

TIBCO Adapter™ for ClarifyCRM

User's Guide

*Software Release 5.4
August 2008*

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 TIBCO ADAPTER FOR CLARIFYCRM USER'S GUIDE) 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.

TIB, TIBCO, TIBCO Software, TIBCO Adapter, Predictive Business, Information Bus, The Power of Now, TIBCO Rendezvous, TIBCO Designer, TIBCO Repository, TIBCO InConcert, TIBCO IntegrationManager, TIBCO ActiveEnterprise and TIBCO Hawk are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or 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.TXT 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 © 1998-2008 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information

Contents

Figures	xi
Tables	xv
Preface	xix
Related Documentation	xx
TIBCO Adapter for ClarifyCRM Documentation	xx
Other TIBCO Product Documentation	xx
Third-Party Documentation	xxi
Typographical Conventions	xxiii
Terminology and Acronyms	xxv
How to Contact TIBCO Support	xxvi
Chapter 1 Concepts	1
ClarifyCRM Overview	2
Adapter Overview	4
Components	4
Adapter Features	6
Adapter Capabilities	8
Message Transports	8
Multithreading	9
Connection Management	9
Adapter Services	11
Publication Service	12
Subscription Service	13
Request-Response Service	14
Request-Response Invocation Service	15
Adapter and ClarifyCRM Interaction	17
Chapter 2 Installation	19
Preparing Your Environment for Installation	20
Operating System Requirements	20
ClarifyCRM Applications Requirements	21
Pre-Installation Worksheet	22

Installer Overview	24
Upgrading an Adapter	24
Uninstalling the Adapter	25
Installation Registry	27
Microsoft Windows Platforms	27
UNIX Platforms	27
Installation History	28
Adapter Components and Compatible Software	29
Adapter Components	29
Required and Optional TIBCO Products	29
Installation on Microsoft Windows	31
Installing the Adapter on Microsoft Windows	33
Combining Options	34
Post-Installation Tasks on Microsoft Windows	34
Installation on UNIX Systems	36
Installing the Adapter on UNIX	38
Combining Options	39
Permission Requirements on UNIX Systems	39
Post-Installation Tasks on UNIX	40
Installer FAQs and Troubleshooting	42
Frequently Asked Questions	42
Running Out of Disk Space	43
Installation Errors on HP-UX 11.00 64 Bit Platform	44
Configuring TIBCO Hawk	44
Cannot Install the Adapter	45
Chapter 3 Publication Service Functionality	47
Publication Service Overview	48
Publication Service Features	48
How the Publication Service Works	52
Prepare ClarifyCRM Components for the Publication Service	53
Add TIBCO Staging Tables to the ClarifyCRM Database	53
Enable the Publication Service for LAN Client	54
Enable the Publication Service for ClarifyCRM Web Client	60
Enable the Publication Service for ClarifyCRM Applications	66
Limitations of the Publication Service	75
Chapter 4 Subscription Service Functionality	77
Subscription Service Overview	78
Subscription Service Features	78
Subscription Service Functionality	78

How the Subscription Service Works	80
CBO Wrapper Invocation	80
Custom JavaBean Invocation	81
Service Error Handling	85
Chapter 5 Request-Response Service Functionality	87
Request-Response Service Overview	88
Request-Response Service Features	88
Supported Request-Response Operations	88
How the Request-Response Service Works	89
Query Operations	89
Workflow Operations	91
Customized Operations	95
Custom JavaBean Invocation	96
Chapter 6 Request-Response Invocation Service Functionality	99
Request-Response Invocation Service Overview	100
Request-Response Invocation Service Using Flexible Deployment	100
Request-Response Invocation Service Using Non-Flexible Deployment	101
How the Request-Response Invocation Service Works	102
ClearBasic Forms	102
Web Forms	109
Chapter 7 Getting Started	115
Overview	116
Prerequisites	117
Configure the Adapter	118
Deploy the Adapter	129
Start the Adapter	130
Stop the Adapter	131
Chapter 8 Adapter Instance Options	133
Overview	134
Start the Design-time Adapter	134
Configuration Tasks	134
Password Handling	135
Adapter Instance Fields	136
Configuration Tab	136
Design-time Connection Tab	137
Run-time Connection Tab	138

Adapter Services Tab	141
General Tab	142
Logging Tab	144
Startup Tab	147
Monitoring Tab	148
Adapter Services	151
Publication Service Fields	152
Configuration Tab	152
Fetch CBO Tab	154
Transport Tab	155
Schema Tab	160
Subscription Service Fields	162
Configuration Tab	162
Fetch CBO Tab	163
Transport Tab	164
Schema Tab	168
Request-Response Service Fields	170
Configuration Tab	170
Fetch CBO Tab	171
Transport Tab	172
Schema Tab	176
Request-Response Invocation Service Fields	179
Configuration Tab	179
Fetch Request CBO Tab	180
Fetch Reply CBO Tab	180
SubscriberOptions Tab	181
Transport Tab	182
Schema Tab	187
Chapter 9 Deploying the Adapter Using TIBCO Administrator Enterprise Edition	189
Create an EAR File in TIBCO Designer	190
Deploy the Project	191
Predefined Properties	191
Start or Stop the Adapter	194
Monitor the Adapter	195
Chapter 10 Advanced Topics	197
Using the Adapter with a Revision Control System	198
Modifying the DTA Rendezvous Connection Properties	201
Defining a TIBCO Hawk Session	202
Using Global Variables	203

Specifying Global Variables Using TIBCO Designer	203
Using Global Variables in the Project	203
Changing Global Variable Values at Runtime	205
Predefined Global Variables	205
Configuring a Remote Adapter	211
Using the Tracking Element	212
Chapter 11 Monitoring the Adapter Using TIBCO Hawk	213
Overview	214
Starting TIBCO Hawk Software	215
The Auto-Discovery Process	216
Invoking Microagent Methods	217
Available Microagents	220
activateTraceRole()	223
deactivateTraceRole()	224
getActivityStatistics()	225
getActivityStatisticsByOperation(Operation)	226
getActivityStatisticsBySchema(SchemaName)	227
getActivityStatisticsByService(ServiceName)	228
getAdapterServiceInformation()	229
getComponents()	230
getConfig()	231
getConfigProperties()	232
getConnectionStatistics()	233
getHostInformation()	234
getPerfMonSetting()	235
getPollingBatchSize()	236
getPollingInterval()	237
getQueueStatistics()	238
getRvConfig()	239
getStatus()	240
getThreadStatistics()	241
getTraceSinks()	242
getVersion()	243
_onUnsolicitedMsg()	244
preRegisterListener()	245
resetActivityStatistics()	246
resetConnectionStatistics()	247
resetThreadStatistics()	248
reviewLedger()	249
setPollingBatchSize()	250
setPollingInterval()	251
setTraceSinks()	252

- stopApplicationInstance() 253
 - unRegisterListener() 254
- Appendix A TIBCO Staging Tables 255**
 - table_tibco_message_queue 256
 - table_tibco_message_fields 257
- Appendix B Frequently Asked Questions 259**
 - Frequently Asked Questions 260
- Appendix C Error Messages 263**
 - Error Message Listing 264
- Appendix D User Exits 305**
 - Overview 306
 - How the Adapter Implements User Exits 307
 - IclarifyUserExit Interface 308
- Appendix E Wrapper Classes 311**
 - Overview 312
 - Wrappers Packaged with the Adapter 313
 - ActEntry_Wrapper 313
 - Address_Wrapper 313
 - BusOrg_Wrapper 313
 - Case_Wrapper 313
 - Communication_Wrapper 314
 - Contact_Wrapper 314
 - CreditCard_Wrapper 314
 - DemandDetail_Wrapper 314
 - DemandHeader_Wrapper 315
 - Dialogue_Wrapper 315
 - DocInst_Wrapper 316
 - Generic_Edr_Com_Role Wrapper 316
 - NotesLog_Wrapper 316
 - OEQuote_Wrapper 316
 - PartPrice_Wrapper 317
 - Part_Wrapper 317
 - ServicePart_Wrapper 318
 - ShopList_Wrapper 318
 - Site_Wrapper 318
 - SubCase_Wrapper 318

Task_Wrapper.	319
User_Wrapper	319
Wrapper Class and TibcoCBOInterface.	320
Writing Wrapper Class for Case Business Object	321
Steps Involved	321
Appendix F Testing the Adapter	331
Testing the Publication Service with ClearBasic Forms	332
Importing ClearBasic Files	332
Testing the Publication Service with Web Forms	336
Testing the Request-Response Functionality with Web Forms	339
Testing the Request-Response Functionality with ClearBasic Forms	340
TIBCO Software Inc. End User License Agreement	341
Index	345

Figures

Figure 1	Adapter services and components	11
Figure 2	Typical Publication Service flow	12
Figure 3	Typical Subscription Service flow	13
Figure 4	Typical Request-Response Service flow	14
Figure 5	Typical Request-Response Invocation Service flow	15
Figure 6	Directive File window	55
Figure 7	Event Configuration window	67
Figure 8	Select Trigger Object	68
Figure 9	Trigger Configuration	68
Figure 10	Add the fields to be logged	70
Figure 11	Define a Publisher	71
Figure 12	Associate the Publisher with the event	72
Figure 13	Publisher Events Configuration	73
Figure 14	Flexible deployment mode	103
Figure 15	Non-Flexible deployment mode	105
Figure 16	Web client deployment mode	110
Figure 17	TIBCO Designer startup panel	118
Figure 18	Save Project dialog	119
Figure 19	TIBCO Designer Configuration tab	120
Figure 20	Design-time Connection settings	121
Figure 21	Run-time Connection settings	122
Figure 22	Adapter Services folder	123
Figure 23	Add Publication Service	124
Figure 24	Fetch CBO list	125
Figure 25	Schema fetched	125
Figure 26	Fetch Contained CBO	126
Figure 27	Transport settings	127
Figure 28	Specify the Termination Subject	128

Figure 29	Change the polling interval	142
Figure 30	Sample General screen.	144
Figure 31	Generate sinks	145
Figure 32	Sample Logging screen.	147
Figure 33	Sample Startup screen	148
Figure 34	Sample Monitoring screen.	150
Figure 35	Enable and configure SSL.	153
Figure 36	Sample Publication Service Configuration screen.	154
Figure 37	Publication Service Transport tab with TIBCO Rendezvous.	157
Figure 38	Publication Service Transport tab with JMS	160
Figure 39	Publication Service Schema tab	161
Figure 40	Subscription Service Configuration tab.	163
Figure 41	Subscription Service Transport tab with TIBCO Rendezvous.	166
Figure 42	Subscription Service Transport tab with JMS	168
Figure 43	Subscription Service Schema tab	169
Figure 44	Request-Response Service Configuration tab	171
Figure 45	Request-Response Service Transport tab with TIBCO Rendezvous	174
Figure 46	Request-Response Service Transport tab with JMS.	176
Figure 47	Change the schema reference	177
Figure 48	Schema mapped to a workflow operation for the Request-Response Service	178
Figure 49	Request-Response Invocation Service Configuration tab.	180
Figure 50	Reply CBO loaded to Request-Response Invocation Service.	181
Figure 51	Request-Response Invocation Service with defined SubscriberOptions.	182
Figure 52	Request-Response Invocation Service Transport tab with TIBCO Rendezvous.	184
Figure 53	Request-Response Invocation Service Transport tab with JMS	187
Figure 54	Request-Response Invocation Service Schema tab.	188
Figure 55	Schema files created by the adapter.	198
Figure 56	Add additional files to revision control system.	199
Figure 57	Hawk session	202
Figure 58	Global variables.	204
Figure 59	TIBCO Hawk Enterprise Monitor	216
Figure 60	Select a method in the Microagents, Methods and Arguments dialog	218

Figure 61	Invocation results dialog.	219
Figure 62	Implement User Exits for Pre-processing of Publication Service event message	308
Figure 63	Add users to resource configuration	333
Figure 64	Create customized form.	333
Figure 65	Add a form to resource configuration	334
Figure 66	Directive window	334
Figure 67	Enter purchase order	336
Figure 68	B2B admin user registration (view 1)	337
Figure 69	B2B admin user registration (view 2)	338
Figure 70	Clarify eOrder cart	339

Tables

Table 1	General Typographical Conventions	xxiii
Table 2	Syntax Typographical Conventions	xxiv
Table 3	Terminology and Acronyms	xxv
Table 4	Adapter Machine Information	22
Table 5	Clarify Server Information	23
Table 6	Clarify User Information	23
Table 7	TIBCO Adapter Components	29
Table 8	Compatible Products	30
Table 9	Supported Platforms, Package Names, and Disk Space for Microsoft Windows.	31
Table 10	ClarifyCRM Matrix for Windows Platforms	32
Table 11	Supported Platforms, Package Names, Patches and Disk Space for UNIX Systems	36
Table 12	ClarifyCRM Matrix for Unix Platforms.	37
Table 13	Incoming Event	49
Table 14	opObject.	50
Table 15	Field	50
Table 16	Relation	51
Table 17	Message Formats that Contain CBOs	51
Table 18	ClearBasic Functions	56
Table 19	Java Methods	61
Table 20	Attributes of the incomingEvent Class	79
Table 21	Relevant Fields for the Custom JavaBean Functionality	79
Table 22	CBOs Provided with Wrapper Classes.	80
Table 23	Incoming Message Formats for Query Operations	90
Table 24	Outgoing Message Formats for Query Operations.	91
Table 25	Incoming Message Formats for Workflow Operations	93
Table 26	Outgoing Message Formats for Workflow Operations	94
Table 27	Incoming Message Formats for Customized Operations	96
Table 28	Outgoing Message Formats for Customized Operations	96

Table 29	Parameter list of function <code>TibcoSetMessageFields()</code>	111
Table 30	Parameter list of function <code>TibcoSetMessageFieldsEMS()</code>	112
Table 31	Predefined Properties	192
Table 32	Predefined Global Variables	205
Table 33	Microagent Methods	220
Table 34	<code>activateTraceRole()</code>	223
Table 35	<code>deactivateTraceRole()</code>	224
Table 36	Input parameter of <code>getActivityStatistics()</code>	225
Table 37	Returns of <code>getActivityStatistics()</code>	225
Table 38	Input parameter of <code>getActivityStatisticsByOperation(Operation)</code>	226
Table 39	Returns of <code>getActivityStatisticsByOperation(Operation)</code>	226
Table 40	Input parameter of <code>getActivityStatisticsBySchema(SchemaName)</code>	227
Table 41	Returns of <code>getActivityStatisticsBySchema(SchemaName)</code>	227
Table 42	Input parameter of <code>getActivityStatisticsByService(ServiceName)</code>	228
Table 43	Returns of <code>getActivityStatisticsByService(ServiceName)</code>	228
Table 44	Input parameter of <code>getAdapterServiceInformation()</code>	229
Table 45	Returns of <code>getAdapterServiceInformation()</code>	229
Table 46	Input parameters of <code>getComponents()</code>	230
Table 47	Returns of <code>getComponents()</code>	230
Table 48	<code>getConfig()</code>	231
Table 49	Input parameter of <code>getConfigProperties()</code>	232
Table 50	Returns of <code>getConfigProperties()</code>	232
Table 51	<code>getConnectionStatistics()</code>	233
Table 52	<code>getHostInformation()</code>	234
Table 53	<code>getPerfMonSetting()</code>	235
Table 54	Input parameter of <code>getPollingBatchSize()</code>	236
Table 55	Returns of <code>getPollingBatchSize()</code>	236
Table 56	<code>getPollingInterval()</code>	237
Table 57	<code>getQueueStatistics()</code>	238
Table 58	Input parameter of <code>getRvConfig()</code>	239
Table 59	Returns of <code>getRvConfig()</code>	239
Table 60	<code>getStatus()</code>	240

Table 61	getThreadStatistics()	241
Table 62	Input parameters of getTraceSinks()	242
Table 63	Returns of getTraceSinks()	242
Table 64	getVersion()	243
Table 65	preRegisterListener()	245
Table 66	Input parameters of reviewLedger()	249
Table 67	Returns of reviewLedger()	249
Table 68	setPollingBatchSize()	250
Table 69	setPollingInterval()	251
Table 70	setTraceSinks()	252
Table 71	unRegisterListener()	254
Table 72	table_tibco_message_queue	256
Table 73	table_tibco_message_fields	257
Table 74	Error Message	264

Preface

This guide describes supported features, platforms, installation, configuration, and overall architecture of TIBCO Adapter™ for ClarifyCRM. The guide assumes familiarity with the basic concepts of the TIBCO environment and Clarify technology.

Topics

- *Related Documentation, page xx*
- *Typographical Conventions, page xxiii*
- *Terminology and Acronyms, page xxv*
- *How to Contact TIBCO Support, page xxvi*

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Adapter for ClarifyCRM Documentation

The following documents form the TIBCO Adapter for ClarifyCRM documentation set:

- *TIBCO Adapter Concepts*: Read this manual for an introduction to TIBCO adapters, adapter components, capabilities, lifecycle and types of adapters available.
- *TIBCO Adapter for ClarifyCRM User's Guide*: Read this manual to gain an understanding of the product that you can apply to the various tasks.
- *TIBCO Adapter for ClarifyCRM Examples Guide*: Read this manual to work through the examples provided along with the adapter.
- *TIBCO Adapter for ClarifyCRM Release Notes*: Read this document for information about new features, changes since last release and a description of any open or closed issues that may affect installing or using the adapter.

Other TIBCO Product Documentation

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

- TIBCO ActiveEnterprise™ software:
 - *TIBCO ActiveEnterprise Concepts*
- TIBCO Designer™ software:
 - *TIBCO Designer User's Guide*
 - *TIBCO Designer Palette Reference*
- TIBCO Administrator™ software:
 - *TIBCO Administrator User's Guide*
 - *TIBCO Administrator Server Configuration Guide*

- TIBCO ActiveMatrix BusinessWorks™ software:
 - *TIBCO ActiveMatrix BusinessWorks Concepts*
 - *TIBCO ActiveMatrix BusinessWorks Getting Started*
 - *TIBCO ActiveMatrix BusinessWorks Process Design Guide*
 - *TIBCO ActiveMatrix BusinessWorks Administration*
 - *TIBCO ActiveMatrix BusinessWorks Palette Reference*
 - *TIBCO ActiveMatrix BusinessWorks Installation*
- 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 Hawk® software:
 - *TIBCO Hawk Installation and Configuration*
 - *TIBCO Hawk Administrator's Guide*
- TIBCO Adapter™ SDK
 - *TIBCO Adapter SDK Concepts*
- TIBCO Runtime Agent™ software
 - *TIBCO Runtime Agent Installation*
 - *TIBCO Runtime Agent Domain Utility User's Guide*

Third-Party Documentation

- *Clarify CBO Programmer's Guide*
- *Clarify CBO Reference Guide*
- *Clarify ClearBasic Programming Guide*
- *Clarify JSP Programming Guide*
- *Documents on Clarify tools*

- *Clarify Integration Gateway Implementation Guide*

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions



Convention	Use
code font	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use <code>MyCommand</code> to start the foo process.
bold code font	Bold code font is used in the following ways: <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, <code>MyCommand</code> is enabled: <code>MyCommand [enable disable]</code>
<i>italic font</i>	Italic font is used in the following ways: <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</code> <i>pathname</i>
Key combinations	Key name separated by a plus sign indicate keys pressed simultaneously. For example: <code>Ctrl+C</code> . Key names separated by a comma and space indicate keys pressed one after the other. For example: <code>Esc, Ctrl+Q</code> .
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	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.

Table 1 General Typographical Conventions (Cont'd)


Convention	Use
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

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 param1 param2 param3</pre>
{ }	<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

Following is a list of terminology and acronyms used in this guide.

Table 3 Terminology and Acronyms

Term	Definition
CBOs	Clarify Business Objects (CBOs) are part of the Clarify eBusiness Framework that are used for developing applications for the CeFO database.
CeFO	Clarify eFrontOffice is a Customer relationship management application that integrates, consolidates, and routes all customer interactions via the web and E-mail.
Clarify eBusiness Framework	Clarify eBusiness Framework is an e-Business Web application that helps users to interact with their customers and business partners.
CRM	Customer Relationship Management
CES	Customer Experience System
DTA	Design-time Adapter
EMS	Enterprise Message Service
JMS	Java Messaging Service
RPC	Remote Procedural Call
RV	TIBCO Rendezvous® protocol; also in certain contexts refers to reliable message quality of service, as opposed to certified message
RVCM	TIBCO Rendezvous certified message quality of service

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, please 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, you can request one.

Chapter 1 Concepts

TIBCO Adapter for ClarifyCRM enables business data exchanges between the ClarifyCRM system and the TIBCO environment. This chapter introduces the adapter by providing background information about its features and the product elements. It also provides an overview of the ClarifyCRM system and explains how the adapter is used to transfer data to and from the ClarifyCRM system.

Topics

- *ClarifyCRM Overview, page 2*
- *Adapter Overview, page 4*
- *Adapter Features, page 6*
- *Adapter Capabilities, page 8*
- *Adapter Services, page 11*
- *Adapter and ClarifyCRM Interaction, page 17*

ClarifyCRM Overview

ClarifyCRM provides customer asset management systems that are typically used in call centers. ClarifyCRM automates front-office functions such as help desk support, customer sales and service.

ClarifyCRM delivers two relationship management solutions:

- Clarify eFrontOffice (CeFO) integrates a company's web site, e-mail, and phone systems to enhance customer support. By maintaining a database of all interactions the company has had with a client, ClarifyCRM can help manage sales leads, route customer calls and offer self-service over the web. The main components of CeFO are Clarify database, Clarify server, Clarify Client, and the Clarify Web Client.
 - Clarify database is a relational database that contains customer data (information about the user's activities, contacts, and so on).
 - Clarify server hosts the Clarify database.
 - Clarify Client is the user interface used to input and retrieve information from the Clarify server.
 - Clarify Web Client is a web application that helps to interact with customers as well as business partners.
- ClarifyCRM eBusiness Framework is an eBusiness Web application that helps in interaction with customers and business partners. The main components of ClarifyCRM eBusiness Framework are CBOs and CBO Data Service.
 - ClarifyCRM Business Objects (CBOs) are part of the ClarifyCRM eBusiness Framework that are used for developing applications for the CeFO database. CBOs consist of Java Beans and ActiveX controls that provide an API to perform create and update operations in the CeFO database. A CBO implements the Clarify data model and encapsulates the application logic by providing methods and properties that can be used to query or update the CeFO database. For example, the Case business object provides access to case-related data and includes methods for finding, updating, and dispatching cases. CBOs also support eBusiness applications for online sales and service. Examples of eBusiness applications are Clarify eOrders and Clarify eSupport.

CBOs can be used in any Java application including Java Server Pages or ActiveX container (such as Visual Basic or Active Server Pages). For example,

the Java Server Pages in ClarifyCRM eSupport and ClarifyCRM eOrder use CBOs to create new cases and orders.

- CBO Data Service manages a connection pool. A CBO uses CBO Data Services to query and update the Clarify database.

Adapter Overview

TIBCO Adapter for ClarifyCRM provides seamless connectivity from the TIBCO environment to the ClarifyCRM system, allows events to be propagated from the ClarifyCRM system to the TIBCO environment and vice versa.

The adapter exports data from and imports data into a ClarifyCRM system in an event-driven fashion. Data is sent out to the TIBCO environment when changes are made in the ClarifyCRM System. When the adapter receives a message containing data to import, the adapter parses the data and inserts it into the ClarifyCRM database.

For a general introduction to adapters and the services they provide, refer to *TIBCO Adapter Concepts*. The book is part of the adapter documentation set.

Components

The adapter has three main components: adapter configuration palette, design-time adapter, and the run-time adapter.

The adapter palette comprises an adapter-specific GUI that seamlessly integrates with TIBCO Designer. It allows you to configure the adapter, and stores the configuration in persistent storage. (You will require TIBCO Designer for this, which is installed as part of the TIBCO Runtime Agent (TRA) installation. TRA must be installed before installing the adapter). The run-time adapter uses this configuration to pass and convert data to and from the applications.

- **Adapter Configuration Palette** — When you install the adapter, a palette containing the configuration resources for the adapter is automatically loaded in TIBCO Designer.

The adapter configuration palette provides an easy-to-use graphic user interface. It allows you to configure adapter services, download CBO schema directly from the ClarifyCRM database, and save resulting configuration in a project.

- **Design-time Adapter** — The design-time adapter runs in the background during your configuration session. The design-time adapter acts as a service, taking requests from TIBCO Designer to connect to a ClarifyCRM system and downloads schema from the ClarifyCRM system.
- **Run-time Component** — Runtime components deliver the core functionality of the adapter. They use configuration information and metadata to exchange messages between ClarifyCRM and applications in the TIBCO environment.

Adapter Services

The adapter functions as an agent to export and import data into the ClarifyCRM system in an event-driven fashion and provides the following services:

- **Publication Service** — This service is used to publish data from ClarifyCRM to the TIBCO environment.
- **Subscription Service** — This service is used to subscribe to messages from the TIBCO environment and post them into the ClarifyCRM database.
- **Request-Response Invocation Service** — This service is used to request data from external applications through the TIBCO environment.
- **Request-Response Service** — This service is used to provide response data by executing query, workflow and customized operations in the ClarifyCRM database using CBOs.

Adapter Features

This section summarizes the features of the adapter. These features are discussed in more detail in subsequent chapters.

- **Support for Multithreading** — The adapter supports a static number of threads. The number of threads is specified at the time of configuration. The number of connections to the application server equals the number of threads. With multiple threads, the adapter processes multiple messages concurrently.
- **Support for ClarifyCRM Business Objects (CBOs)** — The adapter supports CBOs, which are Java Beans and ActiveX controls that provide access to data in the CeFO database.
- **Support for Invoking Custom JavaBean** — The Subscription Service can be used to invoke a Custom JavaBean method.
- **Support for Parent-child Schemas in Design-time** — The Fetch CBO tab can be used to fetch both parent CBOs and child CBOs and it displays the parent-child representation. The same representation is displayed under the AESchemas folder.
- **Support for Insert and Update Operations** — The adapter supports INSERT and UPDATE operations for the Request-Response and Subscription services on Existing, Customized and Extended CBOs.
- **Support for Fetching Multiple Rows by Query Operation in Request-Response Service** — The Request-Response Service now supports QUERY operation resulting in multiple rows of the main CBO.
- **Support for Simple Single CBO Query Operation in Request-Response Service** — The Request-Response Service provides support for simple single CBO QUERY operation.
- **Support for Classic/LAN and Web Client** — The adapter provides features for working with both ClarifyCRM Classic/LAN and Web Clients.
- **Support for Workflow Operations** — ClarifyCRM provides a set of workflow operations to make it easier for the user to create and update the workflow objects (case, OEQuote, etc). Workflow objects also called queue-able objects in the ClarifyCRM system involves the most complicated business logic when workflow objects are created or changed.
- **Support for JMS at Design-time** — Communication between the palette and the design-time adapter is now possible using the JMS transport in addition to the TIBCO Rendezvous transport.

- **Atomicity** — Data is not imported or exported unless the entire business event successfully passes business logic and validation from the ClarifyCRM system.
- **Support for Delayed Acknowledgement** — The adapter ensures transaction integrity by supporting delayed acknowledgement. The adapter sends the acknowledgement to the source of the message received by the Subscription Service only after the message has been consumed in the ClarifyCRM system.
- **A Convenient Configuration Tool** — The adapter is configured using TIBCO Designer. TIBCO Designer connects to a specified ClarifyCRM system, downloads the CBOs for the required services, and transfers the schema and message routing rules for individual services to a central repository. Support for usage of global and client variables ensures easy and quick migration of adapter configurations from development to testing, and from testing to production environments.
- **Support for TIBCO ActiveMatrix BusinessWorks** — The adapter can be used in a TIBCO ActiveMatrix BusinessWorks process. For the latest versions of the TIBCO ActiveMatrix BusinessWorks supported, refer *TIBCO Adapter for Clarify readme.txt*.
- **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 detect 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.
- **Security** — The Obfuscation engine packaged with the adapter enables encryption of sensitive configuration information.

Adapter Capabilities

This section describes in detail how the TIBCO Adapter for ClarifyCRM handles each of the following:

- Message Transports
- Multithreading
- Connection Management

For general conceptual information on these capabilities, refer to *TIBCO Adapter Concepts*.

Message Transports

TIBCO Rendezvous and TIBCO Enterprise Message Service can be used to transport messages to and from the adapter.

- 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 (RVCM) — Ensures delivery of messages at least once. This guarantees that every certified message reaches its intended recipient in the order sent.
 - Distributed Queue — 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 Distributed Queue are ordinary Certified Message senders.
- The JMS mode of transport supports the following messaging protocols (TIBCO Enterprise Message Service 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 messages sent to a queue are consumed by one and only one receiver. 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*.

Multithreading

Multithreaded instance of an adapter will generate additional threads to dispatch events. The number of additional threads to be created for a given instance can be specified at design time and these are created at startup of the adapter.

Publication, Subscription, Request-Response and Request-Response Invocation services configured in an adapter instance can be multithreaded by specifying the required number of additional threads at the instance level. Publication service uses Java threads for dispatching events whereas Subscription, Request-Response and Request Response Invocation services use SDK dispatcher threads for processing events.

The adapter needs to create a connection with the target application for performing CBO operations for Subscription and Request-Response services. The real advantage of parallel processing in multithreaded mode can be achieved only when multiple connections are created with the target application.

Connection Management

This section describes connection pooling and reconnection mechanism.

Connection Pooling

There is a one-to-one mapping between the threads and the Clarify connection. This means that the number of connections to Clarify is the same as the number of threads. Once a connection is associated with a thread, the same connection is used by that thread till it terminates.

The adapter creates a static pool of connections with Clarify on startup. The number of connections is equal to the number of threads. Upon receiving a request, the adapter thread picks up a connection from the pool and uses it for processing. Once a thread picks up a connection, it remains dedicated to that thread till the connection terminates.

Reconnection Mechanism

The adapter uses a reconnection mechanism to handle the network glitches. You can configure the number of times the adapter will attempt to connect to the Clarify database.

Reconnection mechanism starts whenever the adapter faces a Clarify application error while processing an event. It performs a dummy operation to check whether the connection is valid or not. If this connection check is successful, the adapter displays an error for that event with the proper error message. The adapter does not attempt to re-create the connection pool. If the operation is unsuccessful, the adapter activates the reconnection mechanism.

The adapter tries to reconnect to the Clarify database as many times as you have configured it to retry. If it exceeds the parameter `Number of Reconnect Attempts Before Suspending Impacted Service(s)`, the adapter suspends the Subscription Services. Thereafter, the adapter will continue with the reconnection attempts till it crosses the parameter configured for `Maximum Number of Reconnect Attempts`. If the adapter is still not able to reconnect, it stops.

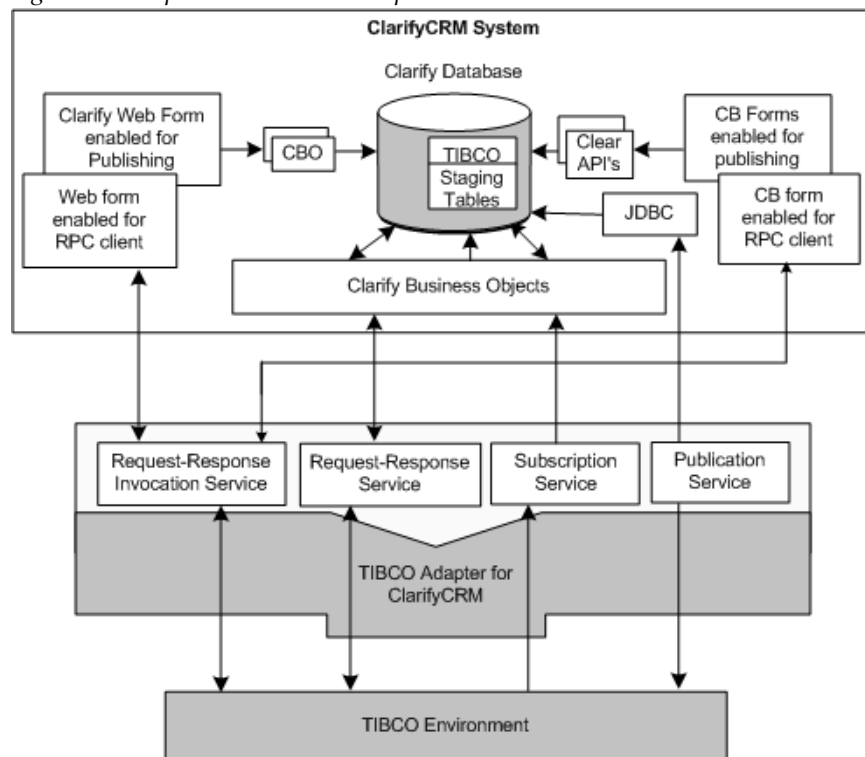
If the adapter is able to reconnect, it drops all the existing connections from the pool and creates the complete connection pool. While re-creating the connection pool, the adapter exits if it fails to establish the first connection. But, if the adapter is able to create one or more connections, it waits for the events.

Adapter Services

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 describes the Publication, Subscription, Request-Response and Request-Response Invocation Services of the adapter in more detail. Figure 1 shows the adapter services in relation to other components in a ClarifyCRM system.

Figure 1 Adapter services and components



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

The following sections describe how these adapter services interact with the ClarifyCRM environment to provide message connectivity.

Publication Service

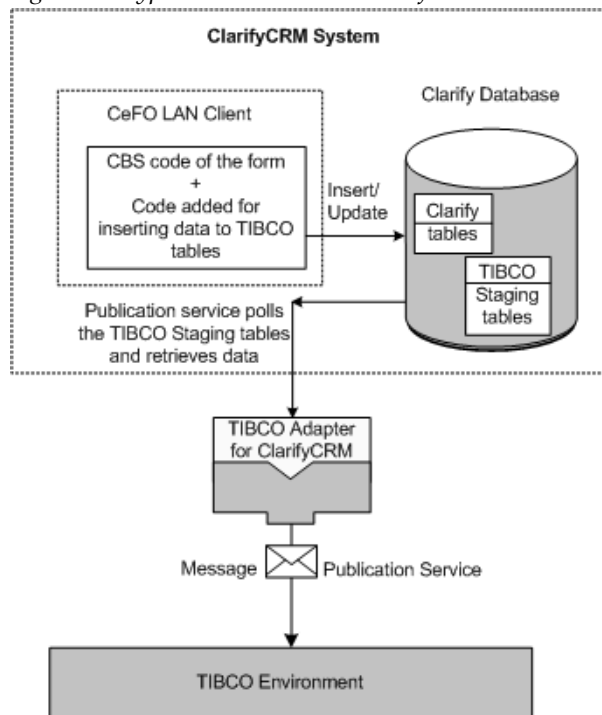
The Publication Service consists of three components:

- TIBCO Staging tables
- ClearBasic code (CBS code) for ClearBasic forms or JSP code for web forms
- a polling process

TIBCO Staging tables are extended into the ClarifyCRM system schema for holding publishing information. The ClearBasic code attached to a ClarifyCRM form or the JSP code for a web form updates the TIBCO Staging tables. The polling process polls the TIBCO Staging tables and publishes Clarify data in the TIBCO environment.

After successfully publishing the message, the Publication Service process deletes the message or updates the status in the message queue depending on the user configuration. For details, see Publication Service Functionality on page 47.

Figure 2 Typical Publication Service flow



Publication Service Example

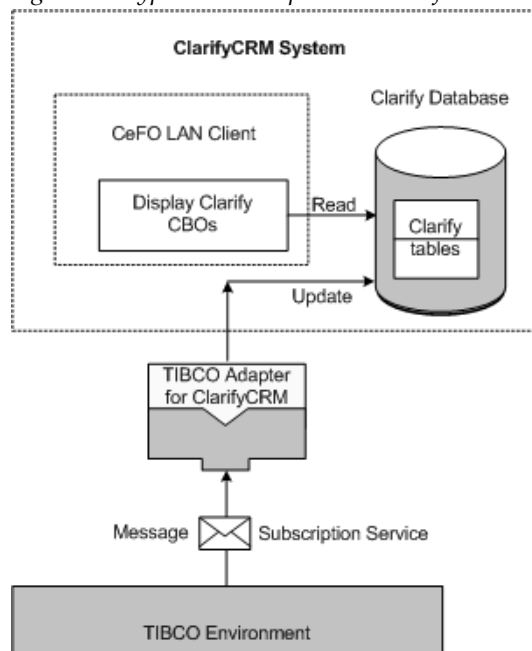
In the ClarifyCRM system, Parts at a site are typically associated with attributes like quantity, status, warranty end dates, etc. To update the status or quantity for an external application like an order management system, the adapter can be configured to publish this data on to the TIBCO environment.

Subscription Service

The Subscription Service receives a message from external applications through the TIBCO environment, processes it and based on the message fields performs the requested operation.

The Subscription Service parses the message, inserts or updates the data in the ClarifyCRM database using CBOs and Wrapper classes. The Subscription Service can be used to invoke a Custom JavaBean method. For details, see Subscription Service Functionality on page 77.

Figure 3 Typical Subscription Service flow



Subscription Service Example

In the ClarifyCRM system, a *site* is associated with a primary address, a ship-to address, and a bill-to address for every customer. If an external billing system makes changes to the bill-to address of a particular customer, then the changed billing information for that customer needs to be updated in the ClarifyCRM system. The Subscription Service can be configured to receive this updated site information published by the external billing system.

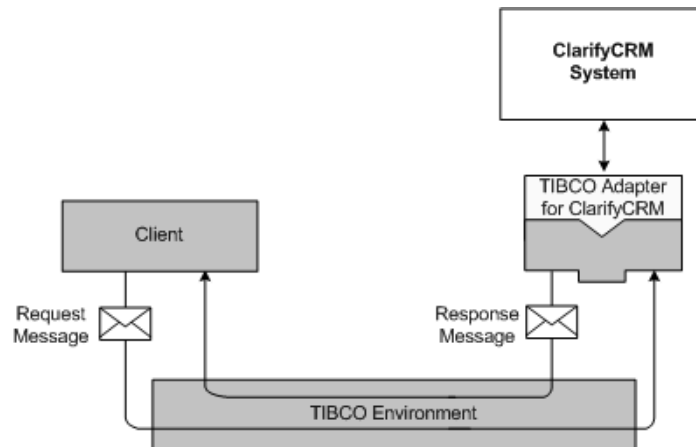
Request-Response Service

The Request-Response Service supports three operations (QUERY, WORKFLOW and CUSTOMIZED) and uses incoming request names to determine the operation type.

1. The adapter receives a request from the TIBCO environment and sends the request to the ClarifyCRM system.
2. When a response is returned to the adapter from the ClarifyCRM system, the adapter sends the response to the TIBCO environment.

The adapter supports request-response scenarios with a Remote Procedural Call (RPC) server. When the adapter receives a request, it takes the requested data, converts it into a formatted ClarifyCRM schema, and sends it to ClarifyCRM by using CBOs. The adapter then returns the response to the client or an external application. For details, see Request-Response Service Functionality on page 87.

Figure 4 Typical Request-Response Service flow



Request-Response Service Examples

In case of a workflow operation, a request is made to the ClarifyCRM system. The adapter creates an object of the specified CBO using the record identifier and executes the workflow operation on that object. A reply containing the status of the workflow operation is sent out to the client.

- **WORKFLOW operation example** — A case registered in an existing Call Support Center can be assigned or forwarded to a Call Service Representative, Sales Representative or Field engineer depending on the nature of the case logged.
- **QUERY operation example** — QUERY operation can be used to retrieve all cases related to a particular site.

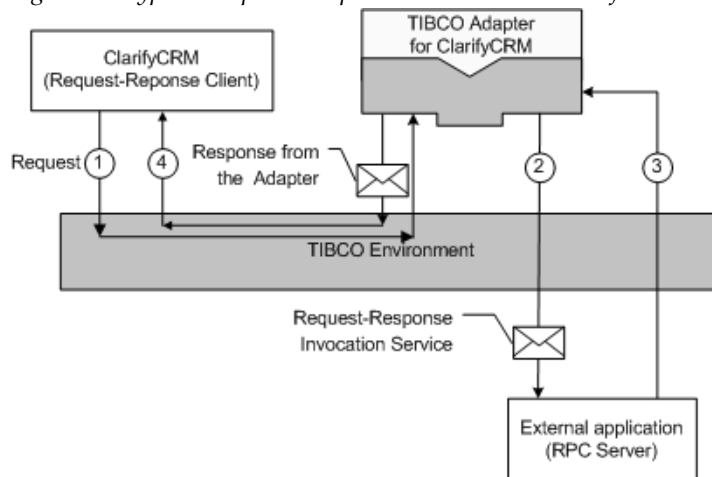
Request-Response Invocation Service

Request-Response Invocation Service receives requests from and sends responses to ClarifyCRM system.

1. The adapter receives a request from the ClarifyCRM system (acting as a Request-Response client).
2. The request is sent to an external application (acting as an RPC server) via the TIBCO environment.
3. When a response is returned to the adapter from the external application, the adapter sends the response back to the ClarifyCRM system.

The Request-Response Invocation service supports the Flexible and Non-Flexible approaches. For details, see Request-Response Invocation Service Functionality on page 99.

Figure 5 Typical Request-Response Invocation Service flow



Request-Response Invocation Service Example

The ClarifyCRM system can QUERY an external application for information such as SiteID or associated subcases for a queue.

Adapter and ClarifyCRM Interaction

The adapter is a bi-directional gateway between the ClarifyCRM system and external applications via the TIBCO environment.

The adapter extracts and stores CBO definitions from the ClarifyCRM system into a project at design-time. Each adapter service can be associated with a single main ClarifyCRM object and multiple contained ClarifyCRM objects. The contained objects are used for look-up, query or creation of the main ClarifyCRM object.

The adapter uses wrappers written for CBOs to perform the above-mentioned operations. A Wrapper class provides more functionality than its underlying business object. It encapsulates business logic, which helps in creating or updating a business object. Wrappers for most of the standard CBOs are packaged and provided with the adapter. For further details on how to write wrappers, refer to Wrapper Classes on page 311.

Examples of how CBOs can be used with the adapter are given next:

- Create Clarify objects in the ClarifyCRM system using the Subscription Service of the adapter.

Example: Create a site in the ClarifyCRM System. A site is associated with address information of the customer. Typically this address could be associated with primary, shipping or the billing address of the customer. If the billing address is changed in an existing billing application, it is critical that corresponding changes be made in the respective ClarifyCRM system. The subscription service of the adapter can be used to achieve this.

- Update Clarify objects in the ClarifyCRM system.

Example: A Contact in the ClarifyCRM system could be a person associated with the customer or any other third party vendor. Each customer can have a single or multiple sites created depending on the business needs. For example, you can have a site for production unit, one for development unit, one for research and development, and so on. Most of the time it could be a single person associated with all the sites. The QUERY operation could be used to list the sites associated with a particular contact and update the information.

- Query a Clarify object using the Request-Response Service of the adapter.

Example: A Case in a ClarifyCRM system is typically used in call centers. The Case contains information about the nature of the call, solution provisioned and the information of the person who has logged the call. You can query and update contact information associated with a case.

- Perform workflow operations provided in the ClarifyCRM system.

Example: A Workflow operation example would be, a case registered in an existing Call Support Center that can be assigned or forwarded to a Call Service Representative, Sales Representative or Field engineer depending on the nature of the call.

- Perform customized operations that a particular ClarifyCRM system may need.

Example: In the example provided with the adapter, the required fields from a business object can be updated in a custom table. See Function — `performYank()`: on page 329 for a detailed description.

Chapter 2 **Installation**

This chapter describes the procedures for installing and uninstalling TIBCO Adapter for ClarifyCRM 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 20*
- *Pre-Installation Worksheet, page 22*
- *Installer Overview, page 24*
- *Installation Registry, page 27*
- *Adapter Components and Compatible Software, page 29*
- *Installation on Microsoft Windows, page 31*
- *Installation on UNIX Systems, page 36*
- *Installer FAQs and Troubleshooting, page 42*

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 complete this process. It provides a checklist of parameters you should obtain from various system administrators within your organization before installing the adapter. Note that obtaining a ClarifyCRM account can take some time depending on your corporate policies — so plan in advance!

Operating System Requirements

Obtain the following information from the administrator of the machine on which you plan to install the adapter.

Do you have the required credentials to run the installer?

- On Microsoft Windows, administrator privileges are required to install.
- On UNIX systems, you can install as root or a regular user. See *Installer Account* on page 38 for details.

Identify the directory where the adapter is to be installed.

- Note that the TIBCO Runtime Agent (TRA) must be installed prior to installing the adapter and a typical adapter installation always places files under the TIBCO root directory that was set when the TRA was installed.

There should be enough space on the disk or partition to install the adapter. The adapter needs space in your temporary area and the directory where it is installed.

- See *Installation Registry* on page 27 for details about temporary folder space requirements on Microsoft Windows and UNIX systems.
- See Table 9 on page 31 for Microsoft Windows installations.
- See Table 11 on page 36 for Unix System installations.

You must have `write` permissions to these directories to install the adapter. In addition, on UNIX systems certain other permissions must be set to run the adapter. See *Permission Requirements on UNIX Systems* on page 39 for details.

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

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

ClarifyCRM Applications Requirements

Determine which ClarifyCRM Applications versions are supported.

- See Supported ClarifyCRM Applications on page 31 for Microsoft Windows.
- See Supported ClarifyCRM Applications on page 37 for UNIX systems.

Install the following software on the machine you intend to install the adapter:

- Clarify eBusiness Framework
- To connect to the Oracle database, install the Oracle client and Oracle JDBC driver (this is included in the Oracle client installation).
- To connect to the Microsoft SQL server, install the Microsoft JDBC driver. This can be downloaded from the Microsoft site.

Pre-Installation Worksheet

Use this form to capture the information you will need to collect before installing TIBCO Adapter for ClarifyCRM.

Table 4 Adapter Machine Information

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

Table 5 Clarify Server Information

Field Name	Field Description	Field Value
Clarify Database Host: (Example: Clarify)	Name of the machine on which the Clarify database instance is running.	
Clarify Database Instance Name: (Example: cly001)	Instance name of the Clarify database.	
Clarify Database Server Name: (Example: clarify)	<ul style="list-style-type: none"> For Oracle, the SQL net connection string. For Microsoft SQL, name of the server machine hosting the database. 	

Table 6 Clarify User Information

Field Name	Field Description	Field Value
Username: (Example: athps5)	Valid Clarify User ID to log on to Clarify.	
Password (Example: Clarify)	Password for the above account.	

Clarify Client Software

Please specify where software for Clarify can be found 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. You can invoke the installer by clicking on the executable.

Console Mode

Console mode allows you to run the installer from the command line. This is useful if your machine does not have a GUI environment or if you wish to install in silent mode.

Silent Mode

Silent mode either installs using default settings or uses a response file that was saved during an earlier installation. Silent mode installs without prompting you for information.

- If no response file has been recorded earlier and you invoke the installer with the `-silent` argument, the default installation parameters are 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, 5.1.0 indicates a minor release and 5.1.1 indicates a patch release.

The installer for a patch release performs an automatic upgrade. For example, the installer automatically upgrades TRA 5.0.0 to 5.0.1 by overwriting the contents of the 5.0 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 5.0.0 release and are upgrading to a 5.2.0 minor release, it will be installed under the 5.2 directory. This allows both the 5.0 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 may 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 wish to uninstall, you are 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><Adapter>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**.
- Use 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`.

Installation Registry

The installer maintains an installation registry. The registry location depends on platform. This section explains where the registry files are located. The files have vpd as a prefix, which stands for Vital Product Database.



