Cost Managers the sequence of steps differs from other PreDefined transactions. The steps are as follows:

1. Run the TIBCO program to export data from the TIBCO Intermediate Table to the Oracle Interface Table.
2. Since the Planning Manager is running, the data in the Oracle Interface Table is processed.
3. Verify that data has been processed by checking the `PROCESS_STATUS` field which changes to the respective value depending on error or success.
4. Once the records are processed, run the `pk_tib_mstschedul_import_new.ErrorLogger` procedure independently to log the error and success messages in the TIBCO Log table.

The procedure `pk_tib_mstschedul_import_new.ErrorLogger` must be registered as a concurrent request.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- `INVENTORY_ITEM_ID`
- `SCHEDULE_DESIGNATOR`
- `ORGANIZATION_ID`
- `LAST_UPDATE_DATE`

The following key fields are logged in the TIBCO Log Table indicating success:

- `MPS_TRANSACTION_ID`
- `LAST_UPDATE_DATE`

Error Logging: Errors are logged into MRP_SCHEDULE_INTERFACE table in the ERROR_MESSAGE column.

Open Replenishment

- Application:** Oracle Inventory
- Oracle Import:** INCRPR
- Concurrent Program Name** Process replenishment counts

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Process Mode (Required)
- Replenishment Header id

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (`main_adb`). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Process Mode
- Replenishment Header id
- Specify Process Mode as 3



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

Table 26 Data Types for Subscription - Open Replenishment

Parameter	Data Type
Process Mode	NUMBER
Replenishment Header id	NUMBER



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- REPLENISHMENT_HEADER_ID

The following key fields are logged in the TIBCO Log Table indicating success:

- REPLENISHMENT_HEADER_ID
- ORGANIZATION_ID
- SUBINVENTORY_CODE
- LAST_UPDATE_DATE

Error Logging: Error flags are populated in the respective interface tables. Each ERROR FLAG value has a distinct error message associated with it.

OpenBudget

- Application:** Oracle Assets
- Oracle Import:** FASCB
- Concurrent Program Name** Capital Budget Interface
- Parameters:**

The Book Type Code parameter is specified in the procedure main of the TIBCO Program.

Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

 - Book Type Code (Required)

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

 - Book Type Code
 - Enter NULL for Book Type Code .



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

Table 27 Data Types for Subscription - OpenBudget

Parameter	Data Type
Book Type Code	VARCHAR2(15)



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- BOOK_TYPE_CODE

The following key fields are logged in the TIBCO Log Table indicating success:

- BUDGET_ID
- LAST_UPDATE_DATE

Error Logging: Errors are logged in the Oracle Application log file.

OpenForecast

Application: Oracle Master Scheduling/MRP

Oracle Import: MRCRLF (Planning Manager)

Concurrent Program Name Planning Manager

Parameters: No specific parameters are required.

For a subscription service add the required parameters during configuration of the concurrent program and ensure that appropriate values are populated.

Subscription with reply is not supported.



For transactions involving the Planning and Cost Managers the sequence of steps differs from other PreDefined transactions. The steps are as follows:

1. Run the TIBCO program to export data from the TIBCO Intermediate Table to the Oracle Interface Table.
2. Since the Planning Manager is running, the data in the Oracle Interface Table is processed.
3. Verify that data has been processed by checking the `PROCESS_STATUS` field which changes to the respective value depending on error or success.
4. Once the records are processed, run the `pk_tib_openforecast_import_new.ErrorLogger` procedure independently to log the error and success messages in the TIBCO Log table.

The procedure `pk_tib_openforecast_import_new.ErrorLogger` must be registered as a concurrent request.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- `INVENTORY_ITEM_ID`
- `FORECAST_DESIGNATOR`
- `ORGANIZATION_ID`
- `LAST_UPDATE_DATE`

The following key fields are logged in the TIBCO Log Table indicating success:

- `TRANSACTION_ID`
- `LAST_UPDATE_DATE`

Error Logging: Errors get logged into MRP_FORECAST_INTERFACE table itself in the ERROR_MESSAGE column.

OpenItem

Application: Oracle Inventory

Oracle Import: INCOIN

**Concurrent
Program Name** Import Items

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Organization Id
- All Organizations
- Validate Items
- Process Items
- Delete Processed Rows
- Process Set
- CreateUpdate Items

In case of Subscription Reply, enter NULL for Organization Id and Process Set (mandatory NULL) and 1 for rest of the parameters.

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Organization Id
- All Organizations
- Validate Items
- Process Items
- Delete Processed Rows
- Process Set
- CreateUpdate Items

While importing data, the adapter requires that relationship keys between the header record and the line records be populated in the `ITEM_NUMBER` and the `ORGANIZATION_ID` fields.

Table 28 Data Types for Subscription - OpenItem

Parameter	Data Type
Organization Id	NUMBER
All Organizations	NUMBER
Validate Items	NUMBER
Process Items	NUMBER
Delete Processed Rows	NUMBER
Process Set	NUMBER
CreateUpdate Items	NUMBER



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- `SEGMENT1/ITEM_NUMBER`

The following key fields are logged in the TIBCO Log Table indicating success:

- `INVENTORY_ITEM_ID`
- `ORGANIZATION_ID`
- `LAST_UPDATE_DATE`

Error Logging:

Errors are logged in the `MTL_INTERFACE_ERRORS` table.

OpenMove

Application: Oracle Work in Process

Oracle Import: WICTMS

Concurrent Program Name WIP Move Transaction Manager

Parameters: No specific parameters are required.

While importing data, the adapter requires that relationship keys between the header record and the line records be populated in the WIP_ENTITY_ID field.

The TRANSACTION_ID field in the Interface tables can be left NULL.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- TRANSACTION_TYPE
- WIP_ENTITY_NAME
- ORGANIZATION_ID/CODE (whichever applicable)



In case of success, TRANSACTION_TYPE cannot be tracked, only Organization_id is tracked.

The following key fields are logged in the TIBCO Log Table indicating success:

- TRANSACTION_ID
- LAST_UPDATE_DATE

The adapter requires that the relationship keys between the incoming header record and the line records (for the TIBCO Intermediate tables) be populated in the WIP_ENTITY_ID field.

Error Logging: Errors are logged in the WIP_TXN_INTERFACE_ERRORS table.



Delete all the records from the WIP_TXN_INTERFACE_ERRORS table so that the non-unique field TRANSACTION_ID in the Error Table can be used to uniquely identify records for a particular run of the program.

OpenResource

Application:	Oracle Bills of Material
Oracle Import:	CMCTCM
Concurrent Program Name	Cost Manager
Parameters:	No specific parameters.

For a subscription service add the required parameters during configuration of the concurrent program and ensure that appropriate values are populated.

Subscription with reply is not supported.



For transactions involving the Planning and Cost Managers the sequence of steps differs from other PreDefined transactions. The steps are as follows:

1. Run the TIBCO program to export data from the TIBCO Intermediate Table to the Oracle Interface Table.
2. Since the Planning Manager is running, the data in the Oracle Interface Table is processed.
3. Verify that data has been processed by checking the PROCESS_STATUS field which changes to the respective value depending on error or success.
4. Once the records are processed, run the `pk_tib_openresource_import_new.ErrorLogger` procedure independently to log the error and success messages in the TIBCO Log table.

The procedure `pk_tib_openresource_import_new.ErrorLogger` must be registered as a concurrent request.



Ensure that the Oracle Interface tables are purged before running the transaction for the first time.

Key Fields	<p>The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:</p> <ul style="list-style-type: none">• WIP_ENTITY_NAME• ORGANIZATION_ID• PRIMARY_ITEM_ID
------------	--

- LAST_UPDATE_DATE

The following key fields are logged in the TIBCO Log Table indicating success:

- TRANSACTION_ID
- LAST_UPDATE_DATE

Error Logging: Errors are logged in the WIP_TXN_INTERFACE_ERRORS. Each header record with an error in WIP_COST_TXN_INTERFACE is related to the corresponding error message in WIP_TXN_INTERFACE_ERRORS through the TRANSACTION_ID column.

OpenTransaction

Application: Oracle Inventory

Oracle Import: INCTCM



In OpenTransaction, in case of Material Movement of Items, the TIBCO Adapter for Oracle Applications only supports inbound Items of type Lots or Lot Serials.

Concurrent Program Name Process transaction interface

Parameters: No specific parameters.

While importing the data, the adapter requires that the relationship keys between the Header record and the Line Records be populated in the TRANSACTION_INTERFACE_ID field.

The following relationships must be maintained when data is sent to the adapter:

- TIB_INT_MTL_TXNS_IFACE.TRANSACTION_INTERFACE_ID must match TIB_INT_MTL_TXN_LOTS_IFACE.TRANSACTION_INTERFACE_ID
- TIB_INT_MTL_TXNS_IFACE.TRANSACTION_INTERFACE_ID must match TIB_INT_MTL_SERIAL_NUM_IFACE.TRANSACTION_INTERFACE_ID
- TIB_INT_MTL_TXN_LOTS_IFACE.TRANSACTION_INTERFACE_ID must match TIB_INT_MTL_SERIAL_NUM_IFACE.SERIAL_TRANSACTION_TEMP_ID
- TIB_INT_MTL_TXNS_IFACE.TRANSACTION_INTERFACE_ID must match TIB_INT_CST_COMP_SNAP_IFACE.TRANSACTION_INTERFACE_ID

While populating the Oracle Interface tables from the TIBCO Intermediate tables, the TRANSACTION_INTERFACE_ID column is populated with the value of mtl_material_transactions_s.NEXTVAL in both the header as well as the detail tables.



It is recommended that the TRANSACTION_INTERFACE_ID field in the Header as well as the detail tables be populated with the Sequence value MTL_MATERIAL_TRANSACTION_S.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- TRANSACTION_INTERFACE_ID



This parameter cannot be tracked in the case of success.

The following key fields are logged in the TIBCO Log Table indicating success:

- TRANSACTION_ID
- LAST_UPDATE_DATE

Error Logging: Errors are logged into MTL_TRANSACTIONS_INTERFACE for Header and Cost records. Items with Lost Control are logged in the MTL_TRANSACTION_LOTS_INTERFACE table. Items with Serial and Lost Serial Control are logged in the MTL_SERIAL_NUMBERS_INTERFACE table.

OrderEntry

Application:	Oracle Order Management
Oracle Import:	OEIMP
Concurrent Program Name	Order Import
Parameters:	<p>Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.</p> <ul style="list-style-type: none">• Order Source• Original System Document Ref• Operation Code• Validate Only (Required)• Debug Level• Number Of Order Import Instances <p>Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (<code>pk_tib_oe_import.main_adb</code>). Ensure that you associate valid values through appropriate value sets with each of these parameters.</p> <ul style="list-style-type: none">• Order Source• Original System Document Ref• Operation Code• Validate Only• Debug Level• Number Of Order Import Instances• Enter N for Validate Only



The optional parameters (which are required by the Oracle Application concurrent program) default to `NULL` in the call to the concurrent program. If needed you can modify the optional values.

Table 29 Data Types for Subscription - OrderEntry

Parameter	Data Type
Order Source	NUMBER
Original System Document Ref	VARCHAR(50)
Operation Code	VARCHAR2(30)
Validate Only	VARCHAR2(1)
Debug Level	NUMBER
Number Of Order Import Instances	NUMBER



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application :

- ORDER_SOURCE_ID
- ORIG_SYS_DOCUMENT_REF
- CHANGE_SEQUENCE
- ORIGINAL_SYS_DOCUMENT_LINE_REF

The following key fields are logged in the TIBCO Log Table indicating success:

- ORDER_NUMBER
- LAST_UPDATE_DATE

Error Logging:

Errors are logged in the OE_PROCESSING_MSGS_TL table. Each record with an error has a REQUEST_ID, which is populated with a set of TRANSACTION_ID's from OE_PROCESSING_MSGS table. An error message is logged for each TRANSACTION_ID in the OE_PROCESSING_MSGS_TL table.

PayablesOpen

Application:	Oracle Payables
Oracle Import:	APXIIMPT
Concurrent Program Name	Expense Report Import
Parameters:	<p>Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.</p> <ul style="list-style-type: none">• Source (Required)• Group• Invoice Batch Name• Hold Name• Hold Reason• GL_date• Purge• Trace Switch• Debug Switch• Summarize Report• Commit Batch Size <p>Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.</p> <ul style="list-style-type: none">• Source• Group• Invoice Batch Name• Hold Name• Hold Reason• GL_date• Purge• Trace Switch

- Debug Switch
- Summarize Report
- Commit Batch Size

In case of Subscription Reply, enter NULL for both Source and Group, N/A for Invoice Batch Name, 'N' for Purge, Trace Switch, Debug Switch, Summarize Report and NULL for the rest.

Table 30 Data Types for Subscription - PayablesOpen

Parameter	Data Type
Source	VARCHAR2(80)
Group	VARCHAR2(80)
Invoice Batch Name	VARCHAR2(30)
Hold Name	VARCHAR2(25)
Hold Reason	VARCHAR2(80)
GL Date	DATE
Purge	VARCHAR2(1)
TraceSwitch	VARCHAR2(1)
Debug Switch	VARCHAR2(1)
Summarize Report	VARCHAR2(1)
Commit Batch Size	VARCHAR2(1)



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

While importing the data the adapter requires that relationship keys between the header record and the line records be populated in the INVOICE_ID field. The following relations must be maintained when data is sent to the adapter:

- TIB_INT_AP_INV_IFACE.INVOICE_ID must match
TIB_INT_AP_INV_LINES_IFACE.INVOICE_ID

While populating the Oracle Interface tables from the TIBCO Intermediate tables, the `INVOICE_ID` column is populated with the value of `AP_INVOICES_INTERFACE_S.NEXTVAL` in both the header as well as the detail tables. The value of the `INVOICE_LINE_ID` in the detail table is populated with the `AP_INVOICE_LINES_INTERFACE_S.NEXTVAL`.



It is recommended that the `INVOICE_ID` field in the Header as well as the detail table be populated with the Sequence value `AP_INVOICES_INTERFACE_S` and the `INVOICE_LINE_ID` field in the Detail table with the Sequence Value `AP_INVOICE_LINES_INTERFACE_S`.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- `INVOICE_ID`
- `SOURCE`
- `VENDOR_ID`
- `INVOICE_LINE_ID`

`INVOICE_LINE_ID` can be tracked only when there are line records.

The following key fields are logged in the TIBCO Log Table indicating success:

- `INVOICE_NUM`
- `LAST_UPDATE_DATE`

Error Logging:

Errors are logged into the `AP_INTERFACE_REJECTIONS` table.

PeriodicCost

- Application:** Oracle Bills Of Material
- Oracle Import:** CMCPIM
- Concurrent Program Name** Period Cost Import Manager

Parameters: Subscription — Add the following parameter during configuration of the concurrent program and ensure that appropriate values are populated.

- Delete Rows

Subscription with Reply—Specify the following parameter in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Delete Rows

In case of Subscription Reply, enter 2 for Delete Rows.

Table 31 Data Types for Subscription - PeriodicCost

Parameter	Data Type
Delete Rows	NUMBER



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

While importing data, the adapter requires that relationship keys between the header record and line Records be populated in the `INTERFACE_HEADER_ID` field. The following relation must be maintained when data is sent to the adapter:

- `TIB_INT_CST_PC_ITM_IFACE.INTERFACE_HEADER_ID` must match `TIB_INT_CST_PC_DET_IFACE.INTERFACE_HEADER_ID`

While populating the Oracle Interface tables from the TIBCO Intermediate tables, the `INTERFACE_HEADER_ID` column is populated with the value of `CST_PC_ITEM_COST_INTERFACE_S.NEXTVAL` in both the header as well as the detail tables.

Also the value of the `INTERFACE_LINE_ID` in the detail table is populated with the `CST_PC_COST_DET_INTERFACE_S.NEXTVAL`.



It is recommended that the `INTERFACE_HEADER_ID` field in the Header as well as the detail table be populated with the Sequence value `CST_PC_ITEM_COST_INTERFACE_S` and the `INTERFACE_LINE_ID` field in the Detail table with the Sequence Value `CST_PC_COST_DET_INTERFACE_S`.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- `INVENTORY_ITEM_ID`

The following key fields are logged in the TIBCO Log Table indicating success:

- `COST_LAYER_ID`
- `LAST_UPDATE_DATE`

Error Logging:

Errors are logged into the tables `CST_PC_ITEM_COST_INTERFACE` and `CST_PC_COST_DET_INTERFACE` where records have `PROCESS_FLAG = 3`.

PurchasingOpenDocs

Application: Oracle Purchasing

Oracle Import: POXPDOI

**Concurrent
Program Name** Import Price Catalogs

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Default Buyer
- Document Type (Required)
- Document SubType
- Create or Update Items
- Create Sourcing Rules (Required)
- Approval Status
- Release Generation Method
- Batch Id
- Global Agreement

Subscription with Reply — Specify the following parameters in the TIBCO program's outermost wrapper procedure (`main_adb`). Ensure that you associate valid values through appropriate value sets with each of these parameters. Enter NULL for Transaction Type and a default NO for Global Agreement.

- Default Buyer
- Document Type
- Document SubType
- Create or Update Items
- Create Sourcing Rules
- Approval Status
- Release Generation Method
- Batch Id
- Global Agreement

Enter NULL for Document Type and Y for Create Sourcing Rules



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

Table 32 Data Types for Subscription - PurchasingOpenDocs

Parameter	Data Type
Default Buyer	NUMBER
Document Type	VARCHAR2(40)
Document SubType	VARCHAR2(40)
Create or Update Items	VARCHAR2(25)
Create Sourcing Rules	VARCHAR2(25)
Approval Status	VARCHAR2(40)
Release Generation Method	VARCHAR2(25)
Batch Id	NUMBER
Global Agreement	VARCHAR2(80)



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- PO_HEADER_ID

The following key fields are logged in the TIBCO Log Table indicating success:

- PO_NUMBER, LAST_UPDATE_DATE

Error Logging: Errors are logged in the PO_INTERFACE_ERRORS table. For each processed record in the Interface Table the errors in the PO_INTERFACE_ERRORS are related through the REQUEST_ID column.



Delete all entries in the PO_INTERFACE_ERRORS table for each run of the TIBCO program to avoid repeated logging of errors for each processed header during subsequent runs of the import program.

RecCustomer

Application: Oracle Receivables

Oracle Import: RACUST

Concurrent Program Name Customer Interface

Parameters: No specific parameters.



Ensure that the Oracle Interface tables are purged before running the transaction for the first time.

Key Fields The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- ORIG_SYSTEM_CUSTOMER_REF

The following key fields are logged in the TIBCO Log Table indicating success:

- PARTY_ID
- PARTY_NUMBER
- LAST_UPDATE_DATE

Error Logging: Errors are logged into the respective Interface tables.



Only the Error Codes are logged into the Error Tables. For details, please refer to the *Oracle Receivables User's Guide*.

Receiving

Application: Oracle Purchasing

Oracle Import: RVCTP

Concurrent Program Name Receiving Transaction Processor

Architecture: In case of the Receiving transaction, the adapter inserts records into the TIBCO Intermediate tables TIB_INT_RCV_HEADERS_IFACE and TIB_INT_RCV_TXNS_IFACE. The TIBCO Validation procedure then picks up the records and inserts them into the Header source table PO.RCV_SHIPMENT_HEADERS and the Line Interface table PO.RCV_TRANSACTIONS_INTERFACE. Only after this is the Oracle Concurrent program called. The reason for such behavior is that records populated in the Header Interface table do not get picked up. Hence the approach of directly inserting into the Source table. Refer to the tar 2257754.996 for further details.

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Transaction Processing Mode
- Group id

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Transaction Processing Mode
- Group id

In case of Subscription Reply, enter NULL for Group Id and BATCH for Transaction Processing Mode.

Table 33 Data Types for Subscription - Receiving

Parameter	Data Type
Transaction Processing Mode	VARCHAR2(50)
Group id	NUMBER



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.



It is recommended that the SHIPMENT_HEADER_ID field in the Header as well as the detail table be populated with the Sequence value RCV_SHIPMENT_HEADERS_S and the INTERFACE_TRANSACTION_ID field in the Detail table with the Sequence Value RCV_TRANSACTIONS_INTERFACE_S.

Key Fields



In case of the Receiving transaction, since the header records are directly inserted into the Source tables, the Errors and Success cases for the Header are not logged into the Log table. Errors are only logged for records in the Detail table RCV_TRANSACTIONS_INTERFACE. The following is a key field in the inbound data that is logged into the TIBCO Log Table. This field is used to track records coming into the Oracle Application:

HEADER_INTERFACE_ID

Purge the Error Table each time you run this program.

The adapter requires that the relationship keys between the header record and the line records be populated in the SHIPMENT_HEADER_ID field.

The adapter inserts the data into the RCV_SHIPMENT_HEADERS and RCV_TRANSACTIONS_INTERFACE tables. Records will not be picked up if they are populated in the RCV_HEADERS_INTERFACE and the RCV_TRANSACTIONS_INTERFACE.

While populating the Oracle Interface tables from the TIBCO Intermediate tables, the `SHIPMENT_HEADER_ID` column is populated with the value of `rcv_shipment_headers_s.NEXTVAL` in both the Header as well as the detail tables.

The value of the `INTERFACE_TRANSACTION_ID` in the detail table is populated with the `RCV_TRANSACTIONS_INTERFACE_S.NEXTVAL`.

Error Logging: Errors are logged into the `PO_INTERFACE_ERRORS` table.

Requisition


Application:	Oracle Purchasing
Oracle Import:	REQIMPORT
Concurrent Program Name	Requisition Import
Parameters:	<p>Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.</p> <ul style="list-style-type: none">• Interface Source Code• Batch Id• Group By (Required)• Last Requisition Number• Multi Distributions (Required)• Initiate Requisition Approval After Requisition Import (Required) <p>Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (<code>main_adb</code>). Ensure that you associate valid values through appropriate value sets with each of these parameters.</p> <ul style="list-style-type: none">• Interface Source Code• Batch Id• Group By• Last Requisition Number• Multi Distributions• Initiate Requisition Approval After Requisition Import <p>Enter All for Group By and Y or N for Multi Distributions as well as Initiate Requisition Approval After Requisition Import.</p>
	<p>The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.</p>

Table 34 Data Types for Subscription - Requisition

Parameter	Data Type
Interface Source Code	VARCHAR2(25)
Batch Id	NUMBER
Group By	VARCHAR2(25)
Last Requisition Number	VARCHAR2(25)
Multi Distributions	VARCHAR2(25)
Initiate Requisition Approval After Requisition Import	VARCHAR2(30)



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- REQ_DIST_SEQUENCE_ID

The following key fields are logged in the TIBCO Log Table indicating success:

- REQUISITION_HEADER_ID
- LAST_UPDATE_DATE



The header fields for inbound data cannot be seen for records marked as successful.

Error Logging:

Errors are logged in the PO_INTERFACE_ERRORS table and TRANSACTION_ID is the key relating the Interface records to the error table entries.

WorkOrder

Application: Oracle Work in Process

Oracle Import: WICMLP

**Concurrent
Program Name** WIP Mass Load

Parameters: Subscription — Add the following parameters during configuration of the concurrent program and ensure that appropriate values are populated.

- Group Id (Required)
- Validation Level
- Print Report (Required)

Subscription with Reply—Specify the following parameters in the TIBCO program's outermost wrapper procedure (main_adb). Ensure that you associate valid values through appropriate value sets with each of these parameters.

- Group Id
- Validation Level
- Print Report

Enter Null for Group Id and 1 for Print Report.



The optional parameters (which are required by the Oracle Application concurrent program) default to NULL in the call to the concurrent program. If needed you can modify the optional values.

Table 35 Data Types for Subscription - WorkOrder

Parameter	Data Type
Group Id	NUMBER
Validation Level	NUMBER
Print Report	NUMBER



The variable data have been listed in the previous table as expected by the TIBCO procedure. You may have to map value sets to the parameter values set at the front end, in order to translate the parameter values to actual values to be passed in the TIBCO program.



Ensure that the Oracle Interface tables are purged before running the transaction for the first time.

Key Fields

The following are key fields in the inbound data that are logged into the TIBCO Log Table. These fields are used to track records coming into the Oracle Application:

- HEADER_ID

The following key fields are logged in the TIBCO Log Table indicating success:

- WIP_ENTITY_ID, LAST_UPDATE_DATE

For a subscription with reply, specify the required parameters in the outermost TIBCO wrapper procedure `pk_tib_oe_import.main_adb`.

Error Logging:

Errors are logged in the `WIP_INTERFACE_ERRORS` table. The `INTERFACE_ID` from the header is logged along with each error entry.

Oracle Application APIs

The following Oracle Application APIs are supported by the adapter:

- Applicant
- Auto Create Deliveries
- Auto Create DelTrip
- Create Update Deliveries
- Create Update Freight Cost
- Bills Of MaterialAPI
- Delete Freight Costs
- Delivery Action
- Detail To Delivery
- Engineering Change Order
- Employee
- Interaction
- Pick Confirm
- Split Line
- Update Shipping Attributes
- Valid Freight Cost Type

Parameters: No specific parameters are required.

For a subscription service, register the TIBCO concurrent program.

For a subscription with reply, no parameters need to be set in TIBCO wrapper procedure `pk_tib_oe_import.main_adb`.

Error Logging: Oracle Errors are directly logged by the TIBCO concurrent program into the TIBCO log table.

An `IMPORT SUCCESS` message is logged on successful import. Key fields are not tracked and have to be verified by checking the appropriate Oracle source tables.

Appendix C **Master Mapping Tables**

This appendix lists the tables, scripts, triggers, view names, and other related information for each PreDefined transaction included in TIBCO Adapter for Oracle Applications.

Topics

- [Publication Mapping Details, page 358](#)
- [API Subscription Mapping Details, page 365](#)

Publication Mapping Details

The table below lists details regarding the PreDefined Outbound Publisher transactions provided by the adapter.

Table 36 Publication Mapping Tables

Transaction Name	Type of Table	Name of Table	Materialized View Log Names
Engg BOM	Header Table	BOM.BOM_BILL_OF_MATERIALS (assembly_type = 2) BOM.BOM_STRUCTURES_B (in Oracle Applications 11.5.10)	MLOG\$_BOM_BILL_OF_MATERIAL
	Details Table(s)	BOM.BOM_INVENTORY_COMPONENTS (BOM_BILL_OF_MATERIALS – bill_sequence_id) BOM.BOM_COMPONENTS_B (in Oracle Applications 11.5.10)	—
		BOM.BOM_SUBSTITUTE_COMPONENTS (BOM_INVENTORY_COMPONENTS – component_sequence_id)	MLOG\$_BOM_SUBSTITUTE_COMPO
		BOM.BOM_REFERENCE_DESIGNATORS (BOM_INVENTORY_COMPONENTS – component_sequence_id)	MLOG\$_BOM_REFERENCE_DESIGN
PickDetails	Header Table	INV.MTL_TXN_REQUEST_HEADERS	MLOG\$_MTL_TXN_REQUEST_HEAD
	Detail Table(s)	INV.MTL_TXN_REQUEST_LINES (INV.MTL_TXN_REQUEST_HEADERS – header_id)	MLOG\$_MTL_TXN_REQUEST_LINE
GLBalances	Header Table	GL.GL_BALANCES	MLOG\$_GL_BALANCES
	Detail Table(s)	N/A	N/A
Journals	Header Table	GL.GL_JE_HEADERS	MLOG\$_GL_JE_HEADERS
	Detail Table(s)	GL.GL_JE_LINES (GL.GL_JE_HEADERS – header_id)	MLOG\$_GL_JE_LINES

Transaction Name	Type of Table	Name of Table	Materialized View Log Names
Items	Header Table	INV.MTL_SYSTEM_ITEMS_B	MLOG\$_MTL_SYSTEM_ITEMS_B
	Detail Table(s)	N/A	N/A
ItemCategory	Header Table	INV.MTL_ITEM_CATEGORIES	MLOG\$_MTL_ITEM_CATEGORIES
	Detail Table(s)	N/A	N/A
On-Hand Quantity	Header Table	INV.MTL_ONHAND_QUANTITIES MTL_ONHAND_QUANTITIES_DETAIL (renamed in Oracle Applications 11.5.8 and 11.5.9)	MLOG\$_MTL_ONHAND_QUANTITIES
	Detail Table(s)	N/A	N/A
Mfg BOM	Header Table	BOM.BOM_BILL_OF_MATERIALS (assembly_type = 1) BOM.BOM_STRUCTURES_B (in Oracle Applications 11.5.10)	MLOG\$_BOM_BILL_OF_MATERIAL
	Detail Table(s)	BOM.BOM_INVENTORY_COMPONENTS BOM.BOM_COMPONENTS_B (in Oracle Applications 11.5.10) (BOM_BILL_OF_MATERIALS – bill_sequence_id)	—
		BOM.BOM_SUBSTITUTE_COMPONENTS (BOM_INVENTORY_COMPONENTS – component_sequence_id)	MLOG\$_BOM_SUBSTITUTE_COMPO
		BOM.BOM_REFERENCE_DESIGNATORS (BOM_INVENTORY_COMPONENTS – component_sequence_id)	MLOG\$_BOM_REFERENCE_DESIGN

Transaction Name	Type of Table	Name of Table	Materialized View Log Names
BOM Revisions BOM	Header Table	BOM.BILL_OF_MATERIALS	MLOG\$_BOM_BILL_OF_MATERIAL
	Detail Table(s)	INV.MTL_ITEM_REVISIONS (BOM.BOM_BILL_OF_MATERIALS – inventory_item_id, organization_id) BOM.BOM_STRUCTURES_B (in Oracle Applications 11.5.10)	MLOG\$_MTL_ITEM_REVISIONS
		INV.MTL_ITEM_REVISIONS has been changed to INV.MTL_ITEM_REVISIONS_B in Oracle Applications 11.5.9	MLOG\$_MTL_ITEMS_REVISIONS_B
BOM Routings	Header Table	BOM.BOM_OPERATIONAL_ROUTINGS	MLOG\$_BOM_OPERATIONAL_ROUT
	Detail Table(s)	BOM.BOM_OPERATION_SEQUENCES (BOM.BOM_OPERATIONAL_ROUTINGS - routing_sequence_id)	MLOG\$_BOM_OPERATION_SEQUEN
		BOM.BOM_OPERATION_RESOURCES (BOM.BOM_OPERATION_SEQUENCES - operation_sequence_id)	MLOG\$_BOM_OPERATION_RESOUR
PurchaseOrders	Header Table	PO.PO_HEADERS_ALL	MLOG\$_PO_HEADERS_ALL
	Detail Table(s)	PO.PO_LINES_ALL (PO_HEADERS_ALL – po_header_id)	MLOG\$_PO_LINES_ALL
		PO.PO_LINE_LOCATIONS_ALL (PO_LINES_ALL – po_line_id)	MLOG\$_PO_LINE_LOCATIONS_ALL
		PO.PO_DISTRIBUTIONS_ALL (PO_LINE_LOCATIONS_ALL – po_line_location_id)	MLOG\$_PO_DISTRIBUTIONS_ALL
Suppliers	Header Table	PO.PO_VENDORS	MLOG\$_PO_VENDORS
	Detail Table(s)	PO.PO_VENDOR_SITES_ALL(PO.PO_VE NDORS_vendor_id)	MLOG\$_PO_VENDOR_SITES_ALL

Transaction Name	Type of Table	Name of Table	Materialized View Log Names
SalesOrders	Header Table	ONT.OE_ORDER_HEADERS_ALL	MLOG\$_OE_ORDER_HEADERS_ALL
	Details Table(s)	ONT.OE_ORDER_LINES_ALL (ONT.OE_ORDER_HEADERS_ALL – header_id)	MLOG\$_OE_ORDER_LINES_ALL
		ONT.OE_LOT_SERIAL_NUMBERS (ONT.OE_ORDER_LINES_ALL – line_id)	MLOG\$_OE_LOT_SERIAL_NUMBER
		ONT.OE_SALES_CREDITS (ONT.OE_ORDER_HEADERS_ALL – header_id)	MLOG\$_OE_SALES_CREDITS
		ONT.OE_PRICE_ADJUSTMENTS (ONT.OE_ORDER_HEADERS_ALL – header_id)	MLOG\$_OE_PRICE_ADJUSTMENTS
Invoices	Header Table	AP.AP_INVOICES_ALL	MLOG\$_AP_INVOICES_ALL
	Details Table(s)	AP.AP_INVOICE_DISTRIBUTIONS_ALL (AP.AP_INVOICES_ALL – invoice_id)	MLOG\$_AP_INVOICE_DISTRIBUT
Payments/Checks	Header Table	AP.AP_CHECKS_ALL	MLOG\$_AP_CHECKS_ALL
	Details Table(s)	AP.AP_INVOICE_PAYMENTS_ALL (AP.AP_CHECKS_ALL – check_id)	MLOG\$_AP_INVOICE_PAYMENTS_

Transaction Name	Type of Table	Name of Table	Materialized View Log Names
Customers	Header Table	AR.HZ_PARTIES	MLOG\$_HZ_PARTIES
	Detail Table(s)	AR.HZ_CONTACT_POINTS	MLOG\$_HZ_CONTACT_POINTS
		AR.HZ_PARTY_SITES	MLOG\$_HZ_PARTY_SITES
		AR.HZ_LOCATIONS	—
		AR.HZ_CUST_ACCOUNT_ROLES	MLOG\$_HZ_CUST_ACCOUNT_ROLE
		AR.HZ_ROLE_RESPONSIBILITY	MLOG\$_HZ_ROLE_RESPONSIBILI
		AR.HZ_CUST_ACCOUNTS	MLOG\$_HZ_CUST_ACCOUNTS
		AR.HZ_CUST_ACCT_RELATE_ALL	MLOG\$_HZ_CUST_ACCT_RELATE_
		AR.HZ_CUSTOMER_PROFILES	MLOG\$_HZ_CUSTOMER_PROFILES
		AR.HZ_CUST_PROFILE_CLASSES	—
		AR.HZ_CUST_PROFILE_AMTS	MLOG\$_HZ_CUST_PROFILE_AMTS
		AR.HZ_CUST_ACCT_SITES_ALL	MLOG\$_HZ_CUST_ACCT_SITES_A
		AR.HZ_CUST_SITE_USES_ALL	MLOG\$_HZ_CUST_SITE_USES_AL
		AP.AP_BANK_ACCOUNT_USES_ALL	MLOG\$_AP_BANK_ACCOUNT_USES
		AP.AP_BANK_ACCOUNTS_ALL	—
		AP.AP_BANK_BRANCHES	—

Transaction Name	Type of Table	Name of Table	Materialized View Log Names
ARTransactions	Header Table	AR.RA_CUSTOMER_TRX_ALL	MLOG\$_RA_CUSTOME R_TRX_ALL
	Detail Table(s)	AR.RA_CUSTOMER_LINES_ALL (AR.RA_CUSTOMER_TRX_ALL – customer_trx_id)	MLOG\$_RA_CUSTOME R_TRX_LINE
EnggChangeNot ification	Header Table	ENG.ENG_ENGINEERING_CHANGES	MLOG\$_ENG_ENGINE ERING_CHAN
	Detail Table(s)	ENG.ENG_CHANGE_ORDER_REVISION S (ENG.ENG_ENGINEERING_CHANGES –organization_id,change_notice)	MLOG\$_ENG_CHANGE _ORDER_REV
		ENG.ENG_REVISIED_ITEMS (ENG.ENG_ENGINEERING_CHANGES –organization_id,change_notice),	MLOG\$_ENG_REVISIED _ITEMS
		ENG.ENG_CURRENT_SCHEDULED_DA TES (ENG.ENG_REVISIED_ITEMS –revised_item_sequence_id)	MLOG\$_ENG_CURREN T_SCHEDULE
		BOM.BOM_INVENTORY_COMPONENTS BOM.BOM_COMPONENTS_B (in Oracle Applications 11.5.10) (ENG.ENG_REVISIED_ITEMS –revised_item_sequence_id)	MLOG\$_BOM_INVENT ORY_COMPON
		BOM.BOM_COMPONENTS_B for Oracle Applications 11.5.10.	MLOG\$_BOM_INVENT ORY_COMPON
		BOM.BOM_SUBSTITUTE_COMPONENTS (BOM.BOM_INVENTORY_COMPONENT S –component_sequence_id) BOM.BOM_COMPONENTS_B (in Oracle Applications 11.5.10)	MLOG\$_BOM_SUBSTIT UTE_COMPO
		BOM.BOM_REFERENCE_DESIGNATORS (BOM.BOM_INVENTORY_COMPONENT S –component_sequence_id) BOM.BOM_COMPONENTS_B (in Oracle Applications 11.5.10)	MLOG\$_BOM_REFERERE NCE_DESIGN

Transaction Name	Type of Table	Name of Table	Materialized View Log Names
ApplicantPub	Header Table	HR.PER_ALL_PEOPLE_F	MLOG\$_PER_ALL_PEO PLE_F
	Detail Table(s)	HR.PER_PERIODS_OF_SERVICE (HR.PER_ALL_PEOPLE_F-person_id)	MLOG\$_PER_PERIODS_ OF_SERVI
		HR.PER_ALL_ASSIGNMENTS_F (HR.PER_PERIODS_OF_SERVICE-person_ id,period_of_service_id)	MLOG\$_PER_ALL_ASSI GNMENTS_
		HR.PER_APPLICATIONS (HR.PER_ALL_PEOPLE_F-person_id)	MLOG\$_PER_APPLICA TIONS
		HR.PER_ADDRESSES (HR.PER_ALL_PEOPLE_F-person_id)	MLOG\$_PER_ADDRESS ES
EmployeePub	Header Table	HR.PER_ALL_PEOPLE_F	MLOG\$_PER_ALL_PEO PLE_F
	Detail Table(s)	HR.PER_PERIODS_OF_SERVICE (HR.PER_ALL_PEOPLE_F-person_id)	MLOG\$_PER_PERIODS_ OF_SERVI
		HR.PER_ALL_ASSIGNMENTS_F (HR.PER_PERIODS_OF_SERVICE-person_ id,period_of_service_id)	MLOG\$_PER_ALL_ASSI GNMENTS_
		HR.PER_APPLICATIONS (HR.PER_ALL_PEOPLE_F-person_id)	MLOG\$_PER_APPLICA TIONS
		HR.PER_ADDRESSES (HR.PER_ALL_PEOPLE_F-person_id)	MLOG\$_PER_ADDRESS ES
InteractionPub	Header Table	JTF.JTF_IH_INTERACTIONS	MLOG\$_JTF_IH_INTER ACTIONS
	Detail Table(s)	JTF.JTF_IH_ACTIVITIES (JTF.JTF_IH_INTERACTIONS-INTERACTI ON_ID)	MLOG\$_JTF_IH_ACTIVI TIES

API Subscription Mapping Details

The following table lists details regarding the API Subscription transactions provided by the adapter.

Table 37 API Subscription Mapping Details

Interface Name	Table Name	Schema Name	Intermediate Table Names	Package Name
ApplicantSub	HR.PER_ALL_PE OPLE_F	HR	TIB_INT_HRMS _APPLICANT	pk_tib_hr_appl icant
AutoCreate Deliveries	WSH.WSH_DELIVE RY_ASSIGNMENTS , WSH.WSH_DELIVE RY_DETAILS	WSH	TIB_INT_WSH_ AUTOCREAT_ DELIV	pk_tib_autocre a_import
AutoCreateDelive riesTrip	DELIVERIES, WSH.WSH_DELIVE RY_DETAILS, WSH.WSH_DELIVE RY_ASSIGNMENTS , WSH.WSH_TRIPS, WSH.WSH_TRIP_S TOPS, WSH.WSH_DELIVE RY_LEGS	WSH	TIB_INT_WSH_ ACREAT_DELT RIP	pk_tib_acrdeltr _import

Table 37 API Subscription Mapping Details

Interface Name	Table Name	Schema Name	Intermediate Table Names	Package Name
Bills of Material	BOM_BILL_OF_MATERIALS_INTERFACE	BOM, INV	TIB_INT_BOM_BILLS_OF_MATERIAL	pk_tib_bom_import
	MTL_ITEM_REVISIONS_INTERFACE		TIB_INT_MTL_ITEM_REV_INTERFACE	
	BOM_INVENTORY_COMPS_INTERFACE		TIB_INT_BOM_INV_COMP	
	BOM_REF_DESGS_INTERFACE		TIB_INT_BOM_REF_DESG	
	BOM_SUB_COMPS_INTERFACE		TIB_INT_BOM_SUBS_COMP	
			TIB_INT_BOM_API_HDR	pk_tib_billsapi_import
			TIB_INT_BOM_API_INV_COMP	
			TIB_INT_BOM_API_ITM_REV	
			TIB_INT_BOM_API_REF_DESG	
			TIB_INT_BOM_API_SUBS_COMP	
Create Update Delivery	WSH.WSH_NEW_DELIVERIES	WSH	TIB_INT_BOM_API_COMP_OPS for Oracle Applications 11.5.9	pk_tib_crupdel_import
			TIB_INT_WSH_NEW_DELIV	

Table 37 API Subscription Mapping Details

Interface Name	Table Name	Schema Name	Intermediate Table Names	Package Name
Create Update Freight Costs	WSH.WSH_FREIGHT_COSTS	WSH	TIB_INT_CR_UPD_FRT_COSTS	pk_tib_crupdfrcst_import
Delete Freight Costs	WSH.WSH_FREIGHT_COSTS	WSH	TIB_INT_DELETE_FRT_COSTS	pk_tib_delfrtcost_import
Delivery Action	WSH.WSH_NEW_DELIVERIES	WSH	TIB_INT_DELIVERY_ACTION_API	pk_tib_delvaction_import
Detail To Delivery	WSH.WSH_DELIVERY_ASSIGNMENTS	WSH	TIB_INT_WSH_DETAIL_TO_DELIV	pk_tib_dettodelimport
EmployeeSub	HR.PER_ALL_PEOPLE_F	HR	TIB_INT_HRMS_EMP	pk_tib_hr_employee

Table 37 API Subscription Mapping Details

Interface Name	Table Name	Schema Name	Intermediate Table Names	Package Name
Engineering Change Order API	ENG.ENG_ENGINEERING_CHANGES	ENG	TIB_INT_ECO_ENGR_CHANGE	pk_tib_engchord_import
			TIB_INT_ECO_ORD_REVISIONS	
			TIB_INT_ECO_REVERSE_ITM	
			TIB_INT_ECO_REV_COMP	
			TIB_INT_ECO_REV_OPERATION	
			TIB_INT_ECO_REF_DESIGN	
			TIB_INT_ECO_SUBS_COMP	
			TIB_INT_ECO_OPN_RESOURCE	
			TIB_INT_ECO_SUBOP_RESOURCE	
			TIB_INT_ECO_CHNG_LIN for Oracle Applications 11.5.9	
Pick Confirm	INV.MTL_MATERIAL_TRANSACTION_TEMP	INV	TIB_INT_MTL_MTRL_TRANSACTION_TEMP	pk_tib_pickconf_import

Table 37 API Subscription Mapping Details

Interface Name	Table Name	Schema Name	Intermediate Table Names	Package Name
Split Line	WSH.WSH_DELIVERY_DETAILS and WSH.WSH_DELIVERY_ASSIGNMENTS	WSH	TIB_INT_WSH_SHIPLINE	pk_tib_splitlin_import
Update Shipping Attributes	WSH.WSH_DELIVERY_DETAILS	WSH	TIB_INT_WSH_DELDET TIB_INT_WSH_SERNG for Oracle Applications 11.5.9 and onwards	pk_tib_deldetup_import
Validate Freight Cost Types	WSH.WSH_FREIGHT_COST_TYPES	WSH	TIB_INT_VAL_FRT_CST_TYPE	pk_tib_valfrcty_import
Interaction	JTF.JTF_IH_INTERACTIONS	WSH	TIB_INT_CRM_INTERACTION TIB_INT_CRM_ACTIVITY	pk_tib_crm_interaction

Non-API Subscription Mapping Details

The table below lists details regarding the Non-API Subscriber transactions provided by the adapter.

Table 38 Non-API Subscription Mapping Details

Interface Name	Interface Table Name	Schema Name	Intermediate Table Names	Package Name	TIBCO Sequence Name
AutoInvoicing	RA_INTERFACE_LINES_ALL	AR	TIB_INT_RA_LINES_IFACE	pk_tib_autoinvo_importh	TIB_INT_RA_LINES_IFACE_S
	RA_INTERFACE_SALESCREDITSDITS_ALL		TIB_INT_RA_SALESCREDITD_IFACE		
	RA_INTERFACE_DISTRIBUTIONS_ALL		TIB_INT_RA_DISTRIBUTUT_IFACE		
AutoLockBox	AR_PAYMENTS_INTERFACE_ALL	AR	TIB_INT_AR_PAYMNT_IFACE	pk_tib_auto_lockbox_import	TIB_INT_AR_PAYMENT_IFACE_S

Table 38 Non-API Subscription Mapping Details

Interface Name	Interface Table Name	Schema Name	Intermediate Table Names	Package Name	TIBCO Sequence Name
BOM Business Object	BOM.BOM_BILL_OF_MATERIALS_INTERFACE	BOM, INV	TIB_INT_BOM_BILLS_OF_MATERIAL	pk_tib_bom_import	TIB_INT_BOM_BILLS_OF_MATERIALS
	BOM.BOM_INVENTORY_COMPS_INTERFACE		TIB_INT_BOM_INV_COMP		
	INV.MTL_ITEM_REVISIONS_INTERFACE		TIB_INT_MTL_ITEM_REV1_IFACE		
	BOM.BOM_SUB_COMPS_INTERFACE		TIB_INT_BOM_SUBS_COMP		
	BOM.BOM_REF_DESGS_INTERFACE		TIB_INT_BOM_REF_DESG		
Budget Upload	GL_BUDGET_INTERFACE	GL	TIB_INT_GL_BUDGET_IFACE	pk_tib_budjupld_import	TIB_INT_GL_BUDGET_IFACE_S
Collection Import	QA_RESULTS_INTERFACE	QA	TIB_INT_QA_RESULT_S_IFACE	pk_tib_collimp_import	TIB_INT_QA_RESULTS_IFACE_S
Customer Item	MTL_CI_INTERFACE	INV	TIB_INT_MTL_CI_IFACE	pk_tib_customer_import	TIB_INT_MTL_CI_IFACE_S
Customer Item X Reference	MTL_CI_XREFS_INTERFACE	INV	TIB_INT_MTL_CI_XREFS_IFACE	pk_tib_customerxref_import	TIB_INT_MTL_CI_XREFS_IFACE_S
Cycle Count Entries	MTL_CC_ENTRIES_INTERFACE	INV	TIB_INT_MTL_CC_ENTRIES_IFACE	pk_tib_cyclecount_import	TIB_INT_MTL_CC_ENTRIES_IFACE_S

Table 38 Non-API Subscription Mapping Details

Interface Name	Interface Table Name	Schema Name	Intermediate Table Names	Package Name	TIBCO Sequence Name
Demand Schedules	RLM_INTERF ACE_HEADER S_ALL	RLM	TIB_INT_R LM_HEAD ER_IFACE	pk_tib_de mandsche dules_imp ort	TIB_RLM_INTER FACE_HEADERS _AL_S
	RLM_INTERF ACE_LINES_A LL		TIB_INT_R LM_LINES_ IFACE		
			TIB_INT_C RM_INTER ACTION	pk_tib_cr m_interacti on	TIB_RLM_INTER FACE_HEADERS _AL_S
Journal Import	GL_INTERFA CE	GL	TIB_INT_G L_INTERFA CE	pk_tib_jou rnimp_im port	TIB_INT_GL_INT ERFACE_S
Mass Additions	FA_MASS_AD DITIONS	FA	TIB_INT_F A_MASS_A DDITIONS	pk_tib_ma ssaddition s_import	TIB_INT_FA_MA SS_ADDITIONS_ S
Open Budget	FA_BUDGET_I NTERFACE	FA	TIB_INT_F A_BUDGET _IFACE	pk_tib_ope nbudget_i mport	TIB_INT_FA_BU DGET_IFACE_S
Open Forecast	MRP_FORECA ST_INTERFAC E	MRP	TIB_INT_M RP_FOREC AST_IFACE	pk_tib_ope nforecast_i mport_ne w	TIB_INT_MRP_F ORECAST_IFAC E_S