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

Microsoft Windows Platforms

ActiveEnterprise 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 temporary folder, typically SystemDrive:\Temp or SystemDrive:\Documents and Settings*<user_name>*\Local Settings\Temp

The installer requires 50 MB of free space in the temporary 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
```

On Solaris and HP-UX, if installation is performed by super-user (root), the installation registry is maintained in the root user's home directory (which is /) with two vpd files.

Installer Disk Space Requirements in Temporary Area

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

On UNIX platforms the following disk space is required:

- On Solaris, 73 MB of free disk space in /var/tmp
- On HP-UX, 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:

```
<install_package_name>.bin -is:tempdir /new_tmp
```

where /new_tmp has sufficient free disk space.

Disk Space Requirements in User's Home Directory

On UNIX platforms, when a regular (non-root) user installs a TIBCO product, 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 Windows: `SystemDrive:\Windows\TIBCOInstallationHistory.xml`

On UNIX: `Users_Home_Directory/TIBCOInstallationHistory.xml`

The file `TIBCOInstallationHistory.xml` therefore contains the 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

You can install different adapter components on different machines. For example, you can run the run-time adapter on one machine and install the design-time components on another. This allows you to configure an adapter on one machine and run it on another.

Adapter Components

Table 7 describes the adapter components on the adapter installation package.

Table 7 TIBCO Adapter Components

Component	Description
Run-time adapter	This process does the actual work of passing and converting data to and from the vendor application. Parameters of data exchanges are stored in projects created using the adapter palette and the design-time adapter (see below).
Adapter palette	Adapter-specific GUI that is loaded in TIBCO Designer (see next section for details) at configuration time.
Design-time adapter	Provides meta-data introspection or schema browsing about business objects in the vendor application at design-time. The design-time adapter is installed on Windows as a service and set to start automatically. On UNIX, the design-time adapter must be started manually.

Required and Optional TIBCO Products

Depending on the tasks to perform, you must install one or more other TIBCO products. Table 8 describes required and compatible products and their purpose. For the latest versions of the listed products, refer *TIBCO Adapter for Clarify readme.txt*.

Table 8 Compatible 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 RVD libraries, SDK libraries, etc. For example, <code>maverick50.dll</code>.</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>
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 exceeds 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 standalone installation and can be installed after installing the adapter.</p>
TIBCO ActiveMatrix BusinessWorks	<p>Optional. TIBCO ActiveMatrix 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 ActiveMatrix BusinessWorks is available as a standalone installation and can be installed after installing the adapter.</p>
TIBCO Enterprise Message Service	<p>Optional. TIBCO Enterprise Message Service allows you to use the Java Messaging Services (JMS) as the message transport for your adapter.</p> <p>TIBCO Enterprise Message Service (EMS) is available as a standalone installation and can be installed after installing the adapter.</p>

Installation on Microsoft Windows

Before starting the installation, review the topics in this section to determine that your system meets the basic requirements and that you have the prerequisite software installed.

Memory requirement for the adapter installation is around 256 MB.

The following is a list of prerequisites for installing the adapter on Microsoft Windows systems. See *Installer Disk Space Requirements in Temporary Area* on page 27 for additional disk space requirements.

Table 9 also lists the platform-specific installation packages, where *<version_num>* is the Adapter release number. For example, the installation package name for the TIBCO Adapter for ClarifyCRM 5.4 on the Microsoft Windows 2003 platform is `TIB_adclycrm-simple_5.4.0_win_x86.exe`.

Table 9 Supported Platforms, Package Names, and Disk Space for Microsoft Windows

Platform	Hardware	Package Names	Disk Space (MB)
Microsoft Windows 2003	x86	TIB_adclycrm-simple_<version_num>_win_x86.exe	30 MB

Supported ClarifyCRM Applications

Before installing the adapter, install the following components from the ClarifyCRM system:

- ClarifyCRM LAN Client
- ClarifyCRM Web Client
- ClarifyCRM Applications

Before proceeding to adapter installation, ensure you can connect to the ClarifyCRM system using one of the clients mentioned above.

The adapter works with the following:

- Amdocs CES 7.5
- Amdocs CRM 7.1
- Amdocs 6.0 (ClarifyCRM 13.1)

Table 10 lists ClarifyCRM Applications versions, Oracle client and Microsoft SQL server combinations for Windows platforms.

Table 10 ClarifyCRM Matrix for Windows Platforms

ClarifyCRM	Platform	Oracle Client	Microsoft SQL Server
Amdocs CES 7.5 Amdocs CRM 7.1	Windows 2003	Oracle 10g	Microsoft SQL Server 2005
Amdocs 6.0 (ClarifyCRM 13.1)	Windows 2003	Oracle 9i	Microsoft SQL Server 2000

TIBCO Runtime Agent Must Be Installed Before the Adapter

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

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

Installer Account

You must have administrator privileges to install this product.

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.

Install from Network Drive

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

Install on Windows 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:\Windows\`.



Windows Terminal Server must be running in remote admin mode, not application sharing mode. The adapter is not supported if installed on a machine that 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 then 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
```

Change back to Execute mode after installation is completed by typing:

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

To check your current mode, type the following:

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

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 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 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_adclycrm-simple_<version_num>_win_x86.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_adclycrm-simple_<version_num>_win_x86.exe -is:javaconsole  
-TIBconsole
```

Install Using Silent Mode

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

```
TIB_adclycrm-simple_<version_num>_win_x86.exe -silent
```

Install and Generate a Response File

You can generate a response file during any installation of the product and 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_adclycrm-simple_<version_num>_win_x86.exe -options-record  
<SystemDrive>\directory\<responseFileName>
```

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_adclycrm-simple_<version_num>_win_x86.exe -options  
<SystemDrive>\directory\<responseFileName>
```

Combining Options

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

```
TIB_adclycrm-simple_<version_num>_win_x86.exe -silent -options  
<responseFileName>
```

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

```
TIB_adclycrm-simple_<version_num>_win_x86.exe -is:javaconsole  
-console -options-record <responseFileName>
```

Post-Installation Tasks on Microsoft Windows

Install JRE

1. After installing the adapter, install the required JRE version.
2. Change the following parameters in the `adclycrm.tra` file available in the `\bin` directory of the adapter installation.

`tibco.env.PATH` should point to the `\bin` and `\lib` folders of the JRE installation.

`java.library` should point to `jvm.dll` in the `client` folder of the JRE installation.

Enable Publishing

To enable publishing, carry out the following steps in the ClarifyCRM Applications and TIBCO Designer before you run the adapter.

1. Use the Clarify tool `dataex` with the `-gensc` option to export your Clarify schema.
2. Add the TIBCO Staging tables in `tibco.sch` to the Clarify schema file generated from the previous step to make the TIBCO Staging tables part of the Clarify extended schema.
3. Use `dataex` with the `-import` option to import `tibco.dat` into the ClarifyCRM system.
4. Use the Clarify tool `cbex` to import the ClearBasic script `tibcoGlobal.cbs` into the ClarifyCRM system.
5. Open the project in TIBCO Designer.
6. Change ClarifyCRM system-related parameters in TIBCO Designer.
7. Start the adapter.



For detailed instructions, refer Add TIBCO Staging Tables to the ClarifyCRM Database on page 53.

Installation on UNIX Systems

Memory requirement for the adapter installation is around 256 MB. Your operating system must meet the minimum patch requirements listed next. See *Installer Disk Space Requirements in Temporary Area* on page 27 for additional disk space requirements.

Table 11 also lists the platform-specific installation packages, where *<version_num>* is the Adapter release number. For example, the installation package name for the TIBCO Adapter for ClarifyCRM 5.4 on the HP-UX 11 platform is `TIB_adclycrm-simple_5.4.0_hpux110_hppa_java_.bin`.

Table 11 Supported Platforms, Package Names, Patches and Disk Space for UNIX Systems

Platform	Package Names	Minimum Patch(es)	Disk Space (MB)
Solaris 10	<code>TIB_adclycrm-simple_<version_num>_sol8_sparc_java_.bin</code>	Make sure you have the latest J2SE Cluster Patches for Solaris. See http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/J2SE	75 MB
Solaris 9	<code>TIB_adclycrm-simple_<version_num>_sol8_sparc_java_.bin</code>	Make sure you have the latest J2SE Cluster Patches for Solaris. See http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/J2SE	75 MB
Solaris 8	<code>TIB_adclycrm-simple_<version_num>_sol8_sparc_java_.bin</code>	Make sure you have the latest J2SE Cluster Patches for Solaris. See http://sunsolve.sun.com/pub-cgi/show.pl?target=patches/J2SE	75 MB
HP-UX 11	<code>TIB_adclycrm-simple_<version_num>_hpux110_hppa_java_.bin</code>	Make sure you have the proper JRE version, refer to Amdocs Platform Guide.	30 MB
HP-UX 11i	<code>TIB_adclycrm-simple_<version_num>_hpux110_hppa_java_.bin</code>	Make sure you have the proper JRE version, refer to Amdocs Platform Guide.	30 MB

Table 11 Supported Platforms, Package Names, Patches and Disk Space for UNIX Systems (Cont'd)

Platform	Package Names	Minimum Patch(es)	Disk Space (MB)
HP-UX 11iv2 on Itanium	TIB_adclycrm-simple_<version_num>_hpux112_hppa_ia64_java_.bin	Make sure you have the proper JRE version, refer to Amdocs Platform Guide.	30 MB

Supported ClarifyCRM Applications

Before installing the adapter, install the following components from the ClarifyCRM system:

- ClarifyCRM LAN Client
- ClarifyCRM Web Client
- ClarifyCRM Applications

Before proceeding to adapter installation, ensure you can connect to the target application using the vendor client.

The adapter works with the following:

- Amdocs CES 7.5
- Amdocs CRM 7.1
- Amdocs 6.0 (ClarifyCRM 13.1)

Table 12 lists ClarifyCRM Applications versions and Oracle client combinations for all the UNIX platforms.

Table 12 ClarifyCRM Matrix for Unix Platforms

ClarifyCRM	Platform	Oracle Client
Amdocs CES 7.5	Solaris 8, Solaris 9, Solaris 10, HP-UX 11.0, HP-UX 11i, HP-UX 11i v2 on Itanium	Oracle 10g
Amdocs CRM 7.1	Solaris 8, Solaris 9, Solaris 10, HP-UX 11.0, HP-UX 11i, HP-UX 11i v2 on Itanium	Oracle 10g
Amdocs 6.0 (ClarifyCRM 13.1)	Solaris 8, Solaris 9, Solaris 10, HP-UX 11.0, HP-UX 11i, HP-UX 11i v2 on Itanium	Oracle 9i

TIBCO Runtime Agent Must be Installed Before the Adapter

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

During installation, the adapter installer checks for the availability of all required products in the system. If any of these are unavailable, the installer does 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 27 for more information.

Windows Environment

A Windows environment such as CDE (i.e. X Windows) is required to run the installer in GUI mode. It is not required for a Console or Silent mode installation.

Installing the Adapter on UNIX

After running 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 prompts you to choose which components of the product are to be installed 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. The examples assume you are installing the adapter on Solaris platform.

Install Using GUI Mode

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

```
% ./TIB_adclycrm-simple_<version_num>_sol8_sparc_java_.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 in a terminal window:

```
% ./TIB_adclycrm-simple_<version_num>_sol8_sparc_java_.bin
-is:javaconsole -console
```

Install Using Silent Mode

Silent mode allows you to install the software without prompts using default values. Type the following in a terminal window:

```
% ./TIB_adclycrm-simple_<version_num>_sol8_sparc_java_.bin -silent
```

Install and Generate a Response File

You can generate a response file during any installation of the product and 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_adclycrm-simple_<version_num>_sol8_sparc_java_.bin
-options-record
/dir/<responseFileName>
```



The response file does not record selections at the component level. It records all other selections, for example, which products you wish to install.

Combining Options

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

```
% ./TIB_adclycrm-simple_<version_num>_sol8_sparc_java_.bin -silent
-options /dir/<responseFileName>
```

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

```
% ./TIB_adclycrm-simple_<version_num>_sol8_sparc_java_.bin
-is:javaconsole -console -options-record /dir/<responseFileName>
```

Permission Requirements on UNIX Systems

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

```
$TIBCO_HOME/adapter/adclycrm/<version_num>/bin
```

```
$TIBCO_HOME/adapter/adclycrm/<version_num>/logs
$TIBCO_HOME/adapter/adclycrm/<version_num>/ledger
$TIBCO_HOME/tra/<version_num>/logs
$TIBCO_HOME/logs
```

For example, if the adapter has been installed in `/opt/tibco`, the user who installed the adapter can make these directories writable for all other users by executing the following commands:

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

Post-Installation Tasks on UNIX



For HP-UX

If the oracle version used is Oracle 9i or Oracle 10g, the `SHLIB_PATH` for the Oracle installation should point to `lib32` and `LD_LIBRARY_PATH` to `lib` folder of the Oracle client.

Install JRE

1. After installing the adapter, install the required JRE version.
2. Change the following parameters in the `adclycrm.tra` file available in the `\bin` directory of the adapter installation.

`LD_LIBRARY_PATH` should point to the `sparc/bin` and `sparc/lib` folders of the JRE installation.

`java.library` should point to `libjvm.so` in the `sparc/client` folder of the JRE installation.

Enable Publishing

To enable publishing, carry out the following steps in the ClarifyCRM Applications and TIBCO Designer before you run the adapter.



Execute the `envvars` script available in the `/bin` directory of the adapter installation to set the ClarifyCRM related environment variables.

1. Use the Clarify tool `dataex` with the `-gensc` option to export your Clarify schema.

2. Add the TIBCO Staging tables in `tibco.sch` to the Clarify schema file generated from the previous step to make the TIBCO Staging tables part of the Clarify extended schema.
3. Use the Clarify tool `SchemaMgr` with the `-upgrade` option to add the TIBCO Staging tables to the ClarifyCRM system.
4. Use `dataex` with the `-import` option to import `tibco.dat` into the ClarifyCRM system.
5. Use the Clarify tool `cbex` to import the ClearBasic script `tibcoGlobal.cbs` into the ClarifyCRM system.
6. Open the project in TIBCO Designer.
7. Change ClarifyCRM system-related parameters in TIBCO Designer.
8. Start the adapter.



The `SchemaMgr` tool and `dataex` tool are present in the `dbadmin` directory of the ClarifyCRM system. The `cbex` tool is available in the `CBToolkit` directory of the ClarifyCRM system. The `tibcoglobal.cbs` and the `tibco.sch` can be found in the path `C:\tibco\adapter\adclycrm\<version_num>\setup`.



For detailed instructions, refer `Add TIBCO Staging Tables to the ClarifyCRM Database` on page 53.

Installer FAQs and Troubleshooting

This section lists some frequently asked questions, common errors along with their causes and solutions.

Frequently Asked Questions

Where is the installation log file located?

Installation and uninstallation log files are created in the `TIBCO_HOME\log` directory.

Will 5.4 installer recognize a 4.x installation?

TIBCO products follow a three digit release numbering scheme:

Major.Minor.Patch

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, 5.4 is a minor release and hence will not recognize either 4.x product installations.

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

For the installer in GUI mode on UNIX, you must open an additional window, generally for graphics. It uses the `DISPLAY` environment variable to let the user 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 name of the computer (on which the installer graphics window is 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
```

Consider a scenario where you need to install the adapter on a remote HP-UX machine (named `itaska`). You have a Solaris 10 machine (named `alaska`) that has a video card and monitor installed, and you can run an X-window application on it. So you decide to telnet to `itaska` from `alaska`.

When you telnet to `itaska`, you will not get access to `itaska`'s monitor and will be unable to display an X-window application. That is why 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_adclycrm-simple_<version_num>_hpux110_hppa_java.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 then creates 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

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

If disk space is consumed by another process while the installer is copying the files, and if the required disk space is thereby reduced, the installer may fail and give a failure message.

Solution

While performing installation, avoid running other processes that consume disk space in product home location.

Installation Errors on HP-UX 11.00 64 Bit Platform

Error Message

Installation on an HP-UX 11.00 64 bit system may crash with the following error message:

```
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. 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
```

Solution

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 it does not have the patch, first install HP-UX kernel patch PHKL_27282 and then proceed with the installation. Installation of patch PHKL_27282 will reboot your system.

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 and do not have a Java Runtime Environment (other than the TIBCO JRE) installed, the TIBCO Hawk Configuration tool is unable to determine the Java home location and the JVM executable. The TIBCO Hawk services will not start correctly and you will be unable to start the TIBCO Hawk Display.

Solution

1. Start the TIBCO Hawk Configuration tool. For example, on Microsoft Windows:
Start>TIBCO>TIBCO Hawk>Hawk Configuration
2. Under the General tab, click **Advanced**.
3. In the Java Home Directory field, provide the path to Java. For example:
C:\tibco\jre\1.5.0
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.

Cannot Install the Adapter

On HP-UX platforms, even though the correct version of TIBCO Runtime Agent version is already installed, installation of an adapter that depends on TIBCO Runtime Agent may 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.

Solution

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
```

where `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
```

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

Chapter 3

Publication Service Functionality

This chapter describes how the Publication Service processes messages received from the ClarifyCRM system and sends the messages to applications configured for the TIBCO environment.

Topics

- *Publication Service Overview, page 48*
- *How the Publication Service Works, page 52*
- *Prepare ClarifyCRM Components for the Publication Service, page 53*
- *Add TIBCO Staging Tables to the ClarifyCRM Database, page 53*
- *Enable the Publication Service for LAN Client, page 54*
- *Enable the Publication Service for ClarifyCRM Web Client, page 60*
- *Enable the Publication Service for ClarifyCRM Applications, page 66*
- *Limitations of the Publication Service, page 75*

Publication Service Overview

The Publication Service consists of the following components:

- A set of TIBCO Staging tables(`table_tibco_message_queue` and `table_tibco_message_fields`), which stores the information to be published. The tables must be added to the Clarify system.

TIBCO Staging tables are used to store the inserted and updated data in the ClarifyCRM system. The Publication Service periodically polls these tables and publishes the new messages. The `table_tibco_message_queue` and `table_tibco_message_fields` are the TIBCO Staging tables that contain the data for an outgoing message. For details, see TIBCO Staging Tables on page 255.

- ClearBasic script that uses ClearBasic APIs to capture the data into TIBCO Staging tables. The script is executed when a user saves data to the ClarifyCRM system by clicking a **Done**, **Replace**, **Add**, or **Save** button on a ClearBasic form.

In case of Web client this is done through Java classes, which uses CBOs to capture data into the database.

In case of ClarifyCRM Applications, data is captured using the Data Change Event feature of Integration Gateway.

- The adapter picks these new records from the database and publishes the same to the TIBCO environment.

Publication Service Features

- The Publication Service allows the publication of events generated from both the web, LAN clients, and ClarifyCRM Applications.
- The Publication Service is multi-threaded. The number of threads can be configured.
- Multiple instances of the adapter can publish from the same Clarify instance, providing load balance.
- Multiple Publication Service endpoints can be defined in an adapter instance.

It is recommended that the subject names (of messages to be published) specified by the ClearBasic or web forms be the same as the subject names configured in the adapter configuration. In the case of ClarifyCRM Applications, the event name configured in Data Change Event should be the same as the one specified in the adapter configuration.

- The number of records that the adapter retrieves from the ClarifyCRM system can be configured.
- The adapter can be configured to publish only a subset of attributes for the business object.
- The adapter uses the user configured CBO schema to publish the messages. If no user-defined schema is found, then the adapter uses a generic schema that contains the object name, operation type and the field name and values as name-value pairs.
- The adapter can be configured to either delete the messages in the ClarifyCRM system once they are published or change the status of the message to published.
- The adapter supports publication of schemas, which contain CBOs.

The schemas can be configured by retrieving the necessary CBOs from the ClarifyCRM system through the design-time adapter. If multiple CBO schemas of the same type have to be published in a message, they need to be given unique names. For example, to publish two address objects, they need to be named AddressR1 and AddressR2 where R1 and R2 are the current object names in the TIBCO Message Queue table.

- If one of the messages in the TIBCO Message Queue table cannot be associated with any of the configured Publication Services (if the subject names do not match), the message will be published using the default Publication Service. The subject name of the message will be the same as the subject name in the table. This message will use the generic schema.

Publication Service can be configured to publish the messages either in a generic or a CBO based schema format.

Message Formats that Use Generic Schema

Table 13 Incoming Event

Instance Name	Field Type	Field Name	Field Value Description
IncomingEvent	M_TREE	Schema	The value of schema field is a sequence of opObject instances. The structure of opObject instance in an incoming message is shown in Table 14.
	M_STRING	Name	Event name. For example, Update contact. You can give the event any name to differentiate the captured events.

Table 14 opObject

Instance Name	Field Type	Field Name	Field Value Description
OpObject	M_STRING	op_type	Not used.
	M_STRING	Name	A valid Clarify object name such as contact or site.
	M_STRING	Identifier	<p>The object instance reference name. It is used to relate this object instance to another object instance because it is possible to have multiple object instances with the same name.</p> <p>A reference name has significance only within the message. It must be unique among the message scope. It is used as a key to uniquely identify an opObject instance within a sequence of opObject instances.</p>
	M_TREE	Fields	A sequence of field instances. The structure of each field instance is shown in Table 15.
	M_TREE	ref_relations	A sequence of relation instances. The structure of each relation is shown in Table 16.

Table 15 Field

Field Name	Type	Value	Value Description
field	M_STRING	Name	A valid Clarify object field name.
	M_STRING or M_INT or M_REAL	Value	M_STRING value if a Clarify field is date-time type or string type; M_INT value if a Clarify field is long or Boolean type; and M_REAL value if a Clarify field is double or float type.

Table 16 Relation

Field Name	Type	Value	Value Description
relation	M_STRING	Name	A valid Clarify relation name such as contact_role2site.
	M_STRING	Parent_ identifier	Refer to a valid opObject instance within this message scope. The value is that of the identifier attribute of an opObject instance.

Message Formats that Contain CBOs

Table 17 Message Formats that Contain CBOs

Field Name	Type	Value	Value Description
Optype	M_STRING	n/a	Name of the operation being performed. This will be the same as the value for the operation field in the TIBCO_MESSAGE_QUEUE table.
Schema	Class	n/a	The CBO, which needs to be part of the message has to be included here.
Lookup	Sequence	n/a	Sequence of look up values.
ref_relations	Sequence	n/a	A sequence of relation instances. Each relation contains the name and parent identifier fields.

How the Publication Service Works

1. The events generated in the ClarifyCRM system are captured in the TIBCO Staging tables by ClearBasic scripts (LAN client) or JSP code (Web forms).
2. The adapter uses two ways to retrieve the new events:
 - Polls the TIBCO Staging tables periodically.
 - Uses the automatic event logging feature configured using Data Change Event editor.

The number of messages to be retrieved from the TIBCO Staging tables is configurable. The retrieved messages are then queued in memory by the adapter.

3. The threads pick up the messages sequentially from the queue and publish them using the configured Publication Service endpoints.

The Publication Service process is multi-threaded, enabling it to process messages that are picked up from the queue, quickly. You can configure the number of threads to be created by the Publication Service process in the adapter instance. However, the poller process, which retrieves the events from the TIBCO Staging tables, is single threaded.

Prepare ClarifyCRM Components for the Publication Service

To prepare ClarifyCRM components for the Publication Service:

1. Log on to the ClarifyCRM system.
2. Add TIBCO Staging tables to the ClarifyCRM database.
3. Enable the Publication Service for the ClarifyCRM LAN client.
4. Enable the Publication Service for the ClarifyCRM Web client.
5. Enable the Publication Service for the ClarifyCRM Applications.

Add TIBCO Staging Tables to the ClarifyCRM Database

To enable publishing, TIBCO Staging tables (`table_tibco_message_queue`, `table_tibco_message_fields`) are required to be added to the ClarifyCRM database. Carry out the following steps to add the tables:

1. Back up the current database.
2. Use the Clarify `dataex` utility to get your current database schema. The command for executing this utility is:

```
dataex -user_name <user name> -password <password> -db_name
<database name> -db_server <server name> -gen_sc <schema name>
```

For example,

```
dataex -user_name sa -password sa -db_name Clarify
-db_server WINGz -gen_sc george.sch
```

3. Check if you have any customized tables with type ID 502, 503, 504 or 510 in the schema file generated in the previous step.
 - If there are no such tables, directly append `tibco.sch` to the end of the generated schema file.
 - If there are such tables, change the three tables in `tibco.sch` to three other unused numbers in the range 470-511 or 2000-4999; then append `tibco.sch` to the generated schema file.

4. Copy the TIBCO Staging tables to Clarify database.
 - a. Run the following command to convert the old .sch format to the new .xml format:
`SchToXml.exe tibco.sch test.xml`
 - b. Append test.xml contents to objectSchema.xml. This is the Clarify Object schema.
 - c. Run the following command to copy the TIBCO Staging tables to Clarify database:
`C:\AmdocsCRM7.5\Server\dbadmin>java -cp ClfySchemaMgr.jar
com.clarify.schemamgr.SchemaMgr -user_name sa -password sa
-db_server cl131qnu.APAC.TIBCO.COM -db_name cl131QNU
-upgrade objectSchema.xml -preview -fullecho -trace config
ntracefile clfy131wdbchanges.txt`
5. Import tibco.dat using the datex utility. For example,
`dataex -user_name sa -password sa -db_name Clarify
-db_server WINGz -imp tibco.dat`



The files `tibco.sch` and `tibco.dat` are installed in
`<TIBCO_HOME>/adapter/adclycrm/<version_num>/setup` directory.



TIBCO Staging tables added by a previous ClarifyCRM system are not removed during installation of the adapter.

Enable the Publication Service for LAN Client

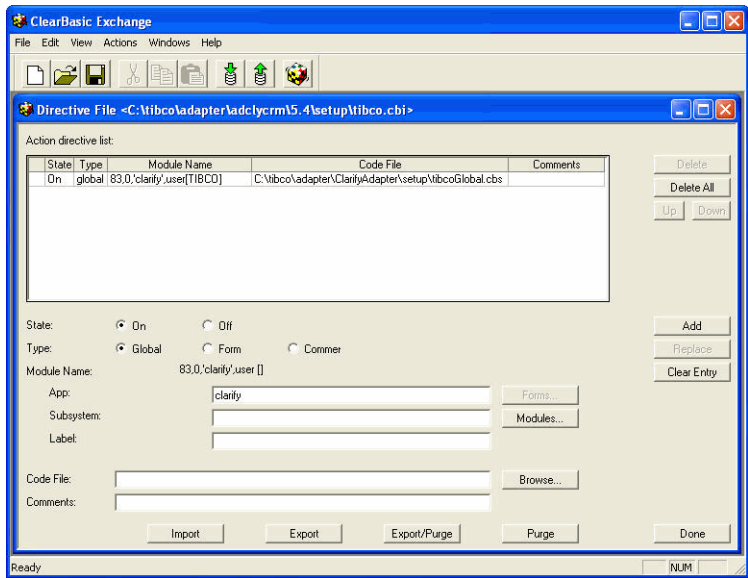
This section describes the procedure to enable the Publication Service for the ClarifyCRM LAN client. The procedure assumes you are familiar with Clarify User Interface Editor and ClearBasic Exchange (cbex). For more information, refer to *Clarify User Interface Editor Guide* and *A ClearBasic Programmer's Guide*.

Import ClearBasic Global Files

Import `tibcoGlobal.cbs` (setup directory), as follows:

1. Execute the following cbex command:
`cbex -dir tibco.cbi`
 The Directive File window is displayed.

Figure 6 Directive File window



- 2. Verify that the file path is correct in the Code file column. Go to step 3. If the path is not correct, use the **Browse** button to select the right path. Then click **Replace**. Continue with step 3.
- 3. Click **Import**.

How the Publication Service Works for LAN Client

ClearBasic Global File

The adapter provides a ClearBasic Global File (`tibcoGlobal.cbs`) to facilitate the capture of business events from the ClarifyCRM system to the TIBCO Staging tables. The file contains a rich set of utility functions, which can be leveraged by the user to customize the ClearBasic forms. The exposed utility functions and their descriptions are given in Table 18.

Table 18 *ClearBasic Functions*

Function Name	Function Description
<code>Sub tibcoRecordWithFilterAndSubject(rec As Record, bulks As BulkSave, ByVal currMsgNum As Long, ByVal refObjName As String, filter As List, ByVal subjectName As String, ByVal closure As String)</code>	Inserts all fields of a Clarify record into TIBCO Staging tables. The function will insert a row into <code>table_tibco_message_queue</code> and multiple rows into <code>table_tibco_message_fields</code> . The number of rows inserted into <code>table_tibco_message_fields</code> is the number of fields that a Clarify object has.
<code>Sub tibcoRecordWithSubject(rec As Record, bulks As BulkSave, ByVal currMsgNum As Long, ByVal refObjName As String, ByVal subjectName As String, ByVal closure As String)</code>	Inserts selected fields of a Clarify record into TIBCO Staging tables. The function will insert a row into <code>table_tibco_message_queue</code> and multiple rows into <code>table_tibco_message_fields</code> . The number of rows inserted into <code>table_tibco_message_fields</code> is the number of fields given by the filter list.
<code>TibcoSetReferenceRelation (ByVal fromRefName As String, ByVal toRefName As String, ByVal from2toRel As String, ByVal currMsgNum As Long, bulks as BulkSave)</code>	Sets the relation between two Clarify records. The <code>fromRelName</code> and <code>toRelName</code> are object reference names rather than Clarify object names. The <code>from2toRel</code> is a valid Clarify relation name.
<code>tibcoPublish (ByVal closure as String)</code>	Internally, it calls <code>tibcoPublishImpl()</code> . Assign this subroutine to a button click callback subroutine. This subroutine creates its own <code>BulkSave</code> object.

Table 18 ClearBasic Functions (Cont'd)

Function Name	Function Description
<code>tibcoPublish2 (BulkS As BulkSave, ByVal closure as String)</code>	This is called by the Clarify subroutine <code>Form_Save1()</code> . Internally, it calls <code>tibcoPublishImpl()</code> . This subroutine does not create its own <code>BulkSave</code> object. Instead, it uses the <code>BulkSave</code> object passed in by <code>Form_Save1()</code> .
<code>tibcoPublishImpl(ByVal currMsgNum As Long, BulkS As BulkSave, ByVal closure as String)</code>	The callback interface that must be implemented for each form that has objects to be published.

For the adapter to publish a message out of the ClarifyCRM system, write a customized ClearBasic script in a ClarifyCRM form. Each ClearBasic script is tailored for a particular form because the ClearBasic code refers to user interface control names for that particular form.

Follow these steps to write your customized ClearBasic (CB) code. It is assumed you have good ClearBasic and Clarify `UIEditor` knowledge.

1. Identify the form and the operations (update or insert) that you wish to publish.
2. Save a copy of the form.
3. Write your CB code to customize the operations.
4. Save the customized CB code to the ClarifyCRM system.

In the following example, form 717 (Site form) is used to illustrate how to write your CB code to customize the site update behavior. For this example, assume you want to publish a site (all fields of site) and its related primary address, bill-to address, ship-to address (city, state, and zip code fields only) whenever the site is updated. The update operation is internally named `MODIFY` (see `UIEditor`) in this form.

Example

```
Private Sub tibcoPublishSite(ByVal closure as String)
    Dim currMsgNum As Long
    Dim BulkS As new BulkSave
    ON Error Goto ErrorHandler
    currMsgNum = tibcoNewMessageNum()
    Select Case currMsgNum
        case -1
            '-- message row is not there
    MsgBox "Error in fetching a new message number. Your changes will
    not be published.", ebOKOnly, "No Table Row"
```

```

        case Else
            call tibcoPublishImp(currMsgNum, BulkS, closure)
        End Select
    BulkS.Save
    Exit Sub

ErrorHandler:
    Dim errDesc as String
    errDesc = Err.Source & "::" & Err.Description
    MsgBox "Error:" & errDesc & "This operation will not be
published", vbOKOnly, "Error"
End Sub

Private Sub tibcoPublishImp(ByVal currMsgNum As Long, BulkS As
BulkSave, ByVal closure as String)
    Dim bulkR As New BulkRetrieve
    Dim billAddrList As New List
    Dim shipAddrList As New List
    Dim siteRec As New Record
    Dim primAddrRec As New Record
    Dim billAddrRec As New Record
    Dim shipAddrRec As New Record
    Dim filter As New List
    Dim subjectName As String

    subjectName = "TEST.SITE" ' insert subject name here

    Set siteRec = Cobj_Location.Contents
    Set primAddrRec = Cobj_primary_add_obj.Contents

    Set siteRec = Cobj_Location.Contents
    Set primAddrRec = Cobj_primary_add_obj.Contents
    bulkr.SetRootByID "site", siteRec.GetField("objid")
    bulkr.TraverseFromRoot 0, "cust_billaddr2address"
    bulkr.TraverseFromRoot 1, "cust_shipaddr2address"
    bulkr.RetrieveRecords
    Set billAddrList = bulkr.GetRecordList(0)
    Set ShipAddrList = bulkr.GetRecordList(1)
    If billAddrList.Count > 0 Then
        Set billAddrRec = billAddrList.ItemByIndex (0)
    Else
        Set billAddrRec = primAddrRec
    End If
    If shipAddrList.Count > 0 Then
        Set shipAddrRec = shipAddrList.ItemByIndex(0)
    Else
        Set shipAddrRec = primAddrRec
    End If

    filter.AppendItem "site_id"
    filter.AppendItem "name"
    filter.AppendItem "site_type"
    Call tibcoRecordWithFilterAndSubject(siteRec, BulkS,
currMsgNum, "R1", filter,subjectName, closure)
    Call tibcoRecordWithSubject(primAddrRec, BulkS,
currMsgNum, "R2",subjectName, closure)

```

```

        Call tibcoRecordWithSubject(billAddrRec, BulkS,
currMsgNum, "R3",subjectName, closure)
        Call tibcoRecordWithSubject(shipAddrRec, BulkS,
currMsgNum, "R4",subjectName, closure)
        call tibcoSetReferenceRelation("R1", "R2",
"cust_primaddr2address", currMsgNum, BulkS)
        call tibcoSetReferenceRelation("R1", "R3",
"cust_billaddr2address", currMsgNum, BulkS)
        call tibcoSetReferenceRelation("R1", "R4",
"cust_shipaddr2address", currMsgNum, BulkS)
End Sub

Sub Form_Load
    Me.DoDefault
End sub

Sub Modify_Click
    isTriggered = false
    Me.DoDefault
    If isTriggered Then
        call tibcoPublishSite("updateSite")
    End If
End Sub

Sub Add_Click
    isTriggered = false
    Me.DoDefault
    If isTriggered Then
        call tibcoPublishSite("addSite")
    End If
End Sub

Sub Form_Save3(bs As BulkSave)
    isTriggered = true
End Sub

```

ClearBasic Script Coding Conventions

This section describes the ClearBasic script coding conventions for writing objects into TIBCO Staging tables. Follow these steps for each form with objects to be published.

1. Declare the global functions in the beginning of each form.
 Declare Sub tibcoRecordWithSubject(rec As Record, bulks As BulkSave, ByVal currMsgNum As Long, ByVal refObjName As String, ByVal subjectName As String, ByVal closure As String)
 Declare Sub tibcoRecordWithFilterAndSubject(rec As Record, bulks As BulkSave, ByVal currMsgNum As Long, ByVal refObjName As String, filter As List, ByVal subjectName As String, ByVal closure As String)
 Declare Sub tibcoSetReferenceRelation(ByVal fromRefName As String, ByVal toRefName As String, ByVal from2toRel As String, ByVal currMsgNum As Long, bulks as BulkSave)
2. Declare a form-wide, boolean-type variable.

```
Dim isTriggered As Boolean
```

3. Find the button-click callback routines for which you need to add code, to publish a message. After identifying the callback routines, write the following lines of code:

```
isTriggered = false
Me.DoDefault
If isTriggered Then
call tibcoPublish"yyyy"("xxx")
End If
```

Where,

yyyy represents the object name. For example, `Site` and `Contact` are the object names.

xxx represents the outgoing event name. For example, `createContact` and `ModifySite` are the event names.

4. Implement the `tibcoPublishImp()` routine. Code the logic to selectively publish a message. This routine is called by `tibcoPublishyyyy()` internally. In this routine, call `tibcoRecord()`, `tibcoRecordWithFilter()`, and `tibcoSetReferenceRelation()` to write your Clarify records into TIBCO Staging tables.
5. Write the following line of code into `Form_Save3()`.

```
isTriggered = true
```

Use the above code to fix a Clarify bug. `Clarify Me.DoDefault` always returns 0, whether the transaction in `Me.DoDefault` succeeds or fails. Do not add rows into TIBCO Staging tables if a transaction is rolled back (fails).

By setting `isTriggered` to `true` in `Form_Save3()`, you enable the callback to help determine if a transaction is committed, because only committed transactions will launch this callback routine.



The above approach works in most forms. However, some Clarify baseline forms (example, 672) use `Clfy_Form_Save1()` internally. In such cases, the above approach will not work. Instead, you must call `tibcoPublishyyyy(BulkSave, 'xxx')` in `Form_Save1()`, where, *yyyy* represents the object name, and *xxx* represents the outgoing event name.

Enable the Publication Service for ClarifyCRM Web Client

This section describes the procedure to enable the Publication Service for the ClarifyCRM Web client. This procedure assumes you are familiar with Java Server Pages and BEA Weblogic. For more information, refer to *Clarify Web Application Implementation Guide*.

For details on enabling the JSP screens to publish from a web portal, see [Enable the Publication Service for ClarifyCRM Web Client](#) on page 60.

How the Publication Service Works for ClarifyCRM Web Client

In the case of web forms, the Publication Service uses JSP code to selectively write business object values to TIBCO Staging tables, which the Publication Service polls periodically. For details, see [TIBCO Staging Tables](#) on page 255.



The adapter supports the following web applications with Oracle database:

- ClarifyCRM eOrder
- ClarifyCRM eSupport

Global Java Class

The Java utility methods used for selectively writing CBO values to TIBCO Staging tables are in `TibcoCboController.class`. You can place this class into the Web server classpath to access this object inside your JSP forms.

In all cases, only six utility methods are required to write rows into TIBCO Staging tables, as listed in [Table 19](#).

Table 19 Java Methods

Method Name	Method Description
<code>String generateCurrentMessageNum (Base bobase)</code>	Generates and returns a new unique message number to be inserted in the TIBCO Staging tables.
<code>void tibcoSetReferenceRelation (Base bobase, String fromRefName, String toRefName, String from2toRel, String curMsgNum)</code>	Sets the relation between two Clarify records. <code>fromRelName</code> and <code>toRelName</code> are object reference names rather than Clarify object names. <code>from2toRel</code> is a valid Clarify relation name.
<code>void tibcoSetTraversalRelation (Base bobase, String fromRefName, String toRefName, String to2FromRel, String curMsgNum)</code>	Sets the traversal relation between two Clarify records. The <code>fromRelName</code> and <code>toRelName</code> are object reference names rather than Clarify object names. The <code>from2toRel</code> is a valid Clarify relation name.

Table 19 Java Methods (Cont'd)

Method Name	Method Description
<code>void tibcoRecordWithSubject (Base bobase, String operation, String refObjectName,String subjectName, String curMsgNum)</code>	Inserts all fields of a Clarify record into TIBCO Staging tables. The function will insert a row into <code>table_tibco_message_queue</code> and multiple rows into <code>table_tibco_message_fields</code> . The number of rows inserted into <code>table_tibco_message_fields</code> are the number of fields that a CBO has.
<code>void tibcoRecordWithFilterAndSubject (Base bobase, String operation, String refObjectName, String[] filter, String subjectName, String curMsgNum)</code>	Inserts selected fields of a Clarify record into TIBCO Staging tables. The function will insert a row into <code>table_tibco_message_queue</code> and multiple rows into <code>table_tibco_message_fields</code> .The number of rows inserted into <code>table_tibco_message_fields</code> are the number of fields given by the filter list.
<code>void tibcoCommit()</code>	Reflects the changes made by TIBCO methods to the TIBCO Staging tables in a single batch transaction.

For the adapter to publish a message out of the ClarifyCRM system, write a customized JSP code in a ClarifyCRM web form. Each JSP code is tailored for a particular form because the JSP code refers CBOs for that particular form.

Follow these steps to write your customized JSP code. It is assumed you have good CBO programming knowledge.

1. Identify the JSP and the operations that you wish to publish.
2. Save a copy of the form.
3. Write your JSP code to customize the operations.
4. Replace the original JSP with the customized JSP.

In the following example, an excerpt from `reg_do_action_1.jsp` is used to illustrate how to write your JSP code to customize the Case creation behavior. In this example, assume you want to publish a Case, its related address, site and contact information whenever the Case is created.

```
<%!  
else if(strSubmit1.equals("/*XLAT*/Submit To CSR"))  
  
{  
strAction="SubmitToCSR";
```



```

MyCaseId = RegisterDoAction.doCreateCase(ClfyForm, ClfySession,
session, request, cboWebSupStr,errorMsg);

String strRedirectTo;

if ( MyCaseId.equals(""))

strRedirectTo="../CommonJSP/ErrorPage.jsp";

else

{

//CUSTOMIZED CODE BEGINS HERE..

try

{

com.clarify.cbo.Case boCase = null;

boCase = (com.clarify.cbo.Case)
ClfyForm.createBO("com.clarify.cbo.Case");

boCase.addNew();

boCase.setFilter("id_number='"+MyCaseId+"'");

boCase.setDataFields("*");

boCase.setBulkName("bulk");

boCase.getBulk().query();

TibcoCboController tc= new TibcoCboController();

String currMsgNum = tc.generateCurrentMessageNum(boCase);

tc.tibcoRecordWithSubject(boCase,"insert","R1","TEST.CASE",
currMsgNum);

tc.tibcoRecordWithSubject(boCase.getContact(),"insert","R2","TEST.
CASE", currMsgNum);

tc.tibcoSetReferenceRelation(boCase,"R1", "R2",
"case_reporter2contact", currMsgNum);

tc.tibcoRecordWithSubject(boCase.getSite(),"insert","R3","TEST.CAS
E", currMsgNum);

tc.tibcoSetReferenceRelation(boCase,"R1", "R3",
"case_reporter2site", currMsgNum);

```

```

tc.tibcoRecordWithSubject(boCase.getAddress(),"insert","R4","TEST.
CASE", currMsgNum);

tc.tibcoSetReferenceRelation(boCase,"R1", "R4", "case2address",
currMsgNum);

tc.tibcoCommit();
}

catch(Exception tibException)
{
System.out.println("Tibco Error: " + tibException);
errorMsg.setDefaultStatusTitle("TIBCO_TEST.CASE_ERR");
errorMsg.setSpecialStatusMessage(tibException.toString());
}

//CUSTOMIZED CODE ENDS HERE
}

strRedirectTo="reg_thanksb2b.jsp?CaseNum=" +
cboWebSupStr.URLEncode(session,MyCaseId);

response.sendRedirect(strRedirectTo);
}

```

JSP Coding Conventions

This section describes the JSP coding conventions for writing objects into TIBCO Staging tables. Follow these steps for each JSP with objects to be published.

1. Import the `TibcoCboController.class` in the beginning of each form.

```
<%@ page import="com.clarify.cbo.*,
com.tibco.cbo.TibcoCboController, com.clarify.common.*"
errorPage="/CommonJSP/ErrorMessage.jsp" %>
```
2. Identify the methods inside the JSPs, where the Clarify objects are updated or inserted using `updateAll()` or `update()` methods of the business objects.

For example: `Do_CreateCase(----`) is a method inside `reg_do_action_1.jsp`, where a new case is created and is updated in the ClarifyCRM system.
3. Identify the business object, which is being updated by the JSP code and published. In this example, it is the `boCase` business object.

4. Add the TIBCO specific code after the object is updated in the database, to publish the same object.

- Create an object of `TibcoCboController` class:
`TibcoCboController tc= new TibcoCboController();`
- Get the next message number by passing the identified business object in step 3 to the following TIBCO method:
`String currMsgNum = tc.generateCurrentMessageNum(boCase);`

5. Call the other TIBCO methods to publish the contents of the identified business object and its related business objects:

```
tc.tibcoRecordWithSubject(boCase,"insert","R1","TEST.CASE",
currMsgNum);
tc.tibcoRecordWithSubject(boCase.getContact(),"insert","R2","TEST.
CASE", currMsgNum);
tc.tibcoSetReferenceRelation(boCase,"R1", "R2",
"case_reporter2contact", currMsgNum);
tc.tibcoRecordWithSubject(boCase.getSite(),"insert","R3","TEST.CAS
E", currMsgNum);
tc.tibcoSetReferenceRelation(boCase,"R1", "R3",
"case_reporter2site", currMsgNum);
tc.tibcoRecordWithSubject(boCase.getAddress(),"insert","R4","TEST.
CASE", currMsgNum);
tc.tibcoSetReferenceRelation(boCase,"R1", "R4", "case2address",
currMsgNum);
```

where:

`boCase` is the identified business object in step 3.

`R1`, `R2`, `R3`, `R4` are the object reference names for the Clarify object being passed.

`insert` is the name of the operation/event.

`TEST.CASE` is the publishing subject name.

`currMsgNum` is the current message number generated in step 4.

`boCase.getContact()`, `boCase.getSite()`, `boCase.getAddress()` will return the related business objects associated with the `boCase` business object.

`case_reporter2contact`, `case_reporter2site` and `case2address` are the Clarify object relation names.

6. Write the following line of code to save the records in a single transaction:

```
tc.tibcoCommit();
```

7. Handle the exceptions using `try` or `catch` block and provide your own implementation to handle the errors. In this case we are adding the error description into the `ErrorMsg` Object of the `reg_do_action_1.jsp`:

```
errorMsg.setDefaultStatusTitle("TIBCO_TEST.CASE_ERR");
```

```
errorMsg.setSpecialStatusMessage(tibException.toString());
```

Enable the Publication Service for ClarifyCRM Applications

The procedure described in this section assumes you are familiar with the Clarify Data Change Event Editor tool. For more information on the Data Change Event Editor tool, refer to the Amdocs Customization Center guide.

How the Publication Service Works for ClarifyCRM Applications

The Data Change Event Editor monitors specific events occurring in the ClarifyCRM system and publishes them to an external system. Any change made to an object using the ClarifyCRM Applications, generates an event, which is populated in the TIBCO Staging tables. The Publication Service polls the Staging tables periodically and updates recent changes in the TIBCO Publishing table.

In order to use data change events, you need to access some of the JAR files packaged with the ClarifyCRM Applications. As a result, you first need to install the ClarifyCRM Applications.

You can select the object on which the Publication Service should be enabled using the Data Change Event Editor.

Using the Data Change Event Editor

The Data Change Event Editor is used to define events and publishers on specific objects.

Task A Define an Event on an Object

1. Double-click Customization Center 13.1. The Amdocs Customization Center window opens.
2. Choose **Tools > Data Change Event Editor**. The Data Change Event Editor opens.

- 3. From the left pane, right-click an object of your choice and choose **Add** from the shortcut menu. The **Event Configuration** window appears on the right pane.

Figure 7 Event Configuration window

Event Details
Name *
TestTibcoEvent
Description

☐ Active
Focus Object
Site
Owner
User
Timebomb Title

Event Option
☒ Log To Event Queue
☐ Generate Timebomb

Publishers For This Event

Please select the publishers for this event:
Available Publishers
TibcoPublisher Add Remove

Triggers *

Trigger Name	Object Type	Relation path to Focus Object	Operation	Update List	Field to Log
--------------	-------------	-------------------------------	-----------	-------------	--------------

Add Edit Delete

* indicates required fields

Save Cancel

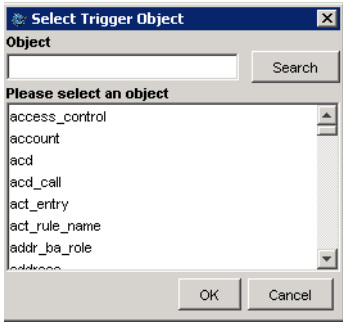
- 4. Enter a name for the event in the **Name** field.
- 5. Enter the name of the **Focus Object**.

The object associated with the change is referred to as the focus object. Other objects related to the focus object are known as trigger objects. An event definition should have a relationship defined between the focus and trigger objects. For example, you can define an event where case is the focus object and subcase is the trigger object. Any change in subcase will trigger an event to be logged in the case object. For more information, refer to the *Clarify Integration Gateway Implementation Guide*.

Task B Associate a Trigger with the Event

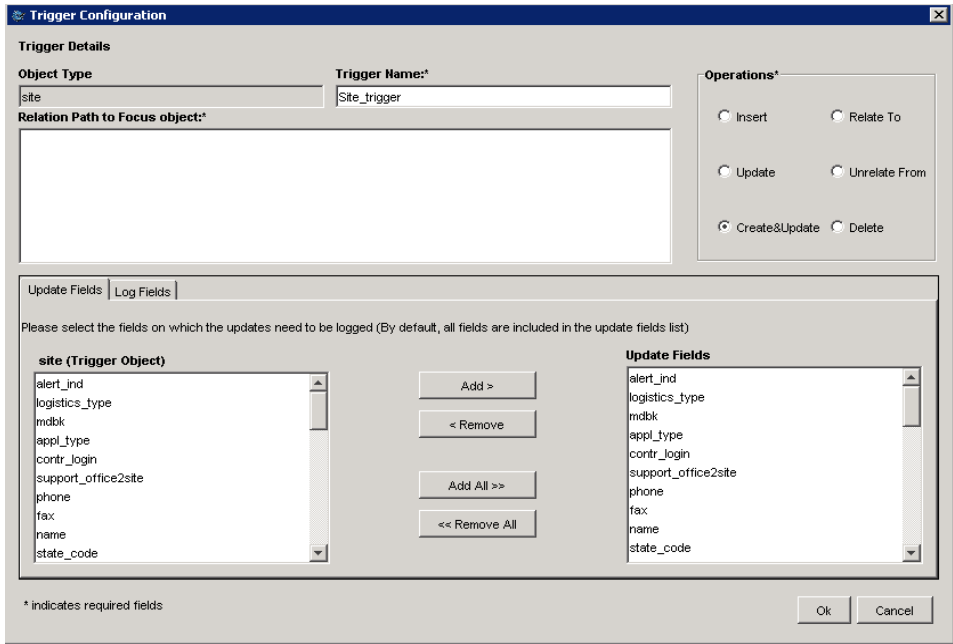
- 1. In the Event Configuration window, under the Triggers group, click **Add**. The **Select Trigger Object** dialog opens.

Figure 8 Select Trigger Object



- 2. In the **Object** field, enter the name of the object, on changing which the event should be logged in the Focus Object. Click **Search** to look for the object you want.
- 3. Select the object of your choice and click **Ok**. The **Trigger Configuration** dialog appears.

Figure 9 Trigger Configuration



4. Enter the name of the trigger object in the **Trigger Name** field.
5. In the **Relation Path to Focus Object**, specify the relation between the trigger object and focus object.

In case the trigger object is the same as the focus object, the relation path can be left empty, otherwise, you must specify the relationship between the two objects. For example, to define a relationship between `acd` and `case`, you can use `'acd2case'` in the relation path. For more information on defining the relation path, refer to *Clarify Integration Gateway Implementation Guide*.

6. From the **Operations** group, select the operation on which the trigger should generate an event.
7. In the **Update Fields** tab, observe that all the fields of the **Trigger Object** list are displayed in the **Update Fields** list.
By default, all changes in the trigger object will be logged in the focus object.
8. From the Update Fields list, select the fields that need not be logged and click **Remove**.
9. Click **Log Fields**. From this tab, you can select fields to be logged from both focus and trigger object.
10. From the **Available Object** list, select **Focus Object**. You can also select trigger or other objects based on the fields you want to log into the Staging table.
11. Select the fields that should be logged into the Staging table and click **Add**. The fields are now displayed in the **Fields to Log** list.

Figure 10 Add the fields to be logged

Trigger Configuration

Trigger Details

Object Type: site Trigger Name: Site_trigger

Relation Path to Focus object:

Operations*

☐ Insert ☐ Relate To

☐ Update ☐ Unrelate From

☒ Create&Update ☐ Delete

Update Fields Log Fields

Please select the object whose fields need to be logged

Available Objects

Focus Object: site (Focus Object)

dev
update_stamp
ship_via
big_evt_gen_sts
child_site2site
support_office2site
cust_primaddr2address
cust_billaddr2address

Add >>

<< Remove

Fields to Log

site_id
name
type
update_stamp
cust_primaddr2address
cust_billaddr2address

* Indicates required fields

Ok Cancel

12. Click **Ok** to close the Trigger Configuration window. In the Event Configuration window, the trigger information appears in the Triggers group.
13. Select **Active**. In the left pane, you will notice that active events appear in boldface.
14. Save the changes.



The event logging tables in Amdocs 6.0 (ClarifyCRM 13.1) has been changed to bg_action instead of da_event_queue and da_event_qstatus.