Table 38 Non-API Subscription Mapping Details

Interface Name	Interface Table Name	Schema Name	Intermediate Table Names	Package Name	TIBCO Sequence Name
Open Item	MTL_SYSTEM_ITEMS_INTERFACE	INV	TIB_INT_MTL_SYS_ITEM_INTERFACE	pk_tib_openitem_import	TIB_INT_MTL_SYS_ITEMS_INTERFACE_S
	MTL_ITEM_REVISIONS_INTERFACE		TIB_INT_MTL_ITEM_REVISION_INTERFACE		
	MTL_ITEM_CATEGORIES_INTERFACE (additional table for Oracle Applications 11.5.9)		TIB_INT_MTL_ITEM_CATEGORIES_INTERFACE (additional table for Oracle Applications 11.5.9)		
Open Master Schedule	MRP_SCHEDULE_INTERFACE	MRP	TIB_INT_MRP_SCHEDULE_INTERFACE	pk_tib_mst_schedule_import_new	TIB_INT_MRP_SCHEDULE_INTERFACE_S
Open Move	WIP_MOVE_TXN_INTERFACE	WIP, BOM	TIB_INT_WIP_MOVE_TXN_INTERFACE	pk_tib_openmove_import	TIB_INT_WIP_MOVE_TXN_INTERFACE_S
	CST_COMP_SNAP_INTERFACE		TIB_INT_CST_COMP_SNAP_INTERFACE		
Open Replenishment	MTL_REPLENISH_HEADER_INT	INV	TIB_INT_MTL_REPLENISH_HEADER_INTERFACE	pk_tib_openrepl_import	TIB_INT_MTL_REPLENISH_HEADER_INTERFACE_S
	MTL_REPLENISH_LINES_INT		TIB_INT_MTL_REPLENISH_LINES_INTERFACE		
Open Resource	WIP_COST_TXN_INTERFACE	WIP	TIB_INT_WIP_COST_TXN_INTERFACE	pk_tib_openresource_import_new	TIB_INT_WIP_COST_TXN_INTERFACE_S

Table 38 Non-API Subscription Mapping Details

Interface Name	Interface Table Name	Schema Name	Intermediate Table Names	Package Name	TIBCO Sequence Name
Open Transaction	MTL_TRANSACTION_INTERFACE	INV, BOM	TIB_INT_MTL_TXNS_IFACE	pk_tib_opntrans_import	TIB_INT_MTL_TXNS_IFACE_S
	MTL_TRANSACTION_LOTS_INTERFACE		TIB_INT_MTL_TXN_LOTS_IFACE		
	MTL_SERIAL_NUMBERS_INTERFACE		TIB_INT_MTL_SERIAL_NUM_IFACE		
	CST_COMP_SNAP_INTERFACE		TIB_INT_CST_COMP_SNAP_IFACE		

Table 38 Non-API Subscription Mapping Details

Interface Name	Interface Table Name	Schema Name	Intermediate Table Names	Package Name	TIBCO Sequence Name
Integrate Order Management Using Order Import Interface	OE_HEADERS_IFACE_ALL	ONT	TIB_INT_OE_HEADER_S_IFACE	pk_tib_oe_import	
	OE_LINES_IFACE_ALL		TIB_INT_OE_LINES_IFACE		
	OE_PRICE_ADJS_IFACE_ALL		TIB_INT_OE_PRICE_ADJS_IFACE		
	OE_CREDITS_IFACE_ALL		TIB_INT_OE_CREDITS_IFACE		
	OE_LOTSERIALS_IFACE_ALL		TIB_INT_OE_LOTSERIALS_IFACE		
	OE_RESERVE_TNS_IFACE_ALL		TIB_INT_OE_RESERVE_TNS_IFACE		
	OE_ACTIONS_IFACE_ALL		TIB_INT_OE_ACTIONS_IFACE		
Open Payables	AP_INVOICES_INTERFACE	AP	TIB_INT_AP_INV_IFACE	pk_tib_payabopn_import	
	AP_INVOICE_LINES_INTERFACE		TIB_INT_AP_INV_LINES_IFACE		
Periodic Cost	CST_PC_ITEM_COST_INTERFACE	BOM	TIB_INT_CST_PC_ITEM_IFACE	pk_tib_pericost_import	
	CST_PC_COST_DET_INTERFACE		TIB_INT_CST_PC_DET_IFACE		

Table 38 Non-API Subscription Mapping Details

Interface Name	Interface Table Name	Schema Name	Intermedi ate Table Names	Package Name	TIBCO Sequence Name
Purchasing Open Docs	PO_HEADERS_INTERFACE	PO	TIB_INT_PO_HEADERS_IFACE	pk_tib_pur copen_docs _import	
	PO_LINES_INTERFACE		TIB_INT_PO_LINES_IFACE		
	PO_DISTRIBUTIONS_INTERFACE		TIB_INT_PO_DISTRIBUTION_IFACE (for Oracle Apps 11.5.9 only)		
Receivables Customer	RA_CUSTOMERS_INTERFACE_ALL	AR	TIB_INT_RA_CUSTOMERS_IFACE	pk_tib_workorder_import	
	RA_CUSTOMER_PROFILES_INTERFACE_ALL		TIB_INT_RA_CUSTOMER_PROFILES_IFACE		
	RA_CONTACT_PHONES_INTERFACE_ALL		TIB_INT_RA_CONTACT_PHON_IFACE		
	RA_CUSTOMER_BANKS_INTERFACE_ALL		TIB_INT_RA_CUSTOMER_BANKS_IFACE		
	RA_CUST_PAY_METHOD_INTERFACE_ALL		TIB_INT_RA_PAY_METHOD_IFACE		

Table 38 Non-API Subscription Mapping Details

Interface Name	Interface Table Name	Schema Name	Intermediate Table Names	Package Name	TIBCO Sequence Name
Receiving Interface	RCV_HEADER_S_INTERFACE	PO	TIB_INT_RCV_HEADERS_IFACE	pk_tib_receiving_import	
	RCV_TRANSACTIONS_INTERFACE		TIB_INT_RCV_TXNS_IFACE		
Requisition Interface	PO_REQUISITIONS_INTERFACE_ALL	PO	TIB_INT_PO_REQ_IFACE_ALL	pk_tib_requisition_import	
	PO_REQ_DIST_INTERFACE_ALL		TIB_INT_PO_REQ_DIST_IFACE_ALL		
Work Order	WIP_JOB_SCHEDULE_INTERFACE	WP	TIB_INT_WIP_JOB_SCHEDULE_IFACE	pk_tib_workorder_import	
	WIP_JOB_DTLS_INTERFACE		TIB_INT_WIP_JOB_DTLS_IFACE		

Appendix D **Trace Messages**

This appendix describes the TIBCO trace message implementation, the validations performed on TIBCO Adapter for Oracle Applications subscription tables, and the errors generated by the validation scripts.

Topics

- [Overview, page 380](#)
- [Trace Message Fields, page 382](#)
- [Log Table Structure, page 386](#)
- [Status Messages, page 389](#)
- [TIBCO Log Table Messages, page 419](#)
- [Validations Performed, page 420](#)

Overview

Like many other TIBCO products, TIBCO Adapter for Oracle Applications can run in conjunction with TIBCO ActiveMatrix Adapter for Database, TIBCO Designer, Adapter SDK libraries, and other TIBCO software. Because of this, TIBCO Adapter for Oracle Applications can report errors that are actually caused by other TIBCO products.

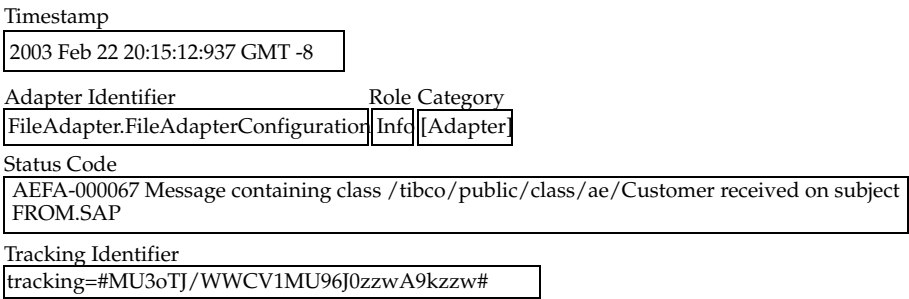
The TIBCO error tracing implementation provides consistent error handling across TIBCO products. It supports end-to-end tracking of messages throughout multiple TIBCO ActiveEnterprise applications, and includes standard information that is enforced by Adapter SDK. This means that all TIBCO software use a similar error message format that allows you to identify the origin and cause of a particular error, regardless of the application that reports the error.

Trace messages provide information about adapter activities. The messages are logged to the console where the runtime adapter was started and to a log file. Trace messages can also be redirected to the TIBCO Hawk Display application, or sent to other applications using the TIBCO Rendezvous transport.

Each trace message can include the following fields:

<Timestamp> <Adapter Identifier> <Role> <Category> <Status Code>
<Tracking Identifier>

The above fields are explained in [Trace Message Fields on page 382](#). The following diagram shows an example trace message and calls out the fields.



Example Trace Messages

The following trace messages were written during a session where TIBCO Adapter for Files received an object from TIBCO Adapter for R/3, then processed the object.

The first message indicates that TIBCO Adapter for Files has started. The timestamp indicates when the adapter started, and the role indicates that the trace message is informational, which means the activity is normal for the adapter. The category is identified, and the corresponding status code is displayed. The status code indicates that the adapter started successfully.

```
2003 Feb 22 20:14:51:718 GMT -8
FileAdapter.FileAdapterConfiguration Info [Configuration]
AEFA-000058 TIBCO Adapter for Files successfully initialized
```

The next set of trace messages indicates the adapter received an object that was sent on the TIBCO Rendezvous subject, FROM.SAP. The #MU3oTJ/WWCV1MU96J0zzwA9kzzw# tracking identifier included in the trace message uniquely identifies the message. The adapter (TIBCO Adapter for R/3) from which the message originated provided the identifier.

```
2003 Feb 22 20:15:12:937 GMT -8
FileAdapter.FileAdapterConfiguration Info [Adapter] AEFA-000067
Message containing class /tibco/public/class/ae/Customer received
on subject FROM.SAP tracking=#MU3oTJ/WWCV1MU96J0zzwA9kzzw#
```

```
2003 Feb 22 20:15:12:937 GMT -8
FileAdapter.FileAdapterConfiguration Info [Adapter] AEFA-000068
Message containing class /tibco/public/class/ae/Customer written
to working file customers.txt in Working Directory
F:\ca\integration\001\data_sets\files\wip
tracking=#MU3oTJ/WWCV1MU96J0zzwA9kzzw#
```

The final trace message indicates the object has been moved to the output directory, which completes the adapter's interaction with the object. Because the trace message is the termination point, the tracking identifier is not displayed.

```
2003 Feb 22 20:15:42:812 GMT -8
FileAdapter.FileAdapterConfiguration Info [Adapter] AEFA-000070
File customers.txt is moved to the Output Directory
F:\ca\integration\001\data_sets\files\solutions\output
```

Trace Message Fields

Each trace message includes the following fields:

Table 39 Tracing Fields

Field Name	Description
Timestamp	Timestamp of occurrence. For example, 2003 Feb 22 20:14:51:718 GMT -8.
Adapter Identifier	Name of the adapter that wrote the trace message. This is a combination of the adapter acronym and adapter configuration name. For example, the application identifier, ADB.publisher1 identifies a TIBCO ActiveMatrix Adapter for Database service named publisher1.
Role	<p>A role can be:</p> <p>Info. Indicates normal adapter operation. No action is necessary. A tracing message tagged with Info indicates that a significant processing step was reached and has been logged for tracking or auditing purposes. Only info messages preceding a tracking identifier are considered significant steps.</p> <p>Warn. An abnormal condition was found. Processing will continue, but special attention from an administrator is recommended.</p> <p>Error. An irrecoverable error occurred. Depending on the error severity, the adapter may continue with the next operation or may stop altogether.</p> <p>Debug. A developer-defined tracing message. In normal operating conditions, debug messages should not display.</p> <p>When configuring the adapter you define what roles should or should not be logged. For example, you may decide not to log Info roles to increase performance.</p>

Table 39 Tracing Fields

Field Name	Description
Category	<p>One of the following:</p> <p>Adapter. The adapter is processing an event.</p> <p>Application. The adapter is interacting with the Oracle Applications.</p> <p>Configuration. The adapter is reading configuration information.</p> <p>Database. The adapter is interacting with a database.</p> <p>DTA (Design-time adapter). The trace message is from the DTA.</p> <p>Metadata. The adapter is retrieving metadata from the Oracle Applications.</p> <p>Palette. The adapter is interacting with the palette.</p> <p>Publisher Service. The publication service is reporting this trace message.</p> <p>Request-Response Client Service. The request-response invocation service is reporting this trace message.</p> <p>Request-Response Server. The request-response service is reporting this trace message.</p> <p>Shutdown. The adapter is shutting down.</p> <p>Startup. The adapter is starting.</p> <p>Subscription Service. The subscription service is reporting this trace message.</p> <p>System. This category is not linked to a specific event process. The trace message may be related to a Windows service related messages, memory allocation, file system error, and so on.</p> <p>TibRvComm. The adapter is communicating with TIBCO Rendezvous.</p> <p>XML. The adapter is parsing XML documents.</p>
Status Code	<p>Unique code for the message and description. Status codes are identified by a unique number and description. If a trace message includes an error or warn role, the status code documentation includes a resolution. See TIBCO Log Table Messages on page 419 for details.</p>

Table 39 Tracing Fields

Field Name	Description
Tracking Identifier	<p>A unique identifier that is "stamped" on each message by the originating adapter. The tracking identifier remains in effect from a message's beginning to its completion as it is exchanged by TIBCO applications. If the adapter is the termination point of the message, the tracking identifier is not displayed in the trace message.</p> <p>You cannot modify the tracking identifier format or configure what information is displayed.</p>
Application Information	<p>Application-specific information added to the tracking info to trace the message back to its source. Set initially by the originating adapter and carried forward. It is augmented by each intermediate component.</p>

The TIBCO Adapter for Oracle Applications error message table format is described in [Log Table Structure on page 386](#).

Identifying the Origin of Error Messages

In an application's log file, each error message has the following format, derived from the tracing facility:

<Timestamp> <ApplicationID> <Role> <Category> <MessageCode> <Message>
<TrackingID>

For example, the result of one trace call might be:
2001 Sep 12 18:13:26:373 zap.inst1 Error [TibrvComm] AESDKJ-000237
AE Operation Request timeout
tracking=#<trackingID>#<info1>#<info2>#<info3>#

The MessageCode in this error message, AESDKJ, identifies this error as having been originated by Adapter SDK (TIBCO ActiveEnterprise SDK, or AESDK). The specific Adapter SDK error, number 000237, is described in the Adapter SDK documentation. Similarly, error messages created by other TIBCO products are described in those products' documentation.

Whether tracking information is added or not depends on the termination point of a message. If the adapter is the termination point a message is logged but tracking Info will not be added.

Error Messages in TIBCO Adapter for Oracle Applications

Error messages originated by the adapter have a `MessageCode` of `AEORAP`. These errors are generated by the subscription table validation scripts and are described in [TIBCO Log Table Messages on page 419](#). Other TIBCO products whose errors may be reported in TIBCO Adapter for Oracle Applications error tables are:

- TIBCO ActiveMatrix Adapter for Database run-time adapter
- TIBCO Designer
- Adapter SDK

Descriptions of the error codes and messages are in the documentation set of each program.

Log Table Structure

The structure of the common error table TIB_INT_LOG_SUB is shown here.



The common_all.sql script contains the create script for the Log Table. It also contains the common procedure for inserting records into the Log Table.

Table 40 Log Table Structure

Column Name	Data Type	Null?	Default Value	Description
TRANS_NAME	VARCHAR2 (10)	No		Name of the transaction in which the error occurred.
UNIQUE_ID	NUMBER	Yes		Sequence ID; Identifies each header record in the TIBCO Intermediate Table. This field is not populated in case of all error messages --which are logged by the Oracle Import program, when data from Oracle Intermediate Table gets processed.
ERROR_SEQUENCE_ID	NUMBER	No		Error Sequence ID, sequence number.

Column Name	Data Type	Null?	Default Value	Description
ERROR_SOURCE	VARCHAR2 (10)	Yes		<p>Indicates the place where the error occurred. It is used only in case of entries for Erroneous records. Possible values are:</p> <p>ORACLE—Indicates that the error occurred while transporting data from Oracle Interface table to the Oracle Source table.</p> <p>TIBCO—Indicates that the error occurred while transporting data from TIBCO Intermediate Tables to Oracle Interface tables.</p>
TABLE_NAME	VARCHAR2 (40)	Yes		<p>In case of valid records, this is the Header Table Name.</p> <p>In case of erroneous records, this is the name of the header or detail table in which the error occurred.</p>
TABLE_FIELD_NAME	VARCHAR2 (50)	Yes		Erroneous field in the header or line table. Used in case of entries for Erroneous records for Header or Line Tables.
ERROR_CODE	VARCHAR2 (20)	Yes		Error code. See TIBCO Log Table Messages on page 419 for more information.
DESCRIPTION	VARCHAR2 (2000)	Yes		Detailed description of the error.
TIB_SYSDATE	DATE	Yes	SYSDATE	Timestamp of when the error occurred.

Column Name	Data Type	Null?	Default Value	Description
STATUS	VARCHAR2 (1)	No		<p>The values are</p> <p>S — If the record is a valid one, where the record has been imported into the Oracle Source Tables.</p> <p>E — If the record is an erroneous one. This includes records which have failed the validations carried out on the Interface Tables by Oracle Concurrent programs.</p>
TIBCO_SOURCE	VARCHAR2(255)	Yes		The source of an external system that publishes data to the Oracle database through the adapter.
TIBCO_SOURCE_ID	VARCHAR2(255)	Yes		This corresponds to different data records from a source specified in TIBCO_SOURCE .
IN_HDR_FLDNAME1 to IN_HRD_FLDNAME10	VARCHAR2 (40)	Yes		A key field name received as part of the inbound data.
IN_HDR_FLDVALUE1 to IN_HRD_FLDVALUE10	VARCHAR2 (100)	Yes		A key field that has been received as part of the inbound data.
OUT_HDR_FLDNAME1 to OUT_HDR_FLDNAME10	VARCHAR2 (40)	Yes		The name of the key field that has been generated by Oracle Import Programs, which is also present in the Oracle Source Table.
OUT_HDR_FLDVALUE1 to OUT_HDR_FLDVALUE10	VARCHAR2 (100)	Yes		A key field that has been generated by Oracle Import Programs, which is also present in the Oracle Source Table.

Status Messages

The following table gives a list of palette errors generated by TIBCO Adapter for Oracle Applications.

For more information on Status Messages, refer Appendix B Error Messages in *TIBCO ActiveMatrix Adapter for Database User's Guide*.

Status Code	Role	Category	Resolution
AEADORAPPS-970001	An error, {0} has occurred while interfacing with target application Oracle Apps Database.		
	Error	Palette	Check the database contains the particular user or schema. Most probably the user does not exist in the database at all. Check the value entered for the user name. Give a correct value and try again.
AEADORAPPS-970002	Field: PreCommit Stored Procedure, Palette error. This is not a valid value.		
	Error	Palette	Enter a valid name for the PreCommit Stored Procedure such as <code>APPS.package_validate.main_adb</code> .
AEADORAPPS-970003	Field: Apps User Details, Palette error. This is mandatory.		
	Error	Palette	Enter a valid value for this field.
AEADORAPPS-970004	Field: SourceTableUserName1, Palette error. This is mandatory.		
	Error	Palette	Enter a valid value for this field.
AEADORAPPS-970005	Field SourceTableUserName2 Password, Palette error. This is mandatory.		
	Error	Palette	Enter a valid value for this field.
AEADORAPPS-970006	Missing Pre SQL script for pre-configured Oracle Apps Transaction.		
	Error	Palette	Check for the existence of the relevant file in the palette jar file provided with the adapter installable.

Status Code	Role	Category	Resolution
AEADORAPPS-970007	Missing Pre Undo SQL script for pre-configured Oracle Apps Transaction.		
	Error	Palette	Check for the existence of the relevant file in the palette jar file provided with the adapter installable.
AEADORAPPS-970008	Missing Post Undo SQL script for pre-configured Oracle Apps Transaction.		
	Error	Palette	Check for the existence of the relevant file in the palette jar file provided with the adapter installable.
AEADORAPPS-970009	Missing Post SQL script for pre-configured Oracle Apps Transaction.		
	Error	Palette	Check for the existence of the relevant file in the palette jar file provided with the adapter installable.
AEADORAPPS-970010	Error Creating Database Objects.		
	Error	Palette	Clear the database of the relevant objects and run the SQL script generated in the [TIBCO_ADORAPPS_HOME]/sql folder.'
AEADORAPPS-970011	An error {0} has occurred while interfacing with target application Oracle Apps Database.		
	Error	Database	Consult your Oracle Applications DBA or Network Administrator for connectivity errors.
AEADORAPPS-000012	An error {0} has occurred while interfacing with {1} schema of Oracle Apps Database.		
	Error	Database	Consult your Oracle Applications DBA for errors.
AEADORAPPS-970013	Database object {0} not found.		
	Error	Database	Create the required database object and try configuring again.

Status Code	Role	Category	Resolution
AEADORAPPS-970014	Exception thrown {0}		
	Error	Palette	Check the Designer error logs for details and contact TIBCO Support if required.
AEADORAPPS-970015	Ill-formatted table name {0}		
	Error	Palette	Enter a valid value for the table name.
AEADORAPPS-970016	Unknown column name {0}		
	Error	Palette	Normal behavior. No action required.
AEADORAPPS-970017	{0} Database connection succeeded.		
	Information	Palette	Normal behavior. No action required.
AEADORAPPS-970018	jdbc connection as : URL {0}, User {1}		
	Information	Palette	Normal behavior. No action required.
AEADORAPPS-970019	SQL statement: {0}		
	Information	Palette	Normal behavior. No action required.
AEADORAPPS-970020	Column type: {0} for field: {1}. Ignoring.		
	Error	Palette	Normal behavior. No action required.
AEADORAPPS-970021	Error reading configuration file %1		
	Error	Configuration	Please check if the properties file is available in the adorapps.jar
AEADORAPPS-970022	The configuration file %1 didn't provide any properties.'		
	Error	Configuration	Please verify the content of the properties file
AEADORAPPS-970023	Field %1 /Palette Error. This field is mandatory.		
	Error	Configuration	Please enter the details for this field
AEADORAPPS-970024	Error deploying %1		
	Error	Configuration	Please Undo the sql scripts present in sql folder

Status Code	Role	Category	Resolution
AEADORAPPS-970025	Error tearing down %1		
	Error	Configuration	Please enter the details for this field
AEADORAPPS-970026	Required schemas for Oracle Applications Adapter could not be loaded.		
	Error	Configuration	Please check the connection
AEADORAPPS-970031	Error stopping the design-time-adapter		
	Error	Configuration	Please check the connection
AEADORAPPS-970032	Field Message Subject/Palette error. Invalid subject:%1		
	Error	Configuration	Please enter the proper subject
AEADORAPPS-970034	%1/Palette Error. Adapter Configuration names must only have alphanumeric characters		
	Error	Configuration	Please give the valid name
AEADORAPPS-970035	Field Service Name		
	Information	Configuration	Please give the field service name
AEADORAPPS-970037	Invalid Selection/Please do not mark a key column as unused.		
	Error	Configuration	Please do not mark a key column as unused
AEADORAPPS-970038	Field %1/Palette Error. Invalid Table Name		
	Error	Configuration	Please give proper table name
AEADORAPPS-970039	Field %1		
	Error	Configuration	Please enter proper field name
AEADORAPPS-970044	Connection Test./Connection test failed.		
	Error	Configuration	Please enter correct values in design time connection
AEADORAPPS-970047	Connection Test./Connection test timed out.		
	Error	Configuration	Please increase the timeout in design time connection tab

Status Code	Role	Category	Resolution
AEADORAPPS-970048	Connection test time out.		
	Error	Configuration	Please increase the timeout in design time connection tab
AEADORAPPS-970052	Error/The Designer cannot write to:%1		
	Error	Configuration	Please ensure that at least one of these directories exists and is writable
AEADORAPPS-970055	Required Configuration Missing./No tables found. Please add tables by editing the Tables' tab.'		
	Error	Configuration	Please check the connection attributes properly
AEADORAPPS-970058	Required Configuration Missing./The number of joins %1 specified for the number of tables is insufficient.		
	Error	Configuration	Please give proper number of joins for the number of tables
AEADORAPPS-970059	Duplicated publishing schema found./The schema class %1 is already used. Please type a different name in PUBLISHING_TABLE field.		
	Error	Configuration	Please check the schema is already present
AEADORAPPS-970061	Restricted Oracle trigger columns found./The publishing table %1 has restricted trigger columns.		
	Error	Configuration	Please set storage mode to "Publish By Reference" for this publication.
AEADORAPPS-970063	Neither primary keys nor user defined keys found./The publishing table %1 has neither primary keys nor user defined keys.		
	Error	Configuration	Please choose the primary key for publishing table
AEADORAPPS-970066	No column to publish./You have to select at least one column to publish.		
	Error	Configuration	Please select at least one column to publish

Status Code	Role	Category	Resolution
AEADORAPPS-970068	No column to subscribe./You have to select at least one column to subscribe.		
	Error	Configuration	Please select at least one column to subscribe
AEADORAPPS-970069	Add Table/The Table Name entered is invalid.		
	Error	Configuration	Please enter a valid Oracle table name
AEADORAPPS-970071	Bad Schema./Publishing tables must be in the default schema.		
	Error	Configuration	Please see that publishing tables must be in the default schema
AEADORAPPS-970072	Required Configuration Missing./The table %1 cannot be used to store opaque messages.		
	Error	Configuration	Please provide a new name, the palette will create this table for you
AEADORAPPS-970073	No additional message was provided.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970074	Delete database objects/The database objects for this %1 cannot be deleted from the database successfully. Please run the generated script to clean up your database manually.		
	Error	Configuration	Please run the generated script to clean up your database manually
AEADORAPPS-970075	Save to database/The changes for instance %1 cannot be saved to the database successfully. You should correct the database error before continue to modify this instance. Please refer to the user guide for steps to recover from this error with the generated script.		
	Error	Configuration	Please correct the database error before continue to modify this instance

Status Code	Role	Category	Resolution
AEADORAPPS-970076	Save to database/The script for reconfiguring the database objects for instance%1 could not be generated successfully. Your changes will not be saved to the database. Please refer to the user guide for steps to recover from this error.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970077	Delete database objects/The script for deleting the database objects cannot be generated successfully. If you continue to delete this instance, your changes will not be saved to the database. Please refer to the user guide for steps to recover from this error.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970079	Error/The table have already been configured by another %1 service.		
	Error	Configuration	Please choose another table and try again
AEADORAPPS-970080	Info/You have chosen a table that has been configured by another service.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970081	Error/Please specify your user name in the adapter configuration resource.		
	Error	Configuration	Please specify your user name in the adapter configuration resource
AEADORAPPS-970082	Warning/You have modified some values in your connection tab without applying the changes.		
	Error	Configuration	Please update the connection setting properly
AEADORAPPS-970083	Error/The table you have selected contains		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970085	Property File Error.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.