Task C Define a Publisher

1. From the left pane, right-click **Publisher** and choose **Add** from the shortcut menu. The **Publisher Configuration** window appears on the right pane.
2. Enter the **Publisher Name** and specify the **Initialization Class** to be `com.tibco.cbo.CustomResource`.

Figure 11 Define a Publisher

Publisher Details

Publisher Name *

TestTibcoPublisher

Description

Initialization Class (leave blank to use default initialization class)

com.tibco.cbo.CustomResource

Initialization Parameters

Connection Retry Count

3

Connection Retry Interval (sec)

600

Poll Interval (sec)

1800

☐ **Active**

Owner

User

Log Level

Automatic logging of all errors and significant events

Event Error Action

☒ Log error, skip this row and continue

☐ Abort only this publisher instance after 3 consecutive errors but continue other publishers

☐ Abort only this publisher instance but continue other publishers

☐ Exit from the application on any error

Publisher Events

Event	Custom Handler Class	Custom Handler Parameters	Acknowledgement	Max Resubmit Count
-------	----------------------	---------------------------	-----------------	--------------------

Edit

* indicates a required field

Save

Cancel

3. Save the changes.

Task D Associate the Publisher with the Event

1. From the left pane, click the event you just created.
2. From the **Event Configuration** window, select the **Available Publishers** list. The Publisher you defined will be visible in the list.
3. Select the Publisher you defined in Task C and click **Add**. You will notice that the Publisher is now included in the Publishers for this Event group.

Figure 12 Associate the Publisher with the event

Event Details

Name *

TestTibcoEvent

☐ Active

Description

Event Option

☒ Log To Event Queue

☐ Generate Timebomb

Focus Object

site

Owner

User

Timebomb Title

Publishers For This Event

Please select the publishers for this event:

Available Publishers

TestTibcoPublisher

Add

Remove

TestTibcoPublisher

TibcoPublisher

TestPublisher

TestServerPublisher

GenericPublisher

AcmePublisher

Triggers *

Trigger Name	Object Type	Relation path to Focus Object	Operation	Field to Log

Task E Configure the Event for the Publisher

- 1. Select the Publisher associated with the event.
- 2. In the **Publisher Configuration** window, verify whether the event you defined earlier is displayed in the **Publisher Events** group.

3. Select the event and click **Edit**. The **Publisher Events Configuration** dialog opens.

Figure 13 *Publisher Events Configuration*

4. In the **Custom Handler Class** field, enter `com.tibco.cbo.CustomPublisher`.
5. Select **Active**.
6. Click **Ok** to save changes.

Setting Up the Event Processor

The Event Processor is a command-line Java Program that is used to process and publish logged events to resources as defined in the `ClarifyEnv.xml` file. The Event Processor Java class is packaged in the `ClfyIGW.jar` file, which is located in the `<clfy_app_root>/WEB-INF/lib` directory.

```
java -classpath <classpath>
com.clarify.daevtpub.ClarifyEventProcessor <username> <password>
-pl <publisher list sperated by a space> [-DCThreads <number of
threads>] [-ShowStatus] [-NologStats] [-OutFile <filename>]
[-Debug]
```

For convenience, a batch file with the above command is packaged with the adapter in the setup directory (`event-proc.bat`). Run the `event-proc.bat` file from the installation directory to launch a service to poll for changes in the Publisher table and copy them to the Staging table.

Limitations of the Publication Service

The following are the limitations of the Publication Service:

- The `obj_ref_name` field in the TIBCO Staging table cannot contain a period (.) because the message transports (TIBCO Rendezvous and JMS) have reserved this character.
- The Publication Service does not publish messages in the format received by the Subscription Service. Some minor changes like adding the wrapper class name need to be made to the message published by the Publication Service for it to be received by the Subscription Service.
- If the number of threads configured is more than 1, the messages may not be published in a sequence.

Chapter 4

Subscription Service Functionality

This chapter describes how the Subscription Service receives messages from the TIBCO environment and updates the ClarifyCRM system.

Topics

- *Subscription Service Overview, page 78*
- *How the Subscription Service Works, page 80*
- *Service Error Handling, page 85*

Subscription Service Overview

The Subscription Service uses CBOs and wrappers to update or create business objects in the ClarifyCRM system. Wrappers are provided with the adapter installation for most of the CBOs. For the entire list of the wrappers provided, see Table 22 on page 80.

Subscription Service Features

The Subscription Service provides the following features:

- Supports Insert and Update operations on Existing, Customized and Extended CBOs.
- Supports Delayed Acknowledgement for the RVCN quality of service.
- The Subscription Service is multi-threaded. The number of threads can be configured.
- If the Subscription Service fails to post a message to the ClarifyCRM system, the adapter publishes this message with a distinct error subject name.
- The Subscription Service can be used to invoke a Custom JavaBean method.

The Subscription Service can process any hierarchical message, including sequences and sub-classes. The incoming hierarchical message to the Subscription Service can be fully accessed by the method defined in the JavaBean as a `java.util.Hashtable` object. The original structure of the incoming data is retained in the `java.util.Hashtable` object.

Subscription Service Functionality

Subscription Service allows for the insert or update of Clarify objects. The Subscription Service on receiving the message from the TIBCO environment parses it and updates the ClarifyCRM system using CBOs.

CBOs are ActiveX controls and Java Bean objects, which provide an API to perform create and update operations on the ClarifyCRM system. The user has to write wrapper classes to perform these operations on business objects. For more information on Wrappers, see Wrapper Classes on page 311.

All incoming messages must be an instance of `incomingEvent`. The `incomingEvent` class should have the following attributes:

Table 20 Attributes of the `incomingEvent` Class

Field Name	Type	Description
<code>optype</code>	string	Name of the operation, either <code>create</code> or <code>update</code> .
<code>wrapper_class</code>	string	Name of the wrapper class.
<code>lookup</code>	sequence[condition]	Used for update operation only. Helps in lookup of CBOs.
<code>schema</code>	class	The CBO schema object which is to be created or updated.

The Custom JavaBean functionality of the Subscription Service also uses the above message structure. The relevant fields for the Custom JavaBean functionality are given next:

Table 21 Relevant Fields for the Custom JavaBean Functionality

Field Name	Type	Description
<code>optype</code>	string	Name of the JavaBean or Java class method to be invoked.
<code>wrapper_class</code>	string	<p>Name of the Java Class or JavaBean. The custom JavaBean can be given any name. But the name stored in the <code>customwrapper</code> field in the message sent to the adapter should be the name of the JavaBean prefixed with <code>TIBADCLYCUST_</code>.</p> <p>For example, if <code>LoadCaseBean</code> is the name of the java bean to be invoked, the <code>customwrapper</code> field in the message should be <code>TIBADCLYCUST_LoadCaseBean</code>.</p>
<code>schema</code>	class	Any class definition. For example, the class defined can include subclasses (with any level of hierarchy) and sequences. The schema need not be fetched from the database. It can be a custom configuration.

How the Subscription Service Works

On receiving a message from the TIBCO environment, the Subscription Service executes a set of steps as per the following rules.

CBO Wrapper Invocation

If the `wrapper_class` field in the incoming data does not start with `TIBADCLYCUST_` , it is a CBO Wrapper invocation and the adapter executes the following steps:

- 1. Checks whether all the mandatory fields are present, like operation name, wrapper class name and the actual Business Object schema.
- 2. Uses the wrapper class specified in the incoming message to perform the operation. In case of any error, the Subscription Service uses the error Publisher to send the same incoming message with an added error attribute. See Custom JavaBean Invocation on page 81 for details.
- 3. If there are no errors and the incoming message contains a reply subject, it sends the same message back with an error attribute set to 0.
- 4. Commits all the changes to the database in one transaction.

Table 22 contains the CBOs for which wrapper classes have been provided with the adapter installation. For details, see Wrapper Classes on page 311.

Table 22 CBOs Provided with Wrapper Classes

Main CBO	Contained CBO
ActEntry	Contact
Address	Not Required
BusOrg	Create Business Object is Not Supported
Case	Contact
Communication	edr_com_role
Contact	Site
CreditCard	Not Required
DemandDetail	Create Business Object is Not Supported

Table 22 CBOs Provided with Wrapper Classes (Cont'd)

Main CBO	Contained CBO
DemandHeader	Contact
Dialogue	Create Business Object is Not Supported
DocInst	Doc_Path
Generic	Not Required
NotesLog	Not Required
OEQuote	Not Required
Part	Part_num
PartPrice	Not Required
ServicePart	Contact
ShopList	Not Required
Site	Address
SubCase	Case
User	Employee, Site, PrivClassCache

Custom JavaBean Invocation

If the `wrapper_class` field in the incoming data starts with `TIBADCLYCUST_`, it is considered as a Custom JavaBean Invocation.



While using the Custom JavaBean Invocation with Subscription Service, the adapter returns a `HashTable`.

The incoming hierarchical message is passed on to the invoked method as a `java.util.Hashtable` object. The hashtable maps keys to values as defined in the incoming message to the adapter. You can iterate through the hashtable to access the various fields of the incoming data. The invoked method can be either boolean or Base type. The invoked method should return `true` for success and `false` in case of failure for boolean and the hashtable of values set in custom code in case of Base type.

In case of any error, the Subscription Service uses the error Publisher to send the same incoming message with an added error attribute. If there are no errors and the incoming message contains a reply subject, it sends the same message back with an error attribute set to 0.

Ensure that the JavaBean method or the Java class method to be invoked by the adapter has `java.util.Hashtable` and `ClarifyAppServer` objects as its parameters. `ClarifyAppServer` class contains all the common utility functions used in the adapter and is available with the package `com.tibco.adapter.clarify`.

Through the `ClarifyAppServer` object you can access and use the CBO Session created by the adapter. The message sent to the adapter is processed and sent to the invoked method in the form of a hashtable. This hashtable retains the hierarchy structure of the incoming message.

Take an example of a simple Java class `InvokeCase` that has to be invoked through the Subscriber. Before starting the adapter, this class should be available in the Classpath. This can be done by appending the location of the class in the `tibco.class.path.extended` variable in the adapter `tra` file (`adclCRM.tra`).

```
import java.util.*;
import com.tibco.adapter.clarify.ClarifyAppServer;
/*****
* <BOCH>
*   Description : This is a Custom Java Class example
*   Derivation  :
*   Virtual members which are redefined: none
*   References   : ClarifyAppServer. This class is available with
*   the adclCRM.jar that comes with the adapter installation.
* <EOCH>
*****/

public class InvokeCase{

/*****
* <BOPH>
* Function      : invoke(Hashtable htIncomingData,ClarifyAppServer
mClAppServer)
* Abstract      : method which will be invoked by the adapter
* Parameters    : ClarifyAppServer - reference to ClarifyAppServer
Object
*               Hashtable - Incoming message sent to the adapter
can be accessed
* through this object
* Return Value  : none
* Known Bugs and/or
* side effects  : none
* Other         : none
* <EOPH>
*****/
}
```

```

        public boolean invoke(Hashtable htIncomingData, ClarifyAppServer
mClAppServer){

        //Getting a handle to FormContext object created by the adapter
        System.out.println("Formcontext object created by the adapter :"+
                                + mClAppServer.getFormContext());
        System.out.println("Iterating through the various keys in the
                                +Hashtable");

        Enumeration enum = htParsedClasses.keys();
        while(enum.hasMoreElements()){
            System.out.println(enum.nextElement());

        }
        return true;
    } //end of method invoke
} //end of class

import java.util.*;
import com.tibco.adapter.clarify.ClarifyAppServer;
/*****
* <BOCH>
* Description : This is a Custom Java Class example
* Derivation :
* Virtual members which are redefined: none
* References : ClarifyAppServer. This class is available with
* the adclyCRM.jar that comes with the adapter installation.
* <EOCH>
*****/
public class PartNumberCreation extends Part implements
AEClarify_AeErrors_en_US,
AEClarify_Constants_en_US {

/*****
* <BOPH>
* Function : CreatePart (Hashtable htIncomingData, ClarifyAppServer
mClAppServer)
* Abstract : method which will be invoked by the adapter
* Parameters : ClarifyAppServer - reference to ClarifyAppServer
Object
* Hashtable - Incoming message sent to the adapter
can be accessed
* through this object
* Return Value : none
* Known Bugs and/or
* side effects : none
* Other : none
* <EOPH>
*****/
public Base CreatePart (Hashtable htIncomingData, ClarifyAppServer
mClAppServer) throws Exception {
    com.clarify.cbo.Generic boModLevel = null;
    //Getting a handle to FormContext object created by the adapter
    System.out.println("Formcontext object created by the adapter :"+
        + mClAppServer.getFormContext());
    System.out.println("Iterating through the various keys in the
+Hashtable");
    Enumeration enum = htParsedClasses.keys();

```

```

while(enum.hasMoreElements()){
System.out.println(enum.nextElement());
}

boModLevel.getFields().getItem("description").setValue("test345");
    return boModLevel;
} //end of method CreatePart
} //end of class

```

The message sent to the adapter to invoke the method `invoke` in the above class looks like the following:

```

-<MessageDataRoot>
- otype           = invoke
- wrapper_class  = "TIBADCLYCUST_InvokeCase"
- schema         = <this can point to any class definition>

```

Service Error Handling

The adapter supports Subscription Service error handling to handle errors that may occur during subscription. There are two options available:

- **Reply subject** — If the Subscription Service encounters an error while processing an incoming message, the message is republished along with the error using the reply subject in the subscribed message.
- **Error Publisher** — If the Subscription Service encounters an error while processing an incoming message that does not contain a reply subject, the message is republished along with the error using the Error handler Publication Service endpoint configured in the project. This error publisher supports both TIBCO Rendezvous and JMS transports.

Chapter 5 Request-Response Service Functionality

This chapter describes how the Request-Response Service performs QUERY, WORKFLOW, and CUSTOMIZED operations using CBOs.

Topics

- *Request-Response Service Overview, page 88*
- *How the Request-Response Service Works, page 89*
- *Query Operations, page 89*
- *Workflow Operations, page 91*
- *Customized Operations, page 95*
- *Limitations of Workflow Operation, page 94*
- *Custom JavaBean Invocation, page 96*

Request-Response Service Overview

The Request-Response Service exposes ClarifyCRM specific operations (QUERY, WORKFLOW, and CUSTOMIZED) to the TIBCO environment thereby providing flexibility in customizing the ClarifyCRM system.

The service uses CBOs and wrappers to perform these operations. Wrappers are provided with the adapter installation for most of the CBOs. For the entire list of the wrappers provided, see Table 22. For more information on Wrappers, see Wrapper Classes on page 311.

Request-Response Service Features

The Request-Response Service provides the following features:

- Supports INSERT, UPDATE, and QUERY operations on Existing, Customized, and Extended CBOs.
- Supports Delayed Acknowledgement for the RVCN quality of service.
- The Request-Response Service is multi-threaded. The number of threads can be configured.
- If the Request-Response Service fails to post a message to the ClarifyCRM system, the adapter publishes this message with a distinct error subject name.
- The Request-Response Service can be used to invoke a Custom JavaBean method.

The Request-Response Service can process any hierarchical message, including sequences and sub-classes. The incoming hierarchical message to the Request-Response Service can be fully accessed by the method defined in the JavaBean as a `java.util.Hashtable` object. The original structure of the incoming data is retained in the `java.util.Hashtable` object.

Supported Request-Response Operations

The adapter supports three types of remote Request-Response operations:

- Generic QUERY operations for querying Clarify objects
- WORKFLOW operations to invoke workflow methods and objects
- CUSTOMIZED operations to be defined by the user

How the Request-Response Service Works

All requests received by the Request-Response Service must conform to three operational metadata classes:

- QUERY
- WORKFLOW
- CUSTOMIZED

The Request-Response Service uses incoming operation names to determine which operation it should use to serve the request. For example, if the operation name is `query`, the Request-Response Service calls a generic `QUERY` operation. All the operations use CBOs to work with the ClarifyCRM system.

All three types of Request-Response operations raise one string type of exception `RPC_SERVER_EXCEPTION`. The various error messages are included in the exception and returned to the client.

The following sections explain how the Request-Response Service works with each of the operations:

- Query Operations on page 89
- Workflow Operations on page 91
- Customized Operations on page 95

Query Operations

When the Request-Response Service receives an incoming request with the operation name as `QUERY`, it calls a generic query operation to fetch information related to the CBOs and service the incoming request.

The request object contains all the information needed to satisfy the request. The incoming request contains the name of the wrapper class (for a business object), method to be invoked, and the lookup information to identify the CBO. Using this information as input, Request-Response Service performs the required operations.

The reply object is constructed from the output of the `QUERY` operation and sent back to the calling function. It contains the input CBO along with the requested fields.

Example of Query Operation

The QUERY operation comes with a request to find contact information associated with a Case business object. The request object will contain lookup values for the Case business object.

The wrapper class provided for each CBO contains methods that can be used to perform a QUERY operation on that CBO.

The wrapper class invokes the `getCBO()` method of the adapter to get a handle for the Case business object to be queried. It then invokes the specific method `getContact()` to get the contact information associated with the Case.

Given next is a sample schema stored in the adapter configuration for the QUERY operation.

`request_RpcQuery_query`

- Input parameters
 MethodName — Name of the method (`getContact`)
 WrapperClass — Name of the wrapper class provided by the adapter with the package name
 (`com.tibco.adapter.clarify.wrapper.Case_Wrapper`)
 query_object — (object of type any) Schema for the object to be queried. (Make this parameter point to the schema being queried, in this example, it is the Case object).
 Class 'com.tibco.adapter.clarify.wrapper.Case_Wrapper'.
- Output Parameters
 schema of the Case Object.
 'com.tibco.adapter.clarify.wrapper.Case_Wrapper'.

The input parameter contains the key fields, which are used to identify a Case business object and the operation name to be invoked. The reply object contains all the information that was a part of the incoming request, in addition to the contact information.

Table 23 Incoming Message Formats for Query Operations

Field Name	Type	Default	Field Value Description
MethodName	M_STRING	n/a	Name of the method to be invoked (in the wrapper class of a business object) to perform a query operation. Example: <code>getsite</code> , <code>getcontact</code>

Table 23 Incoming Message Formats for Query Operations (Cont'd)

Field Name	Type	Default	Field Value Description
wrapperClass	M_STRING	n/a	Name of the wrapper class. Example: <code>com.tibco.adapter.clarify.wrapper.Case_Wrapper</code>
query_object	any	n/a	Schema of the business object on which the query operation has to be performed. Example: <code>Clarify/schema/case</code>
Lookup	Sequence	n/a	Sequence of look up values.

Table 24 Outgoing Message Formats for Query Operations

Field Name	Type	Default	Field Value Description
returnValue	any	n/a	Contains the <code>query_object</code> along with the result of the query operation.
RPC_SERVER_EXCEPTION	string	n/a	If any exception(s) occur while performing a query operation, the server returns the exception to the calling application.

Workflow Operations

Workflow objects can also be called queue-able objects in a ClarifyCRM System. Workflow objects are the most widely used ones and they involve the most complicated business logic when they are created or changed. The usage of CBOs makes the task easier to handle.

When the Request-Response Service receives a request with its operation name as `workflow`, it calls a workflow operation to service the incoming request.

It parses the incoming message to get the request object. The request object contains all the information needed to satisfy the request. The incoming request contains name of the wrapper class (for a business object), method to be invoked, and the lookup information to identify the CBO. In addition, a workflow operation needs information about the `User Object`, `Queue Object` and `WIPBIN` object. This information forms a part of the incoming request.

The server extracts the workflow operation name from the request and invokes the operation's corresponding function, defined in the wrapper class, to meet the requirements.

The adapter uses CBOs to perform the workflow operation. The result of the operation is then returned to the calling adapter instance using a reply object. The reply object in the case of a workflow operation contains a boolean value indicating the status of the operation (True/False). In case of any error, the Request-Response Service returns an exception to indicate that the operation could not be completed.

The list of workflow operations supported is given next:

- Accept
- Assign
- Dispatch
- Forward
- Move
- Reject
- Yank

The objects on which workflow operations can be performed are listed below:

- Case
- Subcase
- OEQuote
- DemandDetail
- Dialogue

Example of Workflow Operation

Given next is a sample schema stored in the adapter configuration for a workflow operation.

`request_RpcWorkflow_workflow`

- Input Parameters
 - MethodName — Name of the workflow method to be invoked
(Assign, Accept, Dispatch ...)
 - wrapperClass — Name of the wrapper class
(`com.tibco.adapter.clarify.wrapper.Case_Wrapper`)
 - workflow_object — (Object of type any) Map this to the schema of the Case Business Object

lookup — This contains information about User, Queue and WIPBIN Object

User_LoginName — Login Name of the user

Queue_Title — Title of the Queue (of the logged in user)

isTemporary — boolean value used for Accept workflow operation

- Output Parameter

boolean - Status of the operation

Table 25 Incoming Message Formats for Workflow Operations

Field Name	Type	Default	Field Value Description
MethodName	M_STRING	n/a	Name of the method to be invoked (in the wrapper class of a business object) to perform a workflow operation. For example: assign, yank, accept, reject.
wrapperClass	M_STRING	n/a	Name of the wrapper class. For example: com.tibco.adapter.clarify.wrapper.Case_Wrapper
workflow_object	any	n/a	Schema of the business object on which the workflow operation has to be performed. For example: Clarify/schema/case
lookup	Clarify\schema\workflow_lookup	n/a	Holds information about User, Queue and WIPBIN business objects. These objects are required to perform workflow operation.
User_LoginName	M_STRING	n/a	Login name of the user. Only the user who owns the object can perform workflow operation(s) on a business object. If the logged in user is not the owner, then workflow operations cannot be performed on that object. In this scenario, the adapter changes the ownership of the object to the logged in user and performs the operation.
Queue_Title	M_STRING	n/a	Title of the queue that belongs to the logged in user. This is required to perform workflow operations like forward, reject, and accept.

Table 25 Incoming Message Formats for Workflow Operations (Cont'd)

Field Name	Type	Default	Field Value Description
isTemporary	boolean	n/a	Boolean value. This is required to perform Accept workflow operation. If it is set to True, the workflow object will be temporarily accepted into the WIPBIN. After the workflow item is saved, the item automatically returns to the original queue.

Table 26 Outgoing Message Formats for Workflow Operations

Field Name	Type	Default	Field Value Description
returnValue	boolean	n/a	True or False. Gives the status of the workflow operation performed on a workflow object.
RPC_SERVER_EXCEPTION	string	n/a	If any exception(s) occur while performing a workflow operation, the server returns the exception to the calling application.

Limitations of Workflow Operation

In case of a workflow operation, the input parameter contains two fields, Login_Name and Queue_Title. It should be populated with the login name of the user (one which the adapter uses to log in to the ClarifyCRM system) and the title of the queue to which the user belongs. The reason is explained below:

Certain workflow operations (reject, forward, accept) can be performed on a business object only by its owner. If the logged in user is different from the owner, then carrying out workflow operations on that object is not possible. To overcome this, the adapter changes the ownership of the object to that of the logged in user (if the current owner is not the logged in user) and performs the operation.



While populating the wrapperClass field in the input parameter, give the package information along with the wrapper class name.

For example: `com.tibco.adapter.clarify.wrapper.Case_Wrapper`

Customized Operations

Customized operations are invoked to perform customized actions on Customized or Extended CBOs.

A customized action is defined to meet specific requirements. The information about the customized CBO should be configured in the project at design time. All the customized methods for a particular CBO are included in its Wrapper Class. For example, the `Case_Wrapper` class contains all customized methods specific to it.

The name of the operation to be performed along with the input values is taken from the incoming request object. Based on the operation name, a customized method is invoked. The result of the operation is then returned to the calling adapter instance after constructing the reply object.

Example of Customized Operation

Given next is a sample schema stored in the project for a customized operation.

- Input Parameters
 - `MethodName` — Name of the Customized method to be invoked.
 - `wrapperClass` — Name of the wrapper class
`(com.tibco.adapter.clarify.wrapper.Case_Wrapper)`
 - `customized_object` — (Object of type any) Map it to the schema of Case Business Object
- Output Parameter
 - `boolean` — status of the operation

Table 27 Incoming Message Formats for Customized Operations

Field Name	Type	Default	Field Value Description
MethodName	M_STRING	n/a	Name of the method to be invoked (in the wrapper class of a business object) to perform a customized operation.
wrapperClass	M_STRING	n/a	Name of the wrapper class. For example: <code>com.tibco.adapter.clarify.wrapper.Case_Wrapper</code>
customized_object	any	n/a	Schema of the business object on which the customized operation has to be performed. For example: <code>Clarify/schema/case</code>

Table 28 Outgoing Message Formats for Customized Operations

Field Name	Type	Default	Field Value Description
returnValue	boolean	n/a	True or False. Gives the status of the customized operation performed on a customized object.
RPC_SERVER_EXCEPT ION	string	n/a	If any exception(s) occur while performing a customized operation, the server returns the exception to the calling application.

Custom JavaBean Invocation

The customized operation can also be used to invoke a Custom JavaBean method.

If the `wrapperClass` field in the incoming data starts with `TIBADCLYCUST_`, it is considered as a Custom JavaBean Invocation. The `MethodName` field in the incoming data contains the name of the method to be invoked in the Custom JavaBean.

Ensure that the JavaBean method or the Java class method to be invoked by the adapter has `java.util.Hashtable` and `ClarifyAppServer` objects as its parameters.

- `ClarifyAppServer` class contains all the common utility functions used in the adapter and is available with the package `com.tibco.adapter.clarify`.

Through the `ClarifyAppServer` object you can access and use the CBO Session created by the adapter.

- The message sent to the adapter is parsed and sent to the invoked method as a `java.util.Hashtable` object. This hashtable retains the hierarchy structure of the incoming message. And the hashtable maps keys to values as defined in the incoming message.

Within the custom JavaBean, you can iterate through the hashtable to access the various fields of the incoming data. The invoked method should return `true` for success and `false` in case of failure.

Example of Custom JavaBean Invocation

Let us take an example of a simple Java class `InvokeCase` that has to be invoked through the Subscriber. Before starting the adapter, this class should be available in the Classpath. This can be done by appending the location of the class in the `tibco.class.path.extended` variable in the adapter `tra` file(`adclyCRM.tra`).

```
import java.util.*;
import com.tibco.adapter.clarify.ClarifyAppServer;
/*****
* <BOCH>
*   Description : This is a Custom Java Class example
*   Derivation  :
*   Virtual members which are redefined: none
*   References   : ClarifyAppServer. This class is available with
*   the adclyCRM.jar that comes with the adapter installation.
* <EOCH>
*****/

public class InvokeCase{

/*****
* <BOPH>
* Function      : invoke(Hashtable htIncomingData, ClarifyAppServer
mClAppServer)
* Abstract      : method which will be invoked by the adapter
* Parameters    : ClarifyAppServer - reference to ClarifyAppServer
Object
*               Hashtable - Incoming message sent to the adapter
can be accessed
* through this object
* Return Value  : none
* Known Bugs and/or
* side effects  : none
* Other        : none
* <EOPH>

*****/

    public boolean invoke(Hashtable htIncomingData, ClarifyAppServer
mClAppServer){
```

```

//Getting a handle to FormContext object created by the adapter
System.out.println("Formcontext object created by the adapter :"+
                    + mClAppServer.getFormContext());
System.out.println("Iterating through the various keys in the
                    +Hashtable");

Enumeration enum = htParsedClasses.keys();
while(enum.hasMoreElements()){
    System.out.println(enum.nextElement());
}
return true;
} //end of method invoke
} //end of class

```

Chapter 6

Request-Response Invocation Service Functionality

This chapter describes how the adapter's Request-Response Invocation Service can be used in the Flexible and Non-Flexible deployments.

Topics

- *Request-Response Invocation Service Overview, page 100*
- *How the Request-Response Invocation Service Works, page 102*

Request-Response Invocation Service Overview

The Request-Response Invocation Service allows users to request data from external applications through the TIBCO environment.

The adapter supports the following approaches:

- Request-Response Invocation Service Using Flexible Deployment
- Request-Response Invocation Service Using Non-Flexible Deployment

Request-Response Invocation Service Using Flexible Deployment

The flexible deployment of the ClarifyCRM system entails the business functionality being implemented in separate servers and the communication between the client and the server taking place over BEA Tuxedo.

BEA Tuxedo is a transaction middleware for building reliable distributed systems. BEA Tuxedo can be used to deploy and manage traditional systems independent of the underlying communication in a distributed environment. The adapter uses the BEA Tuxedo middleware to invoke a Request-Response Invocation Service.

In the Flexible Deployment approach, the following components are used:

- **ClearBasic Module** — passes the request data from the ClearBasic clients to the BEA Tuxedo middleware.

The global module uses the Service Message object to make a request FML (Field Manipulation Language) buffer and invoke a synchronous call to the Tuxedo service (Adapter Agent). The global module reads the reply FML buffer and passes it back to the calling ClearBasic forms.

- **Adapter Agent** — which runs as a Tuxedo service, receives the request from the ClearBasic clients as a FML buffer.

It parses this request, constructs a TIBCO Rendezvous Message, and sends it as a Rendezvous request to the adapter. On receiving the reply Rendezvous message from the adapter, the Adapter Agent parses it and writes it to the reply FML buffer, which goes back to the respective Clarify client. In case the adapter is not available it returns an operation timed out error to the user in the reply FML buffer.

- **TIBCO Adapter for ClarifyCRM** — receives the request (TIBCO Rendezvous Message or JMS Message) from the Adapter Agent, after converting it to the configured schema sends it out as an AE request to external applications.

On receiving the reply from an external RPC Server, the adapter sends it back to the agent as a reply Rendezvous message.

Request-Response Invocation Service Using Non-Flexible Deployment

In the Non-flexible deployment scenario the Clarify ClearBasic client calls a ClearBasic global module, which in turn makes use of the TIBCO Rendezvous ActiveX component to invoke an RPC Client request through the adapter.

Following components are used in this approach:

- **ClearBasic Module** — passes the request data from the ClearBasic clients to the TIBCO Rendezvous ActiveX component or to the Adapter for ClarifyCRM EMS ActiveX component.

The global module parses the request data, uses the TIBCO Rendezvous ActiveX component to make a request Rendezvous message and sends it to the adapter. The Rendezvous message sent by the ActiveX component consists of sequence of name value pairs of the CBO contained in the form. In addition, it also contains the operation name. The response received also has the same structure as the request sent by the ActiveX component, if all the operations are successful. In case of errors, the error description is passed back to the form as a Rendezvous message. On receiving the reply Rendezvous message, the ClearBasic Module passes it back to the calling ClearBasic forms.

- **TIBCO Adapter for ClarifyCRM** — receives the request Rendezvous Message from the ClearBasic Module, after converting it to the configured schema sends it out as an AE request to external applications.

On receiving the reply from an external RPC Server, the adapter sends it back to the agent as a reply Rendezvous message.

How the Request-Response Invocation Service Works

The Request-Response Invocation Service works with the following forms:

- ClearBasic Forms
- Web Forms

ClearBasic Forms

Using ClearBasic forms, the Request-Response Invocation Service functions in two modes:

- Flexible Deployment
- Non-Flexible Deployment

Flexible Deployment

The operations involved in a Flexible deployment are:

1. The client implements the Request-Response Invocation request.
The sample ClearBasic code provided with the adapter installation can be used as a reference, and the Clarify ClearBasic forms can be modified according to the requirement. See Sample Code used for a Flexible Deployment on page 104 for details.
2. The request is sent to the BEA Tuxedo application, which in turn sends the request to the TIBCO environment.
3. The adapter receives the request, does the necessary processing and sends the response back to the TIBCO environment.
4. The BEA Tuxedo application picks up the response from the TIBCO environment and sends the response to the requesting client.

7. Install TIBCO Rendezvous on the Client machine. Refer to the TIBCO Rendezvous documentation for details.
8. Use the sample code provided with the adapter installation to configure the form to perform the Request-Response Invocation functionality. See below for details.

Sample Code used for a Flexible Deployment

A sample ClearBasic code for the implementation of Request-Response Invocation functionality is provided with the adapter installation. The sample and its location are given next.

TIBCO_HOME/adapter/adclycrm/<version_num>/samples/testRPCClient
Option Explicit

'The Data type of the rvmessage that is sent to the adapter.

```
Type Reply_Message_UDT
object_name As String
operation_name As String
field_name(500) As String
field_value(500) As String
error_description As String
End Type
```

'The following declarations are made for the methods in Global Module

```
Declare Sub tibcoSetMessageFields(rec As Record, ByVal opName As String, ByVal objName As String, _
    ByVal service As String, ByVal network As String, _
    ByVal daemon As String, ByVal timeout As Long, _
    ByVal subject As String, _
    ByRef reply_message As Reply_Message_UDT _
)
```

```
Declare Sub tibcoSetTuxedoFields(rec As Record, ByVal serviceFlag As String, ByVal opName As String, ByVal objName As String, _
    ByVal service As String, ByVal network As String, _
    ByVal daemon As String, ByVal timeout As Long, _
    ByVal subject As String, _
    ByRef reply_message As Reply_Message_UDT _
)
```

'in this Sample form a button has been added to the Form and on click of this button(Flexible) a TIBCO Rendezvous message is sent to adapter

```
Sub Flexible_Click()
Dim count As Integer
Dim siteRec As New Record
Set siteRec = Cobj_Location.Contents
```

```

Dim reply_message As Reply_Message_UDT
    Call tibcoSetTuxedoFields(siteRec, "JMS", "update", "location",
    ""7222",_

"127.0.0.1", "", 5000, "TEST.CLIENT", reply_message)

'if the reply is sent back from the adapter without any errors
proceed further

If (reply_message.error_description="") Then
Dim i As Integer
i=0
While reply_message.field_name(i) <> "objid"
i=i+1
Wend
SITE_ID.Value= cstr(count) + reply_message.field_value(i)
Else
msgbox(reply_message.error_description)
End If

End Sub

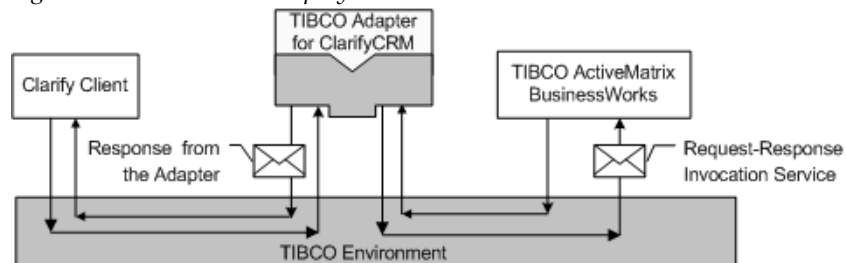
```

Non-Flexible Deployment

The operations involved in a Non-Flexible deployment are:

1. The client implements the Request-Response Invocation request.
The sample ClearBasic code provided with the installation can be used as reference, and the Clarify ClearBasic forms can be modified according to the requirement. See Sample Code used for a Non-Flexible Deployment over TIBCO Rendezvous on page 106 for details.
2. The request is sent to the TIBCO environment.
3. The adapter receives the request, does the necessary processing and sends the response back to the TIBCO environment.
4. The requesting client receives the response from the TIBCO environment.

Figure 15 Non-Flexible deployment mode



Prerequisites for Setting Up Request-Response Invocation Functionality Using Clarify ClearBasic Forms in a Non-Flexible Deployment

1. Install and set up ClarifyCRM LAN client. Refer to the *Clarify Client Installation Guide* for details.
2. Install and set up the adapter.
The files and objects needed for Request-Response Invocation Service implementation for LAN client are installed with the adapter. See Installation on page 19 for details.
3. Import the global modules needed for the Request-Response Invocation functionality using the cbex tool. See Enable the Publication Service for LAN Client on page 54 for details.
4. To support TIBCO Rendezvous, install TIBCO Rendezvous on the Client machine. Refer to the TIBCO Rendezvous documentation for details.
5. To support JMS, install TIBCO EMS on the Client machine. Refer to the TIBCO EMS documentation for details.
6. If you have opted for JMS support, do the following:
Copy the TIBCOEMS.d11 to a folder where you wish to save the COM component.
Execute tib_reg_emscm.bat from the <install_directory>. Use the following syntax to register the COM component:
`tib_reg_emscm.bat <Path for TIBCOEMS.d11>`
7. Use the sample code provided with the adapter installation to configure the form to perform the Request-Response Invocation functionality.
For TIBCO Rendezvous support, see Sample Code used for a Non-Flexible Deployment over TIBCO Rendezvous on page 106 for details. For JMS support, see Sample Code used for a Non-Flexible Deployment over JMS on page 108 for details.

Sample Code used for a Non-Flexible Deployment over TIBCO Rendezvous

A sample ClearBasic code for the implementation of Request-Response Invocation functionality over TIBCO Rendezvous is provided with the adapter installation. The sample code and its location are given next.

`TIBCO_HOME/adapter/adclycrm/<version_num>/samples/testRPCClient`
Option Explicit

'The Data type of the rvmessage that is sent to the adapter.

Type Reply_Message_UDT

```

object_name As String
operation_name As String
field_name(500) As String
field_value(500) As String
error_description As String
End Type

```

'The following declarations are made for the methods in Global Module.

```

Declare Sub tibcoSetMessageFields(rec As Record, ByVal opName As
String, ByVal objName As String, _
    ByVal service As String, ByVal network As String, _
    ByVal daemon As String, ByVal timeout As Long, _
    ByVal subject As String, _
    ByRef reply_message As Reply_Message_UDT _
)

```

```

Declare Sub tibcoSetTuxedoFields(rec As Record, ByVal opName As
String, ByVal objName As String, _
    ByVal service As String, ByVal network As String, _
    ByVal daemon As String, ByVal timeout As Long, _
    ByVal subject As String, _
    ByRef reply_message As Reply_Message_UDT _
)

```

'In this Sample form a button has been added to the Form and on clicking the button (NonFlexible) a TIBCO Rendezvous message is sent to the adapter.

```

Sub NonFlexible_Click()
Dim siteRec As New Record
Set siteRec = Cobj_Location.Contents
Dim reply_message As Reply_Message_UDT

```

'Call the tibcoSetMessageFields with the Record variable, operation name, service,network,daemon,wait time.

```

Call tibcoSetMessageFields(siteRec, "update", "location", "7500", _
    "", "tcp:7500", 100, "TEST.CLIENT", reply_message)
If (reply_message.error_description="") Then
Dim i As Integer
i=0
While reply_message.field_name(i) <> "objid"
i=i+1
Wend
SITE_ID.Value= "" + reply_message.field_value(i)
Else
msgbox(reply_message.error_description)
End If

End Sub

```

Sample Code used for a Non-Flexible Deployment over JMS

A sample ClearBasic code for the implementation of Request-Response Invocation functionality over JMS is provided with the adapter installation. The sample code and its location are given next.

TIBCO_HOME/adapter/adclycrm/<version_num>/samples/testRPCClient
Option Explicit

'The Data type of the JMSMessage that is sent to the adapter.

```
Type Reply_Message_UDT
object_name As String
operation_name As String
field_name(500) As String
field_value(500) As String
error_description As String
End Type
```

'The following declarations are made for the methods in Global Module.

```
Declare Sub tibcoSetMessageFieldsEMS(rec As Record, ByVal opName As String, ByVal objName As String, _
    ByVal service As String, ByVal network As String, _
    ByVal daemon As String, ByVal timeout As Long, _
    ByVal subject As String, _
    ByRef reply_message As Reply_Message_UDT _
)
```

```
Declare Sub tibcoSetTuxedoFields(rec As Record, ByVal opName As String, ByVal objName As String, _
    ByVal service As String, ByVal network As String, _
    ByVal daemon As String, ByVal timeout As Long, _
    ByVal subject As String, _
    ByRef reply_message As Reply_Message_UDT _
)
```

'In this Sample form a button has been added to the Form and on clicking the button(NonFlexible) a JMS message is sent to the adapter.

```
Sub NonFlexibleJMS_Click()
Dim siteRec As New Record
Set siteRec = Cobj_Location.Contents
Dim reply_message As Reply_Message_UDT
```

'call the tibcoSetMessageFieldsEMS with the Record variable, operation name, service, network, daemon, wait time.

```
Call tibcoSetMessageFieldsEMS(siteRec, "update", "location",
"tcp://10.97.97.55:7222", _
    "", "", 5000, "TEST.CLIENT", reply_message)
If (reply_message.error_description="") Then
Dim i As Integer
```

```

i=0
  While reply_message.field_name(i) <> "objid"
    i=i+1
  Wend
  SITE_ID.Value= "SiteID:" + reply_message.field_value(i+1)
Else
  msgboreply_message.error_description)
End If

End Sub

```

Web Forms

Using web forms, the Request-Response Invocation Service functions in the Web Client deployment mode.



The adapter supports the following web applications:

- ClarifyCRM eOrder
- ClarifyCRM eSupport

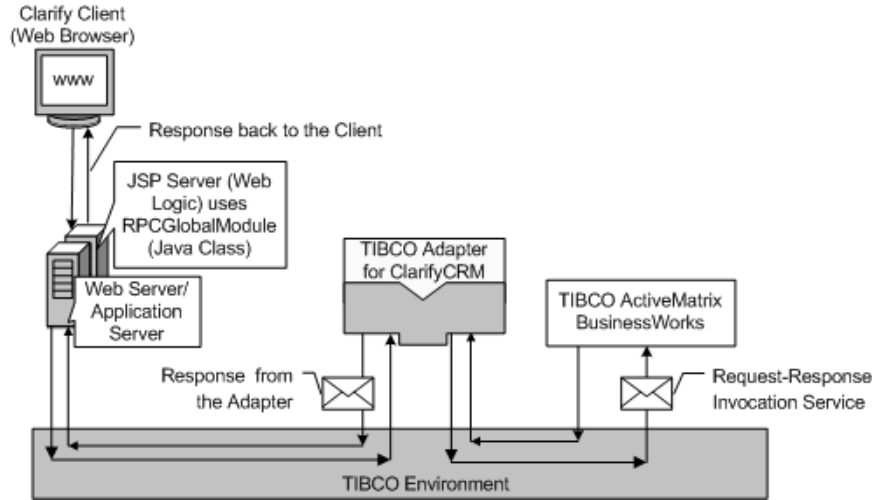
The adapter uses the Java Class Module, `RPCGlobalModule.jar`, to invoke a Request-Response Invocation Service. The Clarify web applications use BEA Weblogic server. The sample implementation (provided with the adapter) uses Clarify objects, JSP pages and the `RPCGlobalmodule` class file.

The operations involved in a Web Client deployment are:

1. The ClarifyCRM client submits the JSP form to the Web server/JSP Server.
2. The web server processes the page and executes the Java code in the requested page. The `RPCGlobalModule` Class method is called in Java code, which sends the request message to the adapter via the TIBCO environment.
3. The adapter receives the request, does the necessary processing and sends the response back to the web server via the TIBCO environment.
4. The requesting client receives the response from the web server.

The sample JSP code provided with the installation can be used as reference, and the JSP forms can be modified according to the requirement. See Sample Code Used for Web Forms on page 112 for details.

Figure 16 Web client deployment mode



Prerequisites for Setting Up Request-Response Invocation Functionality Using Clarify Web Forms

All Clarify web applications use BEA Weblogic server. Carry out the following steps to set up RPC functionality in the Clarify web forms:

1. Install and set up ClarifyCRM Web client. Refer to the *Clarify Client Installation Guide* for details.
2. Install and set up the adapter. The files and objects needed for RPC client implementation for web forms are installed with the adapter. See Installation on page 19 for details.
3. Install and set up eOrder web application, so that you can customize the JSP pages provided with the installation as per the sample code provided with the adapter installation. See Sample Code Used for Web Forms on page 112.
4. Install and set up the BEA Weblogic server. Refer to the *Clarify Client Installation Guide* for details.
5. Install TIBCO Runtime Agent on the machine where BEA Weblogic is installed. TIBCO Runtime Agent supports both RV and JMS libraries. Refer to the TIBCO Rendezvous documentation for details.
6. Add the `RPCGlobalModule.class` and CBOs to BEA Weblogic CLASSPATH environment variable. Refer to the BEA Weblogic documentation for details and information on security.
7. Ensure that the location of the Global module classpath is included in the BEA Weblogic startup script.

- 8. Add location of CBOs and RPCGlobalModule class to the CLASSPATH in your BEA Weblogic startup script.

Programmer Reference to Use the RPC GlobalModule

Class Name	RPCGlobalModule
Function Name	TibcoSetMessageFields()
Declaration	<pre>tibcoSetMessageFields (com.clarify.cbo base, java.lang.String Operation, java.lang.String ObjectName, java.lang.String Service, java.lang.String Network, java.lang.String Daemon, double Timeout, java.lang.String SubjectName, java.util.Vector CurMsgNum) throws TibrvException</pre>
Purpose	Returns the RPC Client response in the vector parameter being passed.

Table 29 Parameter list of function TibcoSetMessageFields()

Parameter	Description
Base	Clarify object.
Operation	Pass the type of operation to be executed.
ObjectName	Object being passed.
Service	RVD service parameter.
Network	RVD network parameter.
Daemon	RVD Daemon parameter.
Timeout	Value should be greater than the Clarify adapter response time for RPC requests.
SubjectName	RVD subject name parameter.
CurMsgNum	Holds the return value of the request response call.

Function Name	TibcoSetMessageFieldsEMS()
Declaration	<pre>TibcoSetMessageFieldsEMS (com.clarify.cbo base,</pre>

```
java.lang.String Operation,  
java.lang.String ObjectName,  
java.lang.String Service,  
java.lang.String Network,  
java.lang.String Daemon,  
double Timeout,  
java.lang.String SubjectName,  
java.util.Vector CurMsgNum)  
throws TibrvException
```

Purpose Returns the RPC Client response in the vector parameter being passed.

Table 30 Parameter list of function TibcoSetMessageFieldsEMS()

Parameter	Description
Base	Clarify object.
Operation	Pass the type of operation to be executed.
ObjectName	Object being passed.
Service	EMS server, port, and address.
Network	N/A
Daemon	N/A
Timeout	Value should be greater than the Clarify adapter response time for RPC requests.
SubjectName	Subject Name.
CurMsgNum	Holds the return value of the request response call.

Sample Code Used for Web Forms

A sample JSP page for the implementation of Request-Response Invocation functionality is provided with the adapter installation. The location of the sample code is given next.

```
TIBCO_HOME/adapter/adclycrm/<version_num>/samples/testRPCClient
```

The RPCGlobalModule interface tibcoSetMessageFields enables the RPC client functionality for the web forms. The sample JSP code demonstrates the usage of the functionality and the changes to be made to the ordersubmit.jsp page that is installed as part of ClarifyCRM web client installation.

```
..... :  
..... :  
boOEQuote.getFields().getItem("total_tax_amt").setValue(boOeQuote2  
.getFields().getItem("total_tax_amt").getValue().toString());
```

```

boOEQuote.getFields().getItem("tot_ship_amt").setValue(boOEQuote2.
getFields().getItem("tot_ship_amt").getValue().toString());
boOEQuote.getFields().getItem("handling_cost").setValue(boOEQuote2
.getFields().getItem("handling_cost").getValue().toString());
boOEQuote.getFields().getItem("total_net").setValue(boOEQuote2.get
Fields().getItem("total_net").getValue().toString());
boOEQuote.submit();
boOEQuote.update();
//CUSTOMIZED CODE BEGINS HERE....
try {
Vector currMsgNum = new Vector();
RPCGlobalModule tc= new RPCGlobalModule();

tc.tibcoSetMessageFields(boOEQuote,"select","R1","","",100,"TES
T.CLIENT", currMsgNum);
}
catch(Exception tibException) {
System.out.println(tibException);
errorMsg.setDefaultStatusTitle("TIBCO_ORDERSUBMIT_ERR");

errorMsg.setSpecialStatusMessage(ClfyForm.getError().getDescriptio
n() + "\r\n" + tibException.toString() );
response.sendRedirect("../CommonJSP/ErrorMessage.jsp");
}
//CUSTOMIZED CODE ENDS HERE

eOrderCommon.SendRoutingRequest(boOEQuote);
pageInit.setAttributeBo("shoppingcart", boOEQuote2 , ClfyForm,
session);

```


Chapter 7 **Getting Started**

This chapter describes the basic steps required to configure an adapter service and run the adapter.

Topics

- *Overview, page 116*
- *Prerequisites, page 117*
- *Configure the Adapter, page 118*
- *Deploy the Adapter, page 129*
- *Start the Adapter, page 130*
- *Stop the Adapter, page 131*

Overview

This chapter attempts to give you a hands-on experience on the various steps involved in configuring and deploying the adapter. This involves the following steps:

1. Configuring an adapter instance
2. Deploying the adapter
3. Running the adapter against the configured adapter instance

You can skip step 1 if you are comfortable with configuring the adapter instance. Pre-configured adapter instances are provided with the adapter installation in the following location.

- TIBCO Rendezvous:
`<TIBCO_HOME>\adapter\adclycrm\<version_num>\samples\im_sheets\adclycrm_publisher_examples_RV.dat`
- TIBCO Enterprise Message Service:
`<TIBCO_HOME>\adapter\adclycrm\<version_num>\samples\im_sheets\adclycrm_publisher_examples_JMS.dat`

Since the adapter and ClarifyCRM are closely integrated, there are some prerequisites to be met on the ClarifyCRM side before you can start TIBCO Designer to configure even minimum settings for the adapter. See Prerequisites on page 117 for details.

The example in this chapter demonstrates the adapter's Publication Service functionality. You should be familiar with TIBCO Designer. TIBCO Designer documentation can be accessed by clicking the **Help** button on the TIBCO Designer startup panel or selecting the **Help>Designer Help** menu option from within the Designer GUI.

Prerequisites

- Ensure that the design-time adapter is running before you start configuration.

On Windows, the design-time adapter is installed and started as a service automatically during installation of the adapter. If the design-time adapter service startup state is set to Manual, you must start the service manually before starting configuration. To start the design-time adapter manually, go to command prompt and change directory to:

`<TIBCO_HOME>\adapter\adclycrm\<version_num>\bin` and type command `adclycrmdta_RV`.

On UNIX, you must start the design-time adapter manually.

- Prepare ClarifyCRM components for publishing. See Prepare ClarifyCRM Components for the Publication Service on page 53 for details.

Install Software

- Install all required software (see Installation on page 19).
- Install the adapter software (see Installation on page 19).