Status Code	Role	Category	Resolution
AEADORAPPS-970086	Palette Error. Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970088	Palette Error./Cannot undo Add/Delete actions on %1. Warning	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970089	Invalid Selection/The selected Business Object has already been used. Error	Configuration	Please select an unused Business Object
AEADORAPPS-970090	Palette Error./The object %1 does not exist in the database or the User Schema does not have grant permissions to object%2. Error	Configuration	Please create this object in the database if not present. If present run the query GRANT SELECT ON <Object_Name> TO USER_SCHEMA or run the common_all.sql script present in <ADORAPPS_HOME>/config/sub. Delete the Business Object and reconfigure it.
AEADORAPPS-970091	Connection Settings Manager/%1 Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970092	Connection Settings Manager/Save password (this may be a security risk)? Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970093	Connection Settings Manager/Really delete %1 ? Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970095	Connection Settings Manager/Select connection to delete Information	Configuration	Indicates normal adapter operation. No action is necessary.

Status Code	Role	Category	Resolution
AEADORAPPS-970096	Connection Settings Manager/	There are no connection setting files.	
	Information	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970097	Connection Settings Manager/	Please apply your settings first!.	
	Error	Configuration	Please apply your settings first
AEADORAPPS-970098	Connection Settings Manager/	Name current configuration.	
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970099	Missing script/	Script for the key:%1 is null.	
	Error	DTA	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970102	Incorrect Input/	Message subject for a selected Business Object cannot be empty.	
	Error	DTA	Please ensure message subject is not empty
AEADORAPPS-970103	Info/	Could not create classes for: %1.	
	Error	Configuration	Please check the error log for details and reconfigure this service.
AEADORAPPS-970104	Info/	Cannot add child tables to the Join Table:%1.	
	Error	Configuration	Please select a view for adding join tables.
AEADORAPPS-970105	Info/	Cannot remove this view.	
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970106	Database Operation/	All the configured views will now be saved.	
	Information	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970107	Operation Failed/	Could not create Views for the service:%1	
	Error	Configuration	Please check the error log for details.

Status Code	Role	Category	Resolution
AEADORAPPS-970108	Incorrect Input/The View Name:%1 entered is incorrect.		
	Error	Configuration	Please provide a valid name.
AEADORAPPS-970109	Missing Values/Please provide a View Name for:%1		
	Error	Configuration	Please provide a View Name
AEADORAPPS-970110	Operation Failed/Could not create Intermediate Tables for the Inbound API service:%1		
	Error	Configuration	Please undo the sql scripts present in sql folder
AEADORAPPS-970111	Error/The join columns could not be remapped for the views.		
	Error	Configuration	Please configure the service once more.
AEADORAPPS-970100	Question/Your selection requires the configuration forms to be regenerated.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970112	Property File Error./Value for storageMode not specified in the properties file		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970113	Property File Error./Storage Mode not specified in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970114	Property File Error./Value for publishingTable not specified in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970115	Property File Error./Publishing Table not specified in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.

Status Code	Role	Category	Resolution
AEADORAPPS-970116	Property File Error./Value for wireFormat not specified in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970117	Property File Error./Incorrect value of wireFormat specified in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970118	Property File Error./Wire Format not specified in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970119	Property File Error./No tables provided in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970120	Property File Error./Wrong number of table relationships specified in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970121	Property File Error./Each table relationship should have		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970122	Property File Error./Tables provided in the relationships must		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970123	Property File Error./Table names in Oracle Apps Database should be		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.

Status Code	Role	Category	Resolution
AEADORAPPS-970124	Property File Error./Invalid Oracle table name.:%1		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970125	Property File Error./The Table: %1 should include schema name also.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970126	Property File Error./Each table relationship should have a		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970127	Property File Error./Each table relationship should have		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970128	Property File Error./Incorrect number of keys specified for		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970129	Property File Error./Incorrect combination of keys specified for		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970130	Info/Table selection panels will reload listings when you add additional tables		
	Information	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970131	Delete Table/Nothing to delete!		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970132	Delete Table/Are you sure you want to delete %1 and related dependencies?		
	Information	Configuration	Indicates normal adapter operation. No action is necessary.

Status Code	Role	Category	Resolution
AEADORAPPS-970133	Add Table/No tables found!		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970134	Add Table/Select table to add:		
	Information	Configuration	please select table to add
AEADORAPPS-970135	Add Table/Table has no columns!		
	Error	Configuration	Indicates normal adapter operation. No action is necessary.
AEADORAPPS-970137	Question/Please enter the Intermediate Table Name for the selected table.		
	Information	Configuration	Please enter the Intermediate Table Name for the selected table.
AEADORAPPS-970138	Table has no columns!/Table %1 has no columns.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970140	Question/Do you want to regenerate subjects set to previous defaults?		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970141	Connection Test/Connection successful		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970142	Question/Enter schema name		
	Information	Configuration	Please enter schema name
AEADORAPPS-970143	Question/The schema class %1 is no longer needed by the operation.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970144	Question/The operation %1 uses the same table as other %2.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970146	MISSING TABLES/No tables provided in the properties file.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970147	Info/The Subject %1 contains substitution variables.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970148	Question/Enter Schema Name:		
	Error	Configuration	Please enter the schema name
AEADORAPPS-970149	Error/No Tables Found		
	Error	Configuration	Please check the design time connection parameters
AEADORAPPS-970150	Question/Select table or view		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970151	Question/Do you want to delete the schema(s) used by %1 and their dependencies?		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970152	Add User/No Users found!		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970153	Add User/Select User to add:		
	Information	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970154	Add Language/No Languages found!		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970155	Add Language/Select Language to add:		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970156	Add Application/No Applications found!		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970157	Add Application/Select Application to add:		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970158	Add Responsibility/No Responsibilities found!		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970159	Add Responsibility/Select Responsibility to add:		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970160	Initialization Errors/Error occurred during initialization of the service.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970161	Question/The schema class %1 already exists.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970162	Warning/Are you sure you want to rename %1 to %2 .		
	Information	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970163	Reading table schema for %1		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970164	Listing tables for %1		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970165	Connecting to database...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970166	Disconnecting design-time adapter client...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970167	Stopping design-time adapter client...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970168	Listing tables for %1		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970169	Generating scripts...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970170	Reconfiguring database objects...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970171	Loading configuration from catalog...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970172	Reading table schema for %1		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970173	Loading schema dependencies...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970174	Select the OracleApps Inbound Business Object you want to configure		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970175	Select the OracleApps Outbound Business Object you want to configure		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970176	Adding operation...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970177	Prepare to delete operation...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970178	Migrating publishers...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970179	Loading preregistered listener info...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970180	Migrating subscribers...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970181	Loading child table mappings...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970182	Generating Oracle Apps Business Object...		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970183	Required schemas for the Adapter Service could not be loaded.Incorrect Schema Information		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970184	The requested service is only available when the Designer application is running		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970185	Unknown database type		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970186	Key and values must match		
	Error	Configuration	Please check that key and values are matching or not
AEADORAPPS-970187	Direction can only be 'in- out', 'in' or 'out'		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970188	Resource was not in expected format!:%1		
	Error	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970189	Required schemas for the Adapter Service could not be loaded:%1		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970190	Required schemas for the Adapter Service could not be loaded: (schema not specified)		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970191	Required schemas for the Adapter Service could not be loaded:Incorrect schema information		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970192	Make sure Tibase.jar is in your classpath		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970193	An error %1 has occurred while interfacing with %2 schema of Oracle Apps Database.		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970194	An error %1 has occurred while interfacing with target application Oracle Apps Database.		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970195	Invalid message format		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970196	Exception thrown %1		
Error	Configuration	Indicates normal adapter operation. No action is necessary	

Status Code	Role	Category	Resolution
AEADORAPPS-970197	Exception thrown %1		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970198	un-supported category save failed		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970199	do and undo sql are not parsed appropriately		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970200	un-supported action saveFailed		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970201	Exception thrown :Field Apps User Details , Palette error. This is mandatory.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970202	publishing table name ill-formatted		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970203	source table name ill-formatted		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970204	unsupported action		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970205	A valid OracleApps Business Object could not be found		
	Error	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970206	Required operations were not found.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970207	adbDateTime type was not found		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970208	Table %1 does not exist.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970209	Creation of database object failed		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970210	Error Creating Database Objects.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970211	The following statement fails.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970212	The following DELETE statement fails because the record is not in ADB catalog tables. %1		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970213	ENDPOINT_ERROR: no type is provided for publisher endpoint		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970214	ENDPOINT_ERROR: no endpoint class is provided for publisher endpoint		
	Error	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970215	ENDPOINT_ERROR: no subject is provided for publisher endpoint		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970216	ENDPOINT_ERROR: no wire format is provided for publisher endpoint		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970217	ENDPOINT_ERROR: %1 is not a standard publisher endpoint		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970218	ENDPOINT_ERROR: %1 is not a supported publisher data format		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970219	ENDPOINT_ERROR: no type is provided for subscriber endpoint		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970220	ENDPOINT_ERROR: no endpoint class is provided for subscriber endpoint		
Error	Configuration	Indicates normal adapter operation. No action is necessary	
AEADORAPPS-970221	ENDPOINT_ERROR: no subject is provided for subscriber endpoint		
Error	Configuration	Please give the subject	
AEADORAPPS-970222	ENDPOINT_ERROR: no wire format is provided for subscriber endpoint		
Error	Configuration	Please give the wire format	
AEADORAPPS-970223	ENDPOINT_ERROR: %1 is not a standard subscriber endpoint		
Error	Configuration	Indicates normal adapter operation. No action is necessary	

Status Code	Role	Category	Resolution
AEADORAPPS-970224	ENDPOINT_ERROR: %1 is not a supported subscriber data format		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970225	missing database Name		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970226	missing table space Name		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970227	missing index Suffix		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970228	missing trigger Suffix		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970229	columns to fire has wrong column names		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970230	user defined key has wrong column names		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970231	Timeout value too small. Please specify a number larger than %1 milliseconds.		
	Error	Configuration	Please specify a number larger than 1000
AEADORAPPS-970232	Error During Delete:Read-Only File		
	Error	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970233	The resource "%1" could not be deleted. Delete requires that "%2" must be checked out. Please checkout the resource and try deleting again. You can select the resource to be checked out by clicking the "Go To Resource" button.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970234	Error During Paste:Read-Only File		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970235	The resource "%1" could not be Pasted. Paste requires that "%2" must be checked out. Please checkout the resource and try Pasting again. You can select the resource to be checked out by clicking the "Go To Resource" button.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970236	Error During Move: Read-Only File		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970237	The resource "%1" could not be Moved. Move requires that "%2" must be checked out.Please checkout the resource and try Moving again. You can select the resource to be checked out by clicking the "Go To Resource" button.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970238	Error During Rename: Read-Only File		
	Error	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970239	The resource "%1" could not be Renamed. Rename requires that "%2" must be checked out.Please checkout the resource and try Renaming again. You can select the resource to be checked out by clicking the "Go To Resource" button.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970240	Global Variables File Read-Only		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970241	Global Variable files needs to be checked out in order to successfully create the adapter.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970242	Error: Read-Only File		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970243	The operation on Resource " %1 " could not be completed.The operation requires that " %2 " must be checked out.Please ensure that the file is checked out.You can select the resource to be checked out by clicking the "Go To Resource" button		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970244	Warning: Add File to RCS		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970245	Warning: "%1" was created during schema generation.		
	Warning	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970246	Error During Configuration: Read-Only File		
	Warning	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970247	The resource "%1" could not be Configured. Configuration requires that "%2" must be checked out.Please checkout the resource and try Configuring again. You can select the resource to be checked out by clicking the "Go To Resource" button.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970248	Error During Rename		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970249	Another service with same name exist in the project.Please specify a unique service name		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970250	Timeout value is too large. Please specify a number less than %1 milliseconds.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970251	Add Concurrent Program Name/No Concurrent Program Names found!		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970252	Add Concurrent Program Name/Select Concurrent Program Name to add.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970253	Warning: The Sql Script %1 already exists.\n Press Yes to OverWrite No to append to the Existing script.		
	Warning	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970254	Error/Palette Error. View Name cannot have more than 30 characters long.\n Please type in a valid View Name For :%1.		
	Error	Configuration	Please specify a View Name that has less than 31 characters
AEADORAPPS-970255	InValid %1 Name/Palette Error. The table name %1 entered is invalid.		
	Error	Configuration	Please ensure that table name does not contain extra spaces, invalid characters and more than 30 characters.
AEADORAPPS-970256	Field %1 /Palette Error. This field should be defined in Global Variables Tab and assigned some value.		
	Error	Configuration	This field should be defined in Global Variables Tab and assigned some value
AEADORAPPS-970257	Add Column1 Name/No Columns found!		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970258	Add Column1 Name/Select Columns to add.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970259	Add Column2 Name/No Columns found!		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970260	Add Column2 Name/Select Columns to add.		

Status Code	Role	Category	Resolution
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970261	InValid %1 Name/Palette Error. Column1 and Column2 Names cannot be the same.Please enter different Column Names.		
	Error	Configuration	Please enter different Column Names for Column1 and Column2
AEADORAPPS-970262	Field %1 /Palette Error. Field Column1 field is mandatory. Enter a valid value for this field.		
	Error	Configuration	Enter a valid value for this field.
AEADORAPPS-970263	Concurrent Program Parameters Not Found./The object %1 does not contain Concurrent Program Parameters.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970264	Appending %1 Concurrent Program Parameters to Header Intermediate table.		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970265	Information/Cannot remove the master table %1.\nPlease Delete the BusinessObject %2 to delete %3.		
	Information	Configuration	Delete the OracleApps BusinessObject configuration instance
AEADORAPPS-970266	Field %1 /Palette Error. This %2 field in Configuration tab is mandatory.		
	Error	Configuration	Enter a valid value for this field.
AEADORAPPS-970267	Question/Enter Table Name Pattern:		
	Information	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970268	Error Deploying Scripts.		

Status Code	Role	Category	Resolution
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970269	Error Saving Scripts.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970270	Error creating Intermediate Tables for Inbound API Service.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970271	Properties could not be found.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970272	Required Configuration Missing.		
	Error	Configuration	Please specify at least one column as a user key in table %1
AEADORAPPS-970273	Required Configuration Missing.		
	Error	Configuration	For child tables, a join column should necessarily be a user key
AEADORAPPS-970274	Cannot create schemas for views.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970275	Could not find the adapter instance.		
	Error	Configuration	Indicates normal adapter operation. No action is necessary

Status Code	Role	Category	Resolution
AEADORAPPS-970276	Oracle Applications Version Not Supported /The Oracle Applications Version %1 is not supported for all Predefined API Subscription Services.\nOnly 11.5.5, 11.5.8 and 11.5.9 Oracle Applications Versions are supported. 11.5.5 Sql scripts will be generated by default.		
	Warning	Configuration	Indicates normal adapter operation. No action is necessary
AEADORAPPS-970278	Grant permissions not given.		
	Error	Configuration	Please ensure that the User_Schema has grant permissions on table APPLSYS.FND_PRODUCT_GROUPS. Please execute the query GRANT SELECT ON APPLSYS.FND_PRODUCT_GROUPS TO USER_SCHEMA or run the common_all.sql script present in <ADORAPPS_HOME>/config/sub. Delete the Business Object and reconfigure it.

TIBCO Log Table Messages

The following adapter errors can occur.

Table 41 TIBCO Log Table Messages

Error Code	Error Description
AEORAP-000014	E-EXPECTED ERROR
AEORAP-000015	U-UNEXPECTED ERROR
AEORAP-000016	INSERT INTO ORACLE APPS INTERFACE TABLES FAILED
AEORAP-000019	KEY FIELDS MISSING
AEORAP-000020	F - FATAL ERROR
AEORAP-000023	UNEXPECTED – ERROR THROWN BY API
AEORAP-000024	DELETION FROM INTERMEDIATE TABLE FAILED

Validations Performed

Data that is subscribed to by Adapter for Oracle Applications using API transactions listed in the following table is validated by the adapter before the Oracle Application can receive new data. An Oracle defined concurrent program then populates validated data into the Oracle database tables. TIBCO Adapter for Oracle Applications validation scripts perform the following checks on subscription transaction parameters.

Table 42 Validations

Transaction Name	Validations Done
EnggChangeOrder	Null, Unexpected Errors, Expected Errors and Fatal Error
BillsOfMtlAPI	Null, Unexpected Errors, Expected Errors and Fatal Error

User-defined Validation

Depending on your requirement, apart from the validations carried out by the adapter, you can perform extra validations or by-pass the validations carried out by the adapter. This is done by using user exits. There are two types of user exits:

- Pre-user Exits
- Post-validation User Exits

You can define parameters for the pre and post-validation user exits to perform validations as per your requirement. The validation procedure using user exits will include the following steps:

1. The Concurrent Manager invokes the TIBCO validation program.
2. A PreDefined procedure (Pre-user Exit) is called for each interface before the TIBCO validation program runs. You can define the following parameters in this pre-user exit:

IN/OUT (Header Record Object, array of child record(s))
OUT (Varchar2 indicator of success or failure)
3. In this user-exit, you can:
 - Map the validations for inbound data according to your requirements.
 - Modify the record attributes if required.

4. If any errors occur, the pre-user exit uses the TIBCO error logging procedure to log errors in the common error tables and modify records in the intermediate tables.
5. The TIBCO validation program runs. If no errors were encountered in the pre-user exit procedure, the standard TIBCO validations are by-passed.
6. After the TIBCO validation program runs or is by-passed, a post-validation user exit is provided in which you can define the following parameters:
 - IN (Header Record Object, array of child record(s))
 - IN/OUT (Varchar2 indicator of success or failure)
 - The post-validation process only checks for referential integrity of records and will not modify attributes of a record. All substitution or modification of record attributes must be performed in the pre-user exit validation process.
7. The validation procedure is complete.

The user exit procedures will run for all subscription transactions, except those transactions which call APIs defined by Oracle. For these transactions, the APIs will perform the required validations before inserting data into the interface tables. For all the APIs, both pre and post-validation user exits are called in succession. In these cases, the API call will not be by-passed, since the API validates and inserts record data. In other transactions, insertion of data into Oracle database tables is independent of by-passing of TIBCO validations.

Appendix E **Clean-up Scripts**

This appendix describes clean-up scripts and clean-up strategy.

Topics

- [Clean-up of Records from MV Log, page 424](#)
- [Clean-up Strategy for Publishing, page 425](#)
- [Clean-up Strategy for Intermediate Tables, page 426](#)
- [Purge TIBCO Records from Oracle Interface Tables, page 427](#)

Clean-up of Records from MV Log

Records are deleted from an MV Log table in either of the following ways:

- By deleting all rows in the MV Log table that have a value of U for both the DMLTYPE\$\$ and OLD_NEW\$\$ fields.
- By deleting all rows older than a specified number of days. All rows in the MV Log table have a TIBCO_SYSDATE field with a default value of sysdate (current timestamp). The period specified in the cleanup script determines which rows are removed, as shown:

```
DELETE FROM ONT.MLOG$_OE_ORDER_HEADERS_ALL WHERE
TO_CHAR(TO_DATE(TIBCO_SYSDATE, 'DD-MON-YYYY'))
BETWEEN TO_CHAR(TO_DATE(SYSDATE, 'DD-MON-YYYY')) AND
TO_CHAR(TO_DATE('25-NOV-2004', 'DD-MON-YYYY'))
```

In the MV Log table, there are two records for every update operation on the Base table: one with the old value and one with the new value. The old value of the update operation is indicated by the record which has DMLTYPE\$\$ = U and OLD_NEW\$\$ = U.

Clean-up Strategy for Publishing

Rows in the publishing table can be deleted by running a clean-up script periodically. All rows in a publishing table where `ADB_L_DELIVERY_STATUS` is **C** (Complete) or **F** (Failed) can be deleted without any adverse effect on the adapter agent. However, the client may decide to keep some or all of the completed or failed records for auditing purposes.

Rows in the MV Log table can also be deleted by running a clean-up script periodically. These rows are deleted without any adverse affect on the publish transactions. An additional field, `TIBCO_SYSDATE`, is added to the MV Log of all the tables when the processing is completed.

An example is given below:

```
Delete from F_PO_VENDORS WHERE ADB_L_DELIVERY_STATUS IN (C, F)
```

Clean-up Strategy for Intermediate Tables

The records in the intermediate tables that are successfully inserted into the Oracle interface tables are deleted by the validation program at the end of each run.

Purge TIBCO Records from Oracle Interface Tables

You can opt to delete the records from the Oracle interface tables inserted by TIBCO Adapter for Oracle Applications.

In each Predefined Subscription transaction, the scripts <*>_all.sql contain code to delete records from the respective Oracle Interface tables.

This code is commented in the <*>_all.sql scripts. To delete records from the Oracle Interface tables, uncomment this code and redeploy the scripts onto the database.

```

/*****Purge program*****/
/*Purging TIBCO Inserted records from interface tables *****/
/*uncomment the below 'Delete codes' if you want this feature*/
/*****

--IF w_request_id IS NOT NULL THEN
--DELETE FROM MTL_ITEM_CATEGORIES_INTERFACE WHERE REQUEST_ID =
--w_request_id;
--DELETE FROM MTL_ITEM_REVISIONS_INTERFACE WHERE REQUEST_ID =
--w_request_id;
--DELETE FROM MTL_SYSTEM_ITEMS_INTERFACE WHERE REQUEST_ID =
--w_request_id;
--END IF;

/*****End of the Purge program*****/

```


Appendix F **Registering a Concurrent Program**

This appendix describes the steps to register a concurrent program in Oracle Applications.

Topics

- [Procedure, page 430](#)

Procedure

Use the following steps to register a concurrent program:

1. Open ORACLE Applications.
2. The Responsibility screen displays.
 - Double-click **System Administrator**.
 - The Navigator-System Administrator screen displays. Double-click **Application**.
 - Double-click **Register**.
 - The Applications screen displays. Enter the name of the application you wish to create, the short name for the application and the basepath for the application. The path is TOP by default.
 - Click **Save** to save the application.
3. Close the Application screen. The Navigator-System Administrator screen displays.
 - Double-click **Security**.
 - Double-click the **ORACLE** option, then double click **Data Group**.
 - The Data Group screen displays.
 - Enter the Data Group and the Description. This is optional. Enter the Application name as specified in the Applications screen in [step 2](#).
 - Select the ORACLE ID APPS from the lookup box.
 - Click **Save** to save the data group.
4. Close the Data Group screen. The Navigator-System Administrator screen displays.
 - Double-click **Application**.
 - Double-click **Menu**.
 - The Menu screen displays.
 - Enter the Menu, User Menu Name (optional) and the description (optional).
 - Enter the Sequence Number (for example: 1), Navigator Prompt (for example, Submit Request), and in the Function field, select View All Concurrent Programs System Administrator from the lookup.
 - Click **Save** to save the menu.