Information You Need

Before starting to configure the adapter, you should have the following information ready:

- Name of the computer hosting the Clarify database server
- User ID and password for logging into the ClarifyCRM system
- Name of the Clarify Database server
- Verification that the Clarify Database Server is up and running

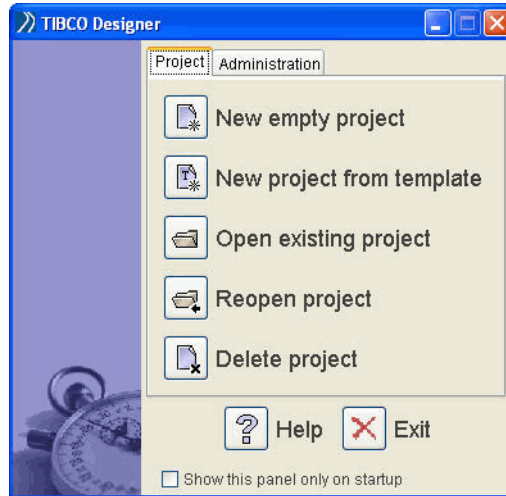
Configure the Adapter

Follow the steps given below to configure the adapter.

Task A Configure an Adapter Instance

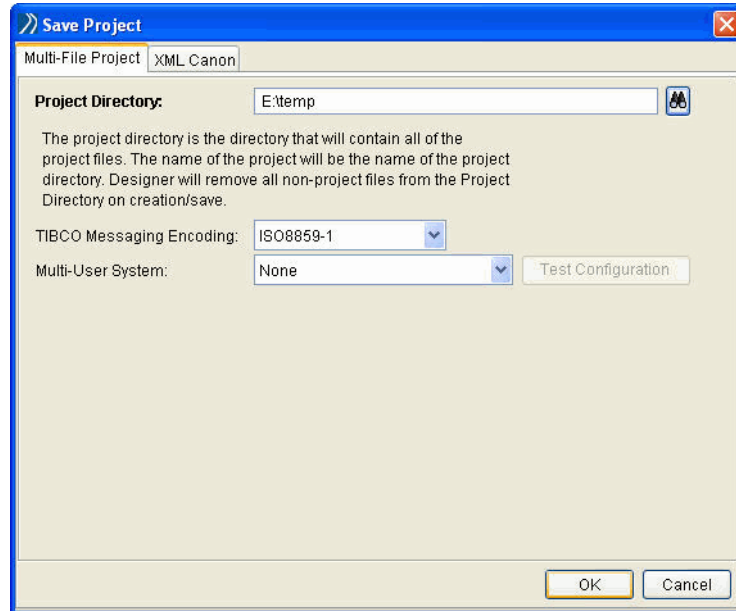
1. Start TIBCO Designer. In the TIBCO Designer startup panel, click **New empty project**.

Figure 17 TIBCO Designer startup panel



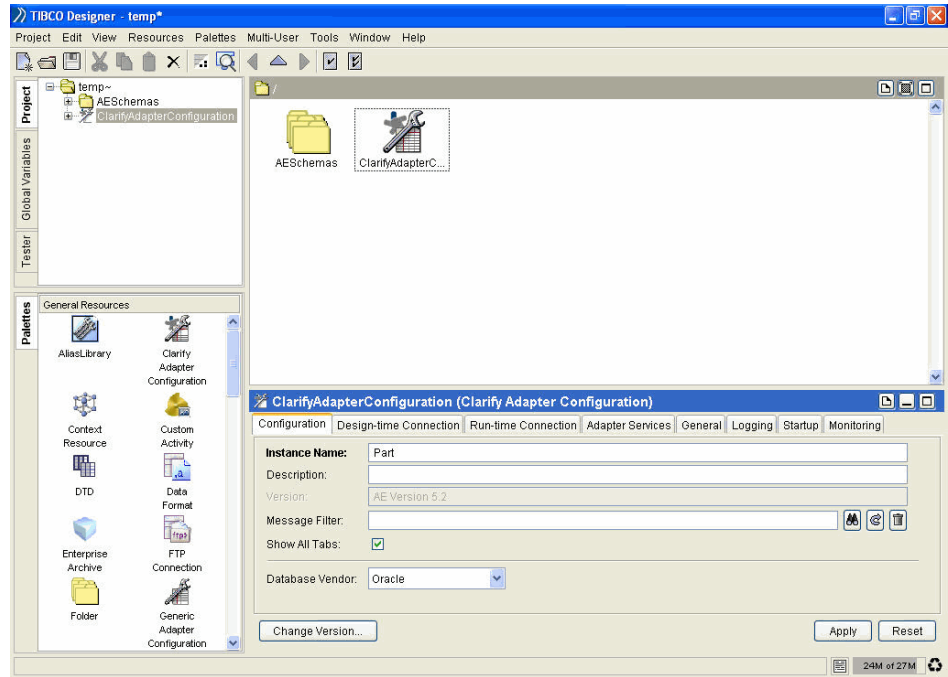
2. Click the **Select a file from the file system** button in the Save Project dialog. Select the location of project where you intend to save the project and specify a name for the project.

Figure 18 Save Project dialog



3. Drag the **ClarifyAdapterConfiguration** icon from the palette panel into the design panel.
4. In the Configuration tab, enter `Part` in the Instance Name field.

Figure 19 TIBCO Designer Configuration tab



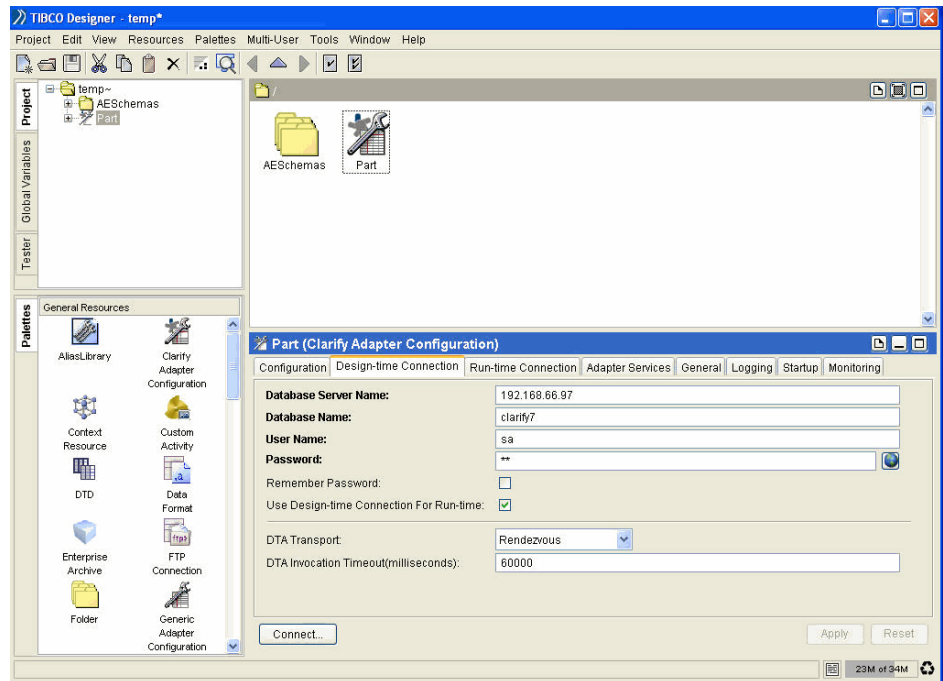
5. Click **Apply**.

Task B Configure Design-time and Run-time Connection Settings

Enter the design-time parameters in the Design-time Connection tab.

1. Enter the Clarify database server name in the Database Server Name field.
2. Specify the Clarify database name in the Database Name field.
3. Specify a valid User Name and Password combination for logging into the Clarify system.
4. Leave other default values unchanged and click **Apply**.
5. You can check the design-time connections by clicking the **Connect...** button and ensure that the design-time adapter (adc1yCRMDTA) is running. Once the connection is established, the **Connect...** button toggles to **Disconnect**.

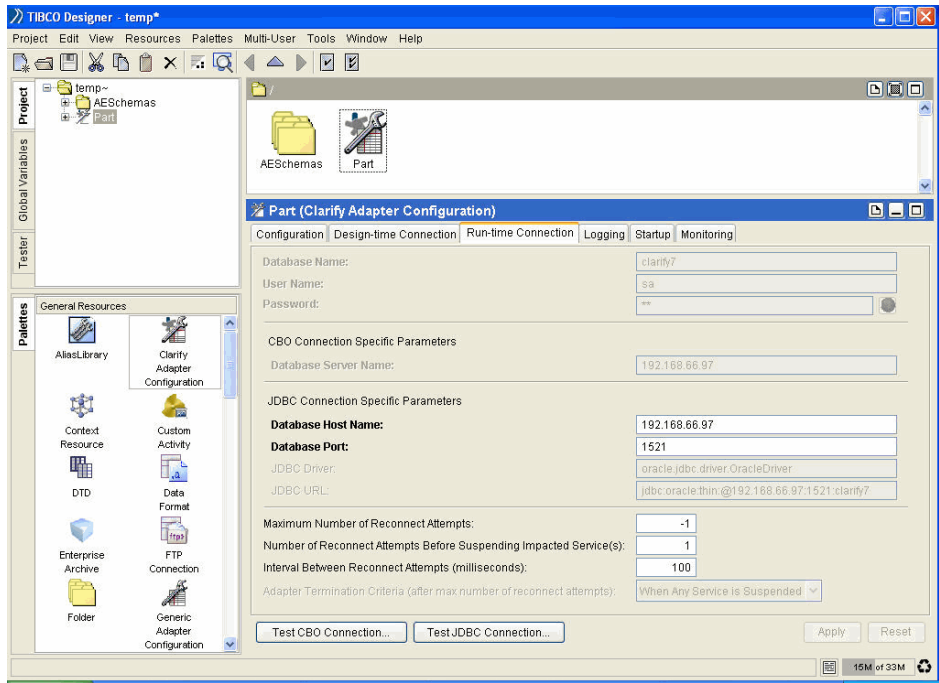
Figure 20 Design-time Connection settings



6. Enter the Database host name and Database port in the Run-time Connection tab.

7. Leave all other settings unchanged and click **Apply**.

Figure 21 Run-time Connection settings

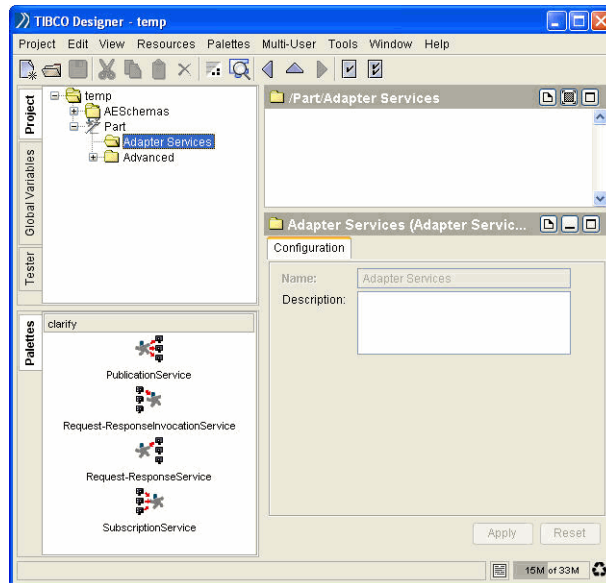


If you select Oracle as the Database Vendor in the Configuration tab, the Database Port field should be 1521. If you select Microsoft SQL Server, the Database Port field should be 1433.

Task C Fetch Schema

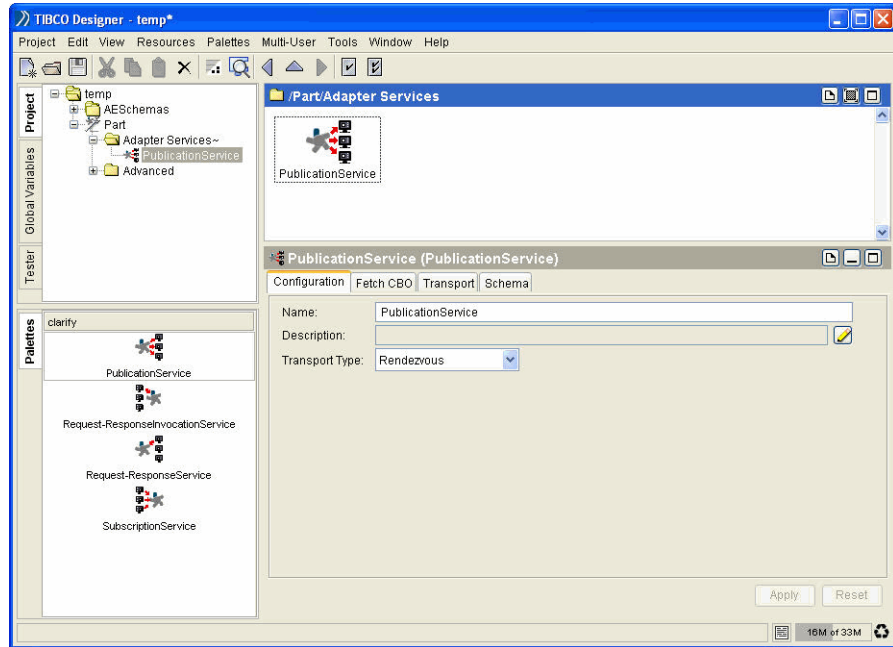
1. Double-click the Adapter Services folder in the project tree panel.

Figure 22 Adapter Services folder



2. Drag the **PublicationService** icon from the palette panel to the design panel.

Figure 23 Add Publication Service



3. Fetch CBO. You can do this in the following two ways:
 - Select the Fetch CBO tab, enter `mod_level` in text field against the **Fetch CBO List...** button and click it.
 - You can also do a fetch by clicking the **Fetch CBO List...** button. Select `mod_level` in the list displayed, click **OK**.

Figure 24 Fetch CBO list

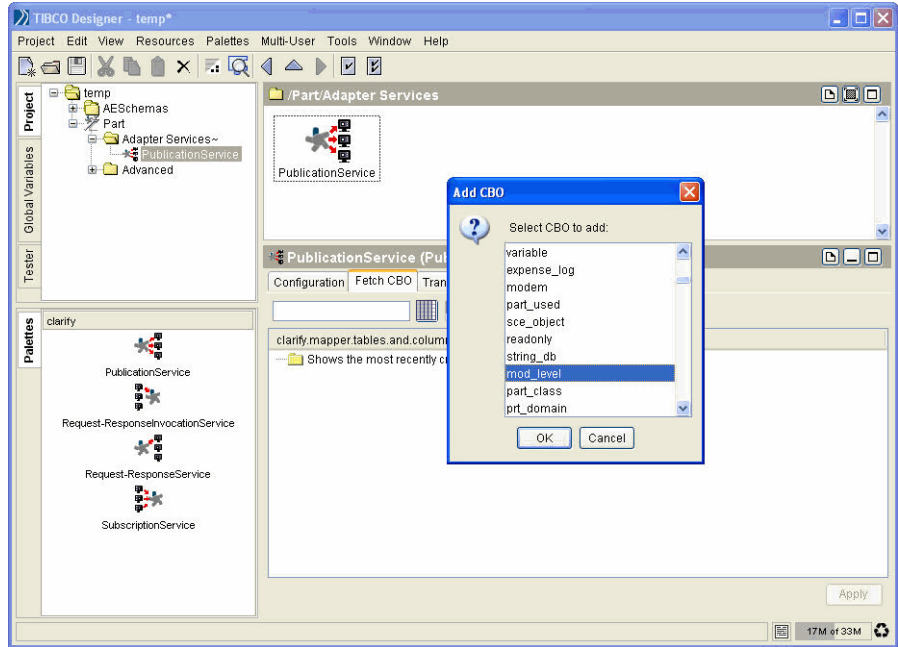
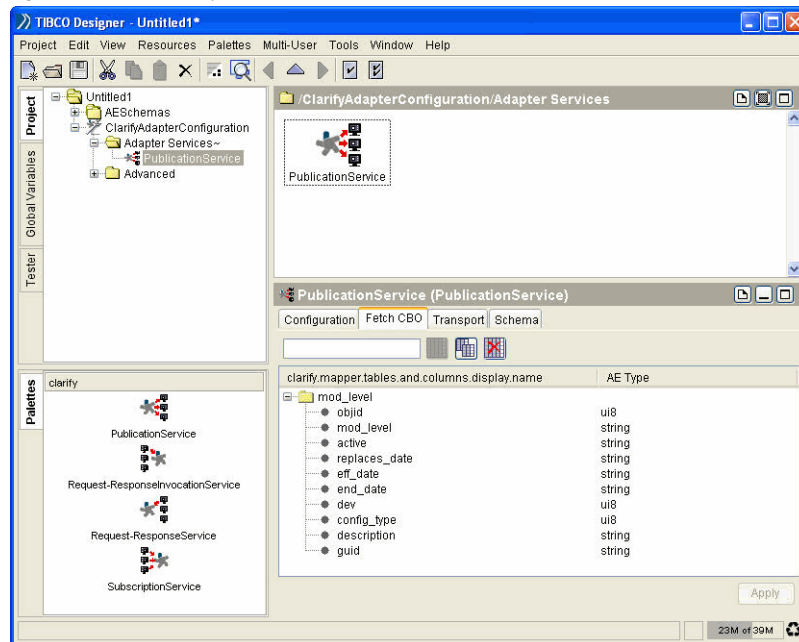
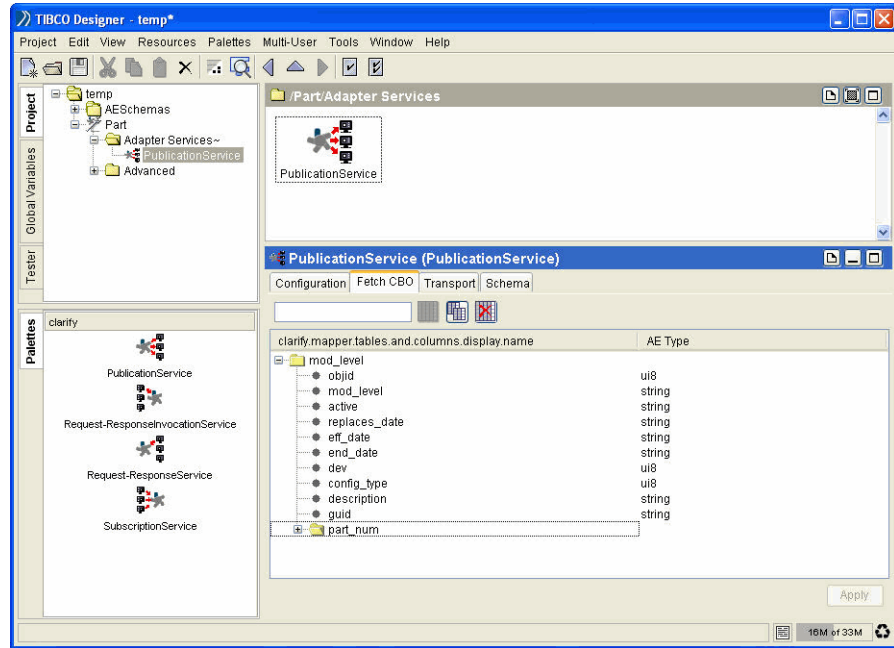


Figure 25 Schema fetched



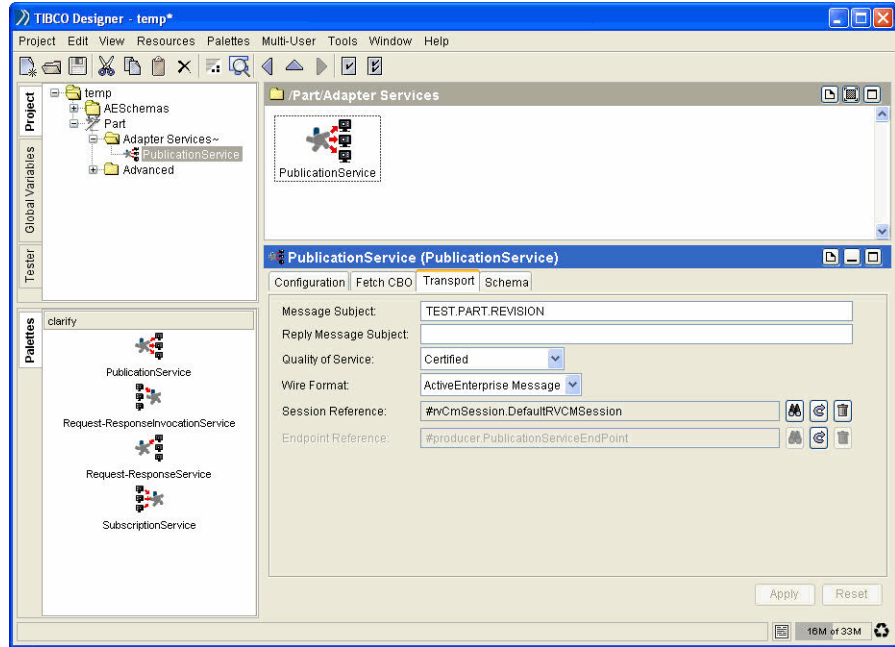
- Click the **Fetch Contained CBO...** button to fetch the `part_num` schema. This creates a `part_num` class inside the `mod_level` schema for the `part_num` business object. `part_num` is populated in the Class field as shown next.

Figure 26 Fetch Contained CBO



- If you want to remove any of the schemas, select that schema and click the **Remove CBO...** button.
- In the Transport tab, enter the appropriate subject in the Message Subject field.
- Select the appropriate quality of service and wire formats in the corresponding fields and click **Apply**.

Figure 27 Transport settings



8. Save the project.

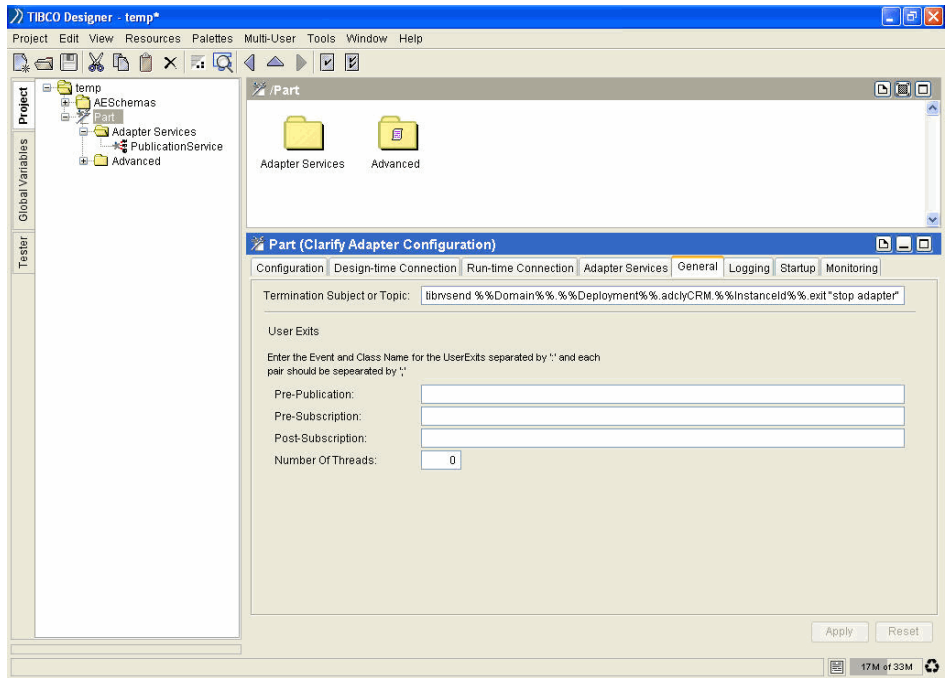
Task D Specify the Termination Subject to be Used to Stop the Adapter

1. Select the Part folder in the project tree panel.
2. Check the **Show All Tabs** checkbox.
3. Select the General tab. In the Termination Subject or Topic field, specify the following termination subject on which the adapter should listen for the stop message.

```
tibrvsend
%%Domain%%.%%Deployment%%.adclyCRM.%%InstanceId%%.exit "stop
adapter"
```

4. Click **Apply**.

Figure 28 Specify the Termination Subject



Deploy the Adapter

1. After configuring the adapter, make a copy of the default properties file (`adcllyCRM.tra`), rename the copy to `adcllyCRM_orig.tra`.

The `.tra` file and the adapter executables are in the `TIBCO_HOME\adapter\adcllycrm\<version_num>\bin` directory.

2. Update the client variable (`tibco.repourl`) in the `adcllyCRM_orig.tra` properties file with the name of the project you just created.

If you have saved your local project as `testconfig.dat`, update `adcllyCRM_orig.tra` by providing the following value for `tibco.repourl`:

`TIBCO_HOME/adapter/adcllyCRM/<version_num>/bin/testconfig.dat`

Start the Adapter

The run-time adapter can be run as a console application on all platforms.

You can run the adapter using a custom properties file. The next command line starts the adapter service using the customized properties file `adclyCRM_orig.tra`, located in the same directory as the executable.

```
adclycrm --run --propFile adclyCRM_orig.tra
```

The absolute path name to the properties file must be given if it is not located in the same directory as the executable.

Stop the Adapter

Use one of the following methods to stop the adapter:

- From a command window, stop the adapter by sending the following message on the terminate subject or terminate topic. Ensure that this is the message specified in the General tab. See Specify the Termination Subject to be Used to Stop the Adapter on page 127.

```
tibrvsend  
%%Domain%%.%%Deployment%%.adcllyCRM.%%InstanceId%%.exit "stop  
adapter"
```

- Using TIBCO Hawk Display, invoke the `stopApplicationInstance()` method.

This chapter explains how to create an adapter instance and assign it services by configuring standard settings. All configuration tasks are performed in TIBCO Designer and the information is stored in a project that is later used by the run-time adapter. The adapter services are used to communicate with ClarifyCRM.

Topics

- *Overview, page 134*
- *Adapter Instance Fields, page 136*
- *Adapter Services, page 151*
- *Publication Service Fields, page 152*
- *Subscription Service Fields, page 162*
- *Request-Response Service Fields, page 170*
- *Request-Response Invocation Service Fields, page 179*

Overview

The adapter can be configured with just a few steps before starting it. Alternatively, you can specify sessions and set configuration parameters to suit your needs before starting the adapter.

All adapter configuration is done using TIBCO Designer. Make sure it has been installed appropriately before continuing (see the *TIBCO Designer User's Guide*).

A set of preconfigured projects involving the frequently used ClarifyCRM Business Objects is shipped along with the adapter. You can configure them according to your requirements or start a new configuration. All the adapter configuration information (of the instances created based on the preconfigured instance) is saved as a project.

Start the Design-time Adapter

Before starting TIBCO Designer, start the design-time adapter if the input files are on a remote system that is not mapped to or mounted on your local system.

- On UNIX systems, start the design-time adapter on the command line.
- On Windows systems, the design-time adapter is installed as a service and starts automatically.

Configuration Tasks

Use the following sequence to create and configure an adapter service.

1. Start TIBCO Designer and open a multi-file project. See the *TIBCO Designer User's Guide* for details.
2. Drag the **ClarifyAdapterConfiguration** icon from the palette panel to the design panel. This creates an adapter instance named, by default, `ClarifyAdapterConfiguration`.
3. Define the adapter instance by assigning a new name and optionally change logging options. See Logging Tab on page 144 for details.
4. Add a service to the adapter instance by dragging the service icon from the palettes panel to the design panel.
5. Under the Configuration tab, set the required combination of options.
6. Save the project and exit TIBCO Designer.
7. Prepare TIBCO Staging tables for publishing, if necessary. See Prepare ClarifyCRM Components for the Publication Service on page 53 for details.

After configuring the adapter, you must create the run-time adapter property file and add the project name and adapter instance name.

Password Handling

At design-time, the adapter uses a password to connect to the backend application and fetch metadata. At run-time, the adapter uses a password to connect to the backend application and interoperate with it.

If you plan to run the adapter locally, define the run-time password value to be a global variable. Before starting the adapter, include the run-time password as client variable in the adapter's `.tra` file and obfuscate it using obfuscate tool. For example, if the password value is defined as `%%myPassword%%`, create a global variable named `myPassword` in the global variables section without value and include the following entry in the adapter's `.tra` file:

```
tibco.clientVar.myPassword
```

If you plan to deploy the adapter using TIBCO Administrator Enterprise Edition, checkmark the `Service` property of the global variable in the global variables section. Before deploying the adapter, 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 Version AE Version 4.0 or AE Version 5.0 or any intermediate version.

Adapter Instance Fields

The following tabs can be used to define an adapter instance:

- Configuration Tab on page 136
- Design-time Connection Tab on page 137
- Run-time Connection Tab on page 138
- Adapter Services Tab on page 141
- General Tab on page 142
- Logging Tab on page 144
- Startup Tab on page 147
- Monitoring Tab on page 148

Configuration Tab

Instance Name

Use the default name or replace it.

- An instance name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used.
- An instance name cannot use global variables.
- An instance name must be unique with respect to other adapter instances for the same adapter in the project. The same instance name can be used to name an adapter instance for a different adapter in the same project. For example, an R/3 adapter instance named TEST and a Siebel adapter instance named TEST can coexist in the same project.
- Each instance name must be unique per adapter within a project even if each instance is defined in a different folder. That is, configuring same-named adapter instances in different folders will not make their names unique.

When you create an adapter instance, the palette automatically creates several resources for it. The names of these resources are derived from the name of the instance they belong to. Changing the adapter instance name results in an automatic regeneration of the resources names. However, if you manually modify any resource name, that particular name will *not* be automatically regenerated the next time you rename the adapter instance.

Description

Provide information about the adapter instance that you want stored in the project. The field is optional.

Version

The version string indicates the ActiveEnterprise (AE) format in which the adapter instance is saved. When a new adapter instance is created in TIBCO Designer, the version string is set to `AE Version 5.2`.

To change versions, click the **Change Version...** button.

Message Filter

Specify a message filter, if you have configured a message filter resource for use with the adapter. The filter allows you to manipulate incoming and outgoing data before sending it on the network or handing it to the target application.

Filters can be written using the TIBCO Adapter SDK. See the *TIBCO Adapter SDK Programmer's Guide* for details.

Show All Tabs

Select this check box to display two additional tabs for configuring advanced options, Adapter Services tab and General tab.

Database Vendor

Select the database type to which the adapter connects. You can either select `Oracle` or `Microsoft SQL Server` from the drop-down list.

Design-time Connection Tab

Many of the following fields make use of global variables. Click the Global Variables tab in the project panel to enter a value for a global variable.

Database Server Name

If you selected `Microsoft SQL Server` in the Configuration tab, then specify the name of the server machine.

If you selected `Oracle` in the Configuration tab, then specify the `SQL*NET` connection string to the Oracle instance.

Database Name

Specify the name of the database you wish to connect to.

User Name

Specify the user name for the account used by the design-time adapter to access the ClarifyCRM system.

Password

Specify the password for the account used by the design-time adapter to access the ClarifyCRM system.

Remember Password

If this check box is cleared, the password must be entered each time the project is opened. If checked, the password will be stored in the project.

Use Design-time Connection for Run-time

Select this check box to change the default design-time connection settings.

On clearing this check box, connection fields are enabled in the Run-time Connection tab. Then you can connect to a different database by entering valid values in the Database Name, User Name, Password, and Database Server Name fields.

DTA Transport

Select the transport to be used by the design-time adapter, JMS or Rendezvous. After selecting the transport, the transport-specific configuration fields display.

DTA Invocation Timeout (milliseconds)

The default value is 60000 milliseconds (60 seconds). If the design-time adapter is unable to connect to the database in 60 seconds, the operation times out.

Run-time Connection Tab**Database Name**

Specify the name of the ClarifyCRM database you wish to connect to.

This field is enabled only if the `Use Design-time Connection For Run-time` check box is cleared in the Design-time Connection tab.

User Name

Specify a valid user name to connect to the ClarifyCRM system.

This field is enabled only if the `Use Design-time Connection For Run-time` check box is cleared in the Design-time Connection tab.

Password

Specify a valid password.

This field is enabled only if the `Use Design-time Connection For Run-time` check box is cleared in the Design-time Connection tab.

Database Server Name

If you selected `Microsoft SQL Server` in the Configuration tab, then specify the name of the server machine.

If you selected `Oracle` in the Configuration tab, then specify the `SQL*NET` connection string to the Oracle instance.

This field is enabled only if the `Use Design-time Connection For Run-time` check box is cleared in the Design-time Connection tab.

Database Host Name

Specify the machine name on which the database is hosted.

Database Port

Specify the port number on which the database is listening.



If you select `Microsoft SQL Server` in the Configuration tab, specify the Database Port to 1433. If you select `Oracle` in the Configuration tab, specify the Database Port to 1521.

JDBC Driver

The name of the JDBC driver the adapter uses to connect to the ClarifyCRM system. This field is disabled by default.

JDBC URL

The URL used by the driver to connect to the ClarifyCRM system. This field is disabled by default.

Maximum Number of Reconnect Attempts

Specify the maximum number of reconnection attempts to make before the run-time adapter or adapter service is stopped. A value of -1 means reconnection attempts will continue indefinitely.

Number of Reconnect Attempts Before Suspending Impacted Service(s)

Specify the number of reconnection attempts to make before suspending a run-time adapter or adapter service. The Adapter Termination Criteria field value determines whether the adapter or an adapter service is stopped.

Interval Between Reconnect Attempts (milliseconds)

Specify the time interval in milliseconds to elapse between each reconnection attempt.

Adapter Termination Criteria (after max number of reconnect attempts)

When Any Service is Suspended stops the adapter if any one service has been unable to re-establish connection after the specified number of reconnection attempts.

This field is unchangeable for TIBCO Adapter for ClarifyCRM.

Test CBO Connection

Clicking the **Test CBO Connection...** button validates the CBO specific parameters entered by the user (Database Server Name, Database Name, User Name, Password) by creating a test connection with the ClarifyCRM system.

Test JDBC Connection

Clicking the **Test JDBC Connection...** button validates the JDBC specific parameters entered by the user (Database Host Name, Database Port, User Name, Password) by creating a test JDBC connection with the ClarifyCRM system.

Adapter Services Tab

Retain Published Messages

On clearing this check box, the published messages will be deleted from the TIBCO Staging tables. By default, this check box is selected and the published messages will not be deleted from the TIBCO Staging tables.



Deletion of messages is not recommended because retracing the history of the messages is impossible. But performance of the adapter may degrade if the size of the TIBCO Staging tables grows too large.

Polling Batch Size

This field is used to specify the number of unprocessed messages to be processed while polling the database. The default polling size is 10.

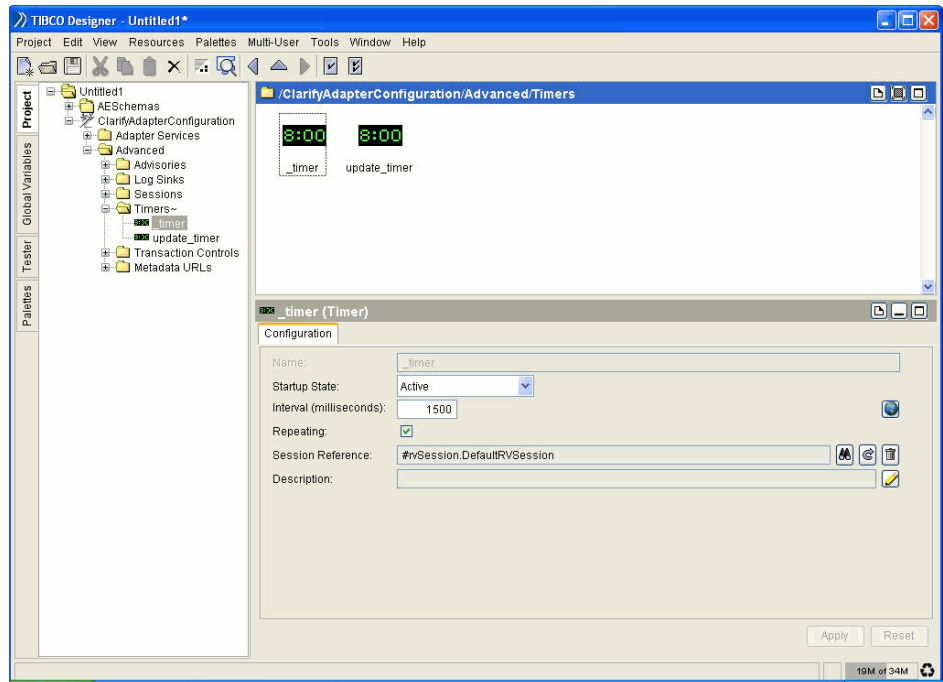
Changing the Polling Interval and the Update Interval

1. Navigate to the **Timers** folder available in the **Advanced** folder of the project panel.

There are two timer icons available: **_timer** icon and **update_timer** icon.

2. Select the **_timer** icon to specify appropriate values in the following fields:
 - **Startup State** — The Startup State refers to the state of the timer during startup. Select the Startup State as **Active** or **Inactive** from the drop-down list as required.
 - **Interval (milliseconds)** — In this field, enter the interval after which the adapter polls the database repeatedly.
 - **Repeating**— Select the Repeating check box if this is a repeating timer. If the check box is cleared, this is an once-only timer.

Figure 29 Change the polling interval



- 3. Select the **update_timer** icon to specify the update interval. In the Interval (milliseconds) field, enter the interval after which the adapter updates the database repeatedly.

General Tab

Termination Subject or Topic

A message sent on the termination subject (if Rendezvous is the transport) or topic (if JMS is the transport) stops the adapter. In most cases, you should use the default value.

See *TIBCO Rendezvous Concepts* for information about specifying subject names. See *TIBCO Enterprise Message Service User's Guide* for information about publishing on a topic.

Pre-Publication

This field is used to implement pre-publication User Exits for the Publication Service. Enter the event name and class name separated by a colon and each pair should be separated by a semi colon.

Example:

```
update_case:Clarifyprepubevent;TEST.CASE:Clarifyprepubevent
```

Pre-Subscription

This field is used to implement pre-subscription User Exits for the Subscription Service. Enter the event name and class name separated by a colon and each pair should be separated by a semi colon.

Example:

```
TEST.CASE:Clarifypresubevent
```

Post-Subscription

This field is used to implement post-subscription User Exits for the Subscription Service. Enter the event name and class name separated by a colon and each pair should be separated by a semi colon.

Example:

```
TEST.CASE:Clarifypostsubevent
```

For more details on User Exits, see User Exits on page 305.

Number of Threads

Specify the number of threads in this field. If not specified, a default number of 0 will be used.

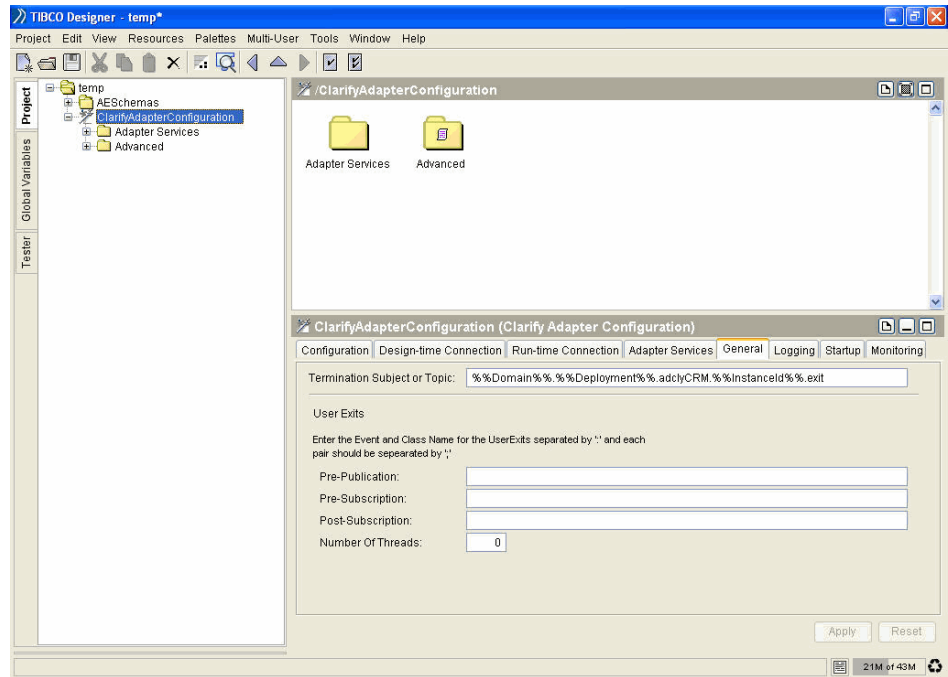


You need to determine the maximum value for the number of threads after doing the necessary performance tests on your hardware/software production environment.



To publish the messages in an ascending order of message numbers, you must configure the thread count as 0 or 1.

Figure 30 Sample General screen



Logging Tab

Use Advanced Logging

When the Use Advanced Logging check box is not selected (the default), you can set two standard output destinations (sinks) for trace messages and set the tracing level for the roles selected.

When Use Advanced Logging is selected, you have complete control on selecting the destinations and associating desired roles with each of the destinations.

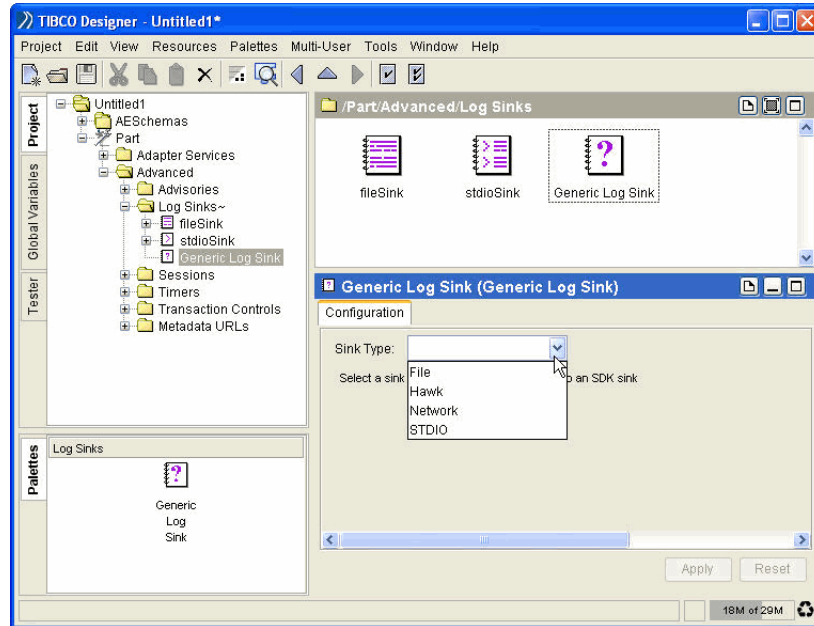
To create and configure the sinks, select the Log Sinks folder under the Advanced folder in the project panel.

To create sinks, drag the **Generic Log Sink** icon from the palette panel to the design panel. From the configuration panel, select the sink type. The following are the sink types available:

- File
- Hawk
- Network

- STUDIO

Figure 31 Generate sinks



When File and STUDIO sinks are created, they offer further configuration options.

- For the File sink, the file name, file limit, file count, and the option to append or overwrite can be specified. When created by default, this is set to 30000 bytes, 3 and append mode respectively.
- For the STUDIO sink, the option to write to stdout or stderr can be selected. When created by default, stdout is selected.

The Hawk sink uses the hawk session, created and used by the adapter for monitoring purposes, to send tracing messages to the TIBCO Hawk monitor or Display. For details on Hawk sessions, see Using Global Variables on page 203. The configuration for the Hawk sink involves specifying the `MicroAgent Name` that must be specified in the configuration panel.

The Network sink is used to publish tracing message on TIBCO Rendezvous. The configuration for the Network sink involves specifying the session, and the subject on which the trace messages needs to be published.

For all the sinks, optionally the name and description for the sinks can be provided.

Log to Standard I/O

(STDIO Sink) When selected, trace messages are displayed in the command prompt window where the adapter is started. When not selected, trace messages do not display in the window.

Log File

Specify the name of the log file (log sink) to which trace messages are written. Global variables can be used to specify the location of the log file. See Using Global Variables on page 203 for more information.

The roles available are Info, Debug, Warning, and Error messages. The trace message generated depends on the roles selected. Turning on the roles can affect the performance of the adapter. Therefore, it is recommended that you turn on the required roles only.

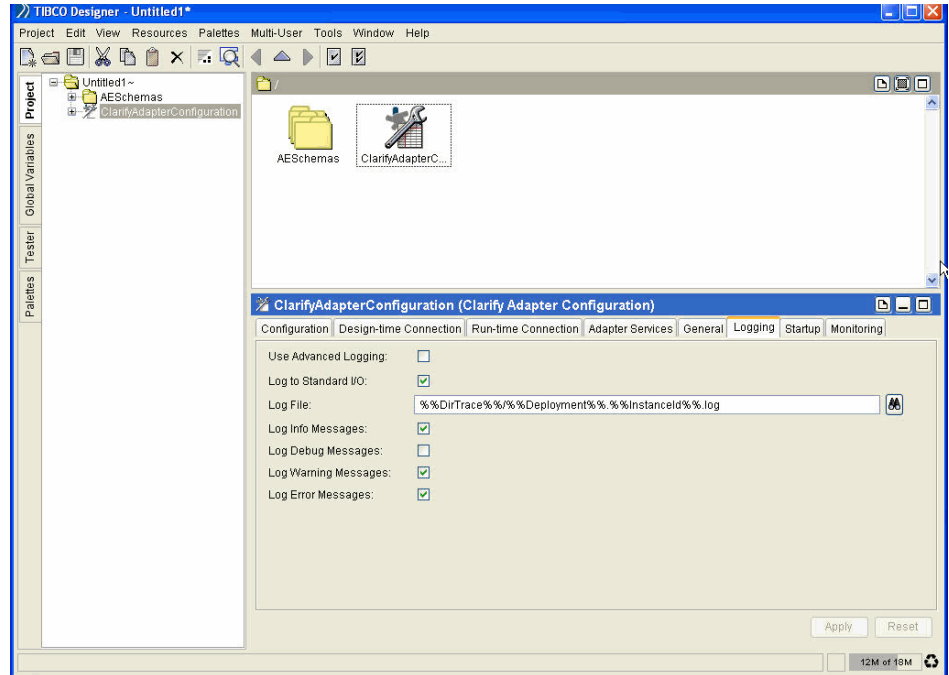
Log Info/Debug/Warning/Error Messages

Trace messages of the selected level(s) will be collected in the named log sink. You can configure which levels of trace messages you want logged, and to where trace messages are sent.

There are three types of logs (log sinks) that you can configure to hold trace messages, corresponding to three levels (roles) of trace messages, Info, Warning and Error.

A fourth level of trace messages, Debug, is reserved and should not be enabled unless requested by the TIBCO Product Support Group. The Debug option writes a lot of information to the log file and significantly degrades the performance of the adapter.

Figure 32 Sample Logging screen



Startup Tab

Show Startup Banner

The startup banner displays the run-time adapter version, the infrastructure version on which the adapter is built, and copyright information in the console window when the adapter is started.

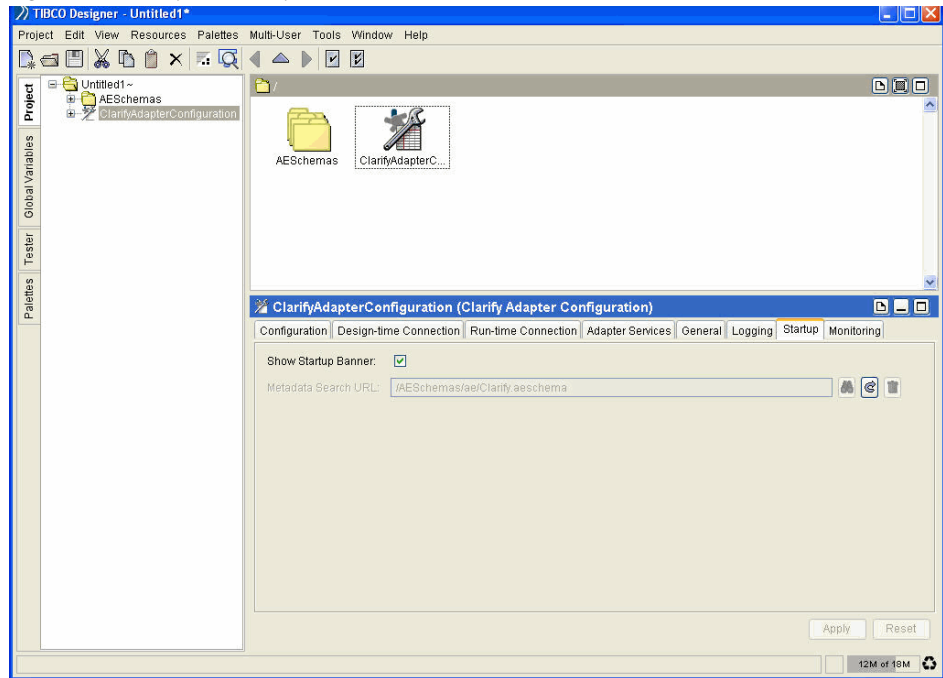
The startup banner is predefined and cannot be changed.

Metadata Search URL

This field specifies the location where the adapter searches for base schemas. The adapter searches for any schema that has been defined and saved at this location, and that should be loaded at startup.

This field is predefined and cannot be changed.

Figure 33 Sample Startup screen



Monitoring Tab

Many of the following fields make use of global variables. Click the Global Variables tab in the project panel to enter a value for a global variable.

Enable Standard MicroAgent

Allows you to turn on or off the standard TIBCO Hawk Microagent.

The way to do it is also configurable. By clicking the **Toggles the display of a field for setting values with global variables.** button, a standard check box or text value (true or false) can be used.

Standard MicroAgent Name

This is the name for the standard microagent to be registered with the TIBCO Hawk system. In most cases, the default value is used.

The InstanceId variable need not be set because it is automatically set at run time by the run-time adapter.

Standard MicroAgent Timeout (ms)

The timeout value for the Standard MicroAgent in milliseconds.

Enable Class MicroAgent

Allows you to turn on or off the instance or class specific standard TIBCO Hawk Microagent.