5. Close the Menus screen. The Navigator-System Administrator screen displays.
 - Double-click **Security**, then double click **Responsibility**.
 - Double-click **Request**.
 - The Request Groups screen displays.
 - Enter the Group and the Application name as specified in the Applications screen in [step 2](#). Enter the Code and Description (optional).
 - Click **Save** to save the request.
6. Close the Request Groups screen. The Navigator-System Administrator screen displays.
 - Double-click **Security** and then double click **Responsibility**.
 - Double-click **Define**.
 - The Responsibilities screen displays.
 - Enter the Responsibility Name, the Application Name as specified in [step 2](#), the Responsibility Key, Description (optional), the Data Group Name as specified in [step 3](#). Enter the Menu as specified in [step 4](#). Enter the Request Group Name (optional) as specified in [step 5](#).
 - Click **Save** to save the responsibility.
7. Close the Responsibilities screen. The Navigator-System Administrator screen displays.
 - Double-click **Security** and then double click **User**.
 - Double-click **Define**.
 - The Users window displays.
 - Query for the Username (for example, OPERATIONS) and then enter the Responsibility name as specified in the Responsibility tab in [step 6](#).
 - Click **Save** to save the user.
8. Go to File Menu and select Switch Responsibility. The Responsibility screen displays.
 - Double-click **Application Developer**.
 - The Navigator-Application Developer screen displays. Double-click **Concurrent**.
 - Double-click **Executable**.
 - The Concurrent Program Executable screen displays. Enter the Executable name. Enter the Short Name and Application name as specified in [step 2](#).

Enter the Description, enter PL/SQL Stored Procedure as the Execution Method, and enter the name of the stored procedure/package which is to be executed in the Execution File Name field.



In the case of TIBCO programs give the appropriate `<packagename>.main` for all transactions.

- Click **Save** to save the concurrent program.
- 9. Close the Concurrent Program Executable screen.
 - Double-click **Program**.
 - Enter the Program name (the program name can be similar to the name specified in the Executable field in [step 8](#)). Enter the Short Name and the Application name as specified in [step 2](#). In the Executable field, enter the Name as entered in Short Name.
 - Some parameters may have to be associated with Value Sets which will associate a List of values or a datatype or both along with each parameter. If this is the case, the value set can also be defined for the parameter appropriately. To add the parameters to be passed to the TIBCO program, click on Parameters. A new screen pops up. Enter the parameter name, description field appropriately.
 - Click **Save** to save the concurrent program.
- 10. Go to File Menu and select Switch Responsibility. The Responsibility screen displays.
 - Double-click **System Administrator**.
 - The Navigator-System Administrator screen displays. Double-click **Security** and then double click **Responsibility**.
 - Double-click the **Request** option.
 - Give the Group Name by selecting an existing group name. Enter the Group Name by selecting an existing group name, as specified in [step 3](#)



Ensure that you register the TIBCO Program as part of the same request group as the Oracle Source Program for that transaction.

In case of subscription mode you have to map value sets to parameter values set at the front end, in order to translate the parameter values set at the front end to actual values that need to be passed to the TIBCO program.

- Click **Save** to save the request group.

11. Go to File Menu and select Switch Responsibility.
 - The Responsibility screen displays.
 - Enter the Application name as specified in [step 2](#).
 - The Navigator-<Application Name> displays. Click the **Functions** Tab. Request comes up. Click on the **Open** button.
 - The Find Request Screen displays. Select the **All My Requests** radio button and click on the **Find** Button.
 - The Request screen displays.
 - Click **Submit a New Request**. The Submit a New Request screen displays.
 - Select the **Single Request** radio button and then click **OK**.
 - Select the Request to be executed from the lookup and then click **Submit**. The Submit Another Request screen displays. Click **No**.
 - You return to the Requests screen. Continue to click the **Refresh Data** button until Phase displays **Completed** and Status displays **Normal**.

Appendix G **Frequently Asked Questions**

This appendix lists answers to the frequently asked questions.

Topics

- [Frequently Asked Questions, page 436](#)

Frequently Asked Questions

Can I bring up TIBCO Designer from a UNIX command-line?

No. TIBCO Designer is a GUI based tool and a UNIX GUI environment is mandatory to run it. It cannot be brought up from a terminal.

When starting the adapter, what if the repository is not found?

Start the TIBCO Repository server before starting the adapter. If you are starting a remote repository ensure that TIBCO Administrator Repository Edition is installed on the remote location. Ensure that a properly configured `.dat` file is available in the path specified (local or remote). Ensure that the `RepoUrl` has been specified accurately in the adapter `.tra` file.

Why does the adapter startup fail?

Either the repository file (`.dat`) is not placed in the `<install_path>\tibco\repository\remoterepos` directory, or the `.dat` file is not properly configured. Ensure that the `RepoUrl` syntax has been specified accurately in the adapter `.tra` file. Ensure that the path specified for the `.tra` file is correct.

Why does the adapter startup fail, even after specifying the appropriate DAT file?

You must start the repository server before you start the adapter. If it is a remote repository ensure that the `RepoUrl` syntax has been specified accurately in the adapter `.tra` file. Ensure that the path specified for the `.tra` file is correct.

When saving an adapter configuration to the project, if an error occurs where is it logged?

TIBCO Designer error messages are logged to the files `stderr.log` and `designer.log` under the `TIBCO_HOME\Designer\<ver>\logs` directory.

When an error occurs in a Subscription Service adapter service, where is it displayed?

Errors that occur in a request-response operation are sent to the client. Errors that occur in a subscription operation are logged to a trace file. The log file path and name is set in the `.tra` file corresponding to the adapter instance. All logs are sent to `<install_path>\adapter\<adapter name>\<version>\logs` unless otherwise specified.

Why does the adapter fail to respond to a request?

The subject name may be inconsistent. The subject name to which the adapter listens may be different from that of the subject name of the client.

Why does the adapter fail to respond to a request after successfully receiving it?

The adapter may fail to respond due to various reasons like errors resulting from class mismatch, records not being available in the target application or, connectivity problems with the target application.

Upon dragging and dropping an Oracle Apps Business Object, errors were encountered and the business object could not be created. What might be the probable reasons?

The following might be the possible reasons for the error to occur:

- One or more of the connection parameters provided in the Connection tab of the adapter instance might be incorrect. Please provide the correct values and try again.
- Sufficient grants (GRANT SELECT ON <TableName> TO <User>) might not have been given on the tables present in Business Object. Here the <User> corresponds to the value provided in the Connection tab of the adapter instance.

Upon dragging and dropping an Oracle Publisher / Oracle Subscriber and selecting a Business Object from the drop-down list, I get a message "Adapter Service names must only have alphanumeric characters and must be at most 80 characters long. Please type in a valid name."

Ignore this message.

How important is it to give values to the TIBCO_KEY field? What if we leave it blank?

The TIBCO_KEY is the field relating the header record to the child record in a Subscription service. Now to relate the children to its grandchildren the TIBCO_KEY is one of the join keys along with the normal Oracle-specific Business Keys. If the above relation is not satisfied the entire message (header-->child-->grandchild) will not get inserted into the Oracle Interface Table. Therefore the TIBCO_KEY field has to be consistent throughout the inbound message that is same for the header, child and grandchild for a single message. This TIBCO key is also used to delete processed records from the TIBCO Intermediate Tables.

What significance do the other Oracle specific Business Keys have?

The Oracle Specific Business Keys are used to relate the records of a child (a child table is one that has a parent table) to that of its grand-child (a grand-child table is one whose parent is in turn a child table) table in any inbound message along with the TIBCO_KEY. If these keys do not match between the child and the grandchild, only the header and child records get inserted into the Oracle Interface Table from the TIBCO Intermediate Table.

What to do if the error "Savepoint found in a distributed transaction environment" is displayed when using "Subscribe with Reply" feature of the adapter?

If you are using Oracle ODBC Driver in the Microsoft Windows platform, then make sure to check the option **Disable MTS Option** in [Oracle ODBC Driver on page 60](#). If this option is not checked, this error is displayed.

In the Oracle Applications adapter I want to export the services as well as the Oracle Application Business Objects configured under an adapter instance in TIBCO Designer. What are the steps that I need to follow in order to achieve this?

When the adapter instance is exported, all the TIBCO ActiveMatrix Adapter for Database services configured under the adapter instance are exported in the XML format specified at the time of export. The exported XML file does not contain the Oracle Application Business Objects. Each Oracle Application Business Object configured under the adapter instance needs to be exported individually into separate XML files in AEXML format.

For example, there is a project `Project1.dat` containing an adapter instance named `Instance1`. This instance has an Oracle Application Business Object (`BusinessObject1`) and a corresponding service (`Service1`). This instance needs to be imported into another project named `Project2.dat`.

Once all the Business Objects have been exported successfully, follow the steps below while importing the adapter instance in another project.

- In the project named `Project1.dat`, select the adapter instance (`Instance1`) and export it as an XML file in the AEXML format. Name this file as `Instance1.xml`.
- In the project named `Project1.dat`, select the Business Object named `BusinessObject1` in the adapter instance (`Instance1`) and export it as an XML file in the AEXML format. Give a name to this file as `BusinessObject1.xml`. A message is displayed during the export process. Ignore this message and click on **Yes**.
- Open `Project2.dat` and import the file `Instance1.xml`. This will import the adapter instance `Instance1` into `Project2.dat`. This instance will NOT have

the Business Object (named `BusinessObject1`) inside the **Adapter Services** folder. This will contain only **Service1**.

- In `Project2.dat`, click on **Adapter Services** folder inside `Instance1` and import the file `BusinessObject1.xml`. This will import the Business Object named `BusinessObject1` into the **Adapter Services** folder of `Instance1`.
- **Save** `Project2.dat`.

This completes the export-import process.

When installing OracleApps, do I need to install the adapter as part of a TIBCO Administrative Domain or stand-alone? Is this installation required as part of an Administrative Domain or not?

If you have TIBCO BusinessWorks installed, install the adapter as a part of the TIBCO BusinessWorks domain. By doing so, the adapter would be a part of the domain along with other engines or adapters. If you do not have TIBCO Administrator or TIBCO BusinessWorks installed, then you can install the adapter in a standalone mode.

Do the `$TIBCO_HOME/bw/1.1/bin`, `$TIBCO_HOME/bw/1.1/logs`, `$TIBCO_HOME/tra/1.0/logs` directories need read-write permissions granted for adapter users? Only the last directory (`.../tra/1.0/logs`) exists in our environment. Am I right to assume that this is the only directory that I need to set permission on, as TIBCO BusinessWorks is not running?

It is recommended that the users be given the read-write access. To do so cascade to sub directories, and to those directories which are directly accessed by the adapter. For example, if you have only the Oracle Apps Adapter installed you should give ownership from the **TIBCO** folder itself, i.e. the entire `/u01/app/tibco` directory.

This is done because apart from the adapter logs the adapter puts logs into other directories too. Hence in order to fetch the log files, you need to give read-write permissions.

Where do I find the `setenv.sh` file?

You can find the `setenv.sh` file in the following location:

`<TIBCO_HOME>adapter/adorapps/5.1`

Should the TIBCO Adapter for Active Database and TIBCO Adapter for Oracle Applications run

on the same machine?

The TIBCO Adapter for Active Database and TIBCO Adapter for Oracle Applications cannot run on the same machine, if they are part of the same domain. If you are installing the TIBCO Adapter for Active Database and TIBCO Adapter for Oracle Applications as a part of TIBCO BusinessWorks domain the adapters cannot run simultaneously.

However, if they are not a part of the same TIBCO BusinessWorks domain, both the adapters can run on the same machine. This is applicable for the 4.X versions of the adapters.

When I install the adapter on Solaris 2.8 by unzipping the tar file, I am unable to find the tibinstall file?

Delete the zip folder already created. Perform the extract procedure again. Note for any error messages during extraction. Perform a checkout if the disk has the requires space for extracting the files. You might also want to try extracting in a different location.

The problem might have been encountered due to a corrupt installer tar file. You could ftp the installer tar file again.

You can also extract the tar file on a Microsoft Windows machine to check for the tar file.

What do I do if the error Savepoint found in a distributed transaction environment is displayed when using Subscribe with Reply feature of the adapter?

If you are using Oracle ODBC Driver on a Microsoft Windows platform, then make sure to check the option Disable MTS Option in Oracle8 ODBC Driver Setup. You encounter this error if this option is unchecked.

Do I need to give values to the TIBCO_KEY field?

The TIBCO_KEY is the field relating the header record to the child record in a Subscription service. To relate the children to its grandchildren the TIBCO_KEY is one of the join keys along with the normal Oracle-specific Business Keys. If the above relation is not satisfied the entire message (**header>child>grandchild**) will not get inserted into the Oracle Interface Table. Therefore the TIBCO_KEY field has to be consistent throughout the inbound message. This is the same for the header, child, and grandchild for a single message. The TIBCO key is also used to delete processed records from the TIBCO Intermediate Tables.

Appendix H **Database Comparison Tool**

This appendix provides information about the Database Comparison Tool. This tool allows you to identify the changes in the structure of the Oracle Application instances.

Topics

- [Functionality, page 442](#)
- [Setting Up and Executing DBTool, page 443](#)
- [Editing the Property file, page 444](#)
- [Editing dbtool.bat, page 445](#)
- [Sample Report Generated from dbtool, page 446](#)

Functionality

This tool lists the table differences between two databases and writes them into a text file. The table differences include the following:

- A column is added to a table.
- A column is deleted from a table.
- A table is deleted, renamed, or moved to another schema.
- The data type of a column is modified.
- The size of a column is modified.

Setting Up and Executing DBTool

The DBTool can be setup using the following procedure:

1. Edit the `Sample.properties` file available in `<adorapps_home>/bin/dbtools`. For details, refer to [Editing the Property file on page 444](#) for details.
2. Edit the `dbtool.bat` file available in `<adorapps_home>/bin/dbtools`. For details, refer to [Editing dbtool.bat on page 445](#).
3. Execute the `dbtool` as follows:

```
dbtool <property file name along with the full path>
```

For example:

```
dbtool <path of properties file>\Sample.properties
```

The tool lists the differences between the databases and writes them in a TXT file as specified in the properties file.

Editing the Property file

Edit the property file as per the requirements. The explanation for the various entries in the properties file follows:

1. Provide values for the old database and new database parameters. For example, the old database entries could be:

```
old_dburl = jdbc:oracle:thin:@192.168.111.15:1550:VIS5
old_username=apps
old_pwd=apps
```

The new database entries could be:

```
new_dburl =jdbc:oracle:thin:@192.168.31.11:8050:VIS
new_username=apps
new_pwd=apps
```

2. Provide the Oracle driver for the parameter driver. For example,
`driver = oracle.jdbc.driver.OracleDriver`
3. The results output from the `dbtool` are written in a text file. Mention the filename with the extension (`.txt`) for the parameter filename.
4. Mention all the transaction names involved in the Oracle Applications Adapter. For example,

```
txn.0 = ARCustomers
txn.1 = WorkOrder
```

5. Mention the table names for each transaction. For example,

```
txn.0.table.0=AR.HZ_PARTIES(schema name.tablename)
txn.0.table.1=AR.HZ_CONTACT_POINTS
```

Editing dbtool.bat

Set the following parameters in the `dbtool.bat` file:

- `TIBCO_HOME`

For example, `TIBCO_HOME` can be set as:

```
SET TIBCO_HOME=c:\tibco
```

- `JAVA_HOME`

For example, `JAVA_HOME` can be set as:

```
JAVA_HOME=%TIBCO_HOME%\jre\1.4.2
```

- `ORACLE_HOME`

For example, `ORACLE_HOME` can be set as:

```
SET ORACLE_HOME=d:\Oracle\ora81
```

- `TIBCO_ADORAPPS_HOME`

For example, `TIBCO_ADORAPPS_HOME` can be set as:

```
SET TIBCO_ADORAPPS_HOME = <adorapps_home>\bin\dbtool
```

Sample Report Generated from dbtool

On execution of DBtool, database comparison report is generated. Following are some portions from a sample report:

```

#####
TxnName      TableName      ColumnName      Description
#####
ApplicantPub PER_ALL_ASSIGNMENTS_F  GRADE_LADDER_PGM_ID  column added in target database
ApplicantPub PER_ALL_ASSIGNMENTS_F  SUPERVISOR_ASSIGNMENT_ID  column added in target database
ApplicantPub PER_ALL_ASSIGNMENTS_F  VENDOR_SITE_ID  column added in target database
.
.
-----
0)Differences were found for the transaction:<<ApplicantPub>>
-----
ARCustomers      HZ_CUST_PROFILE_CLASSES      CREDIT_CLASSIFICATION  column added in target database
ARCustomers      HZ_LOCATIONS      VALIDATION_STATUS_CODE  column added in target database
.
.
-----
1)Differences were found for the transaction:<<ARCustomers>>
-----
ARTransactions      RA_CUSTOMER_TRX_ALL      COMMENTS  column size is changed
                                           [Old column size was:240 --
                                           New column size is:1760]
ARTransactions      RA_CUSTOMER_TRX_ALL      REVERSED_CASH_RECEIPT_ID  column added in target database
-----
2)Differences were found for the transaction:<<ARTransactions>>
-----
AutoInvoicing RA_INTERFACE_DISTRIBUTIONS_ALL      Details for this table not found.
                                           Either table is Deleted OR Renamed
                                           in this new release.
.
.
-----
4)Differences were found for the transaction:<<AutoLockBox>>
-----
BillOfMtlIsENG      BOM_INVENTORY_COMPONENTS      SUGGESTED_VENDOR_NAME  Table moved from 'BOM' to 'APPS'
BillOfMtlIsENG      BOM_INVENTORY_COMPONENTS      SUGGESTED_VENDOR_NAME  column added in target database
.
.
-----
13)No difference was found for the transaction:<<CustomerItem>>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```


Appendix I **Components for Custom Inbound Business Objects**

Topics

- [Creating TIBCO Intermediate Tables for Custom Inbound API Business Objects, page 448](#)
- [Writing a PL/SQL Package for Custom API Inbound Business Objects, page 451](#)
- [Writing a PL/SQL Package for Custom Non-API Inbound Business Objects, page 459](#)

Creating TIBCO Intermediate Tables for Custom Inbound API Business Objects

This section explains how to create the destination tables (TIBCO Intermediate Tables) that are required to be created in the Oracle Application Database before configuring a Custom Inbound API Business Objects using the Palette for the TIBCO Adapter for Oracle Applications.

All references to columns in this section are with regard to intermediate tables for API transactions. Most of the Oracle Application APIs take input in the form of PL/SQL record structures specific to the API. A record structure consists of a set of columns that are required by the API. Therefore, the destination tables (TIBCO Intermediate Tables) configured for an API Subscriber should contain these columns.

With this background, the following steps need to be followed for creating a TIBCO Intermediate Table:

1. Identify all the PL/SQL Record Structures that are involved in the API.
2. For each such record structure identified, create a table consisting of all the columns present in the record structure.

The first column should be `TIBCO_KEY`. Refer to Appendix B for further details on this column. If a record structure corresponds to the header record, add the `TIBCO_ADB_...` fields also. (Details of these columns have been mentioned at the end of this section). There is no need for adding these special columns for child tables.

3. Create these tables in the TIBCO User where all database components required for an Oracle Application Adapter service are created.

TIBCO ADB Columns

There are a number of special columns that are added to the Header Destination Table (main TIBCO Intermediate Table).

The columns and their Oracle data-types are as under:

- `TIBCO_ADB_OPCODE` `VARCHAR2 (1)`
- `TIBCO_INT_PROCESS_FLAG` `NUMBER DEFAULT (2)`
- `TIBCO_SOURCE` `VARCHAR2(255)`
- `TIBCO_SOURCE_ID` `VARCHAR2(255)`

Example

This example is based on one of the API Business Objects supplied as a PreDefined Oracle Application Inbound API Business Object.

The TIBCO Intermediate Table for the Split Line API has the following structure:

- DELIVERY_DETAIL_ID NUMBER
- SPLIT_QUANTITY NUMBER
- TIBCO_SOURCE VARCHAR2(255)
- TIBCO_SOURCE_ID VARCHAR2(255)
- TIBCO_ADB_OPCODE VARCHAR2(1)
- TIBCO_INT_PROCESS_FLAG NUMBER (DEFAULT 2)

Calling the PreCommit Stored Procedure

The PreCommit Stored Procedure is called with the following syntax:

```
{call <sp_name>(?, ?, ?)}
```

The stored procedure can be any name, but must have a fixed parameter list: no return code and three OUT parameters:

```
RETURN_CODE int
SP_TEXT varchar
DO_ROLLBACK int
```

For ORACLE, the PL/SQL declaration would look like:

```
CREATE OR REPLACE PROCEDURE <sp_name>
(RETURN_CODE OUT NUMBER, SP_TEXT OUT VARCHAR2, DO_ROLLBACK
NUMBER) ...
```

The adapter will react differently depending on the values of the OUT parameters.

If `DO_ROLLBACK = 0`, the adapter will commit the transaction and confirm the original message.

If `DO_ROLLBACK <> 0`, the adapter will rollback the transaction and not confirm the message. Note, not confirming the message changes the RVCM behavior. All subsequent RVCM messages will not be confirmed. It should only be done when this RVCM behavior is really desired.

If `RETURN_CODE = 0`, the adapter assumes the procedure was successful and writes a success message to the SDK INFO trace role when the verbose mode is used.

If `RETURN_CODE <> 0`, the adapter assumes the procedure was not successful and writes SP_TEXT to the SDK ERROR trace role (whether verbose mode is on or off).

If the original message was sent with a reply subject, the status and return text is sent back to the sender on the specified reply subject.

In case of error records, only the error message is returned. In case of success records, the key fields which get generated are also returned along with the success message in the following format:

```
KeyField-1:KeyValue-1,KeyField-2:KeyValue-2,KeyField-3:KeyValue-3,
...KeyField-N:KeyValue-N.
```

Example message for a PreDefined Customer Item Subscription Transaction:

```
CUSTOMER_ITEM_ID:110, LAST_UPDATED_DATE: 2003-03-12 12:34:03
```

Writing a PL/SQL Package for Custom API Inbound Business Objects

The structure of the PL/SQL package generated has the structure as is mentioned in “Structure Of PL/SQL Packages” section in Appendix B. We recommend that you follow the same architecture.

Note that you have to put code in all the procedures in the package.

The package should be created with the same name as was mentioned in the PreCommit Stored Procedure field in the Subscriber Options tab for the Custom Subscription Configuration.

You should follow the steps mentioned below in order to write the necessary code in the package. A sample is provided taking into account the Split Line Transaction (configured as a Custom API Transaction).

1. **main_adb:-** This is a wrapper procedure called by the adapter in the subscription with reply scenario. This procedure passes three OUT parameters which are required as the reply parameters by the adapter. This procedure calls the procedure “main” with four parameters:-
 - The general parameters for `chr_errbuf` and `chr_retcode`.
 - `chr_concurrent_flag` which whether the PL/SQL procedure is registered as a Concurrent program or not. Can be set to Y or N depending on whether it is to be registered as a concurrent program or not.
 - `chr_skip_validn_flag` which is defaulted to N . Taking the Split Line transactions as an example,

```
PROCEDURE main_adb(
  out_var1 OUT VARCHAR2,
  out_var2 OUT VARCHAR2,
  out_var3 OUT VARCHAR2)
IS
  l_chr_errbuf VARCHAR2(240);
  l_chr_retcode VARCHAR2(240);
  l_chr_concurrent_flag VARCHAR2(1) := 'N';
  l_chr_skip_validn_flag VARCHAR2(1) := 'N';
BEGIN
  main(
    l_chr_errbuf,
    l_chr_retcode,
    l_chr_concurrent_flag,
    l_chr_skip_validn_flag
  );
EXCEPTION
WHEN OTHERS THEN
  NULL;
END main_adb;
```

2. main:- This is the main procedure called by the adapter in the subscription with reply scenario. In this case, this is called from the procedure main_adb. In the simple subscription scenario, this is the starting procedure.

This procedure requires two OUT parameters and two IN parameters. The procedure calls the procedure "<TransName>_Validation" which contains the actual call to the Oracle API.

Taking the Split Line transactions as an example,