The way to do it is also configurable. By clicking the **Toggles the display of a field for setting values with global variables.** button , a standard check box or text value (true or false) can be used.

Class MicroAgent Name

This is the name for the class microagent to be registered with the TIBCO Hawk system. In most cases, the default value is used.

Class MicroAgent Timeout (ms)

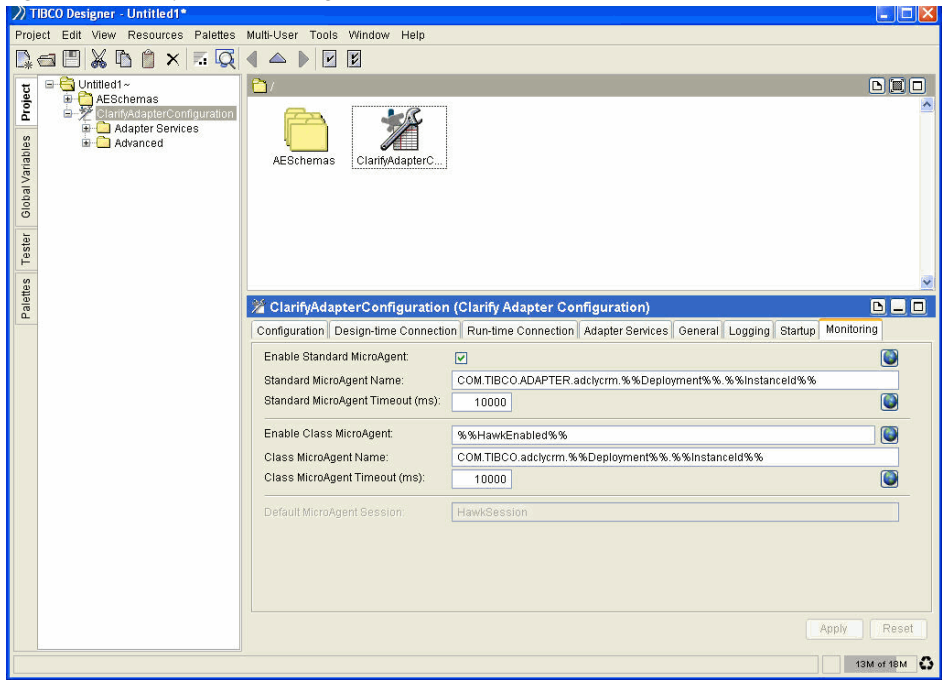
The timeout value for the Class MicroAgent in milliseconds.

Default MicroAgent Session

The session name and the corresponding session is automatically generated by TIBCO Designer. Do not change the session name or the session. However, you can modify the session parameters if required. Navigate to the **Sessions** folder under the **Advanced** folder to modify the session parameters.

Make sure you have set the correct parameter value for the global variables that correspond to the TIBCO Hawk configuration. If the session parameters are not set properly, the microagents will not display in the TIBCO Hawk Display.

Figure 34 Sample Monitoring screen



Adapter Services

After configuring an adapter instance, select one or more adapter services for the instance.

1. In the project panel, click **ClarifyAdapterConfiguration**, then click **Adapter Services**.
2. Drag a **PublicationService**, **SubscriptionService**, **Request-ResponseService** or **Request-ResponseInvocationService** icon to the design palette or select the icon in the project panel.
3. Specify options in the tabs that are displayed and click **Apply**.

The following sections describe the services and fields that are available to the adapter.

- Publication Service Fields on page 152
- Subscription Service Fields on page 162
- Request-Response Service Fields on page 170
- Request-Response Invocation Service Fields on page 179

Publication Service Fields

The following tabs are available:

- Configuration Tab on page 152
- Fetch CBO Tab on page 154
- Transport Tab on page 155
- Schema Tab on page 160

Configuration Tab

Name

You can use the default name or replace it.

- A service name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used.
- A service name cannot use global variables.

Description

Provide information about the adapter service that you want stored in the project. The field is optional.

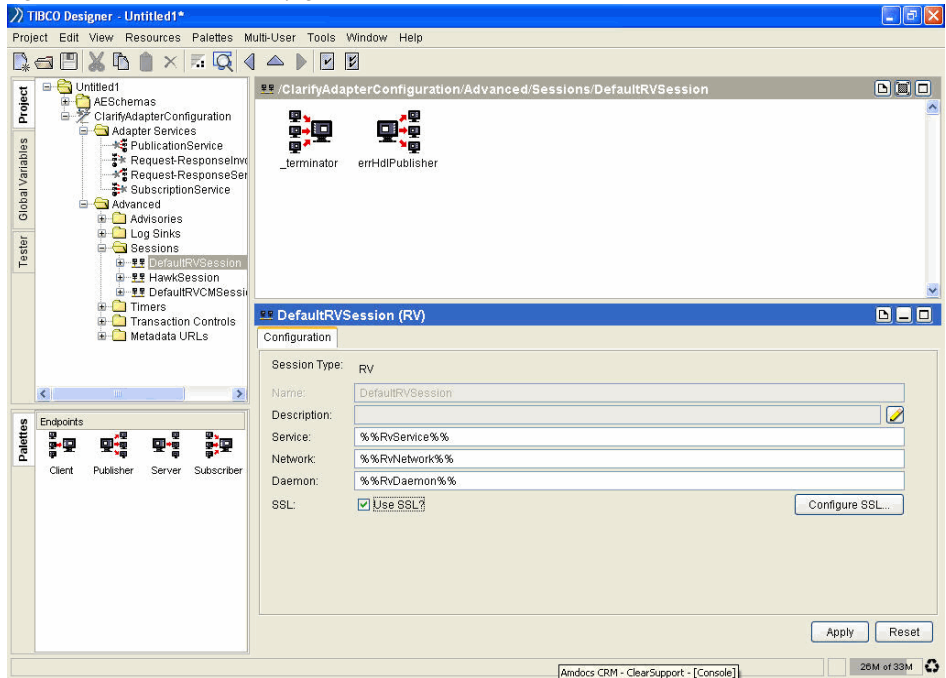
Transport Type

Select the transport to be used by the run-time adapter, JMS or Rendezvous. After selecting the transport, the transport-specific configuration fields display.

The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field, which uses a dialog to perform SSL configuration.

To enable and configure SSL, in the **Project** panel, expand the **Advanced** folder, then expand the **Sessions** folder. Select the TIBCO Rendezvous session or JMS topic and check the **Use SSL?** check box.

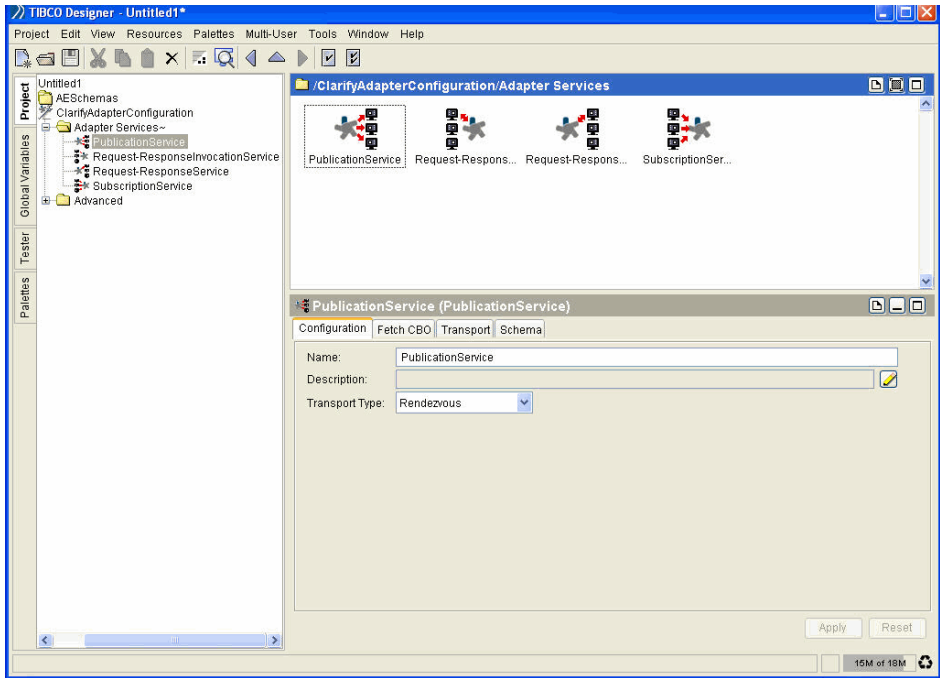
Figure 35 Enable and configure SSL



The SSL configuration options are explained in the online help associated with the session dialog.

Sample Publication Service Configuration Screen

Figure 36 Sample Publication Service Configuration screen



Configuring Multiple Publication Service Endpoints

The publishing component of the adapter allows you to define multiple Publication Service endpoints in an adapter instance. This enables quicker message processing.

For the adapter to sort and publish messages from multiple endpoints, the subject names of messages specified by the ClearBasic or Web forms must be the same as the subject names configured in the project. Otherwise, the Publication Service will handle the messages using a default Publication Service endpoint that publishes the message with the subject name specified by the ClearBasic form.

Fetch CBO Tab

This tab can be used to fetch the CBOs included in a message.

For example, if a message entry in the `TIBCO_MESSAGE_QUEUE` table contains two distinct values under the `object_name` column (Case, Contact). Fetch Case and Contact using the Fetch CBO tab.



You need not set any options in the Fetch CBO tab if you intend to use generic schema for publishing a message. See page 49 for details.

1. To select a schema, do one of the following:

- Type the schema name in the text box adjacent to the **Fetch CBO List...** button and click the button.
- Click the **Fetch CBO List...** button and select the schema case from the list that is displayed.

Click **OK**.



In case of a change in schema in ClarifyCRM, the schema needs to be fetched and loaded again.

2. Click the **Fetch Contained CBO...** button to select contact and click **OK**.

3. The contact CBO is loaded to the Publication Service.

Transport Tab

Message Transport options can be set for the Publication Service depending on the transport type selected in the Configuration tab.

Options Displayed with Rendezvous as the Transport

Message Subject

Message subject for this Publication Service.

Reply Message Subject

Reply subject for this Publication Service.

Quality of Service

If Rendezvous is selected as the transport type, select:

— Certified

Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or the message's time limit expires. This is called certified message delivery.

With certified message delivery, data is stored in a ledger file. The size of the ledger depends on several factors, the most important of which is the retention rate of stored data. That is, the ledger grows fastest in response to the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

— **Reliable**

Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected.

This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated.

Wire Format

Services must use the same wire format to exchange data.

— **XML Message**

The XML Message wire format conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the ActiveEnterprise schema.

— **ActiveEnterprise Message**

Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscriber.

ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO Adapter SDK set of data types. This format allows ActiveEnterprise components to perform extra validation on messages sent or received.

See the *TIBCO Adapter SDK Programmer's Guide* for details.

Session Reference

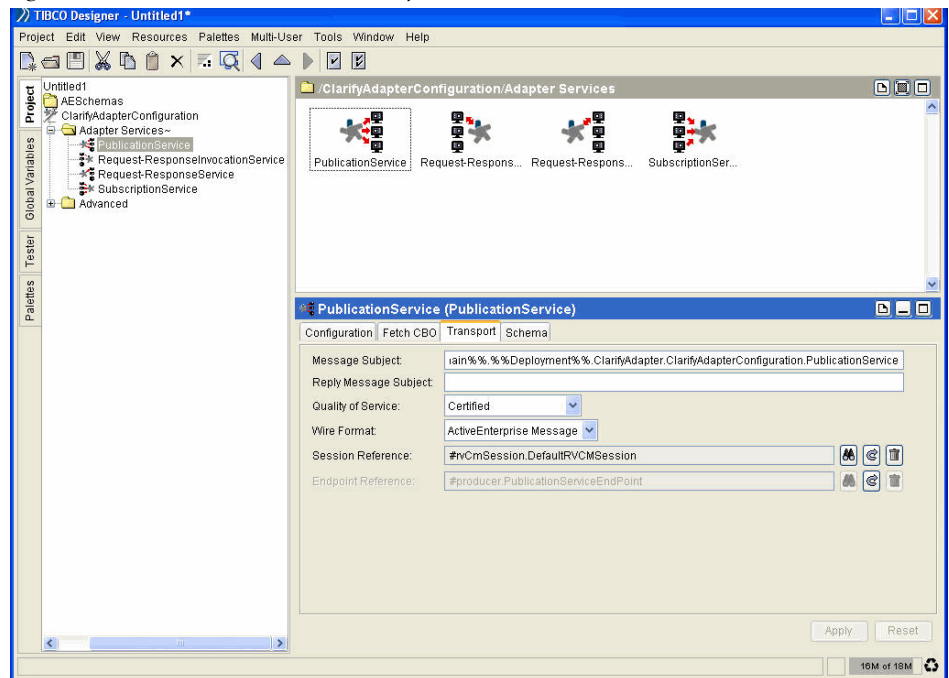
When you create a service, TIBCO Designer creates a corresponding session resource in the `Sessions` folder of the `Advanced` folder and displays it in this field.

If you have explicitly created a custom session of the same type, you can click the **Browse resources...** button to replace the autogenerated session. It is recommended not to change the session for a service.

Endpoint Reference

An endpoint reference for the service. This is a disabled field and points to the corresponding endpoint resource in the **Sessions** folder of the **Advanced** folder. The endpoint resource is automatically created by TIBCO Designer.

Figure 37 Publication Service Transport tab with TIBCO Rendezvous



Options Displayed with JMS as the Transport

Destination

By default, a service uses a dynamic destination that 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 dynamic destination, make sure the values for Domain and Deployment are not empty.

You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the EMS server before it can be used by the run-time adapter.

See the *TIBCO Enterprise Message Service User's Guide* for information about destinations.

Reply Destination

Reply destination of this Publication Service.

Wire Format

The format in which messages are to be published. Publishers and subscribers can only send and receive data if they agree on a specific wire format.

For JMS mode, only the XML Message wire format is available.

- The XML Message wire format conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the ActiveEnterprise schema.

Connection Factory Type

- Topic

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.

- Queue

A message sent to a queue is consumed by one and only one receiver. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver to access the queue receives the message. The other receivers do not. This messaging model is known as point-to-point.

Delivery Mode

- Persistent

In general, a message marked persistent will be available to a JMS client even if the EMS server goes down.

- Non-Persistent

A message marked non-persistent will not be available to a JMS client if the EMS server goes down.

Messages sent with the persistent delivery mode are always written to persistent storage, except when they are published to a topic that has no durable subscribers. When a topic has no durable subscribers, there are no subscribers that need messages re-sent in the event of a server failure. Therefore, messages do not need to be saved, and performance is improved because disk I/O is not required.

See the *TIBCO Enterprise Message Service User's Guide* for more information.

Session Reference

When you create a service, TIBCO Designer creates a corresponding session resource in the `Sessions` folder of the `Advanced` folder and displays it in this field.

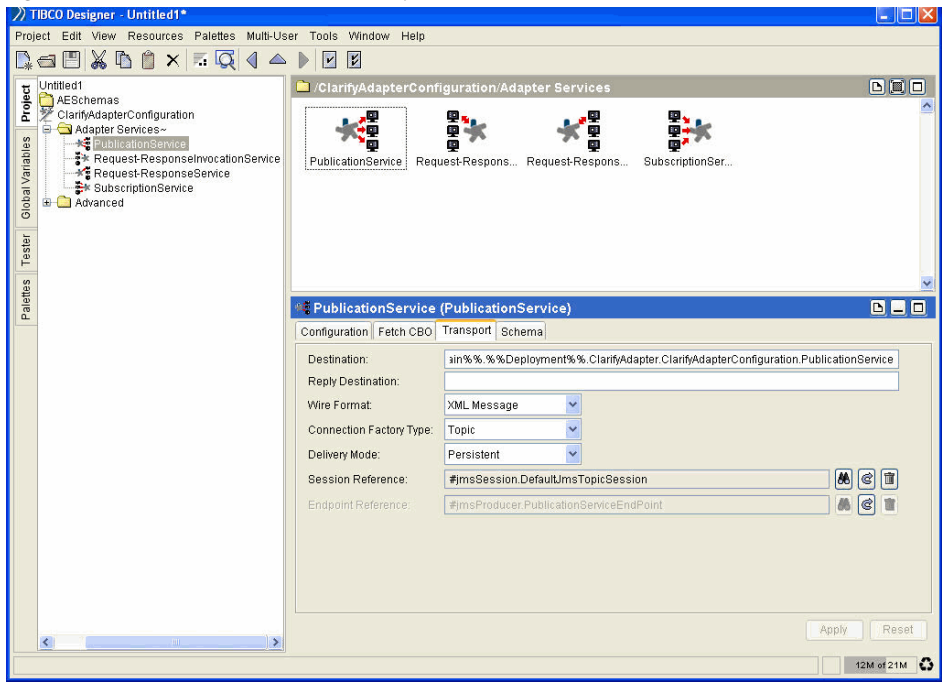
If you have explicitly created a custom session of the same type, you can click the **Browse resources...** button to replace the autogenerated session. It is recommended not to change the session for a service.

Endpoint Reference

An endpoint reference for the service. This is a disabled field and points to the corresponding endpoint resource in the `Sessions` folder of the `Advanced` folder. The endpoint resource is automatically created by TIBCO Designer.

A screen shot of the Publication Service Transport tab with JMS transport options set is shown next.

Figure 38 Publication Service Transport tab with JMS



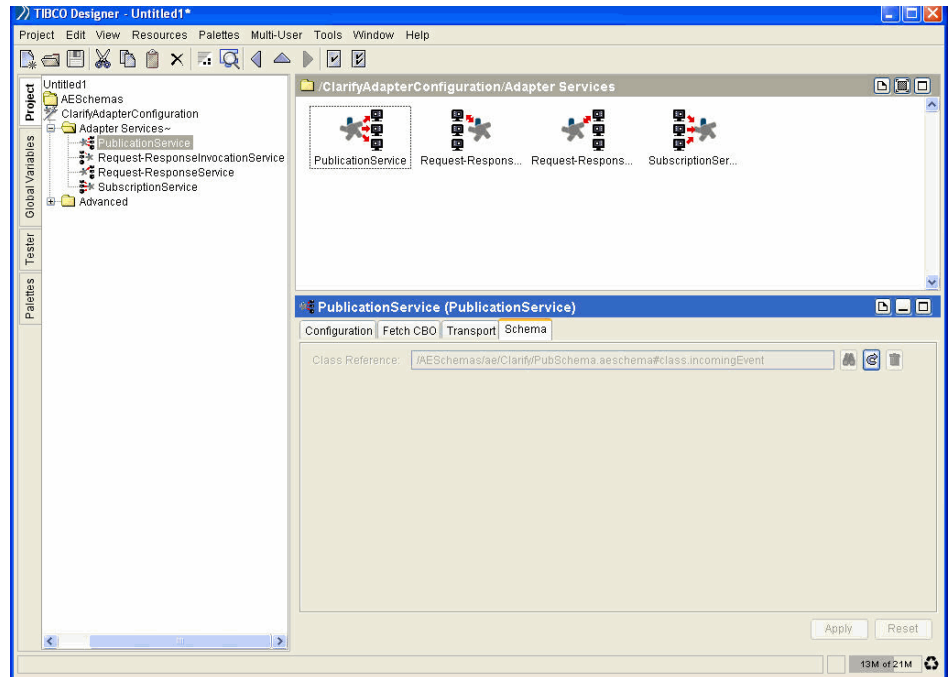
Schema Tab

The Schema tab can be used to view the schema for the Publication Service.

Class Reference

The schema class reference points to the class created for this component. Leave this field unchanged.

Figure 39 Publication Service Schema tab



Conditions When a Generic Schema is Used to Publish a Message

- If no Publication Services are found with the subject name as that in the ClearBasicForm, the message is published using the generic schema.
- If you do not associate any CBOs with the Publication Service endpoint; that is, if you do not use the Fetch CBO tab and if you directly specify the subject name in the Transport tab, the message is published using the generic schema.

Subscription Service Fields

The following tabs are available:

- Configuration Tab on page 162
- Fetch CBO Tab on page 163
- Transport Tab on page 164
- Schema Tab on page 168

Configuration Tab

Name

You can use the default name or replace it.

- A service name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used.
- A service name cannot use global variables.

Description

Provide information about the adapter service that you want stored in the project. The field is optional.

Transport Type

Select the transport to be used by the run-time adapter, JMS or Rendezvous. After selecting the transport, the transport-specific configuration fields display.

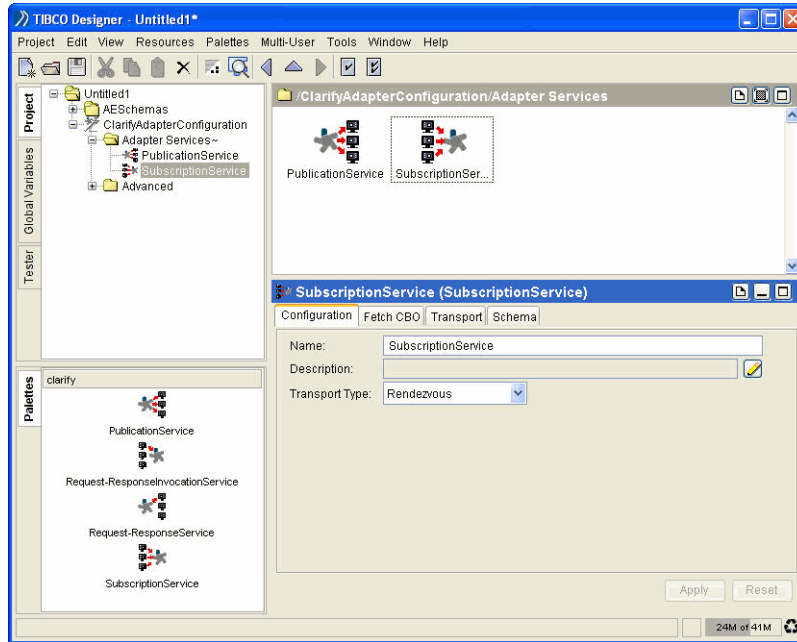
The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field, which uses a dialog to perform SSL configuration.

To enable and configure SSL, in the **Project** panel, expand the **Advanced** folder, then expand the **Sessions** folder. Select the TIBCO Rendezvous session or JMS topic and click **Use SSL?**. The SSL configuration options are explained in the online help associated with the session dialog.

Sample Subscription Service Configuration Tab Screen

The following screen shows a Subscription Service with default configuration parameters set.

Figure 40 Subscription Service Configuration tab



Fetch CBO Tab

You can set schema parameters for the Subscription Service using the Fetch CBO tab.

To select a schema, do one of the following:

- Type the schema name in the text box adjacent to the **Fetch CBO List...** button and click the button.
- Click the **Fetch CBO List...** button and select the schema case from the list that is displayed.

Click **OK**. The case CBO is loaded to the Subscription Service.



In case of a change in schema in ClarifyCRM, the schema needs to be fetched and loaded again.



If you are using the Custom JavaBean feature of the Subscription Service, this schema can be modified to include any level of sub-classes and sequences. Any class which is defined in TIBCO Designer can be included in this schema.

Transport Tab

Message Transport options can be set for the Subscription Service depending on the transport type selected in the Configuration tab.

Options Displayed with Rendezvous as the Transport

Message Subject

Specify the subject name to be used when subscribing.

By default, a service uses a message subject that 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 the values for Domain and Deployment are not empty.

You can type a TIBCO Rendezvous subject name different from the default in this field. See *TIBCO Rendezvous Concepts* for details.

Quality of Service

If Rendezvous is the transport type, select:

— Certified

Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the message's time limit expires. This is called certified message delivery.

If certified message delivery is used, data is stored in a ledger file. The size of the ledger depends on several factors, the most important one is the retention rate of stored data. That is, the ledger grows fastest in response to the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

— Reliable

Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures

because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated.

— Distributed Queue

Indicates load balancing should be enabled. An RVCMQ Session allows applications to use distributed queues for certified delivery, to any number of listeners using queuing member sessions that act together to process inbound task messages.

See *TIBCO Rendezvous Concepts* for details.

Wire Format

Services must use the same wire format to exchange data.

— XML Message

The XML Message wire format conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the ActiveEnterprise schema.

— ActiveEnterprise Message

Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscriber. ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO Adapter SDK set of data types. This format allows ActiveEnterprise components to perform extra validation on messages sent or received.

See the *TIBCO Adapter SDK Programmer's Guide* for details about the control information generated and sent with ActiveEnterprise messages.

Session Reference

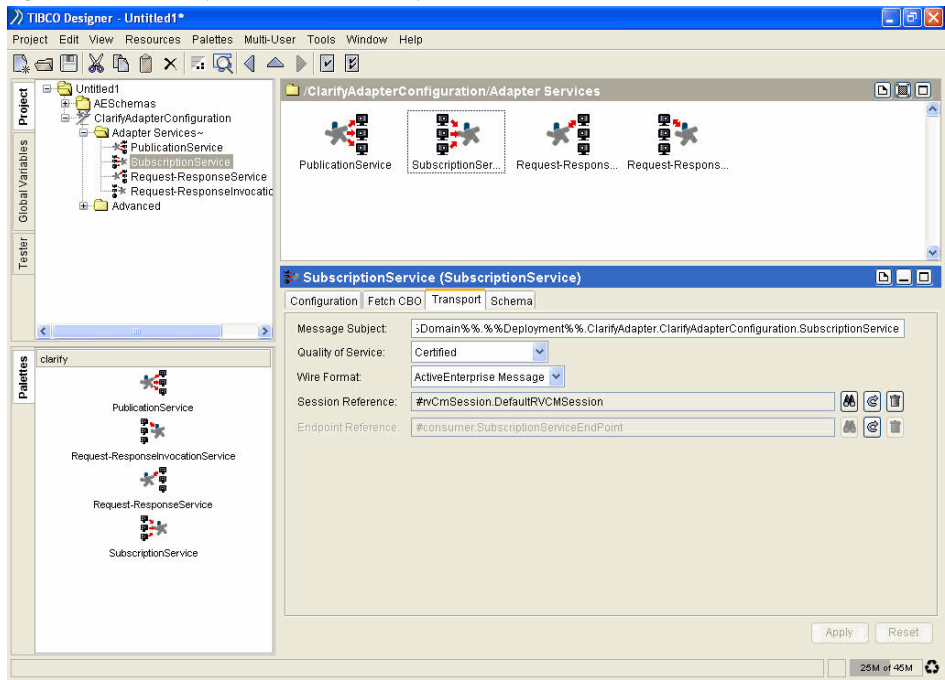
When you create a service, TIBCO Designer creates a corresponding session resource in the Sessions folder of the Advanced folder and displays it in this field.

If you have explicitly created a custom session of the same type, you can click the **Browse resources...** button to replace the autogenerated session. It is recommended not to change the session for a service.

Endpoint Reference

An endpoint reference for the service. This is a disabled field and points to the corresponding endpoint resource in the Sessions folder of the Advanced folder. The endpoint resource is automatically created by TIBCO Designer.

Figure 41 Subscription Service Transport tab with TIBCO Rendezvous



Options Displayed with JMS as the Transport

Destination

By default, a service uses a dynamic destination that 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 dynamic destination, make sure the values for Domain and Deployment are not empty.

You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the JMS server before it can be used by the run-time adapter.

See the *TIBCO Enterprise Message Service User's Guide* for more information.

Wire Format

The format in which messages are to be subscribed. Services must use the same wire format to exchange data.

For JMS mode of transport, only one wire format is available.

- The XML Message wire format conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the ActiveEnterprise schema.

Connection Factory Type

- Topic

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.

- Queue

A message sent to a queue is consumed by one and only one receiver. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver to access the queue receives the message. The other receivers do not. This messaging model is known as point-to-point.

Delivery Mode

An adapter Subscription Service can be durable or non-durable.

- Durable

Indicates that the service is registered with the EMS server. Messages sent to a durable Subscription Service are held by the EMS server until they are consumed by the service. The service can be down and expect to receive its messages when it comes back up.

- Non-Durable

Indicates that the service is not registered with the EMS server. Messages sent to a non-durable Subscription Service are not held by the EMS server. If the service is down, it will not receive the messages that arrived at the EMS server while the service was down.

The semantics for these fields are somewhat more complex than the explanation given here. See the *TIBCO Enterprise Message Service User's Guide* for more information.

Session Reference

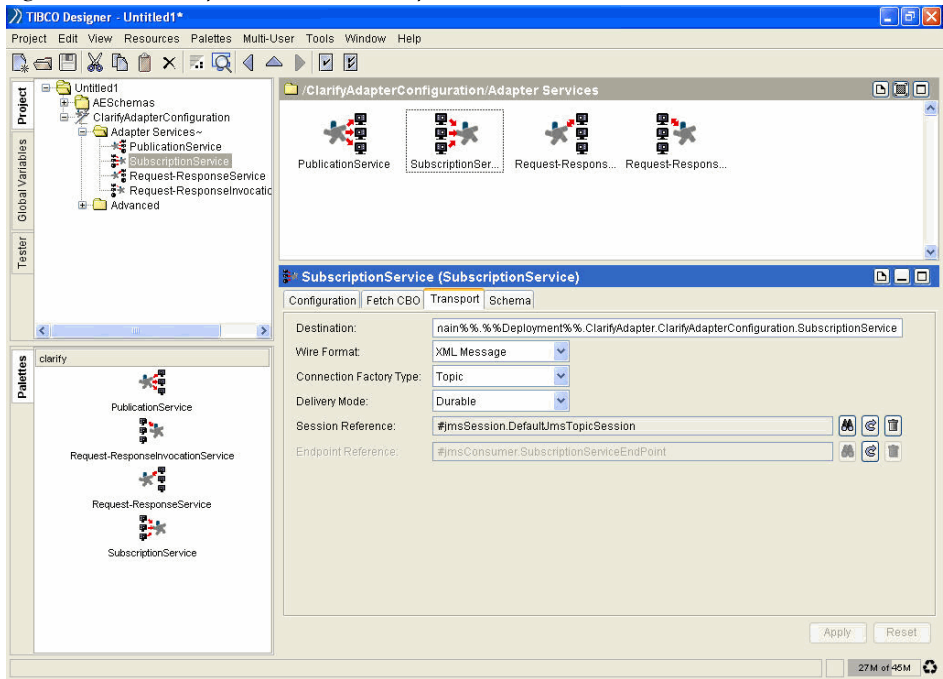
When you create a service, TIBCO Designer creates a corresponding session resource in the Sessions folder of the Advanced folder and displays it in this field.

If you have explicitly created a custom session of the same type, you can click the **Browse resources...** button to replace the autogenerated session. It is recommended not to change the session for a service.

Endpoint Reference

An endpoint reference for the service. This is a disabled field and points to the corresponding endpoint resource in the Sessions folder of the Advanced folder. The endpoint resource is automatically created by TIBCO Designer.

Figure 42 Subscription Service Transport tab with JMS



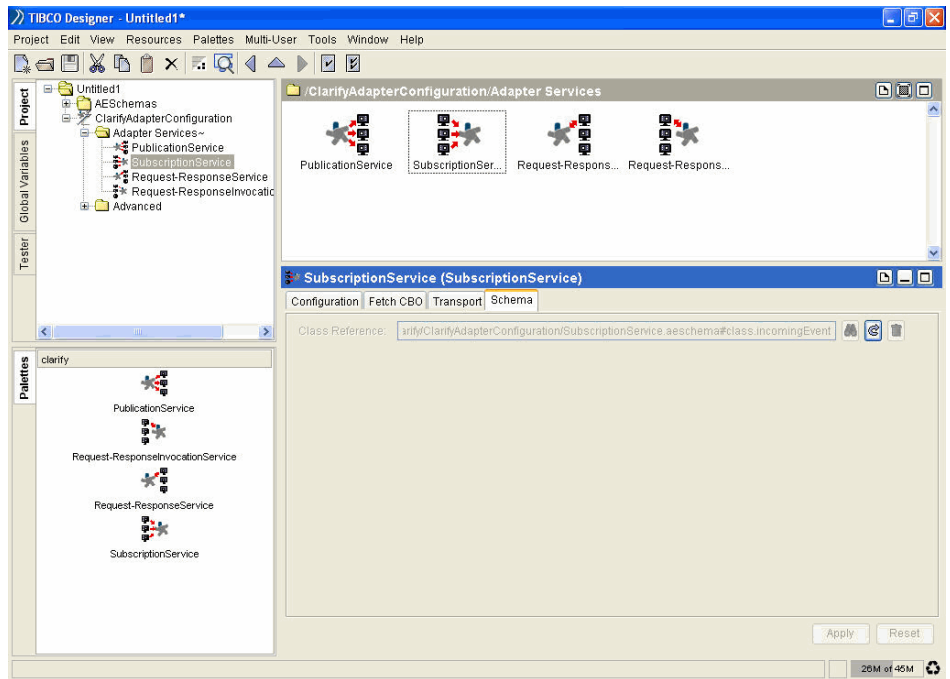
Schema Tab

The Schema tab can be used to view the schema for the Subscription Service.

Class Reference

The schema class reference points to the class created for this service.

Figure 43 Subscription Service Schema tab



Request-Response Service Fields

The following tabs are available:

- Configuration Tab on page 170
- Fetch CBO Tab on page 171
- Transport Tab on page 172
- Schema Tab on page 176

Configuration Tab

Name

You can use the default service name or replace it.

- A service name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used.
- A service name cannot use global variables.

Description

Provide information about the adapter service that you want stored in the project. The field is optional.

Transport Type

Select the transport to be used by the run-time adapter, JMS or Rendezvous. After selecting the transport, the transport-specific configuration fields display.

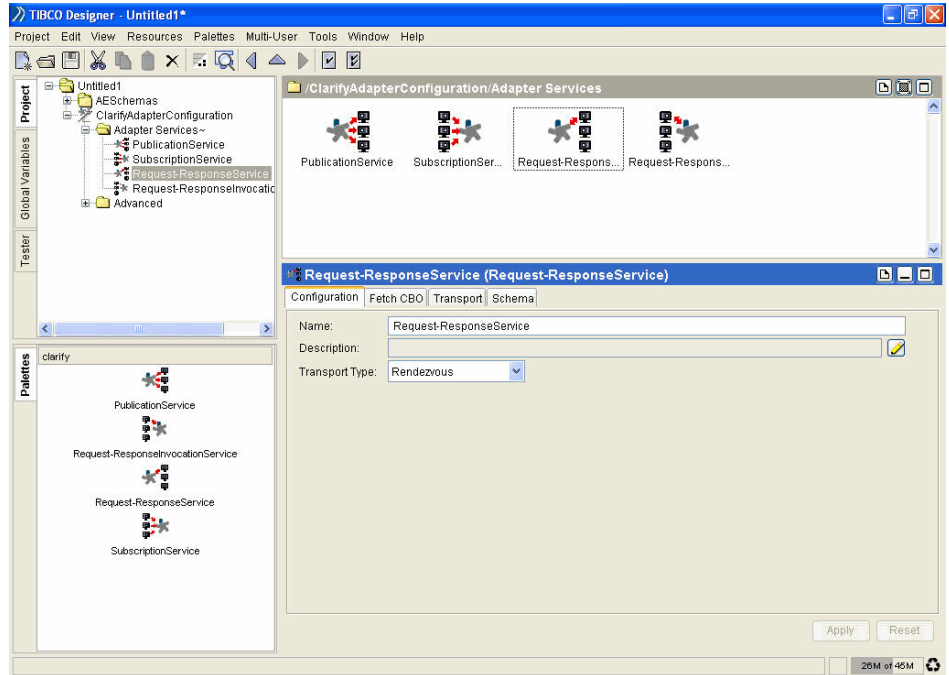
The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field, which uses a dialog to perform SSL configuration.

To enable and configure SSL, in the **Project** panel, expand the **Advanced** folder, then expand the **Sessions** folder. Select the TIBCO Rendezvous session or JMS topic and click **Use SSL?**. The SSL configuration options are explained in the online help associated with the session dialog.

Sample Request-Response Service Configuration Tab Screen

The following screen shows a Request-Response Service with configuration options set.

Figure 44 Request-Response Service Configuration tab



Fetch CBO Tab

You can set schema parameters for the Request-Response Service using the Fetch CBO tab:

To select a schema, do one of the following,

- Type the schema name in the text box adjacent to the **Fetch CBO List...** button and click the button.
- Click the **Fetch CBO List...** button and select the schema case from the list that is displayed.

Click **OK**. The case CBO is loaded to the Request-Response Service.



In case of a change in schema in ClarifyCRM system, the schema needs to be fetched and loaded again.

Transport Tab

Message Transport options can be set for the Request-Response Service depending on the transport type selected in the Configuration tab.

Options Displayed with Rendezvous as the Transport

Message Subject

Specify the subject name to be used by default.

By default a service uses a message subject that 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 the values for `Domain` and `Deployment` are not empty. You can type a TIBCO Rendezvous subject name different from the default in this field. See *TIBCO Rendezvous Concepts* for details.

Quality of Service

— Certified

Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the message's time limit expires. This is called certified message delivery.

If certified message delivery is used, data is stored in a ledger file. The size of the ledger depends on several factors, the most important of which is the retention rate of stored data. That is, the ledger grows fastest in response to the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

— Reliable

Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated.

— Distributed Queue

Indicates load balancing should be enabled. An RVCMQ Session allows applications to use distributed queues for certified delivery, to any number of listeners using queuing member sessions that act together to process inbound task messages.

See *TIBCO Rendezvous Concepts* for details about quality of service.

Wire Format

The format in which messages are to be published.

For Rendezvous mode, only the `ActiveEnterprise Message` wire format is available.

— `ActiveEnterprise Message`

Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscriber.

`ActiveEnterprise` standard wire format provides class information and packing rules for the TIBCO Adapter SDK set of data types. This format allows `ActiveEnterprise` components to perform extra validation on messages sent or received.

See the *TIBCO Adapter SDK Programmer's Guide* for details.

Session Reference

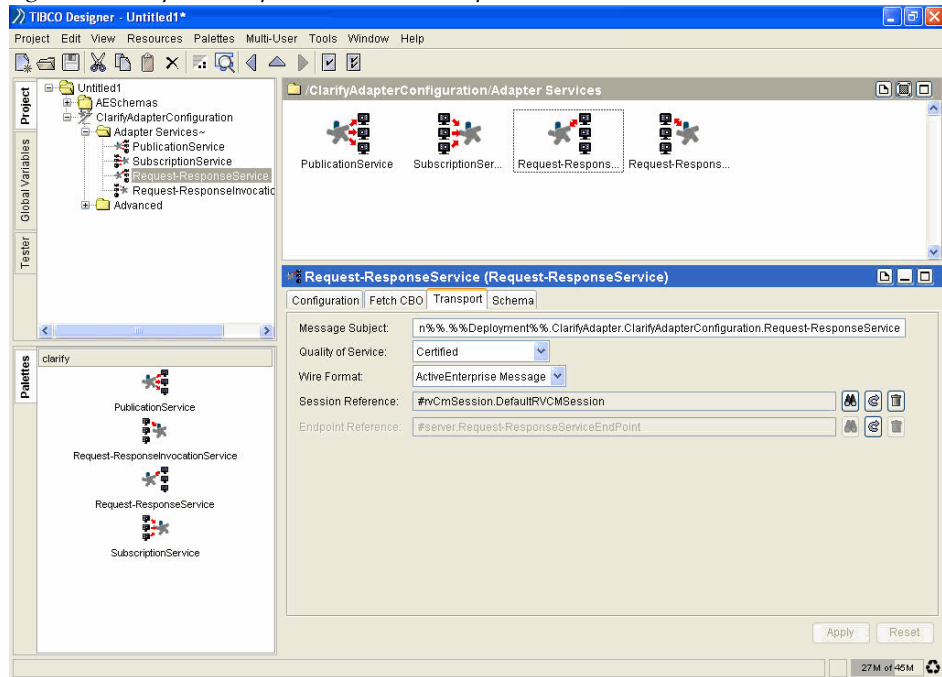
When you create a service, TIBCO Designer creates a corresponding session resource in the `Sessions` folder of the `Advanced` folder and displays it in this field.

If you have explicitly created a custom session of the same type, you can click the **Browse resources...** button to replace the autogenerated session. It is recommended not to change the session for a service.

Endpoint Reference

An endpoint reference for the service. This is a disabled field and points to the corresponding endpoint resource in the `Sessions` folder of the `Advanced` folder. The endpoint resource is automatically created by TIBCO Designer.

Figure 45 Request-Response Service Transport tab with TIBCO Rendezvous



Options Displayed with JMS as the Transport

Destination

By default, a service uses a dynamic destination that 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 dynamic destination, make sure the values for Domain and Deployment are not empty.

You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the EMS server before it can be used by the run-time adapter.

See the *TIBCO Enterprise Message Service User's Guide* for more information.

Wire Format

Services must use the same wire format to exchange data.

For JMS mode of transport, only one wire format is available.

- The XML Message wire format conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the ActiveEnterprise schema.

Connection Factory Type

- Topic

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.

- Queue

A message sent to a queue is consumed by one and only one receiver. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver to access the queue receives the message. The other receivers do not. This messaging model is known as point-to-point.

Delivery Mode

An adapter Request-Response Service can be durable or non-durable.

- Durable

Indicates that the service is registered with the EMS server. Messages sent to a durable Request-Response Service are held by the EMS server until they are consumed by the service. The service can be down and expect to receive its messages when it comes back up.

- Non-Durable

Indicates that the service is not registered with the EMS server. Messages sent to a non-durable Request-Response Service are not held by the EMS server. If the service is down, it will not receive the messages that arrived at the EMS server while the service was down.

See the *TIBCO Enterprise Message Service User's Guide* for more information.

Session Reference

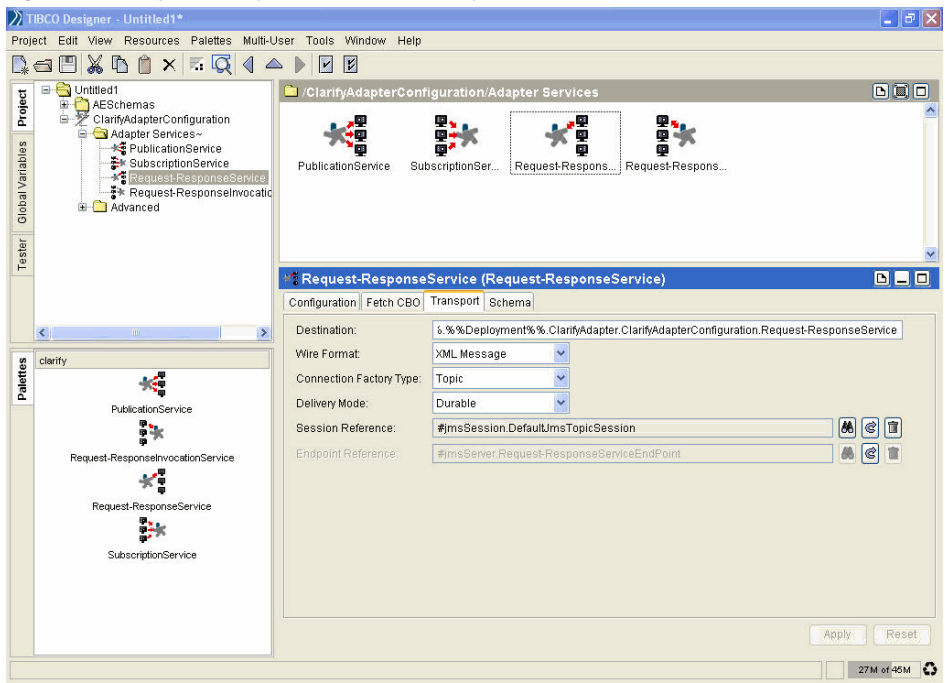
When you create a service, TIBCO Designer creates a corresponding session resource in the Sessions folder of the Advanced folder and displays it in this field.

If you have explicitly created a custom session of the same type, you can click the **Browse resources...** button to replace the autogenerated session. It is recommended not to change the session for a service.

Endpoint Reference

An endpoint reference for the service. This is a disabled field and points to the corresponding endpoint resource in the Sessions folder of the Advanced folder. The endpoint resource is automatically created by TIBCO Designer.

Figure 46 Request-Response Service Transport tab with JMS



Schema Tab

The Schema tab can be used to configure schema for the Request-Response Service.

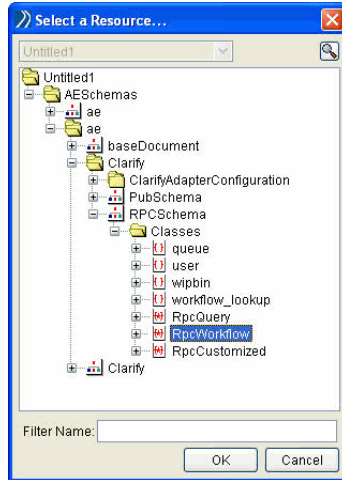
Class Reference

The schema class reference points to the class created for this service. By default, the class reference is set to RpcQuery. It can be changed to RpcCustomized or RpcWorkflow.

Changing Schema Reference

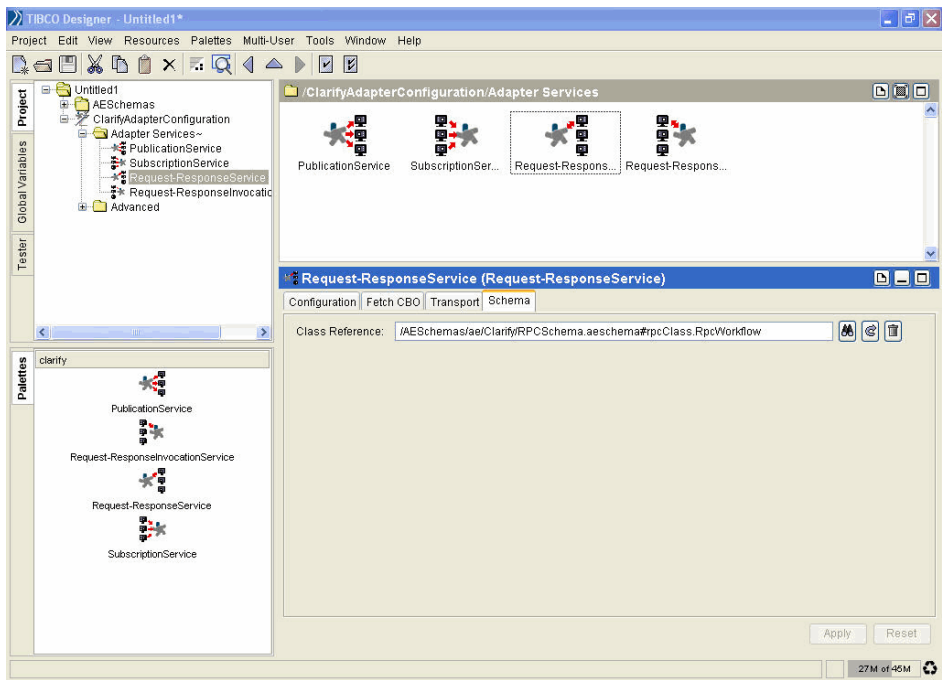
1. Click the **Browse resources...** button. Browse and map the service to an operation using the **Select a Resource** dialog.

Figure 47 Change the schema reference



- 2. Click **OK** and then click **Apply**. A screen shot of the schema mapped to a workflow operation for the Request-Response Service is given next.

Figure 48 Schema mapped to a workflow operation for the Request-Response Service



Request-Response Invocation Service Fields

The following tabs are available:

- Configuration Tab on page 179
- Fetch Request CBO Tab on page 180
- Fetch Reply CBO Tab on page 180
- SubscriberOptions Tab on page 181
- Transport Tab on page 182
- Schema Tab on page 187

Configuration Tab

Name

You can use the default service name or replace it.

- A service name must use alphanumeric characters. An underscore (_) character can be used. The entire instance name must be less than 80 characters. The space character cannot be used.
- A service name cannot use global variables.

Description

Provide information about the adapter service that you want stored in the project. The field is optional.

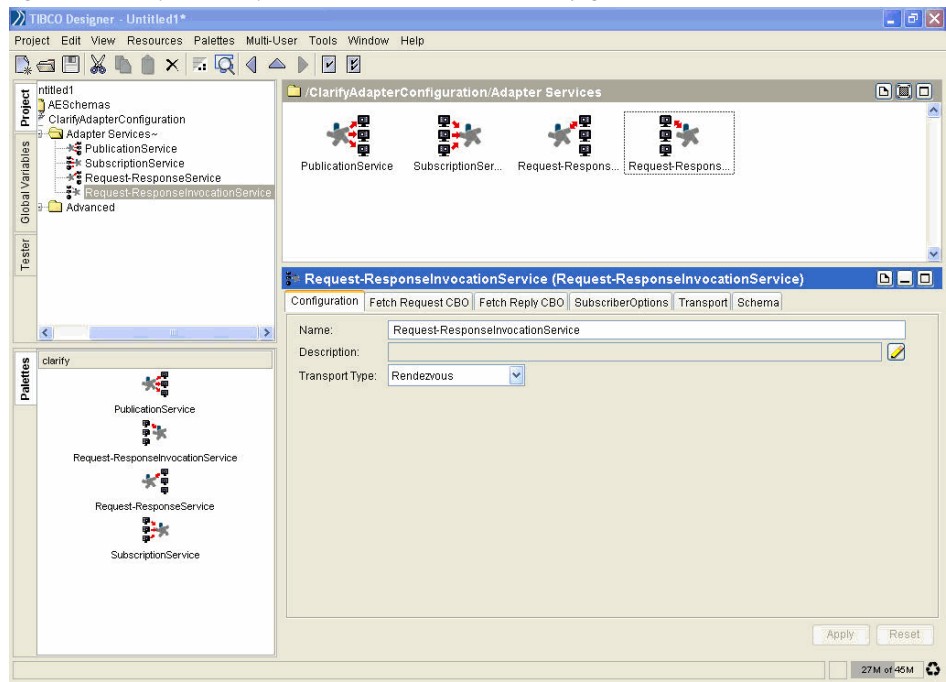
Transport Type

Select the transport to be used by the run-time adapter, JMS or Rendezvous. After selecting the transport, the transport-specific configuration fields display.

The transport can be configured to use a trusted store and identity resource for use in SSL (Secure Sockets Layer) configurations. TIBCO Rendezvous sessions and JMS topics have an SSL configuration field, which uses a dialog to perform SSL configuration.