```
PROCEDURE main (
  out_errbuf OUT VARCHAR2,
  out_retcode OUT VARCHAR2,
  in_concurrent_flagIN VARCHAR2,
  in_skip_validn_flagIN VARCHAR2)
AS
  l_chr_retcodeVARCHAR2(10):= NULL;
  l_chr_errbufVARCHAR2(100):= NULL;

BEGIN
  out_retcode := '0';
  out_errbuf := NULL;

  IF in_skip_validn_flag = 'N' THEN
    --call the validation program to populate the interface table
    SplitLine_Validation(l_chr_retcode, l_chr_errbuf);

  END IF;
  COMMIT;
  EXCEPTION
  WHEN OTHERS THEN
    fnd_file.put_line (
      fnd_file.LOG,
      'Unexpected error occurred ' ||
      ' The oracle error message is : ' ||
      TO_CHAR (SQLCODE) ||
      SQLERRM
    );
    DBMS_OUTPUT.PUT_LINE(' Inside Exception ' ||SQLERRM);
    out_errbuf :=
      'Unexpected error occurred ' ||
      ' The oracle error message is : ' ||
      TO_CHAR (SQLCODE) ||
      SQLERRM;
    out_retcode := '2';
END main;
```

3. <TransName>_Validation:- This is the procedure where the call to the Oracle API is made and the status (Success or Error) of the API call is logged into the Log Table. Before calling the Oracle API, this procedure calls the procedures pro_pre_<TransactionName> and pro_post_<TransactionName> also.

Taking the Split Line transactions as an example,

```

PROCEDURE SplitLine_Validation(
  out_errbuf      OUT   VARCHAR2,
  out_retcode     OUT   VARCHAR2
)
AS
  /*Variable Declaration Section*/
  c_import_Failed          CONSTANT NUMBER :=1;
  c_waiting_to_be_processed CONSTANT NUMBER :=2;
  c_validation_failed      CONSTANT NUMBER :=3;
  c_validation_success     CONSTANT NUMBER :=4;
  c_import_success        CONSTANT NUMBER :=5;
  w_x_msg_count            NUMBER(10);
  w_x_msg_data             VARCHAR2(1000);
  w_x_return_status        VARCHAR2(1);
  w_x_new_detail_id        NUMBER;
  w_api_version_number     NUMBER;
  w_p_from_detail_id       NUMBER;
  w_x_split_quantity       NUMBER;
  w_error_flag             BOOLEAN;
  w_import_success         BOOLEAN;

  /*This is the input record structure: */
  w_uni_seq_id             NUMBER;
  w_app_short_name         VARCHAR2(100);
  w_msg_name               VARCHAR2(1000);
  w_action_code            VARCHAR2(10);
  w_delivery_name          VARCHAR2(30);
  w_init_msg_list          VARCHAR2(1000);
  w_x_delivery_id          NUMBER;
  w_x_name                 VARCHAR2(30);
  w_x_msg_parsed_data      VARCHAR2(2000);
  w_print_msgdata          VARCHAR2(2000);

  w_uexit_error_flag       BOOLEAN;
  w_bypass_validn          BOOLEAN;
  w_pre_uexit_success_flag VARCHAR2(1);
  w_post_uexit_success_flag VARCHAR2(1);

  c_transaction_name CONSTANT VARCHAR2(10):= 'SplitLin';
  c_table_name        CONSTANT VARCHAR2(40):= 'TIB_INT_WSH_SHIPLINE';
  c_err14             CONSTANT VARCHAR2(20):= 'AEORAP-000014';
  c_err15             CONSTANT VARCHAR2(20):= 'AEORAP-000015';

  POS NUMBER;
  NEXTPOS NUMBER;
  w_status_log_rec pk_tib_log.log_rec_type;
  w_error_log_rec pk_tib_log.log_rec_type;
  w_suc_log_rec pk_tib_log.log_rec_type;
  w_error_source VARCHAR2(10):='ORACLE';
  w_msg_source CONSTANT VARCHAR2(10):='TIBCO';

```

```

/*Cursor for selecting records from TIB_INT_WSH_SHIPLINE Table*/

CURSOR cur_select_wsh_SplitLin IS
SELECT * FROM TIB_INT_WSH_SHIPLINE
WHERE TIBCO_INT_PROCESS_FLAG = c_waiting_to_be_processed FOR
UPDATE;

cur_select_wsh_SplitLin_rec cur_select_wsh_SplitLin%ROWTYPE;

BEGIN

out_retcode:='0';
/*Delete the Rows with status as IMPORT_SUCCESS*/
DELETE FROM TIB_INT_WSH_SHIPLINE WHERE TIBCO_INT_PROCESS_FLAG =
c_validation_success;

COMMIT;

/*Update the Intermediate Process Flag to Waiting to be
Processed.*/
UPDATE TIB_INT_WSH_SHIPLINE
SET TIBCO_INT_PROCESS_FLAG = c_waiting_to_be_processed
WHERE TIBCO_INT_PROCESS_FLAG IS NULL;

/*For Each Record in TIB_INT_WSH_SHIPLINE*/
FOR cur_select_wsh_SplitLin_rec IN cur_select_wsh_SplitLin
LOOP

/*Update the Intermediate Table with unique id.*/

--SELECT TIB_INT_WSH_SHIPLINE_S.NEXTVAL INTO w_uni_seq_id FROM
DUAL;
--UPDATE TIB_INT_WSH_SHIPLINE SET TIBCO_UNIQUE_ID = w_uni_seq_id
--WHERE CURRENT OF cur_select_wsh_SplitLin;

pro_pre_SplitLine(cur_select_wsh_SplitLin_rec,
cur_select_wsh_SplitLin_rec.TIBCO_KEY,
w_pre_uexit_success_flag);

/* Assign the current record into the record structure to be passed
*/
w_api_version_number := 1.0;
w_p_from_detail_id:=cur_select_wsh_SplitLin_rec.delivery_detail_id
;
w_x_split_quantity:=cur_select_wsh_SplitLin_rec.split_quantity;

/*Initialise the Error Flag for new Header Record.*/
w_error_flag := FALSE;
w_import_success := TRUE;

w_uexit_error_flag := FALSE;

IF w_pre_uexit_success_flag = '2' THEN /* Error in Pre and do not
bypass */
w_uexit_error_flag := TRUE;

```



```

ELSIF w_pre_uexit_success_flag = '3' THEN /* Error in Pre and
bypass */
w_uexit_error_flag:= TRUE;
END IF;

/*Call the Post User Exit */
pro_post_SplitLine(cur_select_wsh_SplitLin_rec,
    cur_select_wsh_SplitLin_rec.TIBCO_KEY,
    w_post_uexit_success_flag);

IF w_post_uexit_success_flag = '2' THEN /* Error in Post
Validation */
w_uexit_error_flag:=TRUE;
END IF;

WSH_DELIVERY_DETAILS_PUB.split_line
( w_api_version_number,
  FND_API.G_FALSE,
  FND_API.G_FALSE,
  FND_API.G_VALID_LEVEL_FULL,
  w_x_return_status,
  w_x_msg_count,
  w_x_msg_data,
  w_p_from_detail_id,
  w_x_new_detail_id,
  w_x_split_quantity);

IF w_x_return_status = 'E' THEN
fnd_file.put_line (fnd_file.LOG,' Error: Expected Error');
w_error_log_rec.trans_name:=c_transaction_name;
w_error_log_rec.status:='E';
w_error_log_rec.table_field_name:='';
w_error_log_rec.in_hdr_fldname1:='DELIVERY_DETAIL_ID';
w_error_log_rec.error_source:=w_error_source;
w_error_log_rec.error_code:=c_err14;
w_error_log_rec.table_field_name:='TIB_INT_WSH_SHIPLINE';
w_error_log_rec.description:='E-EXPECTED ERROR';

w_error_log_rec.in_hdr_fldvalue1:=cur_select_wsh_SplitLin_rec.DELI
VERY_DETAIL_ID;
w_error_log_rec.unique_id:=cur_select_wsh_SplitLin_rec.TIBCO_KEY;
pk_tib_log.log_interface_status(w_error_log_rec);
w_error_flag := TRUE;
ELSIF w_x_return_status = 'U' THEN
fnd_file.put_line (fnd_file.LOG,' Error: Unexpected Error');
w_error_log_rec.trans_name:=c_transaction_name;
w_error_log_rec.status:='E';
w_error_log_rec.table_field_name:='';
w_error_log_rec.table_field_name:='TIB_INT_WSH_SHIPLINE';
w_error_log_rec.in_hdr_fldname1:='DELIVERY_DETAIL_ID';
w_error_log_rec.error_source:=w_error_source;
w_error_log_rec.error_code:=c_err15;
w_error_log_rec.description:='U-UNEXPECTED ERROR';

```

```

w_error_log_rec.in_hdr_fldvalue1:=cur_select_wsh_SplitLin_rec.DELI
VERY_DETAIL_ID;
w_error_log_rec.unique_id:=cur_select_wsh_SplitLin_rec.TIBCO_KEY;
pk_tib_log.log_interface_status(w_error_log_rec);
w_error_flag := TRUE;
END IF;

FND_MESSAGE.SET_ENCODED(w_x_msg_data);
w_x_msg_parsed_data := FND_MESSAGE.Get;

null;
POS := 1;

NEXTPOS := INSTR(w_x_msg_parsed_data, chr(0), POS);
w_app_short_name := SUBSTR(w_x_msg_parsed_data, POS, NEXTPOS -
POS);
POS := NEXTPOS + 1;

NEXTPOS := INSTR(w_x_msg_parsed_data, chr(0), POS);
w_msg_name := SUBSTR(w_x_msg_parsed_data, POS, NEXTPOS - POS);
POS := NEXTPOS + 1;

w_print_msgdata := SUBSTR(w_x_msg_parsed_data, POS);

fnd_file.put_line(fnd_file.LOG,w_print_msgdata);

IF w_error_flag =TRUE THEN
UPDATE TIB_INT_WSH_SHIPLINE
SET TIBCO_INT_PROCESS_FLAG = c_validation_failed
WHERE TIBCO_KEY = cur_select_wsh_SplitLin_rec.TIBCO_KEY;
ELSIF w_error_flag =FALSE THEN
UPDATE TIB_INT_WSH_SHIPLINE
SET TIBCO_INT_PROCESS_FLAG = c_validation_success
WHERE TIBCO_KEY = cur_select_wsh_SplitLin_rec.TIBCO_KEY;
w_suc_log_rec.trans_name:=c_transaction_name;
w_suc_log_rec.error_source:='';
w_suc_log_rec.description:='IMPORT SUCCESS';
w_suc_log_rec.status:='S';
w_suc_log_rec.table_name:='';
w_suc_log_rec.in_hdr_fldname1:='DELIVERY_DETAIL_ID';

w_suc_log_rec.in_hdr_fldvalue1:=cur_select_wsh_SplitLin_rec.DELIVE
RY_DETAIL_ID;
w_suc_log_rec.unique_id:=cur_select_wsh_SplitLin_rec.TIBCO_KEY;
pk_tib_log.log_interface_status(w_suc_log_rec);

END IF;

END LOOP;

COMMIT;

EXCEPTION
    WHEN OTHERS
    THEN
        fnd_file.put_line (

```

```

        fnd_file.LOG,
        'Unexpected error occurred ' ||
        ' The oracle error message is : ' ||
        TO_CHAR (SQLCODE) ||
        SQLERRM
    );
--    DBMS_OUTPUT.PUT_LINE(' This is the Outside Exception
' || SQLERRM);
    out_errbuf :=
        'Unexpected error occurred ' ||
        ' The oracle error message is : ' ||
        TO_CHAR (SQLCODE) ||
        SQLERRM;
    out_retcode := '2';
END SplitLine_Validation;

```



Take the case of API transactions where the business object to be passed to the API call has a parent -->child-->grandchild table relation. In this case the cursors used to fetch the child and grandchild records need to have the Oracle Business Keys and the TIBCO_KEY in their corresponding WHERE clauses. This is to make sure that correct grandchild records are selected corresponding to their parent.

4. pro_pre_<TransactionName>:- This procedure is called by the <TransactionName>_Validation

procedure. This procedure allows you to carry out your own modifications to records.

Taking the Split Line transactions as an example

```

PROCEDURE pro_pre_SplitLine(
p_header_rec IN OUT TIB_INT_WSH_SHIPLINE%ROWTYPE,
p_unique_id IN NUMBER,
p_return_status OUT VARCHAR2 )
AS
w_count BINARY_INTEGER := 0;
BEGIN
w_count := w_count + 1;
END pro_pre_SplitLine;

```

5. pro_post_<TransactionName>:- This procedure is called by the <TransactionName>_Validation

Procedure. This procedure allows you the flexibility of carrying out referential integrity checks. However, you should not modify record values here.

Taking the Split Line transactions as an example

```

PROCEDURE pro_post_SplitLine(
p_header_rec IN TIB_INT_WSH_SHIPLINE%ROWTYPE,
p_unique_id IN NUMBER,

```

```
p_return_status IN OUT VARCHAR2 )  
AS  
w_countBINARY_INTEGER := 0;  
BEGIN  
w_count := w_count + 1;  
END pro_post_SplitLine;
```

Writing a PL/SQL Package for Custom Non-API Inbound Business Objects

In case of Non API transactions also, the PL/SQL package generated has the structure as is mentioned in Structure Of PL/SQL Packages section in Appendix B. We recommend that you follow the same architecture.

Note that the procedures <TransName_Validation>, pro_post_<TransactionName> and pro_post_<TransactionName> is created fully and that you need not go and modify these.

You have to code for the Submission of the Concurrent Program and the Error/Success Logging mechanism.

You have to code for the following:-

1. main_adb:- This is a wrapper procedure called by the adapter in the subscription with reply scenario. This procedure passes three OUT parameters which are required as the reply parameters by the adapter. In case of simple subscription, this procedure is not required. This procedure calls the procedure main.
2. main:- The first thing a user configuring a Non-API Custom Subscription Transaction should code inside the main Procedure is to set the Profile, APPS and Client Info so that the call to the Oracle Import Program is successful.

Sample Code:

```
IF in_concurrent_flag = 'N' THEN
    IF fnd_request.set_options(
        'NO',          -- Implicit(nature of the request to be submitted)
        'NO',          -- protected(Is the request protected against
updates)
        'AMERICAN', -- language (Important)
        'AMERICA',   -- territory (Important)
        null)
    THEN
        FND_CLIENT_INFO.setup_client_info(
            in_resp_appl_id,    -- Responsibility Appl ID
            in_resp_id,        -- Responsibility ID
            in_user_id,        -- User ID
            null);
        FND_PROFILE.INITIALIZE(
            in_user_id,        -- User ID
```

```

in_resp_id, -- Responsibility ID
in_resp_appl_id, -- Responsibility Appl ID
NULL);
FND_GLOBAL.APPS_INITIALIZE(
in_user_id, -- User ID
in_resp_id, -- Responsibility ID
in_resp_appl_id); -- Responsibility Appl ID
ELSE
dbms_output.put_line(fnd_message.get);
END IF;
END IF;

```

`in_resp_appl_id`, `in_resp_id` and `in_user_id` are set in the outermost `main_adb` procedure in the package. From `main_adb` these parameters are passed into the main procedure.

This procedure will call the Submit Import program which will submit the concurrent request and subsequently do the Error/Success Logging to the Log Table.

— To submit the import program, the `submit_<TransName>` is called.

— Success/Error Logging mechanism:-

To find the errors that were logged after running the concurrent request, you need to identify the table (either the Interface or an Error Table) where the errors are getting logged. This is transaction specific and you may need to open the Oracle Metalink or get in touch with Oracle Support to find out the error table for each transaction.

Then log these into the `TIB_INT_LOG_SUB` table present in the User schema. You can use the procedure `pk_tib_log.log_interface_status` present in the `common_all.sql` script `$TIBCO_ADAPTER_HOME\config\Sub` to log the Errors into the `TIB_INT_LOG_SUB` table.

To find the records that have been successfully added/modified in the Oracle Source Tables, you need to identify the proper Source Tables and log the status of the records that have been added/modified into the `TIB_INT_LOG_SUB` table present in the User schema. You can use the procedure `pk_tib_log.log_interface_status` present in the `common_all.sql` script `$TIBCO_ADAPTER_HOME\config\Sub` to log the Success into the `TIB_INT_LOG_SUB` table.

Appendix J **User Names for PreDefined Oracle Outbound Business Objects**

Topics

- [User Names for PreDefined Oracle Outbound Business Object, page 462](#)

User Names for PreDefined Oracle Outbound Business Object

This appendix gives the User names(Source Table User1 and Source Table User2) to be used for configuring the PreDefined Oracle Outbound Business Objects(Transaction Names).

Table 43 User Names for PreDefined Oracle Outbound Business Objects

Business Object Name	Source Table User1	Source Table User2
Engineering BOM	BOM	
GL Balances	GL	
GL Journals	GL	
Item Category	INV	
Items	INV	
On-hand quantity	INV	
BOM Revisions	BOM	INV
BOM Routings	BOM	
Manufacturing BOM	BOM	
Purchase Orders	PO	
Suppliers	PO	
Sales Orders	ONT	
Invoices	AP	
Checks	AP	
AR Transactions	AR	
AR Customers	AR	AP
Pick Details	INV	
EnggChangeNotification	INV	BOM

Business Object Name	Source Table User1	Source Table User2
ApplicantPub	HR	
EmployeePub	HR	
InteractionPub	JTF	

Appendix K **Troubleshooting Guide for TIBCO Oracle Applications Adapter**

This section gives details on standard errors likely to arise during deployment of the TIBCO Oracle Applications on Subscription mode. A few cases have been listed below with their possible resolutions.

Topics

- [Insertion into Oracle Interface table failed, page 466](#)
- [Oracle Concurrent Program Submission Failed, page 467](#)
- [Insertion into Oracle Source Table Failed, page 468](#)
- [Oracle Application Concurrent Manager is Down, page 469](#)
- [Insertion into TIBCO Intermediate Table Failed, page 470](#)
- [Running adbagent.exe Failed, page 471](#)
- [Running adorapps.dta/adb.dta Failed, page 474](#)
- [Error When Running the Adapter in Publication Mode, page 475](#)
- [Modifying C-Shell to Other Shell Environments, page 479](#)
- [Unable to Set Environment Variables on a Solaris Box, page 480](#)
- [Unable to Create Trigger on MLOG Tables, page 482](#)
- [Errors During Subscription and Subscription with Reply Services, page 483](#)

Insertion into Oracle Interface table failed

Inbound Data gets successfully inserted by the TIBCO Oracle Applications Adapter into the TIBCO Intermediate Table. After the Pre-Commit Stored Procedure execution is over, the data is still in the TIBCO Intermediate Table with `TIBCO_INT_PROCESS_FLAG` updated as 1 for the Header record.

Solution

The Updation of `TIBCO_INT_PROCESS_FLAG` to 1 for the Header record in the TIBCO Intermediate Table means that the Insertion into Oracle Interface Table failed. This is mainly due to violation of some constraint on the Oracle Interface Table either at the header or line level depending on the transaction. In this case check the Oracle Interface Table constraints(Index and Field level) and correct the inbound data accordingly.

Now delete the TIBCO Intermediate Table Entry and re-run the transaction from the Source after making changes to the in bound data.

Oracle Concurrent Program Submission Failed

Inbound Data gets successfully inserted by the TIBCO Oracle Applications Adapter into the TIBCO Intermediate Table. After the Pre-Commit Stored Procedure execution is over, the TIBCO LOG Table has an entry with DESCRIPTION Field as Oracle Concurrent Program Submission Failed. Please check if the values of Responsibility Id, Responsibility Application Id and User Id are correct. The data in the TIBCO Intermediate Table has TIBCO_INT_PROCESS_FLAG updated as 5 for the Header record.

Solution

The Updation of TIBCO_INT_PROCESS_FLAG to 5 for the Header record in the TIBCO Intermediate Table means that the Insertion into Oracle Interface Table succeeded. As the error message suggests you have not given correct values for LANGUAGE, USER, RESPONSIBILITY and APPLICATION from the front end during configuration. If you need to avoid reconfiguring the repository (which is ideal) you have to manually edit the TIBCO Package generated under <TIBCO_HOME>\adapter\adorapps\5.3\sql and recompile it in APPS schema. To do this, open the TIBCO Package and go to the last part of the Package Body which defines the main_adb procedure and make changes to the assignments made to the following variables.

1. l_num_user_id
2. l_num_resp
3. l_num_resp_appl_id



Please provide the correct USER_ID, RESPONSIBILITY_ID and RESPONSIBILITY_APPLICATION_ID for these variables respectively. During configuration you specify LANGUAGE, USER, RESPONSIBILITY and APPLICATION which translate into these IDs within the procedure.

Now re-run the transaction from the Source.

Insertion into Oracle Source Table Failed

After the pre-commit stored procedure execution is complete the TIBCO_LOG Table has an entry Request Errored out. Please check Log File generated for details. The data in the TIBCO Intermediate Table has TIBCO_INT_PROCESS_FLAG updated as 5 for the Header record.

Solution

This error can occur due to the following reasons:

1. The Log files in APPS are full, ask your DBA to clear the Oracle Applications File Logs and then re-run the transaction from the source.
2. In case of Transactions where Logs are getting generated check the APPS Log File which should clearly indicate the problem. For example in case of Open Budget Transaction erroneous data causes the Open Budget Import to fail.

Oracle Application Concurrent Manager is Down

After the pre-commit stored procedure is called, the adapter hangs. The data in the TIBCO Intermediate Table has `TIBCO_INT_PROCESS_FLAG` updated as 5 for the Header record.

Solution

The Updation of `TIBCO_INT_PROCESS_FLAG` to 5 for the Header record in the TIBCO Intermediate Table means that the Insertion into Oracle Interface Table succeeded. Hanging of the TIBCO Oracle applications adapter can occur if the Oracle Applications Concurrent Manager is down. In such cases the adapter has to be brought down and once the Concurrent Manager is active the transaction can be re-run from the source.

Insertion into TIBCO Intermediate Table Failed

The TIBCO Oracle applications adapter starts successfully. On subscribing to the inbound message the adapter fails to insert data into the TIBCO Intermediate Tables.

Solution

TIBCO Intermediate Tables have no constraints on the data. So insertion cannot fail at this stage as a result of constraint violation. The reasons could be one of the following:

1. TIBCO Intermediate Table's Tablespace could be full. In this case ask the database administrator to create adequate space in the Tablespace and re-run the transaction from the source.
2. Sometimes the inbound data type might not match the data type of the fields in the TIBCO Intermediate Table. For example if the adapter tries to insert an `xs` type data into a `VARCHAR` field in the TIBCO intermediate Table this can happen. Such errors are logged in the adapter log file created under `<TIBCO_HOME>\adapter\adorapps\5.1\logs`. In this case please make the required change to the inbound data and re-run the transaction from the source.

Running adbagent.exe Failed

Symptom 1

When I run `adbagent.exe` from the command prompt it returns an error stating:

```
Config Name: ActiveDatabaseAdapterConfiguration
Main thread (name: ActiveDatabaseAdapterConfiguration)
MException thrown in ADBAgent::threadFunction().
Code = AESDKC-0042, Category = Database, Severity = errorRole,
Description = Connection to TIB CR failed. no repoURL given,
File =
D:/build/Maverick/maverick-4.1.2/libmaverick/MAppSupportingClass
Impl.cpp, line = 500
```

Solution

The connection to the specified repository failed because the adapter could not locate the repository.

If you have specified a remote repository for the `tibco.repourl` property in the `.tra` file, make sure the applicable repository server is up. If you have specified a local repository file in the `<repositoryInstance>` parameter, make sure the path is correct and the file is not corrupt.

Symptom 2

When I run `adbagent.exe` from the command prompt it returned an error stating:

```
Main thread (name: ActiveDatabaseAdapterCnfiguration)
MException thrown in ADBAgent::threadFunction().
Code = AESDKC-0064, Category = Database, Severity = errorRole,
Description = Repository Path caused a repository exception.
NodeNotFoundException
Node =
/tibco/private/adapter/ADB/ActiveDatabaseAdapterCnfiguration
RepoException
node does not exist, File =
D:/build/Maverick/maverick-4.1.2/libmaverick/MRepoProperties.cp,
line = 328
```

Solution

The connection to the specified repository failed because the adapter instance name was not specified correctly.

Ensure that a correct instance name is specified both in the TIBCO Designer configuration and the `adbagent.tra`.

Symptom 3

When I run `adbagent.exe` from the command prompt it returns an error stating:

```

2004 Feb 13 11:06:04:797 GMT 5 MetadataAdapter.MetadataAdapter
Info [Adapter] AEADB-700033 call succeed!
com.tibco.sdk.MException: Operation error: unable to create Tibrv
Session for null(MPublisher).
    at com.tibco.sdk.events.pubsub.MRvPublisher.sendWithReply
    (MRvPublisher.java:112)
    at com.tibco.sdk.events.MPublisher.sendWithReply
    (MPublisher.java:403)
    at com.tibco.sdk.events.MPublisher.sendWithReply
    (MPublisher.java:387)
    at com.tibco.ae.adb.metadataadapter.MetadataAdapterShell.
    onInitialization (MetadataAdapter.java:146)
    at com.tibco.sdk.MDefaultApp.onInitialization
    (MDefaultApp.java:535)
    at com.tibco.sdk.MDefaultApp.start(MDefaultApp.java:351)
    at com.tibco.sdk.MApp.start(MApp.java:184)
    at com.tibco.ae.adb.metadataadapter.MetadataAdapter.<init>
    (MetadataAdapter.java:47)
    at com.tibco.ae.adb.metadataadapter.MetadataAdapter.main
    (MetadataAdapter.java:78)
2004 Feb 13 11:06:04:857 GMT 5 MetadataAdapter.MetadataAdapter
Info [Adapter] AEADB-700063
Design Time Adapter starts

```

Solution

The system `classpath` is not set correctly.

Ensure that the `classpath` contains the path to the TRA `bin` directory.

Symptom 4

When I run `adbagent.exe` from the command prompt it returns an error stating:

```

Database driver code: IM002
Database server code: 0
Database driver message: [Microsoft][ODBC Driver Manager] Data
source name not found and no default driver specified

```

Solution

The probable cause maybe incorrect DSN settings.

Ensure that you have an entry in the System DSN settings for your database instance with proper ODBC Driver.

Symptom 5

When I run `adbagent.exe` from the command prompt it returns an error stating:

```

Instance ID: OracleApplication
Config Name: OracleApplication
Main thread (name: OracleApplication )

```

```
MException thrown in ADBAgent::threadFunction().  
Code = AESDKC-0048, Category = Database, Severity = errorRole,  
Description = Repository Exception : Could not connect to  
Repository, File = S/MRepositoryConnection.cpp, line = 78
```

Solution

The repository may be corrupted.

Re-configure the repository and specify proper repository path and the proper instance name.

Running adorapps.dta/adb.dta Failed

Symptom

When I run `adorapps.dta/adb.dta` from the command prompt it returns an error stating:

```
Cannot initialize RV: Version mismatch: tibrvj shared library  
version 7.2 does not match version of Processing  
/tibco/private/adapter/ADOrapps/ADOrappsMetadataAdapter/ADOrapps  
MetadataAdapter ...
```

Solution

The system `classpath` is not set correctly.

Ensure that the `classpath` contains the path to the appropriate TRA `bin` directory.

Error When Running the Adapter in Publication Mode

Symptom 1

When the adapter runs in the Publication mode, an error message appears as follows:

```
2004 Feb 13 11:32:20:045 GMT 5
ActiveDatabaseAdapterConfiguration.ActiveDatabaseAdapterConfigur
ation Error [Database] AEADB-100004
Database driver code: 42S02
Database server code: 942
Database driver message: [Oracle][ODBC][Ora]ORA-00942: table or
view does not exist
```

Solution

The corresponding publishing table is absent in the **adapter user** schema.

Ensure that the `BO_ALL.sql` is run after the Publication configuration.

Symptom 2

When the adapter runs in the Publication mode, an error message appears as follows:

```
MException throw in
ADBDatabaseInterface::handlePollCommandRequest():
tracking=#Nv6--2--E0VVTE6k-/ukzzwkX-zzw#
2004 Feb 10 15:44:46:875 GMT 11 ORACLE11Vendor.ORACLE11Vendor
Error [Metadata] A
ESDKC-0087 Class description not available for:
P_VEND_REF_ITEM_PUB tracking=#Nv 6--2--E0VVTE6k-/ukzzwkX-zzw#
```

Solution

The publication instance might not have been configured correctly.

Ensure that the Publication Service is configured correctly, and the proper `BO_UNDOALL.sql/BO_ALL.sql` scripts are run correctly under the relevant schemas.

Symptom 3

When I do the following:

- Create and configure a **PreDefinedOutboundBusinessObject**.
- Select **Oracle PublicationService**, and select **Items** for the **OracleBusinessObject**.

- Click **Deploy** in the **OracleApplication Business Object**.

I get the following error:

```
"saveFailed: java.sql.SQLException: [tibcosoftwareinc][Oracle
JDBC Driver]Internal error: Net8 protocol error."
```

Solution

The Items PreDefined Outbound transaction or any custom transaction table having more than 255 columns will not work with Merant JDBC Driver packaged with the Adapter installation. This is a known issue with the Merant Driver.

The work around for the issue is as follows:

For Items PreDefined Outbound transaction or any custom transaction table having more than 255 columns, use Oracle JDBC Driver that is packaged with Oracle Client installation. However, for transactions with smaller tables, the Merant JDBC Driver packaged with the Adapter Installation can be used.

Symptom 4

I have installed TIBCO Adapter for Oracle Applications. When I configure the adapter I am unable to find the TIBCO Adapter for Oracle Application Configuration item and the icon in the palette panel of TIBCO Designer. In the **Add Resources** menu, I find the Oracle application adapter menu item and seven sub menu items, for example pub/sub services, out/in bound. But, I am unable to find the TIBCO Adapter for Oracle Applications configuration item.

Solution

TIBCO Adapter for Oracle Applications does not have a palette of its own. It depends on the ADB palette. As such is the case, the installation is correct. For more information refer to Chapter 4, Configuring the Adapter and Chapter 5, Configuration Reference of the *TIBCO Adapter for Oracle Applications, User's Guide*.

Symptom 5

I get the following error when I try to generate the Publication service on the table HR.HR_ALL_ORGANIZATION_UNITS:

```
The following error has occurred:
ORA-00600: internal error code, arguments: [kkzucts], [8], [],
[], [], [], [], []
```

Solution

The Materialized view logs might have got created with columns of LONG type.

This is an Oracle Application limitation.

If you select **Publish by Value** while configuring TIBCO Adapter for Oracle Application, the Publishing tables cannot contain columns with LONG data types. These restrictions do not apply to publishing tables when you publish by reference and LONG or LONG RAW are non-key types.

For more information refer to section Configuration Reference in Chapter 5 of *TIBCO Adapter for Oracle Applications, User's Guide*.

Symptom 6

When I configure the Business Object using Views and associate it with a publication service I get the following error when deploying it on the database:

```
2003 Oct 8 18:12:04:289 GMT -5 CUSTAGENTORAPUB.CUSTAGENTORAPUB
Error [Configuration] AEADB-200003
Class description not found: tib_pub_hz_parties_v.
tracking=#OGBNEM7EDsG9h-6UYWcta7wJ3-90-#
```

Solution

Ensure that the publication configuration is correct. For more information refer to *TIBCO Adapter for Oracle Applications Users Guide*.

Symptom 7

How do I publish data from two publishers that use the same publishing views? For example, I have publisher A that publishes view `tib_pub_a_v`, and a publisher B that publishes view `tib_pub_a_v`.

Solution

This is an adapter limitation.

This limitation exists only when you try and configure two publishers on the same view within the same adapter instance.

If you have these in separate adapter instances this issue will not arise. If your requirement is to have both the publishers in the same adapter instances then it is not possible.

Symptom 8

I try to configure the TIBCO Adapter for Oracle Applications and I encounter the following problem after I configure a Custom Outbound Business Object, click the **get table from other schema** icon, and select the PO schema.

The TIBCO Designer retrieved only two tables: `PO_CONTROL_GROUPS_ALL` and `PO_CONTROL_ROLES`, even though the other tables are available in the database.

Solution

This occurs due to insufficient permissions on the PO tables. The same holds good for other schemas as well. In order to grant right to the TIBCO User, you need to run the following query: `GRANT SELECT ON <TableName> to tibco.`

Modifying C-Shell to Other Shell Environments

Symptom

I am unable to modify the existing C-Shell to other Shell environments.

Solution

The `adorappssetenv.sh` file shipped along with the adapter sets the environment variables for C shell only. If you have a shell environment other than C shell, for example, Korn shell, then you can use this file as reference to create the particular environment file. The changes would be minimal, say, the equivalent for

```
setenv <environment variable name> <environment variable value>  
would be <environment variable name>=<environment variable value>.  
For example, the equivalent of setenv TIBCO_HOME /usr/tibco is  
TIBCO_HOME=/usr/tibco.
```

Unable to Set Environment Variables on a Solaris Box

Symptom

I am unable to set the correct environment variables on a Solaris box.

Solution

Ensure that you specify your environmental settings correctly. Check if the PATH, CLASSPATH and LD_LIBRARY_PATHS are correct and contain the following:

1. **setenv PATH:**

```
/usr/installtest/tra/<version_num>/lib:/usr/installtest/tra/<version_num>/bin:${ODBC_HOME}/bin:${ORACLE_HOME}/bin:${TIBCO_DESIGNER_HOME}/bin:${TIBCO_TRA_HOME}bin:${TIBCO_JAVA_HOME}:${TIBCO_ADORAPPS_HOME}/bin:${TIBCO_TRA_HOME}/tools/bin:${PATH}
```

2. **setenv CLASSPATH:**

```
/usr/installtest/tra/<version_num>/lib/jdbc/TIbase.jar:/usr/installtest/tra/<version_num>/lib/jdbc/TIoracle.jar:/usr/installtest/tra/<version_num>/lib/jdbc/TIutil.jar:${ORACLE_HOME}:/usr/installtest/tra/<version_num>/lib/libtibrvj.so:/usr/installtest/tra/<version_num>/jre/<version_num>/lib:${ODBC_HOME}/lib/libodbc.so:${ODBC_HOME}:/usr/installtest/tra/<version_num>/lib:${CLASSPATH}
```

3. **setenv LD_LIBRARY_PATH:**

```
${ODBC_HOME}/lib:${ORACLE_HOME}/lib:${TIBCO_DESIGNER_HOME}/lib:${ODBC_HOME}/lib/libodbc.so:${TIBCO_TRA_HOME}/lib:${TIBCO_JAVA_HOME}/jre/lib/sparc:${TIBCO_ADORAPPS_HOME}/lib:/usr/installtest/tra/<version_num>/lib:/usr/installtest/tra/<version_num>/lib/ext:/usr/installtest/
```

```
tra/<version_num>/lib/jdbc:${LD_LIBRARY_PATH}
```

Note that the variables used above are as follows:

```
setenv ORACLE_HOME /local/oracle/OraHome1
setenv TIBCO_HOME /usr/installtest
setenv TIBCO_TRA_HOME /usr/installtest/tra/<version_num>
setenv TIBCO_JAVA_HOME
/usr/installtest/tra/1.0/jre/<version_num>
setenv TIBCO_ADORAPPS_HOME
/usr/installtest/adapter/adorapps/<version_num>
setenv TIBCO_DESIGNER_HOME
/usr/installtest/designer/<version_num>
setenv JDBC_HOME /local/oracle/OraHome1/jdbc
setenv ODBC_HOME
/usr/installtest/adapter/adorapps/<version_num>/adb/odbc
```


Unable to Create Trigger on MLOG Tables

Symptom

When I try to create trigger on MLOG tables `ONT.MLOG$_OE_ORDER_HEADERS_ALL`, `ONT.MLOG$_OE_ORDER_LINES_ALL`, `ONT.MLOG$_OE_SALES_CREDITS` I get the following error message: Table or View does not exist.

Solution

Ensure that you have proper grant permissions on the tables and Views that you have configured onto the **adapter user** schema. You might also want to run the `common_all.sql` script before the configuration of the adapter instance. This will give you the necessary access privileges for your **adapter user** schema.

Errors During Subscription and Subscription with Reply Services

Symptom 1

I use a subscription service and the transaction gets inserted in the Employee subscription staging table (TIB_INT_HRMS_EMP). But the TIBCO_UNIQUE_ID and INSERT_UPDATE_MODE fields are not populated.

Solution

Ensure that relevant permissions and grants are given for each table and execution of the package.

Symptom 2

I use a Subscription transaction to insert data into Oracle tables. The child records do not get inserted into the AP.AP_INVOICE_LINES_INTERFACE table even if it is successfully inserted into the TIB_INT_AP_INV_LINES_IFACE table, though the parent record is inserted correctly.

Solution

For the Inbound message to get processed correctly the TIBCO_KEY field has to be compulsorily populated with the same value for the header and the child. This holds good for all the Subscription transactions.

For example, both the AP.AP_INVOICES_INTERFACE and AP.AP_INVOICE_LINES_INTERFACE records should have the same TIBCO_KEY to get identified as a single Invoice Object by the TIBCO Procedure.

Appendix L **End-to-End Traceability**

This appendix provides information on End-to-End Traceability in TIBCO Adapter for Oracle Applications.

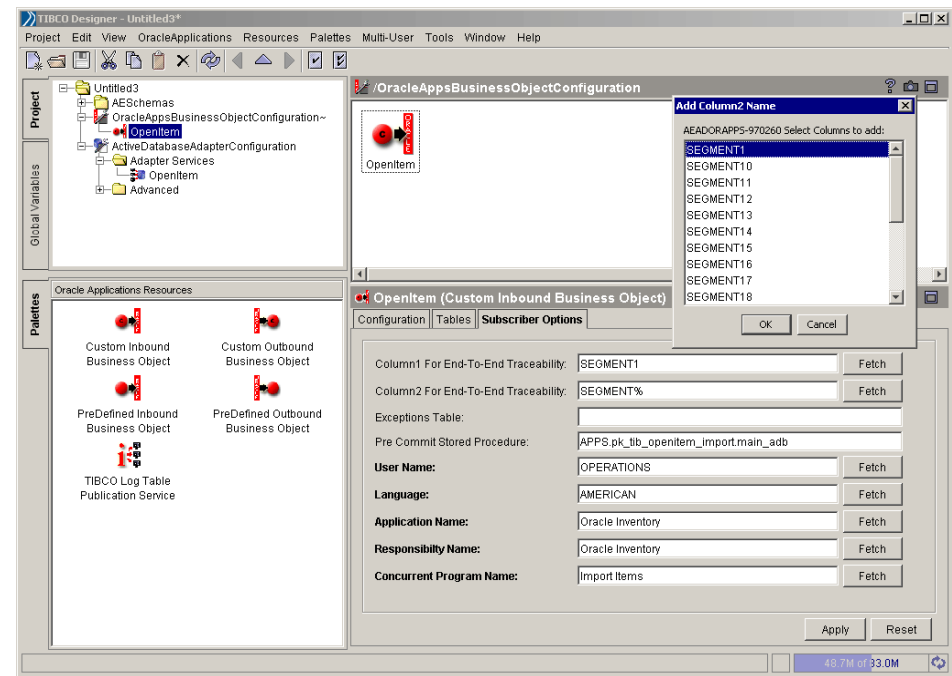
Topics

- [End-to-End Traceability, page 486](#)

End-to-End Traceability

The TIBCO Adapter for Oracle Applications adapter supports end-to-end traceability for the Subscription service. A record inserted by the adapter is mapped onto the Source from which it originates.

Figure 133 End-To-End Traceability



Non-API Subscription Transactions

In case of non-API subscription transactions, the Palette GUI gives the user the option to expose two columns in the Oracle Interface table for *End-to-End* tracking. This is specified in the Subscription Options tab of a PreDefined Inbound Business Object.

The user must expose both columns for End-To-End Traceability. Incoming records are tracked for *End-To-End Traceability* based on the columns selected by the user.

The **Column1 for End-To-End Traceability** drop-down lists all columns common to the interface table and the source table for a particular transaction. A column selected by the user is used to track records which have succeeded. The TIBCO_SOURCE and the TIBCO_SOURCE_ID columns help the user identify records that have succeeded.

The **Column2 for End-To-End Traceability** lists all columns common to the interface table and the error table for a particular transaction. A column selected by the user is used to track records which error out. The TIBCO_SOURCE and the TIBCO_SOURCE_ID columns help the user identify records that have errored out.

In any non-API subscription transaction supporting End-To-End Traceability, the values for **Column1 for End-To-End Traceability**, **Column2 for End-To-End Traceability**, TIBCO_SOURCE and TIBCO_SOURCE_ID are mandatory. If any of the four columns have no value, then End-To-End Traceability will not be supported. However, the user can choose to leave all four fields blank and use the existing subscription feature.

Transactions for which attribute columns are present in both drop-downs

- Autolockbox
- Customeritem
- CustomerItemXRef
- OpenItem
- WorkOrder
- PayablesOpen
- OpenTransactions
- Open Move
- OpenForecast
- OpenReplenishment
- MasterSchedule
- JournalImport
- MassAdditions
- OpenResource

Follow any of the points below to ensure that End-To-End Traceability is achieved for a single record or multiple records involving the transactions listed above.



Each transaction listed above has attribute columns present under both the drop-down lists. These attribute columns may not be used for any Oracle Applications function.

- Select attribute columns in both the drop-down lists and enter unique values. The record will be traced back to the source based on the unique values entered.
- You may opt not to choose any of the attribute columns if one of the columns (other than the attribute columns) in the input data is always unique for each record and is present in both the drop-down lists. Ensure that you choose the unique column in both the drop-down lists. The record will be traced back to the source based on this unique value.
- You may opt not to choose any of the attribute columns if two of the columns in the input data are unique for each record and are present separately in both the drop-down lists (one in each drop-down list).

Transactions for which attribute columns are not present in one or both the drop-down lists

- AutoInvoicing
- Bills of Materials
- BudgetUpload
- CollectionImport
- DemandSchedules
- OpenBudget
- OrderEntry
- PeriodicCost
- PurchasingOpenDocs
- RecCustomer
- Requisition
- CycleCountEntries

Follow any of the points below to ensure that End-To-End Traceability is achieved for a single record or multiple records involving the transactions listed above.

- You may opt not to choose any of the attribute columns if one of the columns (other than the attribute columns) in the input data is always unique for each record and is present in both the drop-down lists. Ensure that you choose the unique column in both the drop-down lists. The record will be traced back to the source based on this unique value.
- If one of the columns in the input data is always unique for each record and is present in only one drop-down list with attribute columns in the other drop-down list, then you can opt to choose one column from one drop-down list and one attribute column from the other. The record will be traced back to the source based on this unique value.
- If none of the columns in the incoming data are unique for each record or the columns that are unique are not present in both drop-down lists, then

End-To-End Traceability for multiple records cannot be achieved. However, End-To-End Traceability for single records can still be achieved. Select two columns from the two drop-down lists (one column from each drop-down list), for which values are always entered in the incoming record. This is based on the assumption that the same set of records are not present in the Intermediate table. Please note that the previous successful records are deleted from the TIBCO intermediate table for each run of the adapter. See, [Clean-up Strategy for Intermediate Tables on page 426](#) for more information.

Errors in End-To-End Traceability for Non-API Transactions

- In the TIB_INT_LOG_SUB table,
 - Value of TIBCO_SOURCE = **WARNING: End to End Traceability not Feasible**
 - Value of TIBCO_SOURCE_ID = **WARNING: End to End Traceability not Feasible**

Reason You have not selected the Oracle Interface table columns in the Subscription options tab during configuration.

Solution To avoid the warning, select the columns in the Subscription Options tab.

- In the TIB_INT_LOG_SUB table,
 - Value of TIBCO_SOURCE = **WARNING: End to End Traceability not Feasible: TIBCO_SOURCE and TIBCO_SOURCE_ID, ATTRIBUTE1 to be Unique and Not NULL in TIB_INT_MTL_SYS_ITEMS_IFACE**
 - Value of TIBCO_SOURCE_ID = **WARNING: End to End Traceability not Feasible: TIBCO_SOURCE and TIBCO_SOURCE_ID, ATTRIBUTE1 to be Unique and Not NULL in TIB_INT_MTL_SYS_ITEMS_IFACE**

Reason You have not entered values for the columns TIBCO_SOURCE, TIBCO_SOURCE_ID and ATTRIBUTE1.

Solution For the columns TIBCO_SOURCE, TIBCO_SOURCE_ID and ATTRIBUTE1, enter the values present in the columns of the header intermediate table TIB_INT_MTL_SYS_ITEMS_IFACE.

- In the TIB_INT_LOG_SUB table,
 - Value of TIBCO_SOURCE = **WARNING: End to End Traceability not Feasible: TIBCO_SOURCE and TIBCO_SOURCE_ID, ATTRIBUTE1 to be Unique and NOT NULL in Intermediate Tables**
 - Value TIBCO_SOURCE_ID = **WARNING: End to End Traceability not Feasible: TIBCO_SOURCE and TIBCO_SOURCE_ID, ATTRIBUTE1 to be Unique and NOT NULL in Intermediate Tables**

Reason You have not entered values for the columns TIBCO_SOURCE, TIBCO_SOURCE_ID and ATTRIBUTE1.

Solution For the columns TIBCO_SOURCE, TIBCO_SOURCE_ID and ATTRIBUTE1, enter the values present in the columns of the Header Intermediate table and Child tables.



The incoming records will be traced back to the source only if submission of the oracle concurrent program has succeeded, i.e., only when the records go to Oracle error table or Oracle source table.

End-To-End Traceability will not be supported when there is an error in invoking the Concurrent Program. In this case, the incoming records remain in the Oracle Interface table. In such cases, the columns TIBCO_SOURCE and the TIBCO_SOURCE_ID in the log table TIB_INT_LOG_SUB will contain the value **concurrent program** indicating that invocation of oracle concurrent program failed and End-To-End traceability will not be supported.

API Subscription Transactions

For an API subscription transaction, the palette GUI does not have drop-down lists (to select columns), as there are no interface tables for API transactions. However, End-to-End traceability is still supported.

Ensure that each incoming record has unique vales of TIBCO_SOURCE and TIBCO_SOURCE_ID. If either of the fields TIBCO_SOURCE or TIBCO_SOURCE_ID is left blank, End-To-End Traceability will not be supported and the log table displays the message **WARNING: End to End Traceability not Feasible: TIBCO_SOURCE and TIBCO_SOURCE_ID cannot be NULL in TIB_INT_ECO_ENGR_CHANGE Table**, in the TIBCO_SOURCE and TIBCO_SOURCE_ID entries. However, you can choose to leave the two fields blank and use existing subscription features.

Appendix M **Logging Enhancements**

This appendix describes enhancements to error logging.

Topics

- [Overview, page 492](#)
- [Usage scenarios for enhanced logging, page 493](#)
- [Changes for enhanced logging, page 495](#)

Overview

Logging, for end-to-end traceability, now has certain changes and enhancements. These are related to the Concurrent Program (CP) Status Logging, details of log records, and log levels.



- Enhanced logging applies to both API and NON-API Subscriber transactions. It is also assumed that the log message length for a typical transaction will be less than 4000 characters.
- This new logging functionality only supports Oracle Applications 11.5.10.

CP Status Logging

The Concurrent Program (CP) status is logged at all the times, that is, in case of both successful submission and submission failure.

Log record details

The Log record of the concurrent program provides details such as:

- Completion Status
- Completion Text
- Log File Location
- The Node on which the log file is located
- Request ID

Log Levels

At the record level, the system provides three levels of logging:

- **Logs for successful records**
Records which were successfully moved to the Oracle Applications tables.
- **Logs for records with errors**
Records which were errored out during CP processing.
- **Logs for Not-Processed records**
Records which were not processed by the CP.

Usage scenarios for enhanced logging

The following are usage scenarios for enhanced logging:

Case 1: Normal Transaction Execution

In this case, the user gets the following details as part of the log:

- A record in the log table describing CP (Concurrent Program) details such as its status, log file, and log file node.
- A record in the log table for each of the records from intermediate tables for end-to-end traceability. This includes success/error/no-processed records.
- A record in the log table with a column TIBCO_LOG, populated with the log message from the PL/SQL code including any exceptions. This can be used for any purpose that any log message serves.

Case 2: Concurrent Program submission Failure

In case of Concurrent Program submission failure:

- A record is inserted into the log table with the message:
"<Transaction Name> Concurrent Program Submission Failed".
- A record is inserted in the log table (as the value of the TIBCO_LOG column) with the log messages collected during the PL/SQL code execution.

Case 3: Exception while program is running

When an exception occurs, you can view log messages in the TIBCO_LOG column of the log table. This helps the user rectify the data so that the transaction runs fine the next time.

Case 4: Option to customize log messages

The option to add additional debug messages is available to users responsible for deploying or customizing the adapter. This is accomplished by calling a procedure. The debug messages are appended at the end of the log messages value of the TIBCO_LOG column of the log table.

Case 5: Provision to alter the Concurrent Program priority

Users have the option to alter the Concurrent Program priority which is set at 10 by default. This value can be changed per specific needs and business requirement.

Case 6: Quicker overall execution completion

The overall time taken to complete a transaction is reduced as a result of reduction in the wait time for CP completion.

Changes for enhanced logging

The TIB_INT_LOG_SUB table has been modified to provide additional logging capabilities. Changes have been made to the STATUS column and additional columns have been added to the table.

Changes to the STATUS Column

The following values can be used as status:

Value	Meaning
SUCCESS_CP	Concurrent Program submitted successfully.
ERROR_CP	Concurrent Program submission failed.
SUCCESS_RECORD	Records successfully imported into Oracle Applications Base tables.
ERROR_RECORD	Records that caused errors while importing from interface tables into Oracle Applications Base tables.
NOT_PROCESSED_RECORD	Records not processed by the Concurrent Program.
INFO	Generic information message – like procedure log messages.
WARNING	Generic warning messages to be logged in the table.
ERROR	Generic error messages to be logged in the table.

Addition of new columns

The following new columns have been added:

Column	Datatype	Usage
CP_REQUEST_ID	NUMBER	The request ID of the CP.
CP_STATUS	VARCHAR2(100)	The meaningful status of the CP.
CP_LOGFILE	VARCHAR2(1000)	Log File location of the CP on the server.
CP_LOGFILE_NODE	VARCHAR2(1000)	Node on which the log file is located. Useful on multi node installation of Oracle Applications.
TIBCO_LOG	VARCHAR2(4000)	The appended log messages of the procedure.

Appendix N **Dynamic Population of Concurrent Program Parameters**

This appendix provides information on the dynamic population of Concurrent Program Parameters in TIBCO Adapter for Oracle Applications.

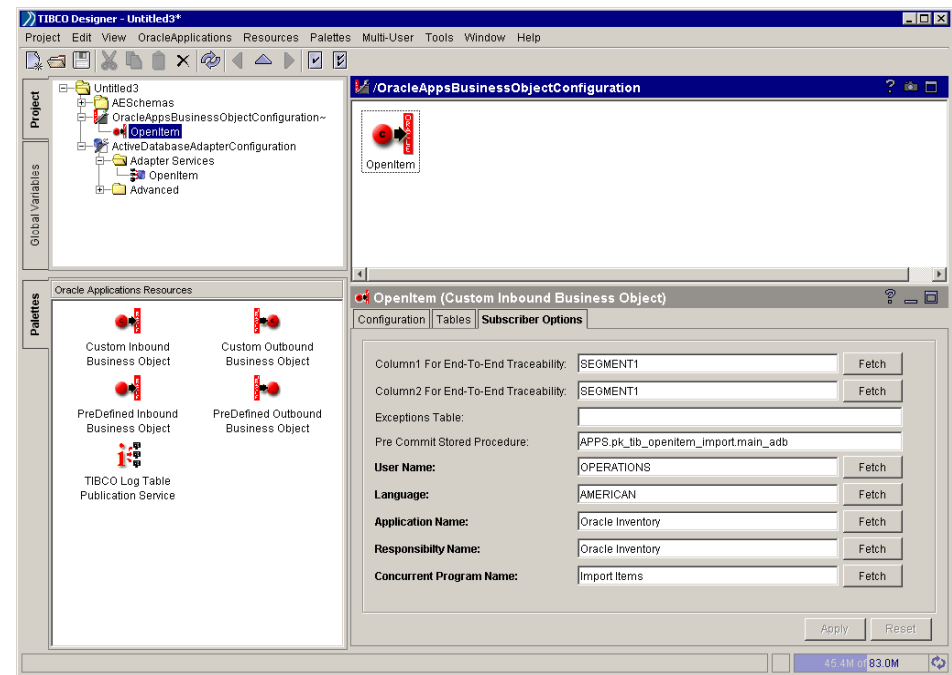
Topics

- [Dynamic Population of Concurrent Program Parameters, page 498](#)

Dynamic Population of Concurrent Program Parameters

The TIBCO Adapter for Oracle Applications supports Run-time population of Concurrent Program Parameters. It is mandatory for the user to enter the Concurrent Program name in the **Subscriber Options** tab of the Designer while configuring the Subscription service.

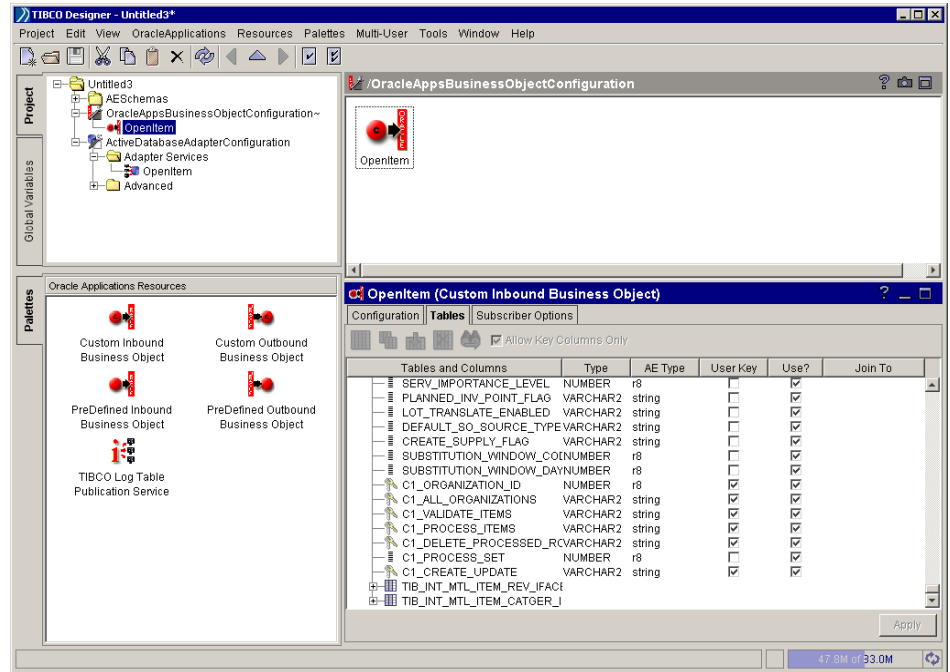
Figure 134 Dynamic Population of Concurrent Program Parameters



The list of Actual Parameters and the details of each of the following parameters is obtained based on the Concurrent Program Name that has been passed.

- If the parameter is mandatory or not.
- Default Value of the parameter.
- Data Type of the parameter
- Data Length of the parameter

Figure 135 Concurrent Program Parameters in Intermediate Table



These concurrent program parameters are added as columns in the header intermediate tables and are identified by `C1_<concurrentparamatername>` in the table. Also, the mandatory parameters are made as user_keys in the table.

Each added column corresponds to a parameter of the Concurrent Program. Using these columns in the TIBCO Intermediate tables, Concurrent Program parameters are passed to the adapter at run-time.

The user should modify the PL/SQL program to populate the values of the Concurrent Program parameters from the columns in the Header Intermediate Tables, and pass the parameters to the `<TransactionName>_Validation` procedure and the main procedure to include the modified call.

An example of dynamically populating the concurrent program for Open Item Transaction is given below.

One of the concurrent program parameters in the Open Item transaction is `l_delete_processed_rows`. The value 1 depicts that the successfully processed rows are deleted from the open item interface tables. The value 2 depicts that the successfully processed records are not deleted.

These parameters are exposed to the user as columns of the header intermediate table. The following scripts serve as an example guideline where the scripts are modified using the PL/SQL cursors to achieve the run-time passing of concurrent program parameters.

```

PROCEDURE main_adb
{
out_var1          OUT          NUMBER,                      --RETURN_CODE
out_var2          OUT          VARCHAR2,                    --SP_TEXT
out_var3          OUT          NUMBER                      --DO_ROLLBACK
}
IS
l_chr_errbuf              VARCHAR2(240);
l_chr_retcode             VARCHAR2(240);
l_chr_concurrent_flag     VARCHAR2(1) := 'N';
l_chr_skip_validn_flag    VARCHAR2(1) := 'N';
l_num_user_id             NUMBER;
l_num_resp_id             NUMBER;
l_num_resp_appl_id        NUMBER;
l_organization_id         NUMBER;
l_all_organizations       NUMBER;
l_validate_items          NUMBER;
l_process_items           NUMBER;
l_delete_processed_rows   NUMBER;
l_process_set             NUMBER;
l_createUpdate_items      NUMBER;
out_var1                  NUMBER;
out_var2                  varchar2(2000);
/*Cursor for concurrent program parameters*/
CURSOR cur_mtl_system_items_conc0 IS
SELECT * FROM TIB_INT_MTL_SYS_ITEMS_IFACE
WHERE TIBCO_INT_PROCESS_FLAG = 2;
BEGIN
dbms_output.put_line('calling main');
/*Initialize user id,responsibility id and application id*/
-----
--Initialize user id,responsibility id and application id
l_num_user_id             := 1318;
l_num_resp_id             := 20539;
l_num_resp_appl_id        := 401;
-----

```

```

-----
-- Perform two separate operations depending on the type of
transaction.
-- (1) Subscription Mode:- Initialize the following to process
records that the User wants(User Enters value in main)
-- (2) Subscription Reply Mode:- This will run with whatever is
given below
-----

/*Dynamic population of concurrent program parameters using the
above cursor*/
FOR cur_mtl_system_conc0_rec IN cur_mtl_system_items_conc0
LOOP
  l_organization_id      :=
cur_mtl_system_conc0_rec.C1_ORGANIZATION_ID;
  -- For Subscription With Reply, put NULL, Org is defaulted from
Profiles.
  -- In case of Subscription, the User will enter the organization
  l_all_organizations     :=
cur_mtl_system_conc0_rec.C1_All_ORGANIZATIONS;
  -- Enter '1' for Yes, '2' for No
  l_validate_items        :=
cur_mtl_system_conc0_rec.C1_VALIDATE_ITEMS;
  -- Enter '1' for Yes, '2' for No
  l_process_items          :=
cur_mtl_system_conc0_rec.C1_PROCESS_ITEMS;
  -- Enter '1' for Yes, '2' for No
  l_delete_processed_rows  :=
cur_mtl_system_conc0_rec.C1_DELETE_PROCESSED_ROWS;
  -- Enter '1' for Yes, '2' for No
  l_process_set            :=
cur_mtl_system_conc0_rec.C1_PROCESS_SET;
  --
  Currently this is NULL for all
  l_createUpdate_items     :=
cur_mtl_system_conc0_rec.C1_CREATE_UPDATE;
  -- Enter '1' for Yes, '2' for No
  dbms_output.put_line('l_delete_processed_rows' || l_delete_proces
sed_rows);
  main(
    l_chr_errbuf,
    l_chr_retcode,
    l_chr_concurrent_flag,
    l_chr_skip_validn_flag,
    l_num_user_id,
    l_num_resp_id,

```

```

        l_num_resp_appl_id,
        l_organization_id,
        l_all_organizations,
        l_validate_items,
        l_process_items,
        l_delete_processed_rows,
        l_process_set,
        l_createUpdate_items,
        out_var1,
        out_var2
    );
END LOOP;
END main_adb;
END pk_tib_openitem_import;

```

For more information, check the `openitem.sql` OracleAppsBusinessObjectConfiguration_OpenItem_all script in `<ADORAPPS_HOME>/examples/BusinessWorks/Sub/OpenItem/sql` and also the OpenItem BW example.

Appendix O **Tips on Using Oracle JPublisher**

This appendix shows how to run Oracle JPublisher to generate the scripts that the adapter service requires. Although the instructions presented in this appendix can be used to execute Oracle JPublisher, Oracle JPublisher documentation is subject to change and a review of Oracle JPublisher documentation is recommended.

Topics

- [An Introduction to Oracle JPublisher, page 504](#)
- [Creating a .bat File for a Specific API, page 505](#)
- [Command Line Parameters, page 506](#)
- [Before Executing the .bat File, page 508](#)
- [Ways for Using a PL/SQL Wrapper File, page 509](#)

An Introduction to Oracle JPublisher

Oracle JPublisher is an Oracle command line tool used to expose their database objects in different formats such as Java and PL/SQL scripts. For TIBCO Adapter for Oracle Applications, the Oracle Applications Inbound Business Object (For R12 Only) Service depends on the PL/SQL wrapper file generated by Oracle JPublisher in order to generate the scripts necessary to run the service.

Creating a .bat File for a Specific API

Assuming that Oracle JPublisher is installed on a Microsoft Windows platform, create a .bat file that contains the following lines:

```
echo on
set
CLASSPATH=%CLASSPATH%;./lib/runtime12.jar;./lib/translator.jar;./lib/classes12.jar
java oracle.jpub.Main -user=apps/apps
-url=jdbc:oracle:thin:@192.168.70.57:1521:VIS -plsqlmap=always
-plsqlfile=<API_NAME> .sql -plsqlpackage=TIB_<API> -sql=<API_NAME>
-plsqlonly=true -dir=<Output>
```

where, <API_NAME> is the name of API Package used,

<API> is the name of the package, and

<Output> is the location where you want store the resulting text file.

Oracle JPublisher can only execute against one API package at one time. If more than one APIs are desired for integration, you need to execute the .bat file multiple times to generate all the associated wrappers for the adapter service configuration.

Command Line Parameters

This section describes the command line parameters required by Oracle JPublisher. The parameter and execution information for Oracle JPublisher is based on Oracle JPublisher version 10.2.0.3.0.

- `classpath`

Oracle JPublisher requires three jar files: `runtime12.jar`, `translator.jar`, and `classes12.jar`. They are installed as part of Oracle JPublisher. You need to make sure the paths to these three files are included in the `classpath` parameter. You can execute the `set classpath` command to set the `classpath` parameter.

- `java oracle.jpub.Main`

The Java class that is set as `java oracle.jpub.Main ...` is used to call Oracle JPublisher. The class is located in the `translator.jar` file. So this library needs to be included in `classpath`.

- `user`

The Oracle Database user that has access to the API package. It need to be a user that has the same permission as the apps user created during an Oracle e-Business Suite installation.

- `url`

The JDBC URL containing the proper destination and credential to log into the Oracle e-Business Suite database.

- `plsmap=always`

This parameter is needed for generate the PL/SQL wrapper file. Do not change this parameter.

- `sql`

The API package name in the Oracle Database which you want to generate PL/SQL wrappers for. The package contains the Oracle API used for integration. You can obtain the name of the package through Oracle's Integration Repository.

- `plsfile`

The name of the output wrapper file that will be created by Oracle JPublisher. It need to be specified by user.

- `dir`

The directory where that the wrapper file will be stored.

- `plspackage`

The package name of the generated wrapper. Oracle JPublisher exposes the internal data types, data structures, and procedure calls of a package so that these objects and calls are accessible at the database level.

- `plsonly=true`

Setting this parameter to `true` specifies to generate a PL/SQL wrapper file only. Do not change this parameter.



Because Oracle JPublisher is a tool provided by Oracle, Oracle reserves the right to change the behavior of the tool. Also note that Oracle JPublisher may not operate properly against some API packages. In addition to the instructions concerning Oracle JPublisher given in this manual, please use the Oracle JPublisher documentation as the ultimate source for configuring and running JPublisher.

Before Executing the .bat File

Modify the .bat file by replacing the API package name, the resulting file name, the destination directory, and the login credentials as needed.

A .bat file can generate a SQL wrapper for only one API package. So, you need to create an individual .bat file for each API package desired for integration and execute the .bat files individually.

To avoid possible issues during the creation process, it is recommended that all the PL/SQL wrappers for integration be generated prior to any adapter configuration.

It may take some time for a PL/SQL wrapper file to be generated. Depending on the load or performance profile of the Oracle database, 15-20 minutes may be required.

Ways for Using a PL/SQL Wrapper File

A PL/SQL wrapper can be used in two ways.

- The adapter can execute the PL/SQL wrapper against the database using a user with the proper privileges to run this script. In this case, the PL/SQL wrapper makes the internal data types and stored procedure accessible through JDBC/ODBC APIs.
- The adapter also uses the PL/SQL script during configuration as a guide to accessing the database objects associated with the API package and creating the correct schema.



This is a prerequisite for the adapter service configuration. Any failure of PL/SQL wrapper generation means that the API package cannot be supported by the adapter without manual PL/SQL intervention.

Selecting APIs and Concurrent Programs from Oracle EBS R12 Integration Repository

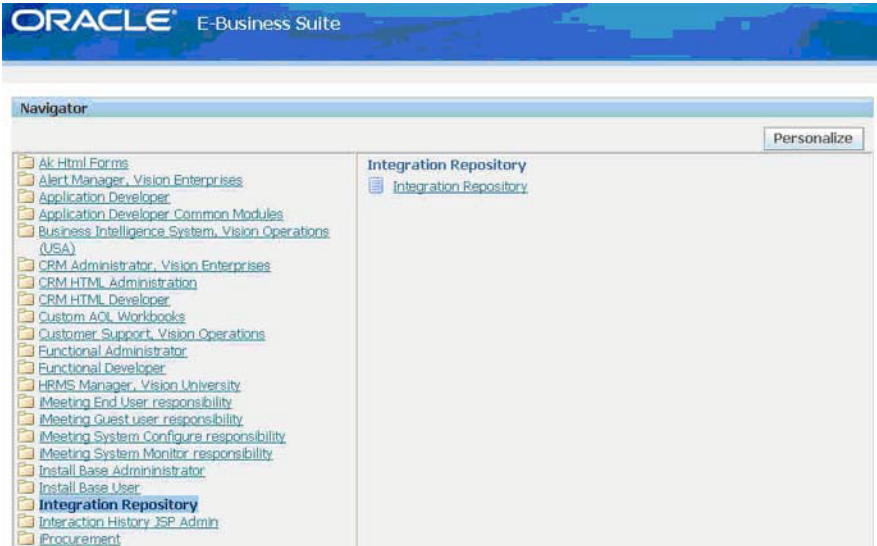
To select APIs and concurrent programs from Oracle EBS R12 integration repository:

1. Launch a Web browser, open the login page of Oracle EBS R12 Web management system (as shown in [Figure 136, The EBS R12 Login Page](#)), and log into the Oracle EBS R12 Web management system as the administrator. After you log in successfully, the Navigator page appears, as shown in [Figure 137, The EBS Navigator Page](#).

Figure 136 The EBS R12 Login Page



Figure 137 The EBS Navigator Page



2. Select **Integration Repository** in the Navigator page and enter the Search page, as shown in [Figure 138, The Search Page](#).

Figure 138 The Search Page



3. To search for an API or a concurrent program, enter the corresponding package name in the Internal Name text box and click the **Browse** button (at the right side of the page). The packages with the given internal name will be listed at the bottom of the page.
4. Click the desired package listed at the bottom of the Search, the package information page appears, where the procedures and functions (for APIs, as shown in [Figure 139, The Package Information Page: Procedures and Functions List](#)) or interface tables (for Concurrent Programs, as shown in [Figure 140, The Package Information Page: Open Interfaces Tables /Views](#))

List) are displayed. Select the desired procedure or interface table in the corresponding page.

Figure 139 The Package Information Page: Procedures and Functions List

Internal Name
Type
Product
Business Entity

WSH_DELIVERY_DETAILS_PUB
PL/SQL
Shipping Execution Common
[Delivery Line](#)

Status
Scope

Active
Public

[Browse](#) [Search](#) [Printable Page](#)

Full Description

This is the public interface for the Delivery Line entity. It allows execution of various Delivery Line functions, including creation, update of delivery lines and other actions.

Source Information

Procedures and Functions

Name	Internal Name	Status	Description
Assign/Unassign Delivery Lines	DETAIL_TO_DELIVERY	Active	Procedure to assign/unassign delivery lines to a delivery.
Autocreate Deliveries	AUTOCREATE_DELIVERIES	Active	Procedure to perform autocreate deliveries for multiple delivery lines.
Autocreate Trips	AUTOCREATE_DEL_TRIP	Active	Procedure to perform autocreate a single trip for multiple delivery lines. It also creates deliveries for the delivery lines.
Autocreate Trips	AUTOCREATE_DEL_TRIP	Active	Procedure to perform autocreate trips for multiple delivery lines. It also creates deliveries for the delivery lines.
Get Delivery Line Status	GET_DETAIL_STATUS	Obsolete	Procedure to get the status of a delivery line. The line status could be one of the three values: 1.
Initiate Delivery Line Changed Attributes	INIT_CHANGED_ATTRIBUTE_REC	Active	This procedure takes in a record of ChangedAttributeType and initializes its attributes to the default FND_API_G values.
Split Delivery Line	SPLIT_LINE	Active	Procedure to split quantity of a delivery line.
Update Shipping Attributes	UPDATE_SHIPPING_ATTRIBUTES	Active	Procedure to change the attributes of multiple delivery lines.
Update Shipping Attributes to enable entry of multiple serial ranges	UPDATE_SHIPPING_ATTRIBUTES	Active	This overloaded version of Update_Shipping_Attributes is to enable entry of multiple serial ranges for a given delivery detail.

[Browse](#) [Search](#) [Printable Page](#)

Figure 140 The Package Information Page: Open Interfaces Tables /Views List

Internal Name
Type
Product
Business Entity

WICMLP
Concurrent Program
Work in Process
[Work Order](#)

Status
Scope

Active
Public

[Browse](#) [Search](#) [Printable Page](#)

Full Description

This program executes the open interface for work orders. It can be used either to create new job/schedule or update existing job/schedule info WIP_JOB_SCHEDULE_INTERFACE, WIP_JOB_DTLS_INTERFACE.

Source Information

Parameters

Name	Type	Required	Displayed	Description
Group ID	FND_NUMBER15_REQUIRED	Yes	Yes	Used to batch interface records
Validation Level	FND_NUMBER15	Yes	No	Indicates what validation should be performed
Print Report	WIP_SRS_YES_NO_MAND	Yes	Yes	Indicates whether report is printed

Open Interface Tables/Views

Name	Direction	Status	Description
WIP_JOB_SCHEDULE_INTERFACE	Inbound	Active	WIP_JOB_SCHEDULE_INTERFACE contains requests to create or modify discrete jobs or rep
WIP_JOB_DTLS_INTERFACE	Inbound	Active	WIP_JOB_DTLS_INTERFACE contains requests to add, delete, and modify material and/or re

[About this Page](#) [Privacy Statement](#)

Integration Repository Home Logout Preferences Help Diagnostics

Index

A

agents [256](#)
alerts [256](#)
API [xxviii](#)
auto-discovery process, TIBCO Hawk [258](#)

C

Certified Messaging, RVCN [6](#)
configuring the database account (Windows) [51](#)
create_user.sql script [51](#)
creating an ODBC system data source (Windows) [57](#)
customer support [xxx](#)

D

database account configuring (Windows) [51](#)
Distributed Queue, RVDQ [6](#)

E

errors
 message format [384](#)
examples
 trace results [384](#)

J

JDBC drivers [43](#)

L

load balancing [251](#)

N

Name [113](#), [125](#), [137](#), [151](#), [171](#), [193](#), [203](#)

O

ODBC
 system data source (Windows) [57](#)
ODBC drivers [43](#)
ODBC drivers documentation [43](#), [67](#)

P

Publication Service typical flow diagram [12](#)

R

RV [xxviii](#), [xxviii](#)
RVCN [xxviii](#)
RVDQ [xxix](#)

S

schema
 examples of use [23](#)

scripts

- create_user.sql [51](#)

service name for ODBC system data source

- (Windows) [44](#)

SQL scripts location [51](#), [51](#)Subscription Mapping Details [365](#)

Subscription Service

- typical flow diagram [18](#)

support, contacting [xxx](#)system data source for ODBC (Windows) [57](#)**T**Tables Tab [146](#)technical support [xxx](#)TIBCO documentation [xxx](#)

TIBCO Hawk

- background information [256](#)

- enterprise monitor components [256](#)

TIBCO_HOME [xxv](#)TIBCOCommunity [xxx](#)Tracing Levels and Fields [382](#), [382](#)

Transactional Delivery

- RVTX [6](#)

typical configuration session [99](#)