To enable and configure SSL, in the Project panel, expand the Advanced folder, then expand the Sessions folder. Select the TIBCO Rendezvous session or JMS topic and click Use SSL?. The SSL configuration options are explained in the online help associated with the session dialog.

Figure 49 Request-Response Invocation Service Configuration tab



Fetch Request CBO Tab

You can set request schema parameters for the Request-Response Invocation Service using the Fetch Request CBO tab:

To select a request schema, do one of the following,

- Type the schema name in the text box adjacent to the **Fetch CBO List...** button and click the button.
- Click the **Fetch CBO List...** button and select the schema case from the list that is displayed.

Click **OK**.



In case of a change in schema in ClarifyCRM system, the schema needs to be fetched and loaded again.

Fetch Reply CBO Tab

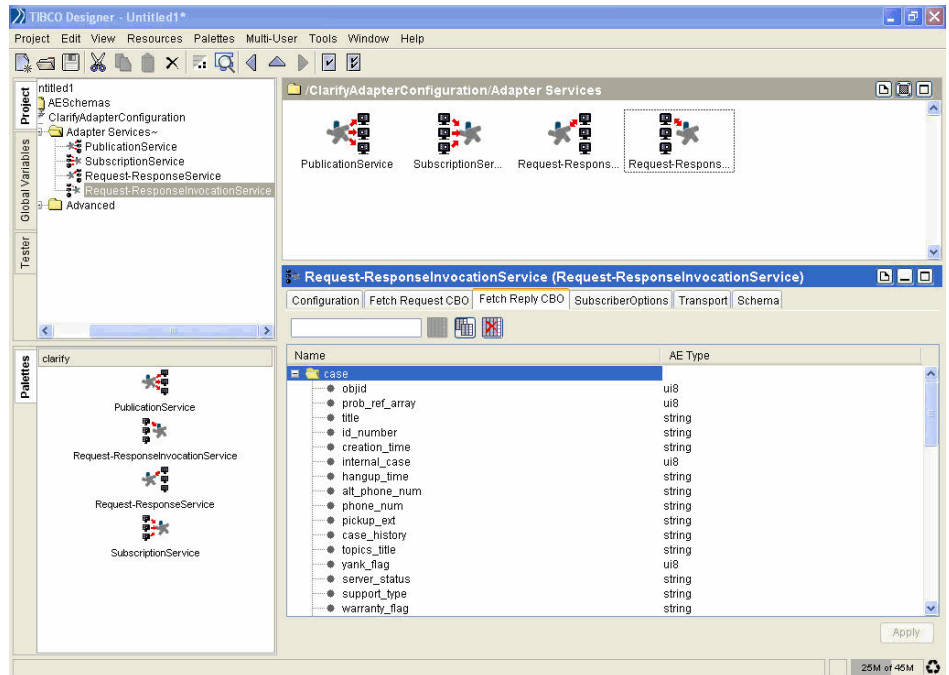
You can set reply schema parameters for the Request-Response Invocation Service using the Fetch Reply CBO tab.

To select a reply schema, do one of the following:

- Type the schema name in the text box adjacent to the **Fetch CBO List...** button and click the button.
- Click the **Fetch CBO List...** button and select the schema case from the list that is displayed.

Click **OK**.

Figure 50 Reply CBO loaded to Request-Response Invocation Service



In case of a change in schema in ClarifyCRM system, the schema needs to be fetched and loaded again.

SubscriberOptions Tab

The SubscriberOptions tab is used to define the various parameters involved in the communication between the adapter and the ClarifyCRM client, such as the Rendezvous subject name. The adapter uses these parameters to create Subscription Services dynamically at run time.

Subscriber Subject

Specify the subject for the Subscription Service to be associated with this Request-Response Invocation Service.

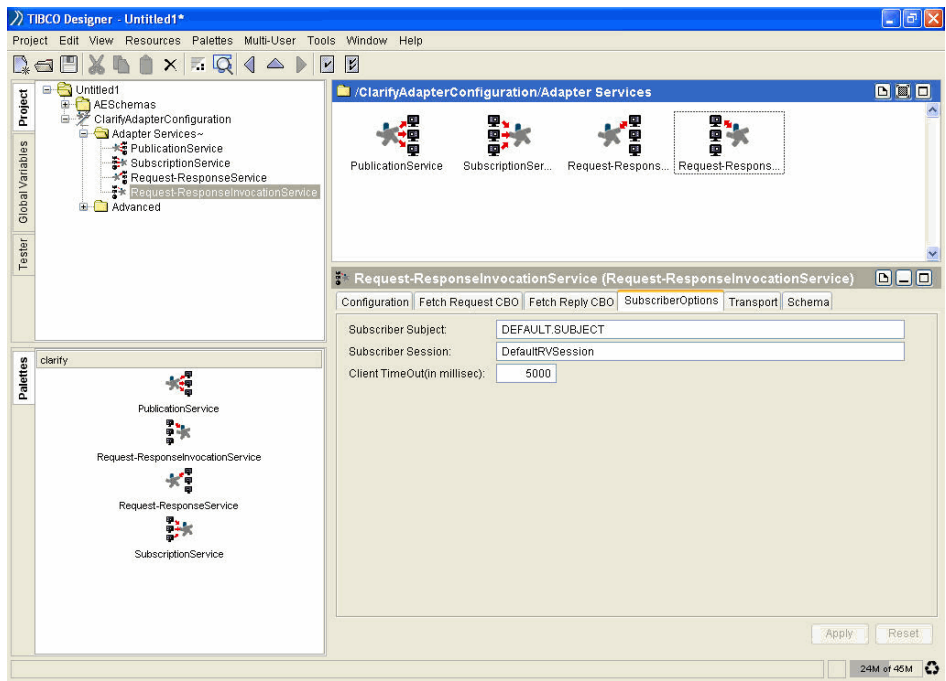
Subscriber Session

Specify the name of the default Rendezvous session (DefaultRVSession) that is created when the adapter configuration is saved. To use another Rendezvous session, create the session first and specify the name of that Rendezvous session.

Client TimeOut (in millisec)

Specify the time interval after which the request times out. The time interval that the client has to wait for the response from the external RPC server.

Figure 51 Request-Response Invocation Service with defined SubscriberOptions



Transport Tab

Message Transport options can be set for the Request-Response Invocation Service depending on the transport type selected in the Configuration tab.

Options Displayed with Rendezvous as the Transport

Message Subject

Specify the subject name to be used by default.

By default, a service uses a message subject that 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 the values for `Domain` and `Deployment` are not empty.

You can type a TIBCO Rendezvous subject name different from the default in this field. See *TIBCO Rendezvous Concepts* for more information.

Quality of Service

— Certified

Guarantees that every certified message reaches its intended recipient in the order sent. The message can be sent across network boundaries, and if a network fails, delivery attempts continue until delivery succeeds or until the message's time limit expires. This is called certified message delivery.

If certified message delivery is used, data is stored in a ledger file. The size of the ledger depends on several factors, the most important of which is the retention rate of stored data. That is, the ledger grows fastest in response to the cumulative length of undeliverable messages. You must ensure that sufficient disk space is available for the expected size of the ledger.

— Reliable

Ensures that each multicast or broadcast message is received as long as the physical network and packet recipients are working, and that the loss of a message is detected. This choice can compensate for brief network failures because it can retransmit a message on request if the first attempt failed. This choice is appropriate when message delivery is expected but some loss can be tolerated.

Wire Format

The format in which messages are to be published.

For Rendezvous mode, only the `ActiveEnterprise Message` wire format is available.

— ActiveEnterprise Message

Control information for validation is sent in the message. If no control information is included, an exception is returned to the subscriber.

ActiveEnterprise standard wire format provides class information and packing rules for the TIBCO Adapter SDK set of data types. This format allows ActiveEnterprise components to perform extra validation on messages sent or received.

See the *TIBCO Adapter SDK Programmer's Guide* for details.

Session Reference

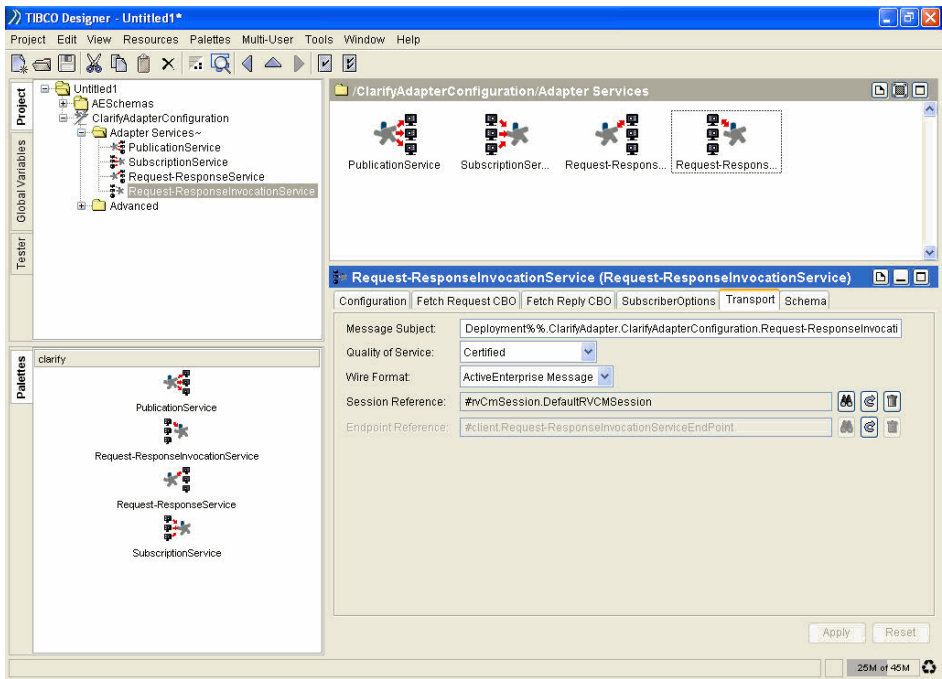
When you create a service, TIBCO Designer creates a corresponding session resource in the Sessions folder of the Advanced folder and displays it in this field.

If you have explicitly created a custom session of the same type, you can click the **Browse resources...** button to replace the autogenerated session. It is recommended not to change the session for a service.

Endpoint Reference

This is a disabled field and points to the corresponding endpoint resource in the Sessions folder of the Advanced folder. The endpoint resource is automatically created by TIBCO Designer.

Figure 52 Request-Response Invocation Service Transport tab with TIBCO Rendezvous



Options Displayed with JMS as the Transport

Destination

By default, a service uses a dynamic destination generated using the `Domain` and `Deployment` global variables, the adapter acronym, the adapter instance name and the service name. To use this default dynamic destination, make sure the values for `Domain` and `Deployment` are not empty.

You can override the default dynamic destination by specifying the static destination in this field. The static destination must be defined on the EMS server before it can be used by the run-time adapter.

See the *TIBCO Enterprise Message Service User's Guide* for more information.

Wire Format

The format in which messages are to be published.

For JMS mode, only the XML Message wire format is available.

- The XML Message wire format conforms to specifically constructed and fully compliant XML Schema (XSD) based on the existing definition of the ActiveEnterprise schema.

Connection Factory Type

- Topic

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.

- Queue

A message sent to a queue is consumed by one and only one receiver. Each message has only one receiver though multiple receivers may connect to the queue. The first receiver to access the queue receives the message. The other receivers do not. This messaging model is known as point-to-point.

Delivery Mode

- Persistent

In general, a message marked persistent will be available to a JMS client even if the EMS server goes down.

- Non-Persistent

A message marked non-persistent will not be available to a JMS client if the EMS server goes down.

Messages sent with the persistent delivery mode are always written to persistent storage, except when they are published to a topic that has no durable subscribers. When a topic has no durable subscribers, there are no subscribers that need messages resent in the event of a server failure. Therefore, messages do not need to be saved, and performance is improved because disk I/O is not required.

See the *TIBCO Enterprise Message Service User's Guide* for more information.

Session Reference

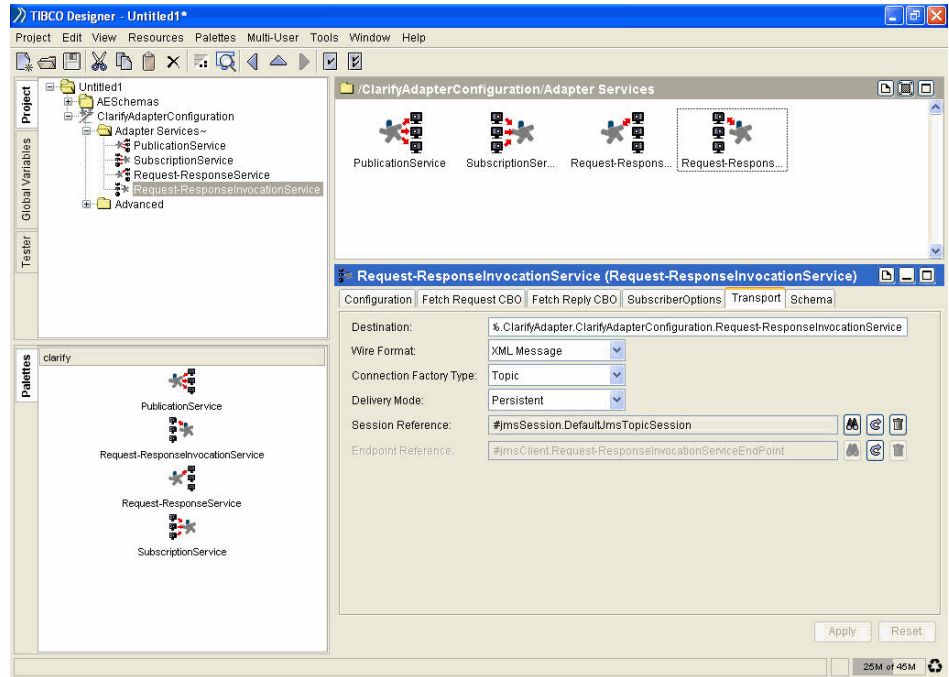
When you create a service, TIBCO Designer creates a corresponding session resource in the `Sessions` folder of the `Advanced` folder and displays it in this field.

If you have explicitly created a custom session of the same type, you can click the **Browse resources...** button to replace the autogenerated session. It is recommended not to change the session for a service.

Endpoint Reference

This is a disabled field and points to the corresponding endpoint resource in the `Sessions` folder of the `Advanced` folder. The endpoint resource is automatically created by TIBCO Designer.

Figure 53 Request-Response Invocation Service Transport tab with JMS



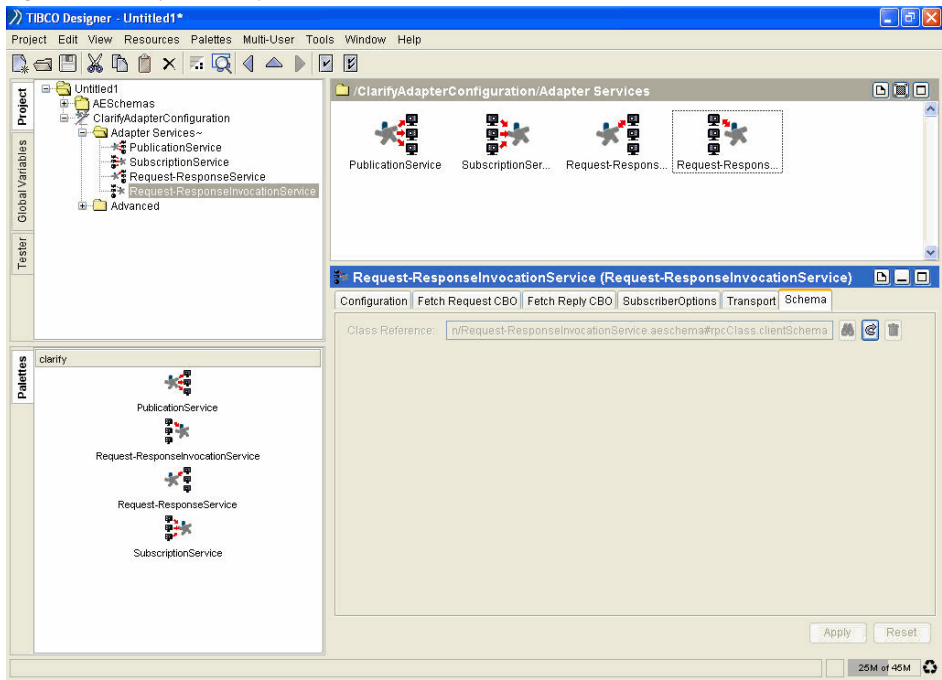
Schema Tab

The schema configured for the Request-Response Invocation Service can be viewed using the Schema tab.

Class Reference

The schema class reference points to the class created for this service. This is a disabled field. You can only view the schema.

Figure 54 Request-Response Invocation Service Schema tab



Chapter 9

Deploying the Adapter Using TIBCO Administrator Enterprise Edition



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 190*
- *Deploy the Project, page 191*
- *Start or Stop the Adapter, page 194*
- *Monitor the Adapter, page 195*

Create an EAR File in TIBCO Designer

The Enterprise Archive file (EAR) contains information about the adapter services you wish to deploy. This could be one or more adapters, one or more TIBCO ActiveMatrix 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 instance.
2. Drag 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. 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 installed with 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 backend application and fetch metadata. At run-time, the adapter uses a password to connect to the back-end application and interoperate with it.

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` or any intermediate version.

Predefined Properties

Table 31 describes predefined properties. Properties that start with `ntservice` are available only on Microsoft Windows platforms.



All paths inside a properties file, including Windows directory names, must use forward slashes.

Table 31 Predefined Properties

Property	Description
<code>tibco.repourl</code>	Identifies the absolute pathname to the Designer project where the adapter configuration is defined.
<code>tibco.configurl</code>	<p>Specifies the location of the adapter configuration inside the project file.</p> <ul style="list-style-type: none">• If a relative path is specified, the adapter service is assumed to be under the default area in the project: <code>/tibco/private/adapter/</code>• If an absolute path is specified, the adapter configuration is looked up in the project as defined by the argument.
<code>tibco.username</code> <code>tibco.password</code>	The user name and password used by the repository server to access the project.
<code>tibco.clientVar</code>	<p>Specifies run-time values to substitute for global variables defined in the project. This value takes precedence over the named global value set in the project. Substitution takes place only at start up.</p> <p>You append the global variable to <code>tibco.clientVar</code>, then give its value. For example, a global variable named <code>DirLedger</code> is specified as follows:</p> <pre>tibco.clientVar.DirLedger TIBCO_HOME/adapter/adclycrm/<version_num>/ledger</pre>
<code>adclycrm.perfMon</code> <on/off>	Turns the performance statistics microagent on or off.
<code>ntservice.name</code>	<p>Name for this Windows Service.</p> <p>This property is useful if you wish to have multiple Windows Services for the same executable. For example, to have two adapters running on the same machine. By specifying different service names and display names for the adapters, you can accomplish this.</p>
<code>ntservice.displayname</code>	<p>Name to display in the Services control for this Windows Service.</p> <p>This property is useful if you wish to have multiple Windows Services for the same executable. For example, to have two adapters running on the same machine. By specifying different service names and display names for the adapters, you can accomplish this.</p>

Table 31 Predefined Properties (Cont'd)

Property	Description
<code>ntservice.starttype</code>	<p>Start type for this Windows Service. Either manual or automatic. For example:</p> <pre>ntservice.starttype automatic</pre> <p>Use this property to initially set the start type for the service. Once the service is installed, use the Windows Services control to change the start type of services.</p>
<code>ntservice.binary.path.absolute</code>	<p>Absolute path to the executable that is run when the service is started. For example:</p> <pre>ntservice.binary.path.absolute C:/tibco/adapters/adclycrm/<version_num>/bin/adclycrm.exe</pre>
<code>ntservice.interactive</code>	<p>Specifies whether the Windows Service is interactive. Either true or false.</p> <pre>ntservice.interactive=true</pre>
<code>ntservice.account</code>	<p>Username to run the Windows Service.</p> <p>Use this property to initially set the account for the service. Once the service is installed, use the Services control to change the user account of services.</p>
<code>ntservice.password</code>	<p>Password for the username in the <code>ntservice.account</code> property.</p> <p>Use this property to initially set the password for the user account. Once the service is installed, use the Services control to change the password.</p>

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 the adapter 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 check box next to the adapter service.
3. Click the **Start Selected** button.

The status changes from `Stopped` to `Starting` up to `Started`.

To stop the adapter:

1. 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 10 **Advanced Topics**

This chapter describes advanced configuration options available in the palette.

Topics

- *Using the Adapter with a Revision Control System, page 198*
- *Modifying the DTA Rendezvous Connection Properties, page 201*
- *Defining a TIBCO Hawk Session, page 202*
- *Using Global Variables, page 203*
- *Configuring a Remote Adapter, page 211*
- *Using the Tracking Element, page 212*

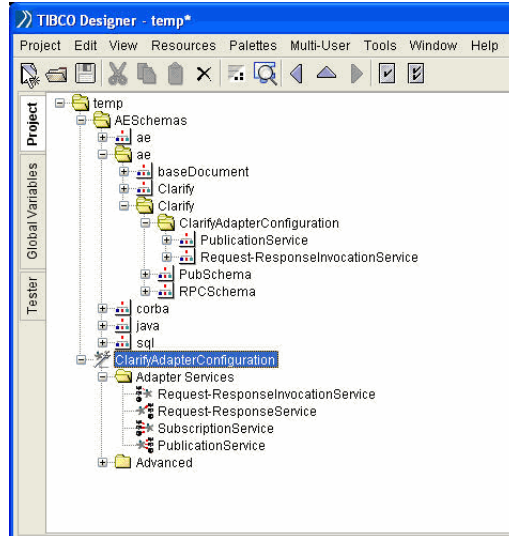
Using the Adapter with a Revision Control System

TIBCO Designer supports revision control systems such as Microsoft Visual SourceSafe and Perforce. To use a revision control system, you must manually add some configured resources to the revision control system and check in the resources when completing the instance configuration.

As part of instance configuration, the adapter creates certain schema files. For example, when you drag an instance `Instance1`, the following files and folders are created:

```
Project_root/AESchemas/ae/Clarify.aeschema
Project_root/AESchemas/ae/Clarify/Instance1
```

Figure 55 Schema files created by the adapter



Next, as a part of the service configuration, the adapter creates schema files in:

```
Project_root/AESchemas/ae/Clarify/Instance1
```

For example, when you configure a Subscription service, `Sub1`, the following file is created:

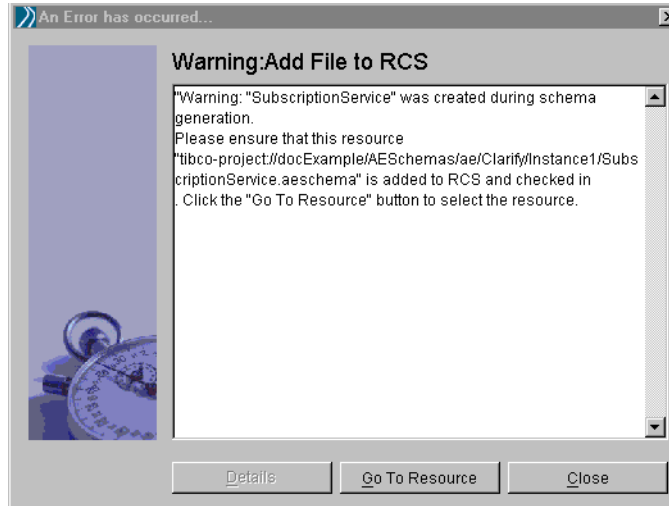
```
Project_root/AESchemas/ae/Clarify/Instance1/Sub1.aeschema
```

Certain sequences and classes are added in:

```
Project_root/AESchemas/ae/Clarify.aeschema
```


When the project is saved and a revision control system specified, the adapter displays a warning that additional files were created and should be added to the revision control system. This warning appears only when the files are created for the first time.

Figure 56 Add additional files to revision control system



The warning displays a **Go To Resource** button that helps in navigating to the resource.

You should use the `Multi-User>Add Resources to RCS` menu command to add these files to the revision control system. For information about how to use the Multi-User feature, 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.

Regeneration When Moving, Copying and Pasting

- Default subjects are not regenerated to reflect the new instance name when a service is moved.
- Manually changed certified messaging and certified messaging queue ledger file names are regenerated to defaults when a service is moved, or copied and pasted to a new instance.
- If a service associated with a custom session is moved, or copied and pasted, the custom session is not moved, or copied and pasted. The session is regenerated as a default session.

Creating Two Adapter Instances in Same Project by Different Users

Use the following steps as guidelines when creating two adapter instances in the same project by two users:

1. Create a new project, `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 yes 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, you cannot configure the instances.

Modifying the DTA Rendezvous Connection Properties

You can modify the TIBCO Rendezvous connection properties used by the design-time adapter in the `adclycrmDTA.tra` properties file or in the `adclycrmDTA.dat` project. Both files are located in the `bin` directory.

Modifying the Properties File

The `adclycrmDTA.tra` properties file contains properties that specify, for example, the pathname to the project where configuration information for the design-time adapter is defined, the location of the directories where trace files and the certified messaging ledger files are stored, and so on. Properties specified in the `adclycrmDTA.tra` properties file overwrite the same properties specified in the project.

To specify custom Rendezvous connection properties for the design-time adapter:

1. Change directory to the `bin` directory. For example:
2. Open the `adclycrmDTA.tra` properties file with a text editor.
3. Add or edit the following properties:

```
tibco.clientVar.dtaRvDaemon=<value>
tibco.clientVar.dtaRvNetwork=<value>
tibco.clientVar.dtaRvHost=<value>
```

where `<value>` is a valid TIBCO Rendezvous daemon, network or host value. See the TIBCO Rendezvous documentation for correct values to use.

4. Restart the design-time adapter to make these modifications take effect.

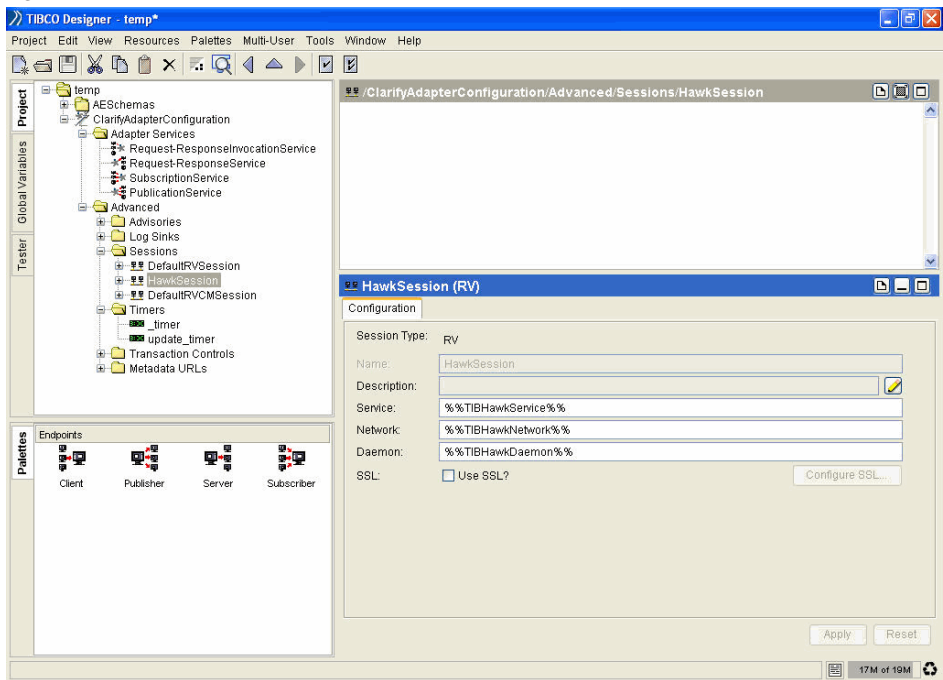
Defining a TIBCO Hawk Session

A default TIBCO Rendezvous session, `HawkSession`, is defined in the project whenever a new instance of the adapter is configured. You can use this session to monitor the adapter using TIBCO Hawk.

To modify the parameters of `HawkSession`:

1. In the project tree panel, click the **ClarifyAdapterConfiguration** button defined for your adapter instance.
2. Select the **Advanced>Sessions>HawkSession**.
3. The default values in the Service, Network and Daemon fields are 7474, None and tcp:7474. Change the default values if required and click **Apply**.

Figure 57 Hawk session



Using Global Variables

The variable substitution mechanism can override global variables predefined in the project in a restricted manner. Predefined variables can be viewed and set in TIBCO Designer. Variables are specified as %%VARNAME%% and contain no white space.

Variable substitution allows you to:

- Substitute string variables specified in the project at startup time.
- Locally define the value for a variable for a specific project. The local value takes precedence over any global value.
- Specify the value for a variable in a properties file. This overrides the project repository and values set in code, but not variables set on the command line.
- Enforce the pre-defined variables listed in Predefined Global Variables on page 205.

Variables can be used anywhere in the configuration and will be replaced by the locally-defined adapter instance.

Specifying Global Variables Using TIBCO Designer

Global variables provide an easy way to set defaults for use throughout your project. There are several ways to use them:

- Define a variable using TIBCO Designer, then override the value for individual applications at deployment time using TIBCO Administrator. You can also override values for predefined variables, unless the GUI does not allow you to set them later.
- Predefine a variable using TIBCO Designer, then override the value for individual services (for example, Publication Service or TIBCO ActiveMatrix BusinessWorks process) at deployment time using TIBCO Administrator. The values are then used at runtime. You can also override values for predefined variables, unless the GUI does not allow you to set them later.

For example, you could assign the value 7474 to the predefined global variable `RvDaemon`. Then use the variable in different sessions in your adapter. To change the TIBCO Rendezvous daemon for your adapter, you can globally set it to a different value or override it from the command line.

Using Global Variables in the Project

To use global variables in your project:

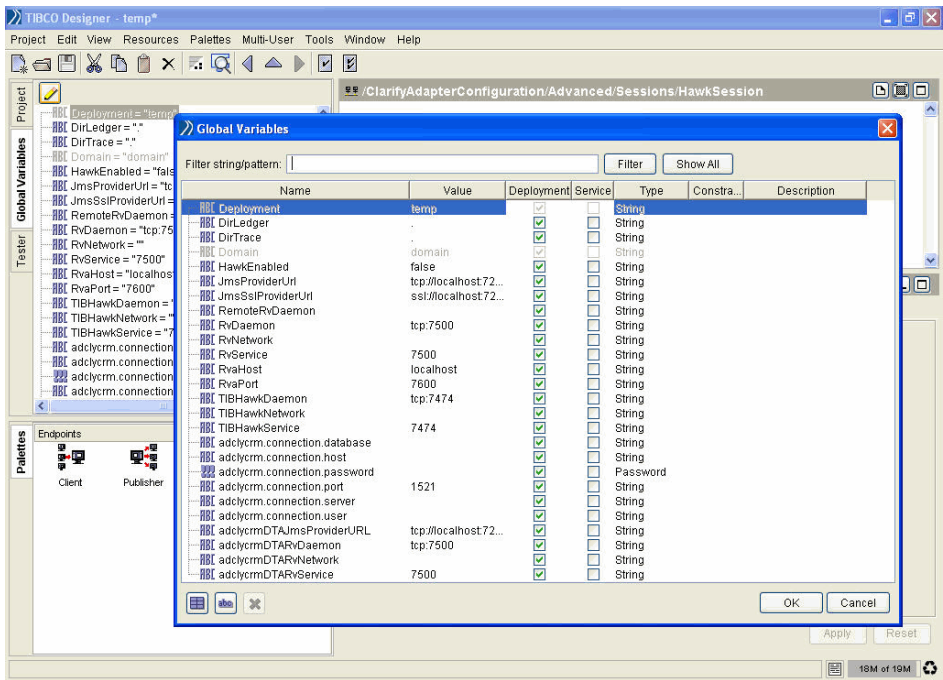
- 1. In the project panel, select the Global Variables tab.

The project panel is updated to display all currently defined global variables. Click **Open Advanced Editor** (pencil button at the top left corner). You now have these choices:

- To assign or change a variable value, select that region and triple-click the variable. The variable expands so you can change either the variable name or the variable value. Press **Enter** when you are done.
- To add a new global variable group, click the leftmost button, **Add a Variable Group**, at the bottom of the dialog box. Specify the name of the group, then press **Enter**. With the group icon selected, you can click the abc icon to add variables to the group.
- To add a global variable, click the abc button. A new global variable item is added to the bottom of the list. Supply the variable name and, optionally, the value. Press **Enter** when you are done.

The global variable is now displayed in the global variables list.

Figure 58 Global variables



- 2. To use the global variable in the fields of a resource, enter the variable name surrounded by %% on both sides.

When the project is deployed and the configured components are running, all occurrences of the global variable name are replaced with the global variable value (unless it was overridden in a way that had higher precedence).

A number of global variables are predefined. See Predefined Global Variables on page 205 for information. You may add definitions of any variables you need to the predefined variables.

Changing Global Variable Values at Runtime

You can change the value of a global variable when you deploy your project in TIBCO Administrator. See the section on modifying runtime variables in the *TIBCO Administrator User's Guide* for more information.

You can also specify values for global variables when starting a process engine on the command line. To do this, specify the following as a command line argument when starting the process engine:

```
-tibco.clientVar.<variablePathAndName> <value>
```

where *variablePathAndName* is the name of the variable you wish to set, including the path to the variable if it is contained in a folder. *value* is the value you wish to set the variable to.

For example, if you have a global variable named `item1` contained in a folder named `myGroup` and you wish to set its value to 500, add the following argument to the command line when starting the process engine:

```
-tibco.clientVar.myGroup/item1 500
```

Predefined Global Variables

Table 32 lists and explains the predefined global variables. Some global variables are automatically used within the system when an adapter instance is configured.

Table 32 Predefined Global Variables

Variable	Description
Deployment	Defaults to the TIBCO Designer project name. This global variable is used by the system to partially define the subject name defined for a service.
DirLedger	Specifies the path name of the TIBCO Rendezvous certified messaging ledger file. The default is the root installation directory.
DirTrace	Specifies the path name for log file used by the adapter. The default is the root installation directory.

Table 32 Predefined Global Variables (Cont'd)

Variable	Description
Domain	The default value for file-based local projects is <code>MyDomain</code> . The value for server-based projects is the domain to which the project was saved.
HawkEnabled	Indicates whether TIBCO Hawk is used to monitor the adapter. <code>True</code> indicates that a Hawk microagent is defined for the adapter. <code>False</code> indicates the microagent is not to be used.
JmsProviderUrl	Specifies where the JMS server is located. Setting this value mostly makes sense in early stages of a project, when only one JMS daemon is used.
JmsSslProviderUrl	Specifies where the JMS SSL daemon is located.
RemoteRvDaemon	TIBCO Rendezvous routing daemon (<code>rvrd</code>) to be used. See <i>TIBCO Administrator Server Configuration Guide</i> for details about setting up a domain using <code>rvrd</code> .
RvDaemon	TIBCO Rendezvous daemon. Sessions use this daemon to establish communication. The default value is <code>7500</code> .
RvNetwork	<p>TIBCO Rendezvous network. This variable needs only be set on computers with more than one network interface. If specified, the TIBCO Rendezvous daemon uses that network for all outbound messages.</p> <p>In most cases, you can leave the default.</p>
RvService	<p>TIBCO Rendezvous service. The Rendezvous daemon divides the network into logical partitions. Each transport communicates on a single service. A transport can communicate only on the same service with other transports.</p> <p>Unless you are using a non-default TIBCO Rendezvous configuration, you should leave the default (<code>7500</code>).</p>
RvaHost	Computer on which the TIBCO Rendezvous agent runs. This variable is only relevant if you are using the TIBCO Rendezvous Agent (<code>rva</code>) instead of the TIBCO Rendezvous daemon, and if you have configured a non-default setup. See <i>TIBCO Rendezvous Administration</i> for details about specifying the <code>rva</code> parameters.
RvaPort	TCP port where the TIBCO Rendezvous agent (<code>rva</code>) listens for client connection requests. See <i>TIBCO Rendezvous Administration</i> for details about specifying the <code>rva</code> parameters. Defaults to <code>7600</code> .

Table 32 Predefined Global Variables (Cont'd)

Variable	Description
TIBHawkDaemon	TIBCO Rendezvous daemon used in the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details.
TIBHawkNetwork	TIBCO Rendezvous network used by the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details.
TIBHawkService	TIBCO Rendezvous service used by the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details.
adclycrm.connection.database	Used by the system to identify the Clarify database the adapter needs to connect to.
adclycrm.connection.host	Used by the system to identify the Clarify database host name. Used by the Publisher during run-time only.
adclycrm.connection.password	Password to be used by the system to log into the Clarify database.
adclycrm.connection.port	The JDBC port on which the adapter connects to the database.
adclycrm.connection.server	Used by the system to identify the Clarify database server name.
adclycrm.connection.user	User name or User ID to be used by the adapter to log into the Clarify database.
adclycrmDTAJmsProviderURL	Used by the system to identify the JMSProviderURL to connect to the design-time adapter.
adclycrmDTARvDaemon	Used by the system to identify the TIBCO Rendezvous daemon parameter to connect to the design-time adapter. The parameter instructs the transport object about how and where to find the Rendezvous daemon and establish communication. See <i>TIBCO Rendezvous Concepts</i> for details.
adclycrmDTARvNetwork	Used by the system to identify the TIBCO Rendezvous network parameter to connect to the design-time adapter. Every network transport communicates with other transports over a single network interface. On computers with more than one network interface, the network parameter instructs the TIBCO Rendezvous daemon to use a particular network for all outbound messages from this transport. See <i>TIBCO Rendezvous Concepts</i> for details.

Table 32 Predefined Global Variables (Cont'd)

Variable	Description
Domain	The default value for file-based local projects is <code>MyDomain</code> . The value for server-based projects is the domain to which the project was saved.
HawkEnabled	Indicates whether TIBCO Hawk is used to monitor the adapter. <code>True</code> indicates that a Hawk microagent is defined for the adapter. <code>False</code> indicates the microagent is not to be used.
JmsProviderUrl	Specifies where the JMS server is located. Setting this value mostly makes sense in early stages of a project, when only one JMS daemon is used.
JmsSslProviderUrl	Specifies where the JMS SSL daemon is located.
RemoteRvDaemon	TIBCO Rendezvous routing daemon (<code>rvrd</code>) to be used. See <i>TIBCO Administrator Server Configuration Guide</i> for details about setting up a domain using <code>rvrd</code> .
RvDaemon	TIBCO Rendezvous daemon. Sessions use this daemon to establish communication. The default value is <code>7500</code> .
RvNetwork	<p>TIBCO Rendezvous network. This variable needs only be set on computers with more than one network interface. If specified, the TIBCO Rendezvous daemon uses that network for all outbound messages.</p> <p>In most cases, you can leave the default.</p>
RvService	<p>TIBCO Rendezvous service. The Rendezvous daemon divides the network into logical partitions. Each transport communicates on a single service. A transport can communicate only on the same service with other transports.</p> <p>Unless you are using a non-default TIBCO Rendezvous configuration, you should leave the default (<code>7500</code>).</p>
RvaHost	Computer on which the TIBCO Rendezvous agent runs. This variable is only relevant if you are using the TIBCO Rendezvous Agent (<code>rva</code>) instead of the TIBCO Rendezvous daemon, and if you have configured a non-default setup. See <i>TIBCO Rendezvous Administration</i> for details about specifying the <code>rva</code> parameters.
RvaPort	TCP port where the TIBCO Rendezvous agent (<code>rva</code>) listens for client connection requests. See <i>TIBCO Rendezvous Administration</i> for details about specifying the <code>rva</code> parameters. Defaults to <code>7600</code> .

Table 32 Predefined Global Variables (Cont'd)

Variable	Description
TIBHawkDaemon	TIBCO Rendezvous daemon used in the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details.
TIBHawkNetwork	TIBCO Rendezvous network used by the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details.
TIBHawkService	TIBCO Rendezvous service used by the TIBCO Hawk session. See the <i>TIBCO Hawk Installation and Configuration</i> manual for details.
adclycrm.connection.database	Used by the system to identify the Clarify database the adapter needs to connect to.
adclycrm.connection.host	Used by the system to identify the Clarify database host name. Used by the Publisher during run-time only.
adclycrm.connection.password	Password to be used by the system to log into the Clarify database.
adclycrm.connection.port	The JDBC port on which the adapter connects to the database.
adclycrm.connection.server	Used by the system to identify the Clarify database server name.
adclycrm.connection.user	User name or User ID to be used by the adapter to log into the Clarify database.
adclycrmDTAJmsProviderURL	Used by the system to identify the JMSProviderURL to connect to the design-time adapter.
adclycrmDTARvDaemon	Used by the system to identify the TIBCO Rendezvous daemon parameter to connect to the design-time adapter. The parameter instructs the transport object about how and where to find the Rendezvous daemon and establish communication. See <i>TIBCO Rendezvous Concepts</i> for details.
adclycrmDTARvNetwork	Used by the system to identify the TIBCO Rendezvous network parameter to connect to the design-time adapter. Every network transport communicates with other transports over a single network interface. On computers with more than one network interface, the network parameter instructs the TIBCO Rendezvous daemon to use a particular network for all outbound messages from this transport. See <i>TIBCO Rendezvous Concepts</i> for details.

Table 32 Predefined Global Variables (Cont'd)

Variable	Description
adclycrmDTARvService	Used by the system to identify the TIBCO Rendezvous service parameter used to connect to the design-time adapter. The Rendezvous daemon divides the network into logical partitions. Each transport communicates on a single service; a transport can communicate only with other transports on the same service. See <i>TIBCO Rendezvous Concepts</i> for details.

Configuring a Remote Adapter

If only TIBCO Designer is installed on your computer and the adapter is installed on a remote computer, you can still configure the adapter from your computer. To do this:

1. In a command prompt window, copy the two adapter palette files (`adclycRMpalette.dat` and `adclycRMpalette.jar`), located in `TIBCO_HOME/adapter/adclycrm/<version_num>/lib/palettes`, from the remote computer to your computer.
2. Go to the `TIBCO_HOME/Designer/<version_num>/bin` directory and open the `designer.tra` file. Modify the path specified for the property that enables Designer to find the adapter's palette as shown:

```
#application.args -d  
java.property.palettePath  
<install_path>/tibco/adapter/adclycrm/<version_num>/lib/palettes
```
3. Save and close the `designer.tra` file.
4. Open the project in TIBCO Designer. This project must be server based. In the project tree panel, select the **ClarifyAdapterConfiguration** icon and configure it as desired. This icon represents the remote adapter configuration.

Using the Tracking Element

Trace messages contain a tracking element, which is helpful in troubleshooting because it tracks all the components, products, and processes in your installation related to the problem.

The tracking element has the format:

```
tracking=#<trackingID>#<infoItem1>#<infoItem2>...[#<infoItemn>#
```

The tracking ID has the format:

```
tracking=#<IDpart1>--<IDpart2>--<IDpart3>--<IDpart4>#
```

An example of tracking ID is given below:

```
2003 May 15 15:09:38:277 GMT +5 Clarify Info [Adapter]
```

```
AEClarify-0000445 "Message to be published :1"
```

```
tracking=#QdUThbTh8n/Uak9u9Azzw6BUzzw#
```

Tracking Element ID

The trace message above indicates the adapter published a message. The #QdUThbTh8n/Uak9u9Azzw6BUzzw# tracking identifier uniquely identifies the message.

Chapter 11 **Monitoring the Adapter Using TIBCO Hawk**

This chapter explains how to use TIBCO Hawk microagents to monitor and manage the adapter.

Topics

- *Overview, page 214*
- *Starting TIBCO Hawk Software, page 215*
- *The Auto-Discovery Process, page 216*
- *Invoking Microagent Methods, page 217*
- *Available Microagents, page 220*

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 e-mail, paging, running executables
- 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 following 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** — Feeds information back to TIBCO Hawk and exposes 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* for details.

The *TIBCO Hawk Administrator's Guide* explains how to start the TIBCO Hawk Display.

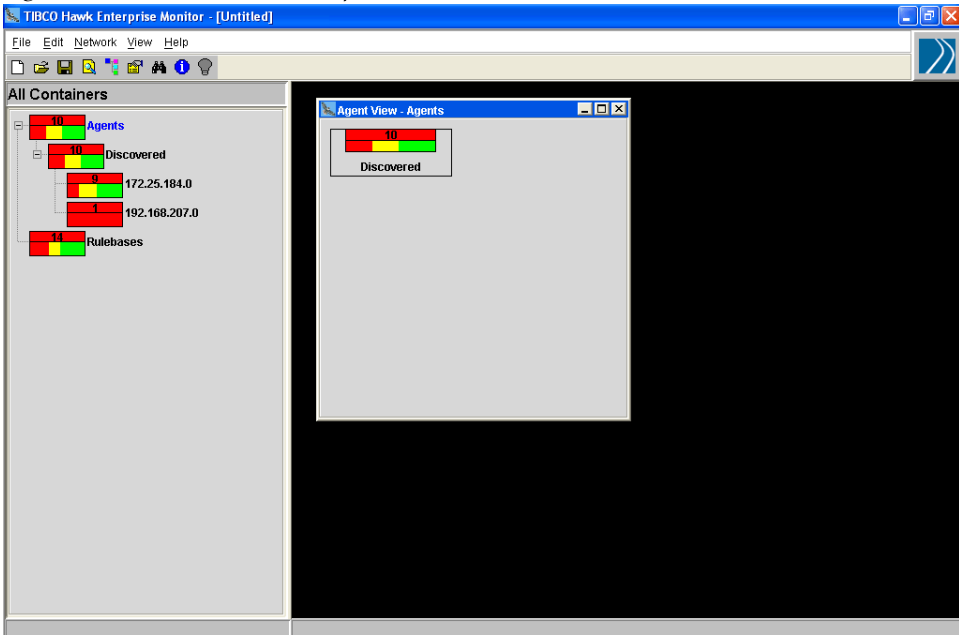
The guides are included in your TIBCO Hawk software installation area.

The Auto-Discovery Process

Once starting 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 59 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 information of icon elements and characteristics, see *TIBCO Hawk Administrator's Guide*.

Invoking Microagent Methods

A set of default microagents, platform-specific and platform-independent, is loaded when a TIBCO Hawk Agent is started. When you install and start the adapter, microagents for the adapter are dynamically added to the local agent.

To invoke a microagent method on a TIBCO Hawk Agent:

1. In TIBCO Hawk Display, right-click the agent icon and select **Get Microagents**.

If TIBCO Hawk security is implemented on your system and you have no access to microagents on this agent, an error dialog displays. Select another agent, or contact your system administrator to obtain access.

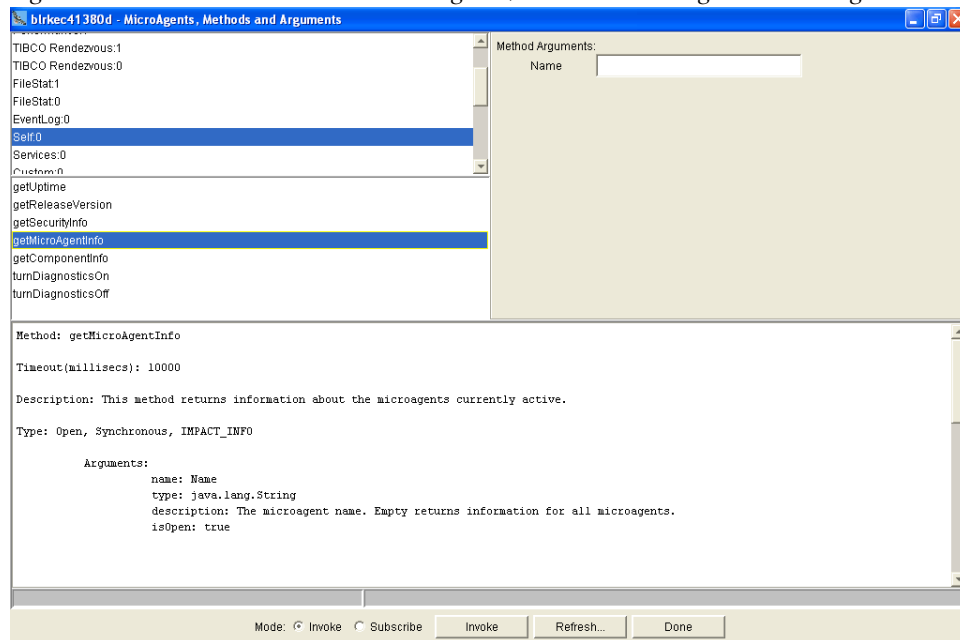
The Microagents, Methods and Arguments dialog displays. The panel on the upper left lists microagents you can access on the current agent. This dialog has two modes, Invoke and Subscribe:

- Invoking a method immediately returns a single set of current results.
- Subscribing provides updates of current results at regular intervals.

Radio buttons at the bottom of the dialog control these modes.

2. Click a microagent name, such as **Self**, to display a list of associated methods and text descriptions in the panels below.
3. Click the name of the method to invoke, such as **getMicroAgentInfo**.

Figure 60 Select a method in the Microagents, Methods and Arguments dialog



If the method accepts arguments, fields for each argument display in the upper right panel. Detailed help text displays in the lower panel.

4. Specify any arguments for the method invocation.
5. Verify that the **Invoke** radio button is selected.
6. Click the **Invoke** button to invoke the selected method.

The Invocation Results dialog displays the results returned by the method.

Figure 61 Invocation results dialog

blirkec41380d - Invocation Results for - getMicroAgentInfo(Name=)

Name	Display Name	Count	Help
COM.TIBCO.hawk.hma.EventLog	EventLog	2	TIBCO Hawk Windows Event Log Microagent
COM.TIBCO.hawk.hma.Process	Process	2	TIBCO Hawk Process Microagent
COM.TIBCO.hawk.hma.FileStat	FileStat	2	TIBCO Hawk File Status Microagent
COM.TIBCO.hawk.hma.TibRendezvous	TIBCO Rendezvous	2	TIBCO Hawk Rendezvous Microagent
COM.TIBCO.hawk.hma.Services	Services	2	TIBCO Hawk Windows Services Microagent
COM.TIBCO.hawk.microagent.Self	Self	1	TIBCO Hawk built-in Microagent
IM Engine - AE Tracing Hawk Sink for imed_...	IM Engine - AE Tracing Hawk Sink for imed_...	1	Provides trace messages from IM Engine
COM.TIBCO.hawk.microagent.Custom	Custom	1	TIBCO Hawk built-in Microagent
IM Engine Administration for imed_debug_e...	IM Engine Administration for imed_debug_e...	1	HAWK agent for the shell
COM.TIBCO.hawk.hma.Registry	Registry	2	TIBCO Hawk Windows Registry Microagent
COM.TIBCO.hawk.hma.Performance	Performance	2	TIBCO Hawk Windows Performance Microa...
IM Engine Monitor for imed_debug_engine1	IM Engine Monitor for imed_debug_engine1	1	Various Engine monitoring funtions
COM.TIBCO.hawk.microagent.Logfile	Logfile	1	TIBCO Hawk built-in Microagent
COM.TIBCO.hawk.microagent.RuleBaseEn...	RuleBaseEngine	1	TIBCO Hawk built-in MicroAgent
COM.TIBCO.hawk.microagent.SysInfo	SysInfo	1	TIBCO Hawk built-in Microagent

Click a cell to display its value in this area!

Done

7. Click **Done** to close the dialog.

These steps describe how to interactively invoke a microagent method and receive a single set of results in TIBCO Hawk Display. You can also use a microagent method as the data source of a TIBCO Hawk rule. Rules automatically receive method results, apply tests to evaluate them, then take action if necessary. For more information on building TIBCO Hawk rules and rule bases, see *TIBCO Hawk Administrator's Guide*.

Available Microagents

The adapter has two microagents, a standard TIBCO Hawk microagent named `COM.TIBCO.ADAPTER.xyz` where `xyz` is the adapter configuration name and a class microagent. The microagents provide:

- Business level statistics. Statistics that report the progress of the adapter as it interacts with the vendor application.

For example, in a database adapter such statistics might indicate whether objects were successfully or unsuccessfully inserted, updated, or deleted in the database.

- Queries that return information about the state of the adapter. This can be an important tool for seeing the internals of an adapter and debugging it if something appears wrong.

For example, methods can return information about threads, internal queues, or connections to the target system. Using these methods, one might be able to identify certain bottlenecks or gauge how successfully an adapter is scaling with respect to the current environment.

- Updates of the adapter run-time parameters. This includes retrieving the current run-time parameters and setting new run-time parameters without restarting the adapter. An example is getting and setting the polling interval. Updating a run-time parameter through the Hawk microagent only affects the setting of the running instance. It does not make a permanent change of the setting in either the project or the `.tra` file.

By default, all microagents, standard and class microagents are available at run-time.

The `perfMon` property value set in the adapter's property file affects the business statistics related methods. If this property is set to `on`, the adapter does all the performance related calculations. When you invoke the methods, if the `perfMon` property is set to `off`, default values are displayed and not the valid values.

Table 33 lists each method available for the adapter and page on which the method is explained.

Table 33 Microagent Methods

Method	Description	Page
<code>activateTraceRole()</code>	Activates a mapping of a role to a sink at runtime.	223
<code>deactivateTraceRole()</code>	Deactivates a mapping of roles to a sinks at runtime.	224

Table 33 Microagent Methods (Cont'd)

Method	Description	Page
<code>getActivityStatistics()</code>	Returns the total number of objects processed for all the schemas.	225
<code>getActivityStatisticsByOperation(Operation)</code>	Returns the total number of objects processed for all the schemas by each service associated with a specified operation.	226
<code>getActivityStatisticsBySchema(SchemaName)</code>	Returns statistics about any activities on a particular object or schema.	227
<code>getActivityStatisticsByService(ServiceName)</code>	Returns statistics about the data handled by a particular adapter service.	228
<code>getAdapterServiceInformation()</code>	Returns information about the services implemented by this adapter.	229
<code>getComponents()</code>	Returns information about the publisher, subscriber and IODescriptor.	230
<code>getConfig()</code>	Returns basic configuration information. More specific information is accessed by the more specific methods.	231
<code>getConfigProperties()</code>	Returns a list of publishers and subscribers.	232
<code>getConnectionStatistics()</code>	Returns the state and statistics for all the current connections used by the adapter.	233
<code>getHostInformation()</code>	Returns standard and extended application information.	234
<code>getPerfMonSetting()</code>	Returns the setting of the perfMon option.	235
<code>getPollingBatchSize()</code>	Returns the polling batch size.	236
<code>getPollingInterval()</code>	Returns the current polling interval setting.	237
<code>getQueueStatistics()</code>	Returns the current count of elements in any internal queue used by the adapter.	238
<code>getRvConfig()</code>	Returns information about all TIBCO Rendezvous sessions defined.	239

Table 33 *Microagent Methods (Cont'd)*

Method	Description	Page
<code>getStatus()</code>	Returns general status information, such as the number of TIBCO Rendezvous messages received and published, the number of errors since the last call, the PID of the application, and more.	240
<code>getThreadStatistics()</code>	Returns the operation counts of current threads.	241
<code>getTraceSinks()</code>	Returns information about sinks to which traces currently go.	242
<code>getVersion()</code>	Returns the instance ID, application name, version, and date for this adapter instance.	243
<code>_onUnsolicitedMsg()</code>	Displays alert messages sent to the current adapter.	244
<code>preRegisterListener()</code>	Preregisters an anticipated listener.	245
<code>resetActivityStatistics()</code>	Resets all the counts for the activity statistics.	246
<code>resetConnectionStatistics()</code>	Resets all the counts for the connection statistics.	247
<code>resetThreadStatistics()</code>	Resets all the counts for the thread statistics.	248
<code>reviewLedger()</code>	Returns information retrieved from the ledger file of a certified messaging session for a publisher adapter.	249
<code>setPollingBatchSize()</code>	Sets the polling batch size.	250
<code>setPollingInterval()</code>	Sets the polling interval for the publication service.	251
<code>setTraceSinks()</code>	Adds a role or changes the file limit of a previously specified sink.	252
<code>stopApplicationInstance()</code>	Stops the running adapter instance.	253
<code>unRegisterListener()</code>	Unregisters a preregistered listener.	254

activateTraceRole()

Activates a mapping of a role to a sink at runtime. This replaces the now deprecated `setTraceSink()` Hawk method.

Table 34 *activateTraceRole()*

Parameters	Type	Description
roleName	string	Name of the role to activate.
sinkName	string	Name of a sink for which to activate the role.

deactivateTraceRole()

Deactivates a mapping roles to a sinks at runtime.

Table 35 deactivateTraceRole()

Parameters	Type	Description
roleName	string	Name of the role to deactivate.
sinkName	string	Name of a sink for which to deactivate the role.

getActivityStatistics()

Returns the total number of objects processed for all the schemas, based on the request type. Also, returns the number of success and error objects.

Table 36 Input parameter of *getActivityStatistics()*

Input Parameter	Type	Description
GetSubTotalBy	string	Indicates how to group the subtotals, by <code>Service</code> or <code>Operation</code> .

Table 37 Returns of *getActivityStatistics()*

Returns	Type	Description
Name	string	Service name or <code>All Services</code> which represents the final tally of all the services.
Total	integer	Total number of objects processed including both success and failures.
Success	integer	Total number of objects successfully processed.
Failure	integer	Total number of objects that caused an error during processing.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

getActivityStatisticsByOperation(Operation)

Returns statistics about one operation.

Table 38 Input parameter of getActivityStatisticsByOperation(Operation)

Input Parameter	Type	Description
Operation	string	Name of the operation.

Table 39 Returns of getActivityStatisticsByOperation(Operation)

Returns	Type	Description
Operation	string	Name of the operation.
Service Name	string	Name of the service.
Total	integer	Total number of objects processed, both success and failures.
Success	integer	Total number of objects successfully processed.
Failure	integer	Total number of objects that caused an error during processing.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.
LineIndex	string	Concatenated string of Service Name and Operation separated by a comma.

getActivityStatisticsBySchema(SchemaName)

Returns the total number of objects processed for the given schema by each service that uses the schema. Also, returns the number of success and error objects.

Table 40 Input parameter of getActivityStatisticsBySchema(SchemaName)

Input Parameter	Type	Description
SchemaName	string	Name of the schema.

Table 41 Returns of getActivityStatisticsBySchema(SchemaName)

Returns	Type	Description
ServiceName	string	Name of the service associated with the specified schema.
Total	string	Total number of objects processed for this schema for a Publication service. Total number of objects received for this schema for a Subscription service.
Success	string	The number of objects that were successfully identified for this schema which will be published or written to a file.
Error	string	The number of objects that were identified for this schema but were not published because the header of the schema failed validation for the Publication service, or was written to a file because the schema was not associated with the subscriber for a Subscription service.

getActivityStatisticsByService(ServiceName)

Returns statistics about the data handled by a given adapter service or all adapter services since the time the adapter was started.

Table 42 Input parameter of getActivityStatisticsByService(ServiceName)

Input parameter	Type	Description
ServiceName	string	Name of service to get the statistics for. If no service name is given, performance statistics for all services is returned.

Table 43 Returns of getActivityStatisticsByService(ServiceName)

Returns	Type	Description
ServiceName	string	Service name.
SchemaName	string	Name of top level schema processed by this service.
Operation	string	Type of operation this service provides.
Total	integer	Total number of objects processed, both success and failures.
Success	integer	Total number of objects successfully processed.
Failure	integer	Total number of objects that caused an error during processing.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.
LineIndex	string	Concatenated string of Service Name and Operation separated by a comma.

getAdapterServiceInformation()

Returns information about the services implemented by this adapter. The information is a summary of available adapter services.

Table 44 Input parameter of getAdapterServiceInformation()

Input Parameter	Type	Description
serviceName	string	Name of the service from which to get information. Default is ALL.

Table 45 Returns of getAdapterServiceInformation()

Returns	Type	Description
Line	Integer	Sequential row number.
Name	string	Name of the Service.
Endpoint	string	Name of the endpoint used for this service.
Type	string	Type of the endpoint, for example, Publisher, Subscriber.
Quality of Service	string	Quality of service for the endpoint, for example, RV, RVCM.
Adapter Name	string	Name of the application for this sink.
Session Name	string	Name of the TIBCO Rendezvous session.
Subject	string	Subject defined for this endpoint.
Number of Messages	Integer	Number of messages processed for this endpoint.

getComponents()

Returns information about the currently active TIBCO Hawk components such as publishers, subscribers, or timers.

Table 46 Input parameters of getComponents()

Input Parameters	Type	Description
Component Name	string	Name of the component. Default is all components.
Component Type	string	Any of Publisher, Subscriber, Timer, or IODescriptor. The default value is All.

Table 47 Returns of getComponents()

Returns	Type	Description
Component Name	string	Name of the TIBCO Hawk component.
Instance ID	string	Name of this adapter instance.
Adapter Name	string	Name of the adapter.
Session Name	string	Name of the TIBCO Rendezvous session.
Component Type	string	The name of the TIBCO Adapter SDK class for this TIBCO Hawk component, such as MPublisher, MSubscriber, or MIODescriptorSource. For more information, see your TIBCO Adapter SDK documentation.
Description	string	Information about this TIBCO Hawk component, for example, time interval, signal type, validating publisher (or subscriber) etc.

getConfig()

Retrieves generic configuration information. More specific configuration information is accessed through separate methods.

Table 48 getConfig()

Returns	Type	Description
Instance ID	string	Instance ID of this adapter.
Adapter Name	string	Name of the adapter.
Repository Connection	string	URL of the repository used for adapter configuration.
Configuration URL	string	Location of the adapter repository instance; either a file name or configuration URL.
Command	string	Command line arguments used to start the adapter.

getConfigProperties()

Returns all attributes and elements for the given repository object.

Table 49 Input parameter of getConfigProperties()

Input Parameter	Type	Description
Property	string	Name of the property for which elements (tags) and attributes are desired. For example, agentone/startup. If no value is given, all properties are returned.

Table 50 Returns of getConfigProperties()

Returns	Type	Description
Element Name	string	Repository directory for the property.
Attribute Name	string	Name of the repository object attribute.
Attribute Value	string	Value of the repository object attribute.
Line	integer	Line number in which this property is defined in the repository instance file.

getConnectionStatistics()

Returns the state and statistics for all the current connections used by the adapter.

Table 51 *getConnectionStatistics()*

Returns	Type	Description
Connection ID	string	Unique identification of a particular connection.
Connection Type	string	Type or key that will match this connection to a thread or queue.
State	string	Current state: CONNECTED or DISCONNECTED.
NumRetries	integer	Total number of times this connection had to be reestablished.
TotalNumOperations	integer	Total number of operations processed by this connection since the adapter started.
CurrentNumOperations	integer	Total number of operations processed by this connection since the last reconnection.
NumLostConnections	integer	Total amount of time that this connection has been lost.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

getHostInformation()

Returns standard and extended application information set.

Table 52 *getHostInformation()*

Returns	Type	Description
Name	string	Name of the property.
Value	string	Value of the property.

getPerfMonSetting()

Returns the setting of the perfMon option.

Table 53 *getPerfMonSetting()*

Returns	Type	Description
Setting	string	Value of the perfMon option.

getPollingBatchSize()

Returns the current polling batch size setting.

Table 54 Input parameter of `getPollingBatchSize()`

Input Parameter	Type	Description
ServiceName	string	Name of the service to query.

Table 55 Returns of `getPollingBatchSize()`

Returns	Type	Description
PollingBatchSize	integer	Polling batch size in milliseconds.

getPollingInterval()

Returns the current polling interval setting.

Table 56 *getPollingInterval()*

Returns	Type	Description
PollingInterval	integer	Polling interval in milliseconds.

getQueueStatistics()

Returns the current count of elements in any internal queue used by the adapter. This includes the TIBCO Rendezvous event queues automatically spawned by Rendezvous for each adapter.

Table 57 *getQueueStatistics()*

Returns	Type	Description
QueueID	string	Unique identification of a particular queue.
QueueType	string	Type or key that will match this queue to a thread or connection.
QueueCount	integer	Current number of elements in the queue.
MaxQueueSize	integer	Maximum number of elements in the queue.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

getRvConfig()

Returns information about the TIBCO Rendezvous session defined by this adapter. Information about all currently defined sessions is returned if no `sessionName` is provided.

Table 58 Input parameter of `getRvConfig()`

Input Parameter	Type	Description
Session Name	string	Name of the TIBCO Rendezvous session for which configuration is required (default is <code>all</code>).

Table 59 Returns of `getRvConfig()`

Returns	Type	Description
Instance ID	string	The instance ID of this adapter.
Adapter Name	string	Name of the adapter.
Session Name	string	Name of the session.
Service	string	Service parameter for this session.
Daemon	string	Daemon parameter for this session.
Network	string	Network parameter for this session.
Synchronous	boolean	Returns 1 if this is a synchronous session, 0 otherwise.
Session Type	string	Type of session; one of <code>M_RV</code> , <code>M_RVCM</code> , or <code>M_RVCMQ</code> .
Certified Name	string	Name of this certified session.
Ledger File	string	Ledger file for this certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.
CM Timeout	string	Timeout for this certified messaging session. Returns the empty string for sessions that are not certified messaging sessions.

getStatus()

Retrieves basic status information about the adapter.

This information is fairly limited. Additional methods are provided in getConfig() on page 231 and getRvConfig() on page 239.

Table 60 *getStatus()*

Returns	Type	Description
Instance ID	string	Instance ID for this adapter instance.
Adapter Name	string	Name of the adapter.
Uptime	integer	Number of seconds since startup.
Messages Received	integer	Number of TIBCO Rendezvous messages received.
Messages Sent	integer	Number of TIBCO Rendezvous messages published.
New Errors	integer	Number of errors since the last call to this method.
Total Errors	integer	Total number of errors since startup.
Process ID	integer	Process ID of the application.
Host	string	Name of host machine on which this adapter is running.

getThreadStatistics()

Returns the operation counts of the current threads.

Table 61 *getThreadStatistics()*

Returns	Type	Description
ThreadID	string	Unique identification of a particular thread.
ThreadType	string	Type that tells what part of the adapter this thread belongs to. Valid types include "Publisher", "Subscriber", "RPC", or "Connection".
TaskType	string	One-word description of the tasks this thread processes.
TaskCount	integer	Number of tasks processed by this thread.
MeasurementInterval	integer	Displays the time (in seconds) since last time the adapter was reset, or if never reset, since the adapter started.

getTraceSinks()

Returns information about sinks to which traces currently go.

Table 62 Input parameters of *getTraceSinks()*

Input Parameters	Type	Description
Sink Name	string	Name of the sink for which you need information. If no name is specified, information about all sinks is returned. Default is <code>all</code> .
Role Name	string	Name of the role for which you need information for the specified sink or sinks. Default is <code>all</code> .

Table 63 Returns of *getTraceSinks()*

Returns	Type	Description
Instance ID	string	Name of this adapter instance as a string.
Adapter Name	string	Name of the application for this sink.
Sink Name	string	Name of the sink.
Sink Type	string	Type of this sink. One of <code>fileSink</code> , <code>rvSink</code> , <code>hawkSink</code> , <code>stderrSink</code> .
Roles	string	Roles this sink supports, as a string. For example, “warning, error, debug”.

getVersion()

Retrieves version information for the current application. Two lines may be returned, one for the TIBCO Adapter SDK, one for the adapter.

Table 64 *getVersion()*

Returns	Type	Description
Instance ID	string	The instance ID as a string, for example, SDK.
Adapter Name	string	Name of the adapter as a string, for example, agentone.
Version	string	Version number as a string, for example, 1 . 1.

`_onUnsolicitedMsg()`

Displays all alert messages sent from the adapter or an error if not successful.

preRegisterListener()

Preregisters an anticipated listener.

Some sending applications can anticipate requests for certified delivery even before the listening applications start running. In such situations, the sender can preregister listeners, so TIBCO Rendezvous software begins storing outbound messages in the sender’s ledger. If the listening correspondent requires old messages, it receives the backlogged messages when it requests certified delivery.

Table 65 preRegisterListener()

Input Parameters	Type	Description
Session Name	string	Name of the session that anticipates the listener.
Publisher Name	string	Name of the component for which the listener should be preregistered.
Listener Session Name	string	Name of the listener to preregister.

Returns OK if the subscription service was preregistered successfully, false otherwise.

resetActivityStatistics()

Resets all the counts for the activity statistics.

resetConnectionStatistics()

Resets all the counts for the connection statistics.

resetThreadStatistics()

Resets all the counts for the thread statistics.

reviewLedger()

Returns information retrieved from the ledger file of a TIBCO Rendezvous certified messaging session.

Before invoking this method, ensure that the certified messaging publisher adapter has established a certified delivery agreement with its subscriber agents.

Table 66 Input parameters of reviewLedger()

Input Parameters	Type	Description
Session Name	string	Name of the TIBCO Rendezvous session for which ledger information is desired (default is a11).
Subject	string	Name of the subject for which ledger information is desired.

Table 67 Returns of reviewLedger()

Returns	Type	Description
Session Name	string	Name of the TIBCO Rendezvous CM session to which this information applies.
Subject	string	Subject name for this session.
Last Sent Message	integer	Sequence number of the most recently sent message with this subject name.
Total Messages	string	Total number of pending messages with this subject name.
Total Size	integer	Total storage (in bytes) occupied by all pending messages with this subject name. If the ledger contains ten messages with this subject name, then this field sums the storage space over all of them.
Listener Session Name	string	Within each listener submessage, the Listener Session Name field contains the name of the delivery-tracking listener session.
Last Confirmed	string	Within each listener submessage, the Last Confirmed field contains the sequence number of the last message for which this listener session confirmed delivery.
Line	integer	Row number in the ledger file.

setPollingBatchSize()

Sets the polling batch size for a service.

Table 68 *setPollingBatchSize()*

Input Parameter	Type	Description
ServiceName	string	Name of service where the polling batch size is set.
PollingBatchSize	integer	Polling batch size in milliseconds.

setPollingInterval()

Sets the polling interval for the Publication service.

Table 69 *setPollingInterval()*

Input Parameter	Type	Description
PollingInterval	integer	Polling interval in milliseconds.
ServiceName	string	Name of service where the polling interval is set.

setTraceSinks()

Adds a role or changes the file limit of a previously specified sink.

Table 70 *setTraceSinks()*

Input Parameters	Type	Description
Sink Name	string	Name of the sink for which to add a role or change the file limit.
Role Name	string	Name of the role to add to this sink (warning, error, debug, or user defined). Default is all.
File Size	integer	Maximum file size for this sink. This parameter is ignored if the sink specified by sinkName is not a file sink.

Returns OK if successful or an error if not successful.

stopApplicationInstance()

Stops the specified adapter by calling the internal `stop()` method. This method returns OK if successful or an error if not successful.

unRegisterListener()

Unregisters a currently preregistered listener.

Table 71 *unRegisterListener()*

Input Parameters	Type	Description
Session Name	string	Name of the session that anticipates the listener.
Publisher Name	string	Name of the component for which the listener should be preregistered.
Listener Session Name	string	Name of the listener to unregister.

This method returns `true` if the listener was unregistered successfully, `false` otherwise.

Appendix A **TIBCO Staging Tables**

TIBCO Staging tables are used to store the inserted and updated data in the ClarifyCRM system. The Publication Service periodically polls these tables and publishes the new messages.

The following are the private TIBCO Staging tables that contain the outgoing messages and message numbers.

- `table_tibco_message_queue`
- `table_tibco_message_fields`

Topics

- *table_tibco_message_queue, page 256*
- *table_tibco_message_fields, page 257*

table_tibco_message_queue

table_tibco_message_queue contains the names of objects in the message queue.

Table 72 table_tibco_message_queue

Column Name	Description
message_num	Used to group the rows that form a message as a logic record.
Operation	The outgoing event name, for example createContact or updateContact.
object_name	A valid Clarify object name, for example, site or contact.
object_ref_name	<p>Any string that uniquely identifies a row within a message scope. This field is required because several rows of a message may have the same object_name.</p> <p>For example, a site update message may have several address rows, one for the shipment address, one for the billing address, and one for the primary address. In such a case, object_ref_name can be used to distinguish the different addresses.</p>
flag_existing_object	Currently not in use.
status	A long type field to indicate the status of a "logic" record.
status 0 (UNPUBLISHED)	Represents messages yet to be published.
status 1 (POLLED)	The message status is updated to 1 after the message has been retrieved for publishing.
status 2 (PUBLISHED)	Represents a successfully published record.
status 3 (PUBLISH_ERROR)	Represents that an error occurred and that the message was not published.
x_pub_id	A variable string type which holds the subject name.

table_tibco_message_fields

table_tibco_message_fields contains field values of the objects within a message.

Table 73 table_tibco_message_fields

Column Name	Description
message_num	Combined with object_ref_name to identify the fields of a row in table_tibco_message_queue.
obj_ref_name	Combined with message_num to identify the fields of a row in table_tibco_message_queue.
field_name	Valid Clarify field name.
field_value	The value of this field.
field_type	An integer indicating if this field represents an object field (field_type = 1) or an object relation (field_type = 2).
field_data_type	<p>The Clarify database type defined by Clarify metadata table adp_sch_info.</p> <p>512 = Clarify long type; 514 = Clarify floating type; 516 = Clarify string type; 609=Clarify date-time type; 766 = Clarify decimal type.</p> <p>In an outgoing message, Clarify data-time type is published as a string; Clarify floating type is published as a real number with 7 significant digits; a Clarify decimal type is published as a real number with 15 digits of precision.</p>

Appendix B **Frequently Asked Questions**

This appendix contains answers to some frequently asked questions.

Topics

- *Frequently Asked Questions, page 260*

Frequently Asked Questions

Can I bring up TIBCO Designer from a Unix command-line?

TIBCO Designer is a GUI based tool and a Unix GUI environment is mandatory to run it.

When starting the adapter, what if the repository is not found?

Start the repository server before starting the adapter.

If you are starting a remote repository, ensure that TIBCO Administrator 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's `.tra` file.

Why does the adapter startup fail?

Ensure that the `RepoUrl` syntax has been specified accurately in the adapter's `.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's `.tra` file. Ensure that the path specified for the `.tra` file is correct.

When saving an adapter configuration to the project, why does it result in error messages indicating network problems?

If your installation is a stand-alone installation, check if the `<install_path>\TRA/bin` directory is present in your `PATH` settings.

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?

If the request comes with a reply subject, then the Subscription Service sends back a reply with the exception message as a part of the reply.

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 `TIBCO_HOME/adapter/adclycrm/<version_num>/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.

Appendix C **Error Messages**

This appendix explains the error messages that are logged to the location specified when the adapter was configured.

Topics

- *Error Message Listing, page 264*

Error Message Listing

This table lists the various messages.

Table 74 Error Message

Message	Role	Category	Resolution
AEClarifyDTA-0000001	Waiting for Request...		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000002	Exception in creating the Connection object:%1		
	Error	Adapter	Please refer to the exception message.
AEClarifyDTA-0000003	Invoked ConnectClfy		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000004	Successfully connected to Clarify		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000005	Could not connect to Clarify Database		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000007	CBO Exception in getting the list of CBOs:%1'		
	Error	Adapter	Check if the database is up. Look for the error code in Clarify documentation.
AEClarifyDTA-0000008	Query executed in getting the typeid :%1		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000009	Query executed in getting the field names :%1		
	Information	Adapter	Indicates normal adapter operation. No action required.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarifyDTA-0000010	CBO Exception in getting the field names and type :%1		
	Error	Adapter	Check if the database is up. Look for the error code in Clarify documentation.
AEClarifyDTA-0000011	replied...		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000012	CBO Exception in creating the application object for database connection :%1		
	Error	Adapter	Check if the database is up. Look for the error code in Clarify documentation.
AEClarifyDTA-0000013	Creating session....		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000014	Logging into the database....		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000015	Successfully logged in....		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000016	CBO Error in creating session and logging into the database		
	Error	Adapter	Check if the database is up. Look for the error code in Clarify documentation.
AEClarifyDTA-0000017	Timer is not defined in the repository		
	Information	Adapter	Check whether the timer exists in the repository.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarifyDTA-0000018	Exception in activating the Subscriber :%1		
	Error	Adapter	Check if the CLFYDTADiscSub subscriber is present in the repository.
AEClarifyDTA-0000019	Sending discovery message to find other running instances of CLFYDTA on subject %1 , service %2		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000020	Exception in sending the discovery message :%1		
	Error	Adapter	Check if CLFY_DTARpcServer server is present in the repository.
AEClarifyDTA-0000021	Assuming master status.		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000022	Received Reply to Discovery Request		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000023	%1		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000024	An instance of CLFYDTA is already available on : %1		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClarifyDTA-0000028	Exception in getting the address of the machine :%1		
	Error	Adapter	Please refer to the error message.
AEClarifyDTA-0000029	Exception in sending the reply :%1		
	Error	Adapter	Please refer to the error message.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClearifyDTA-0000030	Creating Application Object		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClearifyDTA-0000031	Logged out from the database		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClearifyDTA-0000032	Exited the Clarify application		
	Information	Adapter	Indicates normal adapter operation. No action required.
AEClearify-100001	Palette error/Adapter Configuration names must have only alphanumeric characters with no embedded spaces and can be up to 80 characters long.		
	Error	CONFIG	Please type in a valid name. Adapter Configuration names can only have alphanumeric characters and can be up to 80 characters long.
AEClearify-100002	Palette error/The [%1] must be greater than [%2], and less than [%3].		
	Error	CONFIG	The value specified must be greater than or equal to 0, and less than or equal to 65535.
AEClearify-100003	Palette Error/Names of adapters of the same type must be unique.		
	Warning	CONFIG	Specify a unique name for the adapter instance.
AEClearify-100004	Connection Retry Mechanism Warning/This adapter version does not suspend services on connection failure. The configured value Number of Reconnect Attempts Before Suspending Impacted Service(s) will be ignored.		
	Warning	CONFIG	Indicates normal adapter operation. No action required.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-100006	JMS Service Configured/This adapter version %1 does not support JMS services, but one was found.		
	Warning	CONFIG	You have configured a service with transport type as JMS. But the AE Version of this adapter instance does not support JMS. You may change the AE Version or make sure that the runtime version is higher than this version.
AEClarify-100007	XML Wire Format found/This adapter version %1 does not support XML Wire Format		
	Warning	CONFIG	You have configured a service with XML Wire Format. But the AE Version of this adapter instance does not support XML format. You may change the AE Version or make sure that the runtime version is higher than this version.
AEClarify-100008	Mandatory Field Missing/The field [%1] is a mandatory field.		
	Error	CONFIG	Please specify a value for the field.
AEClarify-100009	Palette error/The field [%1] must have only numeric values.		
	Error	CONFIG	Specify a valid numeric value for the field.
AEClarify-100011	Invalid Service name:/Service name can only contain alphanumeric characters with no embedded spaces and cannot exceed maximum length of 80 characters.		
	Error	CONFIG	Please type in a valid name. Adapter Configuration names can only have alphanumeric characters and can be up to 80 characters long.
AEClarify-110001	DTA Connection Failure/The Design Time Connection has failed %1.		
	Error	DTA	Verify if the Design Time Adapter is up and running.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-110002	Class Not Found/Could not find the Class. Error: %1. Error during connection to DTA.		
	Error	DTA	Please verify the palette jar which comes along with the installation is picked up by the adapter.
AEClarify-110003	DTA Connection Successful/The Design-Time Adapter Connection is successful.		
	Information	DTA	Indicates normal adapter operation. No action necessary.
AEClarify-110005	No DTA Connection/Connection to Design-Time Adapter is not established.		
	Warning	DTA	Check the connection parameters.
AEClarify-120003	Palette Error/Get Schema Failed.%1.		
	Error	SRVC	Please check whether the Clarify database is up.
AEClarify-120012	DTA Connection Successful/CBO Connection parameters are valid. CBO test connection is successful.		
	Information	DTA	Indicates normal adapter operation. No action necessary.
AEClarify-120013	DTA Connection Successful/JDBC Connection parameters are valid. JDBC test connection is successful.		
	Information	DTA	Indicates normal adapter operation. No action necessary.
AEClarify-120014	DTA Connection Failed/Please check whether the database is up and Connection parameters are valid.		
	Information	DTA	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-120015	DTA CBO Connection Failed/CBO Connection parameters are invalid. Please check the parameters entered.		
	Information	DTA	Indicates normal adapter operation. No action necessary.
AEClarify-120016	DTA JDBC Connection Failed/JDBC Connection parameters are invalid. Please check the parameters entered.		
	Information	DTA	Indicates normal adapter operation. No action necessary.
AEClarify-120017	Do you want to Reload Schema?/Schema already Exists. Do you want to Reload Schema?		
	Warning	SRVC	Select the required option.
AEClarify-120018	Palette error/Please enter the value in the specified format		
	Error	CONFIG	Enter the value in the format specified in the UI.
AEClarify-120019	No CBOs found/There are no tables in the database starting with the given string.		
	Information	SRVC	Indicates normal adapter operation. No action necessary.
AEClarify-120020	CBO not found/The selected CBO is not found in the database		
	Information	SRVC	Indicates normal adapter operation. No action necessary.
AEClarify-100016	Get Schema failed/The schema was not fetched for the service. Please ensure that the aeschema file is writable.		
	Error	CONFIG	Please make sure that the aeschema file created for the service is writable.
AEClarify-100017	Cannot Delete/The resource could not be deleted. Please ensure that the aeschema file is writable.		
	Error	CONFIG	Please make sure that the aeschema file created for the service is writable.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-100018	Cannot Paste Service/The service could not be pasted. Please ensure that the aeschema file is writable.		
	Error	CONFIG	Please make sure that the aeschema file created for the service is writable.
AEClarify-100019	Cannot Rename/The adapter configuration could not be renamed. Please ensure that the aeschema file is writable.		
	Error	CONFIG	Please make sure that the aeschema file created for the service is writable.
AEClarify-0000001	SDK Exception in Main : %1		
	Error	Adapter	Ensure all the required environment variables are set properly.
AEClarify-0000002	Exception in Main : %1		
	Error	Adapter	Ensure all the required environment variables are set properly.
AEClarify-0000003	Advisory Error Message : %1		
	Error	Adapter	Check configuration details for advisory.
AEClarify-0000004	Advisory Info Message : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000005	Advisory Warn Message : %1		
	Warning	Adapter	The details are displayed in the warning message.
AEClarify-1000005	Adapter Successfully Initialized		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000006	Exception while processing the advisory event : %1		
	Error	Adapter	Ensure all the required environment variables are set properly.
AEClarify-0000008	MException in Initializing the Adapter : %1		
	Error	Adapter	Ensure whether the required environment variables are set properly. Refer to the <i>TIBCO Adapter for ClarifyCRM User's Guide</i> for the required environment settings.
AEClarify-0000009	Exception in Initializing the Adapter : %1		
	Error	Adapter	Ensure whether the required environment variables are set properly. Refer to the <i>TIBCO Adapter for ClarifyCRM User's Guide</i> for the required environment settings.
AEClarify-0000010	The Adapter is shutting down		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000011	Connecting to Clarify with the parameters : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000012	Exception in Connecting to Clarify during Initialization : %1		
	Error	Adapter	Please check for valid database connection parameters.
AEClarify-910007	Unable to create connection with the clarify application using connection parameters [login=%1,password=****,DBServer=%2,DBName=%3] Clarify Application error is : %4		
	Error	Adapter	Please check for valid database connection parameters.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000013	Got the Subscriber : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000014	Activated the Subscriber (Name:Subject) = %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000015	No subscribers are configured in the Repository		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000016	Total Number of subscribers Activated : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000017	No components in Repository to activate		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000018	Exception in Initializing Subscriber Components : %1		
	Error	Adapter	Check the Repository for proper configuration.
AEClarify-0000019	Exception in Initializing Timer Components : %1		
	Error	Adapter	Check the Repository for proper configuration.
AEClarify-0000020	Got the Publisher (Name:Subject) = %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000021	No Publishers are configured in the Repository		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000022	Total Number of Publishers Activated : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000023	Failed in Initializing Publisher Components		
	Error	Adapter	Check configuration details for Publisher.
AEClarify-0000024	Failed in Initializing Publisher Components		
	Error	Adapter	Check configuration details for Publisher.
AEClarify-0000026	Exception in Initializing Publisher Components : %1		
	Error	Adapter	Check configuration details for Publisher.
AEClarify-0000027	Terminate Subscriber not found in repository so shutting down the adapter		
	Error	Adapter	Please check the repository if terminate subscriber is configured.
AEClarify-0000028	Got RPC Server : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000029	No RPC Server are configured in the Repository		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000030	Total Number of RPC Servers Activated : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000031	Exception in Initializing RPC Server Components : %1		
	Error	Adapter	Check configuration details for RPC Server.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000032	Initializing Adapter Services Info and Advisory Listeners		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000033	Adapter Services Info and Advisory Listeners initialized		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000034	Exception in Initializing Adapter Services Info and Advisory Listeners : %1		
	Error	Adapter	Check configuration details.
AEClarify-0000035	Exception in initializing the user defined classes for user exits : %1		
	Error	Adapter	Check if the User Exits are properly configured in the repository and the user exit classes are there in the classpath.
AEClarify-0000036	Registering Publisher Event %1 for Preprocessing		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000037	Registering Subscriber Event %1 for Preprocessing		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000038	Registering Subscriber Event %1 for Post processing		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000039	No User Exits defined for Publisher Preprocessing		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000040	No User Exits defined for Subscriber Preprocessing		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000041	No User Exits defined for Subscriber Post processing		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000042	Class Not Found Exception in Registering User Exits : %1		
	Error	Adapter	Please check if User Exit Class is included in the Classpath.
AEClarify-0000043	Instantiation Exception in Registering User Exits : %1		
	Error	Adapter	Please check if User Exit Class is written as per directions in the <i>TIBCO Adapter for ClarifyCRM User's Guide</i> .
AEClarify-0000044	Illegal Access Exception in Registering User Exits : %1		
	Error	Adapter	Please check if User Exit Class is written as per directions in the <i>TIBCO Adapter for ClarifyCRM User's Guide</i> .
AEClarify-0000045	Exception in getting the User Exit Configuration from Repository : %1		
	Error	Adapter	Please check if the association attribute associated with user exits is present in the Repository.
AEClarify-0000046	Successfully Logged into Clarify database		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000047	CBO Error occurred while Connecting to database : %1		
	Error	Adapter	Please check the Clarify documentation for the CBO Errors, check for proper connection parameters like login, password, database name and database server name.
AEClarify-0000048	Exception occurred while Connecting to database : %1		
	Error	Adapter	Check for proper connection parameters like login, password, database name and database server name.
AEClarify-0000050	CBO Error occurred while Creating a FormContext Object : %1		
	Error	Adapter	Please check the Clarify documentation for the CBO Errors.
AEClarify-0000051	Exception occurred while Creating a FormContext Object : %1		
	Error	Adapter	Please check the Clarify documentation for the CBO Errors.
AEClarify-0000052	CBO Error occurred while Checking the Clarify Connection : %1		
	Error	Adapter	Please check the Clarify documentation for the CBO Errors.
AEClarify-0000053	Exception occurred while Checking the Clarify Connection : %1		
	Error	Adapter	Check database connectivity.
AEClarify-0000054	Successfully Logged into Clarify database during reconnect		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000055	CBO Error occurred while Reconnecting to database : %1		
	Error	Adapter	Please check the Clarify documentation for the CBO Errors. Also check whether database is up and running.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000056	Exception occurred while Reconnecting to database : %1		
	Error	Adapter	Please check whether database is up and running.
AEClarify-0000057	Exception occurred while Waiting for the Connection Retry Interval : %1		
	Error	Adapter	Check database connectivity.
AEClarify-0000059	Exited the Clarify application		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000061	Exception occurred in Polling the database : %1		
	Error	Adapter	Check database connectivity.
AEClarify-0000062	Exception in getting Error Publisher : %1		
	Error	Adapter	Please check if the Adapter Configuration in Repository Contains Error Handling Publisher.
AEClarify-0000063	Adapter shutdown in progress		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000064	Inside OnEvent of Subscriber		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000065	Exception while extracting data from the incoming message		
	Error	Adapter	Please check if the incoming message is proper as per the Adapter documentation.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-920002	Subscription error. Subscription service %1 failed to deserialize the event received on subject %2 and SDK exception thrown is %3. The Repository URL is %4 and the Configuration URL is %5.		
	Error	Adapter	Please check if the incoming message is proper as per the Adapter documentation.
AEClarify-920003	Subscription error. Subscription service %1 listening on subject %2 received an inbound event with null data. The Repository URL is %3 and the Configuration URL is %4.		
	Error	Adapter	Please check if the incoming message is proper as per the Adapter documentation.
AEClarify-920007	Subscription error. Subscription service %1 listening on subject %2 received an inbound event with null class description. The Repository URL is %3 and the Configuration URL is %4.		
	Error	Adapter	Please check if the incoming message is proper as per the Adapter documentation.
AEClarify-0000066	Exception while getting the optype field in the incoming message		
	Error	Adapter	Check if incoming message has the optype field.
AEClarify-0000067	Exception while getting the wrapper_class field in the incoming message		
	Error	Adapter	Check if incoming message has the wrapper_class field.
AEClarify-0000068	Exception while getting the main CBO name from the incoming message		
	Error	Adapter	Check if incoming message has the schema field.
AEClarify-0000069	Exception while getting the lookup field from the incoming message		
	Error	Adapter	Check if incoming message has the lookup field.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000070	Exception while building the filter from the lookup in the incoming message		
	Error	Adapter	Check if the values passed in the lookup field are correct as per adapter documentation.
AEClarify-0000071	Exception while extracting the main and contained CBO details		
	Error	Adapter	Check if incoming message has the schema field and is populated with correct values.
AEClarify-0000072	Main CBO is : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000073	%1		
	Error	Adapter	Check if the business data in the incoming message is proper.
AEClarify-920015	Subscription error. Subscription service %1 listening on subject %2 failed due to Clarify application invocation error %3. The target application specific commands and parameters are %4		
	Error	Adapter	Check if the business data in the incoming message is proper.
AEClarify-920016	Subscription error. Subscription service %1 listening on subject %2 received error %3 in PostProcessing user exit. The user exit parameters are %4		
	Error	Adapter	Check if the business data in the incoming message is proper.
AEClarify-0000074	Reconnection Attempts failed. So shutting down the adapter		
	Error	Adapter	Check if Clarify Application is up and running.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-920017	Subscription error. Subscription service %1 listening on %2 could not send reply for the Clarify application invocation. The Repourl is %3 and the configuration url is %4		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-940001	Request Response error. Request Response service %1 listening on %2 received unexpected null data in incoming request. The Repository URL is %3 and the Configuration URL is %4		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-940002	Request Response error. Request Response service %1 failed to deserialize the received MServerRequest to MInstance: Received event on subject %2, event = %3, SDK exception = %4. The Repository URL is %5 and the Configuration URL is %6		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-940007	Request Response error. Error in incoming data for RPC service: %1 on subject: %2. Missing mandatory parameter %3 for RPC input class %4		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-940008	Request Response error. Connection error in invocation of RPC service:%1 on subject:%2. Connection parameters are [login name=%3, password = ****, DBName =%4]		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-940009	Request Response error. Request Response service %1 listening on subject %2 failed due to Clarify application invocation error %3. The inbound event is %4.		
	Error	Adapter	Check if Clarify Application is up and running.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-940010	Request Response error. Request Response service %1 listening on subject %2 failed to create Reply Object with error %1.		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-950001	Request Response Invocation error. Request Response Invocation service %1 with subject as %2 received event from target application %3. It failed while converting event to Request, as it could not get the class description for %4. Repository URL is %5 and the Configuration URL is %6.		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-950002	Request Response Invocation error. Request Response Invocation service %1 with subject %2 received error while requesting event over the wire. The Request Response invocation endpoint details are [operationName =%3, className = %4].		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-950003	Request Response Invocation error. Request Response Invocation service %1 with subject %2 received null data while requesting event over the wire. The Request Response invocation endpoint details are [operationName =%3, className = %4].		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-950004	Request Response Invocation error. Request Response Invocation service %1 with subject %2 received timeout error while requesting event over the wire. The Request Response invocation endpoint details are [operationName =%3 , className = %4].		
	Error	Adapter	Check if Clarify Application is up and running.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-950005	Request Response Invocation error. Request Response Invocation service %1 with subject %2 received error while processing while reply. The Request Response invocation endpoint details are [operationName =%3, className = %4].		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-920014	Subscription error. Subscription service %1 listening on subject %2 could not process the inbound event due to connection error %3 against the Clarify application with parameters %4 after %5 connection retries. The connection timeout is %6.		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-0000075	MException Received by Subscriber : %1		
	Error	Adapter	The sending and receiving applications may be using different repositories with different definitions of the same class or the sending application does not conform to the wire format.
AEClarify-920001	Subscription error. Subscription service %1 listening on %2 received an Exception event type. The Repository URL is %3 and the Configuration URL is %4 Error received = %5		
	Error	Adapter	Check the configuration of the application that is publishing the event and make sure that it matches the inbound event definition for the above subscription service.
AEClarify-0000076	Exception while processing the Event received by Subscriber : %1		
	Error	Adapter	The sending and receiving applications may be using different repositories with different definitions of the same class. Or the sending application does not conform to the wire format.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000077	Event Processed. Waiting for Next Event		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000080	Contained CBO is : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000081	Exception Occurred while sending back the Subscriber Success Reply		
	Error	Adapter	Confirm that the subscriber is created.
AEClarify-0000082	Exception Occurred while sending back the Subscriber Failure Reply		
	Error	Adapter	Confirm that the subscriber is created.
AEClarify-0000083	Exception Occurred while sending back the Reply		
	Error	Adapter	Confirm that the subscriber is created.
AEClarify-0000088	Error in preprocessing : %1		
	Error	Adapter	Check if user exit code is behaving properly.
AEClarify-0000091	Error in post processing : %1		
	Error	Adapter	Check if user exit code is behaving properly.
AEClarify-0000092	Error in Creating the Utility Class ClarifyAppServer : %1		
	Error	Adapter	The program may be out of java virtual memory or another internal jvm exception occurred.
AEClarify-0000093	Error in resetting the FormContext : %1		
	Error	Adapter	Check the database connectivity.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000094	Error in getting a reference to the CBO : %1		
	Error	Adapter	Please check in the incoming message if it contains the main and the contained CBO properly.
AEClarify-0000095	Create Business Object succeeded Create Id is : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000096	Create Business Object Failed		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000097	Update Business Object succeeded Update Id is : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000098	Update Business Object Failed		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000099	"Requested operation %1 not supported"		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000100	Error in getting a reference to the Clarify queue : %1		
	Error	Adapter	Please check whether the business object is associated with a Queue.
AEClarify-0000101	Error in getting a reference to the Login : %1		
	Error	Adapter	Please check in the incoming message if it contains the main and the contained CBO properly.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000102	Error in getting a reference to the Main CBO : %1		
	Error	Adapter	Please check in the incoming message if it contains the Main CBO properly
AEClarify-0000103	"Field %1'could not be set in the Reply Object for the queried contained BO'%2"		
	Warning	Adapter	Check whether the schema used in the project is in sync with the database being used. If not, reload the schema.
AEClarify-0000104	Error in building the Reply Message		
	Error	Adapter	Make sure that the server has sent the message correctly, make sure the parameters or the operation is specified properly.
AEClarify-0000105	Trying to connect to clarify application with the following parameters: %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000106	Error in getting the terminate Subscriber		
	Error	Adapter	Check if terminate subscriber endpoint exists.
AEClarify-0000108	"Field %1'could not be set in the Reply Object for the queried contained BO'%2"		
	Warning	Adapter	Check whether the schema used in the project is in sync with the database being used. If not, reload the schema.
AEClarify-0000201	Exception occurred in Clarify RpcServer Constructor : %1		
	Error	Adapter	The program may be out of java virtual memory or another internal jvm exception occurred.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000202	Exception occurred in OnInvoke of Clarify RpcServer : %1		
	Error	Adapter	Check for the correctness of operation name in the incoming request.
AEClarify-0000203	Exception occurred while sending reply due to OnInvoke Error in Clarify RpcServer: %1		
	Error	Adapter	Check the TIBCO Rendezvous Settings.
AEClarify-0000204	The request does not contain all the required parameters to perform a rpc operation		
	Error	Adapter	Check whether the incoming request contains all the required parameters.
AEClarify-0000205	Reconnection Attempts failed. Attempting to Shut down the adapter		
	Error	Adapter	Check if Clarify Application is up and running.
AEClarify-0000206	Exception Occurred due to mismatched / missing output parameter. Exception : %1		
	Error	Adapter	Check whether the output parameter in the schema is configured correctly.
AEClarify-0000207	Exception Occurred while sending the reply back to the calling application in Clarify RpcServer. Exception : %1		
	Error	Adapter	Check the adapter and TIBCO Rendezvous Settings.
AEClarify-0000208	Exception Occurred while invoking RpcServer Query Operation. Exception : %1		
	Error	Adapter	Check if the method name is correct/ Ensure that the database contains proper data.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000209	Exception Occurred while invoking RpcServer Workflow Operation. Exception : %1		
	Error	Adapter	Check if the method name is correct/ Ensure that the database contains proper data.
AEClarify-0000210	Exception Occurred while invoking RpcServer Customized Operation. Exception : %1		
	Error	Adapter	Check if the method name is correct/ Ensure that the database contains proper data.
AEClarify-0000211	Exception occurred while parsing the incoming request to retrieve input parameters. Exception : %1		
	Error	Adapter	Ensure that the incoming request schema conforms to the configured schema.
AEClarify-0000212	Incoming request schema does not match with the configured schema. Exception : %1		
	Error	Adapter	Ensure that the incoming request schema conforms to the configured schema.
AEClarify-0000213	RpcServer Query Operation invoked.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000214	RpcServer Workflow Operation invoked.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000215	RpcServer Customized Operation invoked.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000216	Invocation of RpcServer Operation is completed. Waiting for Next Operation.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000306	Query Operation is not supported for a %1 Business Object		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000307	WorkFlow Operation cannot be performed on a %1 Business Object		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000308	Customized Operation is not supported for a %1 Business Object		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000309	ActEntry Object cannot be created as query for contact returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000310	Adapter does not support creation of BusOrg CBO		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000311	Case Object cannot be created as query for contact returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000312	Contact Object cannot be created as query for site returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000313	Adapter does not support creation of DemandDetail CBO		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000314	Adapter does not support updating a DemandDetail CBO		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000315	DemandHeader Object cant be Created as the query for contact returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000316	Adapter does not support creation of Dialogue CBO		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000317	Adapter does not support updating a Dialogue CBO		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000318	DocInst Object cannot be Created as the query for contact returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000319	Part Object cannot be Created as the query for part_num returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000320	ServicePart Object cannot be Created as the query for contact returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000321	Site Object cannot be created as query for address returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000322	SubCase Object cannot be created as query for case returned no rows		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000323	%1 operation on %2 Business Object failed since the specified User does not exist in the database		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000324	%1 operation on %2 Business Object failed since the specified Queue does not exist in the database		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000325	%1 Business Object accepted in the default WIPbin		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000326	%1 Business Object accepted in the specified WIPbin.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000327	Assign operation on %1 Business Object failed since the logged in user does not own the %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000328	%1 Business Object assigned to specified User and default WIPbin		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000329	%1 Business Object assigned to specified User and specified WIPbin		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000330	Dispatch operation on %1 Business Object failed since the logged in user does not own the %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000331	%1 Business Object dispatched to specified Queue		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000332	Forward operation on %1 Business Object failed since the logged in user does not own the %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000333	%1 Business Object forwarded to specified Queue		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000334	Move operation on %1 Business Object failed since the logged in user does not own the %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000335	%1 Business Object moved to default WIPbin		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000336	%1 Business Object moved to specified WIPbin		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000337	Reject operation on %1 Business Object failed since the logged in user does not own the %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000338	%1 Business Object rejected		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000339	%1 Business Object Yanked		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000340	Exception occurred in the init method of %1 wrapper. Exception : %2		
	Error	Adapter	Check Clarify CBO related settings
AEClarify-0000410	Published messages will be deleted from the database		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000411	Published messages will be retained in the database		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000414	Exception in getting the Publisher options from the Repository :%1		
	Error	Adapter	The association attribute Publisher options may have been deleted from the repository. Enter the values in the Publisher options tab and save the repository.
AEClarify-0000416	Exception in polling the database :%1		
	Error	Adapter	Please check the exception message for details.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000420	SQLException in updating the publish status in the TIBCO_MESSAGE_QUEUE table :%1		
	Error	Adapter	Check if the database is up. Look for the error code in Clarify documentation.
AEClarify-0000422	General Exception in clearing the queue and updating the status in the staging table : %1		
	Error	Adapter	Please check the exception message. Your system may be running short of memory. Please close some applications and restart the adapter.
AEClarify-0000423	Exception in creating the string of published message numbers whose status is to be updated : %1		
	Error	Adapter	Please check the exception message. Your system may be running short of memory. Please close some applications and restart the adapter.
AEClarify-0000424	Exception in updating the queue containing the published message numbers : %1		
	Error	Adapter	Please check the exception message. Your system may be running short of memory. Please close some applications and restart the adapter.
AEClarify-0000427	Exception in getting message from the message queue :%1		
	Error	Adapter	Please check the exception message. Your system may be running short of memory. Please close some applications and restart the adapter.
AEClarify-0000430	Exception in deleting message from the message queue :%1		
	Error	Adapter	Please check the exception message.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000431	SQL Exception in getting distinct message numbers from the queue table :%1		
	Error	Adapter	Check if the database is up. Look for the error code in Clarify documentation.
AEClarify-0000432	General Exception in getting distinct message numbers from the queue table :%1		
	Error	Adapter	Please check the exception message.
AEClarify-0000436	Exception in creating the message queue :%1		
	Error	Adapter	Please check the exception message.
AEClarify-0000437	SQLException in querying of message fields from queue and field tables :%1		
	Error	Adapter	Check if the database is up. Look for the error code in Clarify documentation. If datatype found is CLOB turn "adclycrm.oracle.useCLOB" option in tra to ON.
AEClarify-0000438	General Exception in bulk querying of messages with status 0 from queue and field tables :%1		
	Error	Adapter	Please check the exception message.
AEClarify-0000440	Exception in initializing the publisher threads :%1		
	Error	Adapter	Please check the exception message.
AEClarify-0000441	Interrupted thread exception :%1		
	Error	Adapter	Please check the exception message. Restart the adapter.
AEClarify-0000445	Message to be published :%1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000447	MException in getting the Publisher specific schemas :%1		
	Error	Adapter	Please make sure the schemas created under the folder PubSchema in the repo are present.
AEClarify-0000448	Exception in Processing the message to be published :%1		
	Error	Adapter	Please check the user-defined class written for implementing PrePublisher User exit.
AEClarify-0000452	Exception in casting the field values :%1		
	Error	Adapter	Please make sure field_value in the TIBCO_MESSAGE_FIELDS table is of the same data type as specified in the field _data_type column.
AEClarify-0000453	Exception in getting the Publisher specific sequences which are required to construct the message:%1		
	Error	Adapter	Please make sure the default sequences created under the SchemasSequences.....Clarify folder in the repo are present.
AEClarify-0000458	Exception in invoking the onInvoke method of the object of UserDefined class :%1		
	Error	Adapter	Please check the user-defined class written for implementing PrePublisher User exit.
AEClarify-0000461	Exception in querying queue/field table or in creating the MInstance due to incorrect data in field table : %1		
	Error	Adapter	Refer to the exception message for details.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000462	Exception in freeing database connection on SQLException in getting distinct message numbers : %1		
	Error	Adapter	Check if the database is up. Look for the error code in Clarify documentation.
AEClarify-0000463	Server name is not provided in the TRA file		
	Error	Adapter	Please provide the server name.
AEClarify-0000464	Port number is not provided in the TRA file		
	Error	Adapter	Please provide the Port number.
AEClarify-0000465	No CBO connection is made with the database as no subscriber and rpc are configured		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000467	Trying to connect using jdbc with the following parameters : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000468	Exception in registering the driver :%1		
	Error	Adapter	Check if the classes12.zip is present in the classpath.
AEClarify-0000469	SQLException in obtaining connection cache :%1		
	Error	Adapter	Check the connection parameters.
AEClarify-0000470	Exception in obtaining connection cache :%1		
	Error	Adapter	Check the connection parameters.
AEClarify-0000471	Exception in connecting to the database using JDBC : %1		
	Error	Adapter	Check the connection parameters.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000472	Duplicate subject names have been configured for RPC Clients. The adapter will shut down.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000473	Error in creating run time subscribers for RPC client :%1		
	Error	Adapter	Check the RvSession name given in the palette already exists.
AEClarify-0000474	RpcClient : %1 associated with Subscriber listening on Subject :%2 and Session :%3 found		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000475	ClientRequest for RPC Client Endpoint :%1 created		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000476	Type of client operation : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000477	Error in creating ClientRequest object :%1		
	Error	Adapter	Please check the ClassName and OperationName configured for the endpoint.
AEClarify-0000478	Error in creating MInstance for ClientSchema :%1		
	Error	Adapter	Check the configuration details.
AEClarify-0000479	Request class associated with the clientRequest schema : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000480	Error in creating MInstance of requestSchema class :%1		
	Error	Adapter	Check the configuration in the repository.
AEClarify-0000482	Error in parsing the incoming TIBCO Rendezvous message and setting the values for the request schema: %1		
	Error	Adapter	Check the incoming TIBCO Rendezvous message is of the proper format.
AEClarify-0000484	Error in setting the values for the MClientRequest parameters : %1		
	Error	Adapter	Check the configuration in the repository.
AEClarify-0000485	Reply from the external server has exception		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000487	Request timed out before the reply arrived : %1		
	Error	Adapter	Check whether the external RPC server is up and running.
AEClarify-0000488	Error due to missing class description or failure to marshall/unmarshall data : %1		
	Error	Adapter	Metadata class information of the repository and the MB should be the same.
AEClarify-0000489	Error in parsing the server reply and creating the reply TIBCO Rendezvous message : %1		
	Error	Adapter	Check whether the reply is in the required format.
AEClarify-0000490	Error :%1		
	Error	Adapter	Check the incoming TIBCO Rendezvous message is of the proper format.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000491	Error in creating the errorReply message : %1		
	Error	Adapter	Check whether the proper reply schema is configured.
AEClarify-0000492	RvReply message : %1		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000493	RPC client operation invoked successfully		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000494	Error in invoking the client operation :%1		
	Error	Adapter	Check the configuration in the repository
AEClarify-0000495	No CBO connection is made with the database		
	Information	Adapter	Indicates normal adapter operation. No Subscription or Request-Response Services are configured.
AEClarify-0000496	Main CBO is not set in the schema. Message wont be published.		
	Error	Adapter	Use the Schema tab in the Publication service to set the Main CBO.
AEClarify-0000497	Contained CBO %1 is not found in the schema		
	Warning	Adapter	Please refer to the <i>TIBCO Adapter for ClarifyCRM User's Guide</i> for the steps to be followed to attach a contained CBO.
AEClarify-0000498	Schema mismatch problem : %1		
	Warning	Adapter	Check whether the schema used in the project is in sync with the database being used. If not, reload the schema.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000499	Error in trying to remove an expired connection :%1		
	Error	Adapter	Check if the DB is up.
AEClarify-0000500	Error in invalidate connection :%1		
	Error	Adapter	Check the validity of DB connection parameters.
AEClarify-0000501	Error while clean up of connection pool :%1		
	Error	Adapter	Check if the DB is up.
AEClarify-0000502	Error in removing connection event listener :%1		
	Error	Adapter	Check the validity of DB connection parameters.
AEClarify-0000503	Not able to connect to database. Please check the database parameters in the tra and repository		
	Information	Adapter	Check the database parameters in the tra and repository.
AEClarify-0000504	Exception in initializing the RPC Client components :%1		
	Error	Adapter	Please check the repository for proper configuration.
AEClarify-0000505	%1 field is not found in the configured request schema		
	Warning	Adapter	Check whether the schema used in the project is in sync with the database being used. If not, reload the schema.
AEClarify-0000507	Startup Warning. Password decryption was unsuccessful.		
	Warning	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000508	Using CLOB data type instead of LONGVARCHAR		
	Warning	Adapter	Indicates normal adapter operation. It also indicates that Oracle 9i or Oracle 10g database is being used. No action necessary.
AEClarify-0000511	Exception occurred while getting connection from the connection pool.		
	Error	Adapter	Check for connection pool size. The size should not be less than zero.
AEClarify-0000512	Exception occurred in associating connection to a thread.		
	Error	Adapter	Please check if the specified number of threads and connections have been created.
AEClarify-0000515	%l Connections created with Clarify Application.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000517	Stopped the dispatchers.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000519	%1 dispatchers created.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000524	Exception occurred in associating connection to a thread.		
	Warning	Adapter	Please check if the specified number of threads and connections have been created.
AEClarify-0000526	Exception in adding threadId to table":%1.		
	Warning	Adapter	Indicates normal adapter operation. No action necessary.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-0000527	Reconnect Attempt %1 succeeded.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000528	Suspending Subscribers since reconnection failed in transient attempts..		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-0000529	Activating Subscribers after reconnection success.		
	Information	Adapter	Indicates normal adapter operation. No action necessary.
AEClarify-930001	Publication error. Publication service %1%2 encountered error %3 while trying to connect to target application %4. Connection parameters are %5, the connection timeout is %6 milliseconds, and the number of retry efforts is %7.		
	Error	Adapter	Check the target application and make sure it is up and running. Check the connection parameters for right syntax and values.
AEClarify-930003	Publication error. It failed while converting event to Minstance as it could not get the class description. Repository URL is %3 and the Configuration URL is %4.		
	Error	Adapter	Please verify the configuration of the publication service and check that the schema/class definitions are present in the repository.
AEClarify-930009	Publication error. Publication service %1 with publication subject %2 received an event from the target application but encountered error %3 in pre-processing user exit invocation. The target application details are %6.		
	Error	Adapter	Make sure that the parameters passed to UserExit are valid and the User Exit can be invoked by the adapter.

Table 74 Error Message (Cont'd)

Message	Role	Category	Resolution
AEClarify-930014	Publication error. Publication service %1 with publication subject %2 received error while sending event over the wire.%3		
	Error	Adapter	Please check repository settings for valid configuration of the publish endpoint for this service.



The error messages for Request-Response Invocation Service cannot exceed 255 characters when used in the flexible deployment scenario. This is a limitation of the Tuxedo middleware.

Appendix D **User Exits**

This chapter describes custom processing that can be performed for messages depending on the adapter component.

Topics

- *Overview, page 306*
- *How the Adapter Implements User Exits, page 307*

Overview

User exits are hooks provided by the adapter. They allow you to perform custom processing on each message depending on the adapter (Subscription or Publication Service) and for the message, which is currently being processed. Custom processing can be done before (pre-processing) and after (post-processing) the original message is processed.

How the Adapter Implements User Exits

The adapter contains an interface implemented for the UserExits functionality. The supported functionalities are:

- Pre-processing of the Subscription Service event message
- Post-processing of the Subscription Service event message
- Pre-processing of the Publication Service event message

To provide a user exit for a Subscription Service, or a Publication Service event, you need to define a class that implements the `IclarifyUserExit` Interface. You have to compile the class and include it in the adapter's classpath.

Pre-processing examples include data validation, message format conversion. An example of post-processing would be generating additional trace messages, validation of the reply object and any format modifications.

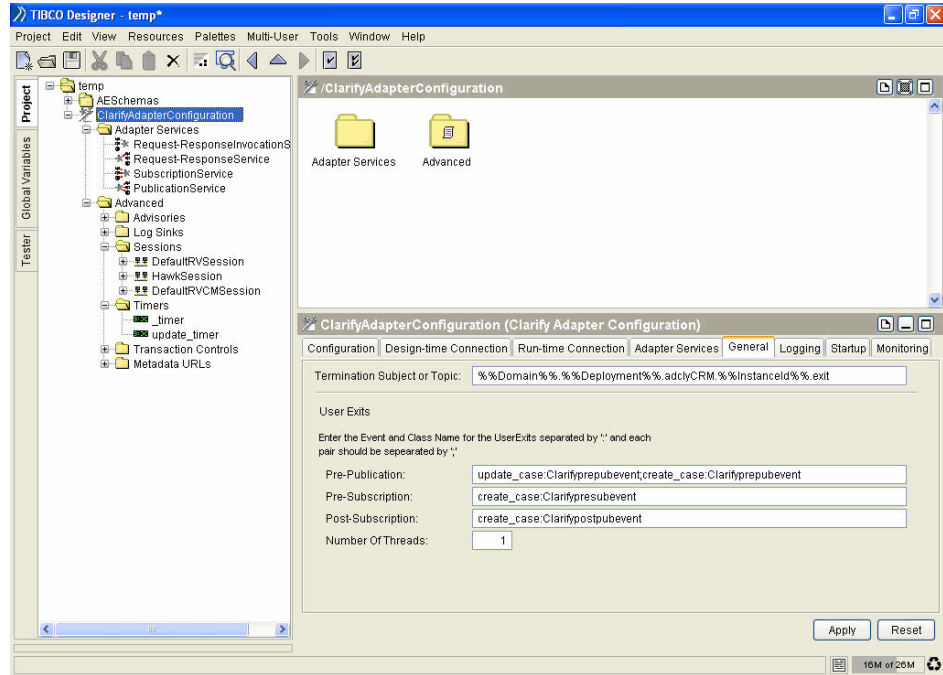
To implement a user exit for the adapter component:

1. For the adapter components (Subscription Service and Publication Service), select the event(s) to provide the user exit. For example, `TEST.CASE` event.
2. Choose when you want the adapter to call the user exit: before or after the event is processed.
3. Define a new class, which implements the `IclarifyUserExit` Interface (example: `Clarifypresubevent.java` for implementing pre-processing of Subscription Service event message).
4. Implement the methods defined in the interface (example: `onInitialization`, `onInvoke`, `onTermination`).

Implement user exits for pre-processing of Publication Service event message

1. In the adapter configuration, select the Configuration tab and select the Show all Tabs check box.
2. Select the General tab.
3. In the Pre-Publication, Pre-Subscription and Post-Subscription fields, enter the event name and class name separated by a semi colon in the respective fields. Example: `TEST.CASE:Clarifypresubevent`

Figure 62 Implement User Exits for Pre-processing of Publication Service event message



Each pair of event and class should be separated by a semicolon. Example:
 update_case:Clarifyprepubevent;TEST.CASE:Clarifyprepubevent

4. Click **Apply** to save the event and the class in the project.

IclarifyUserExit Interface

Given next is the IclarifyUserExit Interface.

```
public interface IclarifyUserExit
{
    void onInitialization(ClarifyAdapterCore m_mApp);
    void onTermination();
    UserExitResult onInvoke(MInstance incomingEvent);
    void onInvoke(MInstance incomingEvent, boolean
bSuccess,Exception Ex);
}
```

Example:

The following steps show how to create a user exit implementation for pre-processing, for the event TEST.CASE for the Subscription Service component.

1. Define a class that implements the `IclarifyUserExit` interface. Following is the definition for the example:

```
import com.tibco.adapter.clarify.*;
import com.tibco.sdk.metadata.*;

public class Clarifypresubevent implements IclarifyUserExit{
    ClarifyAdapterCore m_mApp;
    UserExitResult userExitRes;

    public void onInitialization(ClarifyAdapterCore m_mApp){
        this.m_mApp = m_mApp;
        System.out.println("Inside OnInitialization in ClarifyPubEvent");
    }

    public void onTermination(){
        System.out.println("Inside onTermination in ClarifyPubEvent");
    }

    public UserExitResult onInvoke(MInstance incomingEvent){
        System.out.println("Inside onInvoke in ClarifyPubEvent");
        userExitRes=new UserExitResult();
        userExitRes.setStatus(true);
        userExitRes.setInstance(incomingEvent);

        return userExitRes;
    }
}
```

2. Implementation for the methods `onInitialization`, `onTermination`, and `onInvoke` (`onInvoke` is the method that is called for every registered event) should be provided as above. `onInitialization` and `onTermination` are called only once for every event exit implementation.

In the case of the Preprocessing of Publication Service and Subscription Service event message, `onInvoke` method takes an `MInstance` object and returns an object of class `UserExitResult`. `UserExitResult` class provides three functions: `setStatus(boolean status)`, `setInstance(MInstance incomingInstance)`, and `setException(Exception)`.

In the case of Postprocessing of Subscription Service event message, `onInvoke` method takes an `MInstance` object, `boolean` and `Exception` object as parameters and returns none.

3. Compile this class.

Before compiling, make sure that `adclycrm.jar` which comes with adapter installation is in the classpath. The compiled class is included in the adapter classpath. This can be done by appending the location of the class in the

`tibco.class.path.extended` variable in the adapter tra file
(`adcllyCRM.tra`).

Appendix E **Wrapper Classes**

This chapter describes Wrapper classes, how to write wrapper classes and how they work with the adapter.

Topics

- *Overview, page 312*
- *Wrappers Packaged with the Adapter, page 313*
- *Wrapper Class and TibcoCBOInterface, page 320*
- *Writing Wrapper Class for Case Business Object, page 321*

Overview

A Wrapper class provides more functionality than its underlying business object. It encapsulates business logic which helps by extending the base CBOs, which enables creating or updating CBOs. All the wrapper classes should implement `TibcoCBOInterface`, which contains all the operations that can be performed. The usage of `TibcoCBOInterface` ensures that all the wrapper classes have a similar structure.

Objectives of Wrapper Classes

- Provide more functionality than its underlying CBO
- Encapsulate Business logic to create or update CBOs
- Enable customization to implement the required functionality

Wrappers Packaged with the Adapter

This section explains the wrapper classes that are packaged with the adapter installation and the functionality provided for each of the wrapper classes.

Wrappers are available in the following location:

`TIBCO_HOME\adapter\adclycrm\<version_num>\samples\wrapper`

ActEntry_Wrapper

This class is used as a wrapper over the `ActEntry` CBO and it extends the `ActEntry` class. It can be used to perform activities like update, queryRPC, workflowRPC operations on the CBO.

The `ActEntry_Wrapper` class is used to create new activity log entry for business objects (like installed part or workflow item). The wrapper is customized to associate new activity log entry with a Business Object (BO) such as Case, get activity log entries for Business Object, and associate a Business Object as a participant in an activity.

Address_Wrapper

This class is used as a wrapper over the `Generic` CBO and it extends the `Generic` class. It can be used to perform activities like Create, Update and QueryRPC operations on the CBO.

BusOrg_Wrapper

This class is used as a wrapper over the `BusOrg` CBO and it extends the `BusOrg` class. The `BusOrg` wrapper class is used to perform create, update and QueryRPC operations on the CBO that stores information about the organization that you do business with.

Case_Wrapper

This class is used as a wrapper over the `Case` CBO and it extends the `Case` class. The `Case` wrapper class is customized to perform create, update, QueryRPC, and WorkflowRPC operations on the CBO.

The wrapper class can fetch `billto/shipto` address, `commitlog`, `contact`, `emaillog`, `noteslog`, `onsitelog`, `phonelog`, `researchlog`, `site` and `sitepart` associated with a given case. The wrapper class also performs `Accept`, `Assign`, `Dispatch`, `Forward`, `Move`, `Reject` and `Yank` workflow operations for a given case.

Communication_Wrapper

This class is used as a wrapper over `Communication CBO` and it extends the `Communication` class. It can be used to perform `Create`, `Update` and `QueryRPC` operations on the CBO.

The `Communication` wrapper class can fetch and modify communication information from the `Communication` table (like address, com role or information related to email communication).

Contact_Wrapper

This class is used as a wrapper over the `Contact CBO` and it extends the `Contact` class. The `Contact` wrapper class is customized to perform `create`, `update`, `QueryRPC`, and `WorkflowRPC` operations on the CBO.

The `Contact` wrapper class can fetch and modify contact information from the `contact` table (like account information, action items, role, credit card information, email logs, alerts, opportunities, phone logs, Quote, sites associated with contact). The wrapper can also be used to associate or disassociate a contact with a site with a given specific role. Updates site information for a particular contact. Logs a phone call.

CreditCard_Wrapper

This class is used as a wrapper over the `CreditCard CBO` and it extends the `CreditCard` class. The `CreditCard` wrapper class is customized to perform `create`, `update`, `QueryRPC`, and `WorkflowRPC` operations on the CBO.

DemandDetail_Wrapper

This class is used as a wrapper over the `DemandDetail CBO` and it extends the `DemandDetail` class. The `DemandDetail` wrapper class is customized to perform `QueryRPC`, and `WorkflowRPC` operations on the CBO.

Limitation

Create and update operations are not supported by the wrapper class because of the limitation of the adapter to perform these operations. The `DemandDetail` CBO is basically intended for use with eOrder web application and provides basic functionality for fulfilling online orders. The CBO is basically used in conjunction with the `demandHeader` Business Object to define part requests for a customer order.

The wrapper class can be customized to only fetch contained objects which provide details like current state of part request etc. The wrapper class can be customized to update the existing content of the contained class only.

DemandHeader_Wrapper

This class is used as a wrapper over the `DemandHeader` CBO and it extends the `DemandHeader` class. The `DemandHeader` wrapper class is customized to perform Create, Update, QueryRPC, and WorkflowRPC operations on the CBO.

The wrapper class is also customized to create a new demand header, establish user's shipping and billing information, and generate part requests for the items in the user's order. The wrapper class can fetch information like shipping and billing information and the details of the person who requested the parts.

Limitation

The behavior of the `DemandHeader` business object has been tailored for use with eOrder. You should use it only to initiate order fulfillment. Do not attempt to use the business object to create other types of part requests in ClearLogistics.

Dialogue_Wrapper

This class is used as a wrapper over the `Dialogue` CBO and it extends the `Dialogue` class. The `Dialogue` wrapper class is customized to perform QueryRPC, and WorkflowRPC operations on the CBO.

The class can fetch the `Communication` Business Object associated with the `Dialogue` Business Object. The wrapper class also supports the workflow operations like Assign, Dispatch and Yank workflow operations for a given case.

Limitation

The wrapper class cannot be used to support create and update operations because of the limitation of the adapter to support these operations on the `Dialogue` CBO.

The assign operation can be performed only if the incoming request contains information about the user to whom the object has to be assigned.

Dispatch and Forward operations can be performed only if the incoming request contains information about a Queue to which the object has to be dispatched.

DocInst_Wrapper

This class is used as a wrapper over the DocInst CBO and it extends the DocInst class. The DocInst wrapper class is customized to perform create, update, QueryRPC, WorkflowRPC and CustomizedRPC operations on the CBO. The wrapper class can create new attachments, get attachment objects such as case, subcase, change request and site. The wrapper class fetches information like path of storage of the attachment etc.

Generic_Edr_Com_Role Wrapper

This class is used as a wrapper over Generic CBO and it extends the Generic class. It is customized to perform Create, Update and QueryRPC operations on the CBO.

This wrapper class can fetch the information about Address BO associated with a ComRole BO.

NotesLog_Wrapper

NotesLog wrapper is used as a wrapper over NoteLog CBO and it extends NotesLog class. It can be used to perform create, update and queryRPC operations on the CBO.

This can create a new notes log or fetch the existing noteslog for a case, contract or subcase.

OEQuote_Wrapper

This class is used as a wrapper over the OEQuote CBO and it extends the OEQuote Class. The OEQuote wrapper class is customized to perform create, update, QueryRPC, WorkflowRPC and CustomizedRPC operations on the CBO.

The wrapper class can be used to access the contract table. It can be customized to provide a list of products and options being purchased, price of items, price adjustments like discounts and surcharge, organization or individual purchase item, shipping and billing information for the organization or the user, payment information, phone logs, email logs and notes associated with the order.

The wrapper class can be customized to access information in OEQuote table in addition to the contract table. It is also customized to perform workflow operations like dispatch, assign, accept, and yank quotes and part requests.

Limitation

The Dispatch and Forward operations can be performed only if the incoming request contains information about a Queue to which the object has to be dispatched.

Assign operation can be performed only if the incoming request contains information about the user to whom the object has to be assigned.

PartPrice_Wrapper

This class is used as a wrapper over the PricePart CBO and it extends the PricePart class. The PricePart wrapper class is customized to perform create, update, QueryRPC, WorkflowRPC and CustomizedRPC operations on the CBO.

The wrapper class is customized to get pricing information for a specific part. The wrapper class can be customized to access contained objects of the PricePart to list the parts, currency, and price book associated with price instance.

Limitation

Although you can use this business object to add new price instance information to the database, you must be very careful while doing so. This business object does not validate the price instance data you add. To make sure your price instance data is validated properly, use the Product Manager application to enter your data.

Part_Wrapper

This class is used as a wrapper over the Part CBO and it extends the Part Class. The Part wrapper class is customized to perform create, update, QueryRPC, WorkflowRPC and CustomizedRPC operations on the CBO. The class is also customized to get information from part_num table about the part number or child part.

ServicePart_Wrapper

This class is used as a wrapper over the ServicePart CBO and it extends the ServicePart Class. The ServicePart wrapper class is customized to perform create, update, QueryRPC, WorkflowRPC and CustomizedRPC operations on the CBO. The wrapper class can be customized to access the site_part table in the ClarifyCRM system. The wrapper can also get the related sites and related products for a service part.

ShopList_Wrapper

This class is used as a wrapper over the ShopList CBO and it extends the ShopList Class. The ShopList wrapper class is customized to perform create, update, QueryRPC, WorkflowRPC and CustomizedRPC operations on the CBO.

Site_Wrapper

This class is used as a wrapper over the Site CBO and it extends the Site Class. The Site wrapper class is customized to perform create, update, QueryRPC, WorkflowRPC and CustomizedRPC operations on the CBO.

The wrapper class is also customized to perform operations like create new site, get and modify site information in the site table of the ClarifyCRM system. The class can get and change the address and primary business organization associated with the site, install or link a part at a site and view existing log entries for a site.

SubCase_Wrapper

This class is used as a wrapper over the SubCase CBO and it extends the SubCase class. The SubCase wrapper class is customized to perform create, update, QueryRPC, WorkflowRPC and CustomizedRPC operations on the CBO. The SubCase wrapper class is customized to fetch address, closelog, commitlog, contact, emaillog, noteslog, onsitelog, phonelog, researchlog, site and sitepart associated with the SubCase. Also customized to perform workflow operations like Accept, Assign, Dispatch, Forward, Move, Reject, and Yank operations.

Task_Wrapper

This class represents the task table of the Generic Business Object. It extends Generic class, which contains information about other business objects and action items associated with business organizations. The Task_Wrapper class can be used to perform Create and QueryRPC operations on the CBO.

It has a **task_for2bus_org**, relationship with Bus_Org CBO. The Bus_Org CBO is a contained CBO of the Task_Wrapper class.

User_Wrapper

This class is used as a wrapper over User CBO and it extends User class. It can be used to perform Create, Update and QueryRPC operations on the CBO.

It can also be used to fetch the details of current user's site,skills and also agent, employee, wipbin and actions of the current user.



The wrappers can be enhanced to do certain customized operations by implementing the customizedRPC method in the wrapper. The Case wrapper provides a reference custom operation implementation. The custom operation in the Case wrapper populates the `tibco_ex_case_history` table with the values from case table.

Wrapper Class and TibcoCBOInterface

Wrapper classes are developed to make the task of creating or updating a CBO easier.

The wrapper class extends an interface called `TibcoCBOInterface`. The usage of `TibcoCBOInterface` ensures that all the wrapper classes have a similar structure. The defined structure acts as a template, enabling you to write a new wrapper class.

The interface contains a list of methods, which a wrapper class must implement. These methods are:

1. **Init** — This method creates an object of the wrapper class and the handle is used throughout the session. For wrappers extending `Generic` class, after creating an object of the wrapper class and creating a handle to it, call `setDBObjectName()` present in `Generic` class to specify the table or view to which this CBO provides access. Please refer `Generic_Edr_Com_Role_Wrapper` class provided in the adapter samples for more details.
2. **Create** — This method, when implemented by a wrapper class, should contain the business logic to create a CBO. For create operation, call `GenericHandle2CBO` defined in `ClarifyAppServer` class to get an object reference for the contained CBO. The details provided for Contained CBO in the input should be that of an existing entry in the database. For example, to create a new contact, the details of an existing site have to be provided in the input.
3. **Update** — This method, when implemented by a wrapper class, should contain the business logic to update a CBO.
4. **queryRPC** — This method should be implemented by all the wrapper classes in such a way that it should incorporate the logic to extract details from the `ClarifyCRM` system for a particular Business Object based on the user request.
5. **workflowRPC** — This method should be implemented by all the wrapper classes in such a way that it should incorporate the logic to perform workflow operations on `Clarify` workflow Business Objects.
6. **customizedRPC** — This method should be implemented by all the wrapper classes in such a way that it should incorporate the logic to perform customized operations on CBOs.

Writing Wrapper Class for Case Business Object

Steps Involved

Example:

1. The class implements `TibcoCBOInterface`.
2. The class extends the derived CBO `Case`.

```
Public class Case_Wrapper implements TibcoCBOInterface extends
Case
```

3. Implement the method `Init()` of the Interface. The method creates an object of the wrapper class and creates a handle for it.

```
Case_Wrapper boCase = (Case_Wrapper)this;
```

4. Implement the method `create()` of the interface. When invoked, this method inserts a new row for a business object, along with the details in the corresponding table in the ClarifyCRM system. The input parameters for this method carries all the information required to perform the operation.

Algorithm:

- Get an object reference for the contained CBO. For `Case`, the contained Business Object is `Contact`. This can be done by calling a method `getHandle2CBO` defined in `ClarifyAppServer` class. `ClarifyAppServer` class contains all the common utility functions used in the adapter. The function `getHandle2CBO` takes two parameters, one holding the underlying table name of the business object and the other pointing to its Derived Business Object Name. For contact business object, the table name is `contact` and its business object name is `Contact`.
- Invoke `addNew()` method to add a new row to the business object. This method assigns a temporary object `ID` to the new row. When the object is updated for the first time, a permanent object `ID` will be assigned to the row before committing the changes to the database.
- Call `setCBOValues` method defined in the `ClarifyAppServer` to fill the field values in the newly added row. The field values will be taken from the incoming request. If the incoming request does not contain a value for a field, its value is set to the default value (if there is any). Otherwise it is set to null.
- To associate the newly created case with the contact, use `getContact().add()` passing the contact object reference.
- Call `update()` method to commit the changes to the database.

5. Implement the `update()` method of the interface. When invoked, this method will update the information associated with a business object. The adapter provides functionality to change the field values associated with a business object. It does not support changing the reference associated with contained/child business objects.

Algorithm:

- Get a handle to the Case CBO to be updated. This can be done by calling `getMainCBO()` of `ClarifyAppServer` class by passing the Case business object handle as a parameter.
 - Call `setCBOValues()` to update the field values.
 - Invoke `update()` to commit the changes to the database.
6. Implement the `queryRPC()` method of the interface. This method fetches information about the child business objects (or contained CBO(s)) based on the request and sends the result back to the calling function. The adapter provides support to fetch information about the contained CBOs and if you want to get data associated with field values of a CBO, you have to customize the wrapper class. The incoming request should contain the name of the method to be invoked to perform the query operation. For example, if you want to get information about the Site associated with a Case, the method name to be used is `getsite`. The method name should reflect the name of the method defined in the Case business object's list of methods.

Algorithm:

- Based on the method name, invoke the corresponding method. For example, if the method to be invoked is `getsite`, invoke `fetchSite()`.
- Return the result of the operation to the calling function.

Function — fetchSite():

- Algorithm:**
- Get a handle to Case Business object by calling `getMainCBO()`. This handle should point to a specific row in the business object.
 - Use Site property defined in the Case Business object to complete the operation. The site property contains a reference to a Generic business object, which points to the Site that reported the Case. `boCase.getSite()`;



When a query operation is performed on a business object, it returns information about the main business object as well as its child business object. However, for some child business objects, the query mode is disabled by default. The reason is the query might fetch many rows. In that case, the query mode for that child business object must be enabled before doing the query operation.

For example, the email log associated with a case business object can be extracted by using the `EmailLog` property. However, the query mode of this business object is set to `Disabled` by default. When a query operation is performed in Case business object, the result of the query does not contain any reference to the email log business object. To access the email logs, you have to explicitly set the Query Mode property to `cboSubmitEnabled`.

```
boCase.getEmailLog().setQueryMode(setQueryMode(CboConstants.cboSubmitEnabled));
```

7. Implement the `workflowRPC()` method of the interface. This method performs workflow actions on CBOs. Workflow operations require information about user, queue and WIPBIN objects.

- Algorithm:**
- Get a handle to the Case Business Object on which the workflow operation has to be performed.
 - Get a handle to user, queue and WIPBIN objects.
 - Based on the incoming request, invoke a function defined in the wrapper. For example, if the incoming request requires the `Assign` workflow operation to be performed on a Case business object, invoke `performAssign()` defined in the Case Wrapper. This method contains the business logic to perform `Assign` operation.
 - Return the result to the calling function.

Function — performAccept():

This method accepts a workflow item into a specified WIPBIN.

- Check whether the incoming request contains any data about the WIPBIN.
- If it has any, accept the workflow business object in the specified WIPBIN.

The `accept` operation can be invoked by using `accept` method of the workflow abstract class:

```
boCase.accept (Base WIPBIN, Boolean isTemporary)
```

where

`boCase` is a handle to the `Case` business object.

`WIPBIN` represents a handle to the `WIPBIN` object.

`Boolean` when set to `true`, the workflow item will be temporarily accepted in the specified `WIPBIN`. Else move the object to the default `WIPBIN`.

`boCase.accept ()` returns the status of the operation to calling function.

Function — `performAssign()`:

This method assigns a workflow item to a specified user and places the item in the specified `WIPBIN`. Input parameters are:

- Handle to a User Object
- Handle to a `WIPBIN` Object



The `assign` operation can be performed on a workflow object only by the owner of the object. If the logged in user is not the owner, the adapter will not be able to perform the workflow operation on that object.

Before performing a workflow operation the adapter checks whether the logged in user is the owner. If the owner is the logged in user, the adapter invokes the operation to satisfy the request. If not, the adapter changes the ownership of the workflow object and then performs the workflow operation.

The `assign` operation requires information about the `WIPBIN` and the user. If the incoming request does not have details about the `WIPBIN` or if the details furnished by the incoming request do not point to any specific `WIPBIN`, the adapter assigns the workflow object to the user's default `WIPBIN`.

Algorithm:

- Get a handle to the user and `WIPBIN` to which the workflow object has to be assigned.
- Check whether the logged in user is the owner of the object. If not change the ownership of the workflow object.
- If the incoming request does not contain information about the user, send a reply back to the calling function specifying that the request does not contain the required information.
- Check whether the request contains information about the `WIPBIN`. If it points to a specific `WIPBIN`, assign the workflow object to it. Assign

operation can be performed by calling `assign` method of the workflow abstract class.

```
boCase.assign (Base User, Base WIPBIN)
```

- Else assign the workflow object to the default WIPBIN.

```
boCase.assign (Base User)
```

- Return the status of the operation to the calling function.

Function — `performDispatch()`:

This method dispatches a workflow item to a specified queue. The input parameter is

- Handle to a Queue Object



The dispatch operation can be performed on a workflow object only by the owner of the object. If the logged in user is not the owner, the adapter will not be able to perform the workflow operation on that object.

Before performing a workflow operation, the adapter checks whether the logged in user is the owner. If the owner is the logged in user, the adapter invokes the operation to satisfy the request. If not, the adapter changes the ownership of the workflow object and then performs the workflow operation.

The dispatch operation requires information about a Queue object to which the workflow object has to be dispatched.

Algorithm:

- Get a handle to the Queue to which the workflow object has to be dispatched.
- Check whether the logged in user is the owner of the object. If not change the ownership of the workflow object.
- If the incoming request does not contain information about the queue, send a reply back to the calling function specifying that the request does not contain the required information.
- Else dispatch the workflow object to the specified queue

```
boCase.dispatch (Base Queue)
```
- Return the status of the operation to the calling function.
- Implement the `customizedRPC()` method of the interface. This method calls a function based on the user request, which then performs the action defined in it. The adapter supports customized operation on Case Business Object. If you want to implement any functionality which the adapter

doesn't support, you have to write a method implementing your requirements and that method can be invoked as and when required.

Example: The adapter supports a method `insert2CaseHistory`, which adds a new row in a table called `table_tibco_ex_case_history` based on the field values that come with the incoming Case object.

Function — `performForward()`:

This method forwards a workflow item from its current queue to a specified queue.

The input parameter is:

- Handle to a Queue Object



Forward operation can be performed on a workflow object only by the owner of the object. The object must also be in a queue possessed by the owner. If the logged in user is not the owner, the adapter will be unable to perform the workflow operation on that object.

Before performing a workflow operation, the adapter checks whether the logged in user is the owner. If the owner is same as logged in user, the adapter invokes the operation to satisfy the request. If not, the adapter changes the ownership of the workflow object and also changes the queue in which the object resides to the one associated with the logged in user. It then performs the workflow operation.

- Algorithm:**
- Get a handle to the Queue to which the workflow object has to be forwarded.
 - Check whether the logged in user is the owner of the object. If not, change the ownership of the workflow object.
 - Change the queue associated with the object.
 - If the incoming request does not contain information about the queue, send a reply back to the calling function specifying that the request does not contain the required information
 - Else forward the workflow object to the specified queue. Forward operation can be performed by invoking forward method of the abstract workflow class.
- ```
boCase.forward(Base Queue)
```
- where
- boCase is a handle to the Case business object.
- Queue is a handle to the Queue object.
- Return the status of the operation to the calling function.

### Function — performMove():

This method moves a workflow item from its current WIPBIN to a specified WIPBIN.

The input parameters are:

- Handle to a WIPBIN Object



The move operation can be performed on a workflow object only by the owner of the object. If the logged in user is not the owner, the adapter does not perform the workflow operation on that object.

Before performing a workflow operation, the adapter checks whether the logged in user is the owner. If the owner is the logged in user, the adapter invokes the operation to satisfy the request. If not, the adapter changes the ownership of the workflow object and then performs the workflow operation.

The move operation requires information about a WIPBIN object to which the workflow object has to be moved. If the incoming request does not possess information about the WIPBIN object, the workflow object will be forwarded to the default WIPBIN.

- Algorithm:**
- Get a handle to the WIPBIN to which the workflow object has to be moved.
  - Check whether the logged in user is the owner of the object. If not change the ownership of the workflow object.
  - If the incoming request contains information about the WIPBIN, move the object to the specified WIPBIN. Move operation can be performed by using move method of the workflow abstract class.
- ```
boCase.move(Base WIPBIN)
```
- where
- boCase is a handle to the Case business object.
- WIPBIN represents a handle to the WIPBIN object.
- Else move the workflow object to the default WIPBIN
- ```
boCase.Move()
```
- Return the status of the operation to the calling function.

### Function — performReject():

This method rejects a workflow item from the current queue. This operation does not take any input.



The reject operation can be performed on a workflow object only by the owner of the object. The object must be in a queue possessed by the owner. If the logged in user is not the owner, the adapter does not perform the workflow operation on that object.

Before performing a workflow operation, the adapter checks whether the logged in user is the owner. If the owner is the logged in user, the adapter invokes the operation to satisfy the request. If not, the adapter changes the ownership of the workflow object and also changes the queue in which the object resides to the one associated with the logged in user.

Before performing a reject operation, RejectMsgLog business object has to be updated. Use RejectMsg property to update the RejectMsgLog business object. After updating it, perform reject workflow operation.

- Algorithm:**
- Check whether the logged in user is the owner of the object. If not change the ownership of the workflow object.
  - Change the queue associated with the object.
  - Update the RejectMsgLog business object
  - Reject the workflow object. This can be done by using reject method of the workflow abstract class
- ```
boCase.reject(Base Queue)
```
- where
- boCase is a handle to the Case business object.
- Queue is a handle to the Queue object.
- Return the status of the operation to the calling function.

Function — performYank():

This method yanks a workflow item and places it in the current user's default WIPBIN. It does not take any parameters.

- Algorithm:**
- The Yank operation can be invoked by using yank method of the workflow abstract class.
- ```
boCase.yank()
```
- Return the status of the operation to calling function.
  - Implement the customizedRPC() method of the interface. This method calls a function based on the user request, which performs the action defined in it. The adapter supports customized operation on Case Business Object. If you want to implement any functionality which the adapter does not support, you have to write a method implementing your requirements and invoke that method as and when required.

**Example:** The adapter supports a method insert2CaseHistory which adds a new row in a table called `table_tibco_ex_case_history` based on the field values that come with the incoming Case business object.

**Function — insert2CaseHistory:**

- Algorithm**
- Get a handle to a Base business object pointing to the table `table_tibco_ex_case_history`.
  - Add a new row to the business object using `addNew()`.
  - Set the values for all the fields by extracting the data from the Case business object.
  - Update the field values in the current row of the business object to the database by invoking `update()` method.

## Appendix F    **Testing the Adapter**

This chapter describes how to verify that the adapter is installed and configured properly. Tests to verify importing and updating a site in ClarifyCRM are provided in this chapter.

### Topics

---

- *Testing the Publication Service with ClearBasic Forms, page 332*
- *Testing the Publication Service with Web Forms, page 336*
- *Testing the Request-Response Functionality with Web Forms, page 339*
- *Testing the Request-Response Functionality with ClearBasic Forms, page 340*

## Testing the Publication Service with ClearBasic Forms

---

Five ClearBasic files are provided along with the adapter to test the Publication Service functionality. Before you run the Publication Service, you must import all ClearBasic files using ClarifyCRM tools.

The ClearBasic forms are listed below and are available in the `samples/testPublisher` directory.

- `tibco616.cbs`
- `tibco712.cbs`
- `tibco717.cbs`
- `tibco624.cbs`
- `tibco672.cbs`

You must use ClarifyCRM client-side tools to import the ClearBasic forms into the ClarifyCRM system to be used by the Publication Service.

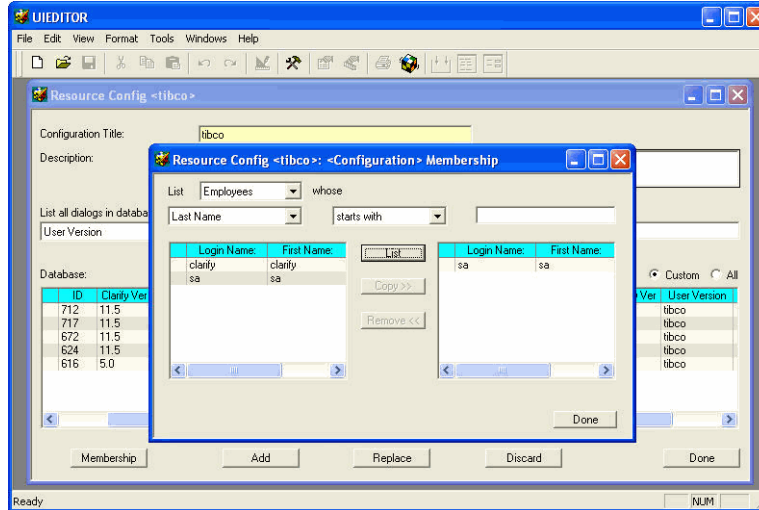
See `readme.txt` in the `samples/testPublisher` directory for additional information.

### Importing ClearBasic Files

The installation procedure assumes you are familiar with the Clarify User Interface Editor and ClearBasic Exchange (cbex). For more information about these tools, see the *Clarify User Interface Editor Guide* and *ClearBasic's Programmer's Guide*.

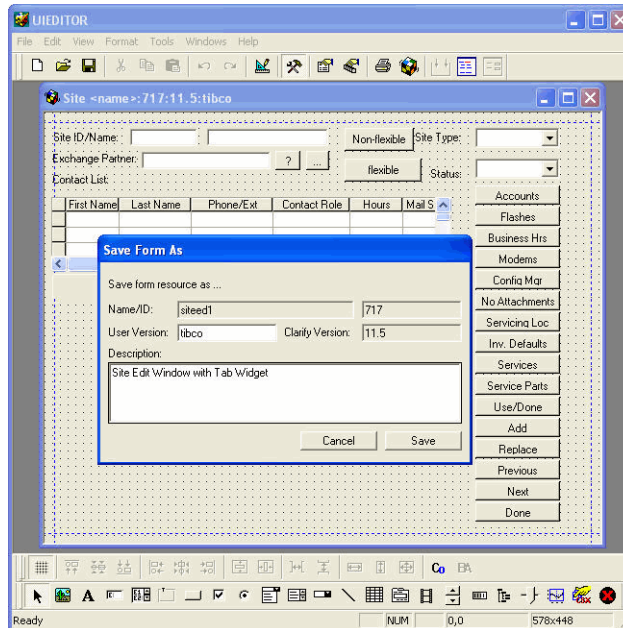
1. Use the `UIEditor` to create a Resource Configuration for those users who will publish data from your forms.
2. Add those users to the Resource Configuration.

Figure 63 Add users to resource configuration



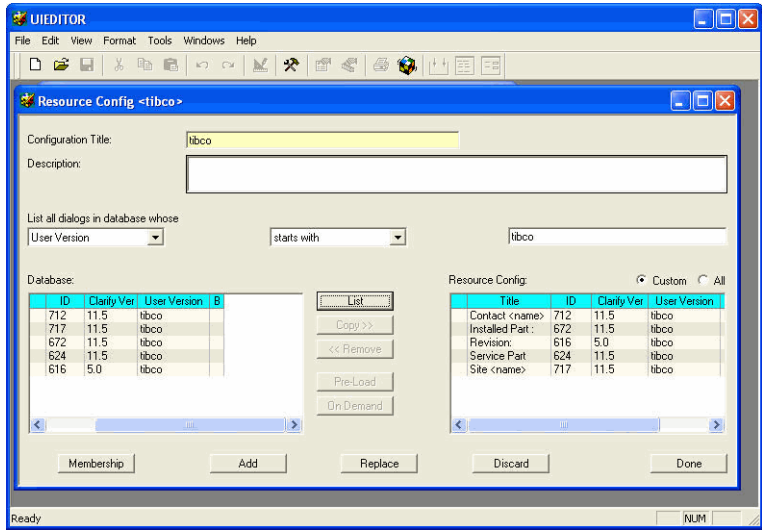
3. Use the UIEditor to create a customized version of the form for which TIBCO customized code is provided in the samples/testPublisher directory as separate cbs files.

Figure 64 Create customized form



4. Add these forms to the Resource Configuration you created in step 1.

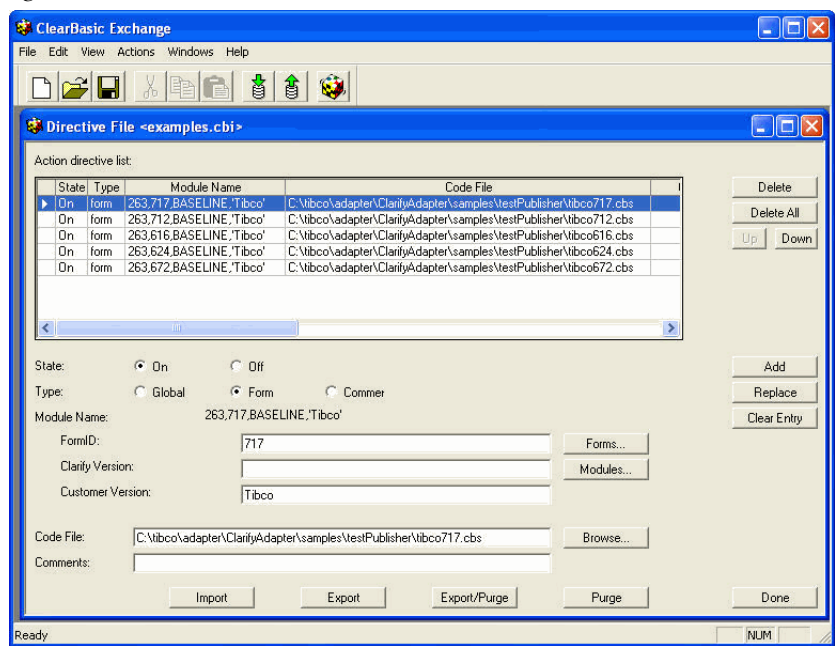
Figure 65 Add a form to resource configuration



5. Import the forms into the Clarify database by entering the command:  
**cbex -dir examples.cbi**

The following Directive File window appears.

Figure 66 Directive window





6. Verify that the code file path is correct. If it is correct, go to step 8.
7. If the code file path is not correct, use the **Browse** button to select the correct path. Then click **Replace** button to replace the previous path. Continue with step 8.
8. Click **Import**.



If the ClarifyCRM system already has ClearBasic customization on the selected forms, you must merge the script provided by TIBCO with your customization.

## Testing the Publication Service with Web Forms

Two JSP files are provided along with the adapter to test the Publication Service functionality. Before you run the Publication Service, you must put the two JSP files into the eBusiness Framework installation.

JSP forms are:

- `OrderSubmit.jsp`
- `reg_do_action_1.jsp`

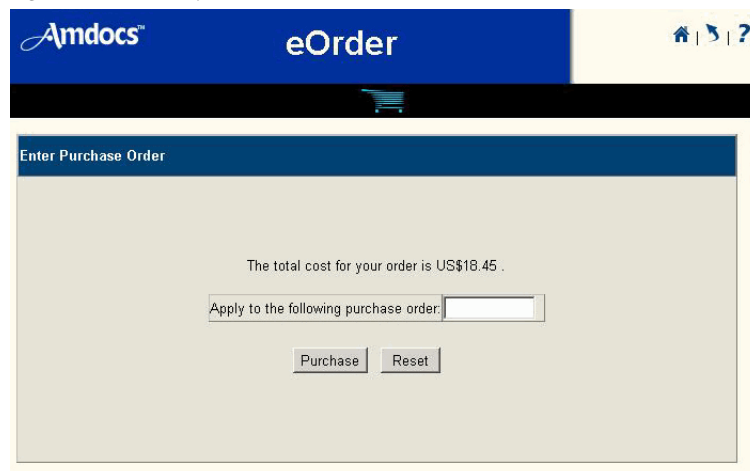
See `readme.txt` in the `samples/testPublisher` directory for details.

1. Make sure that your eBusiness framework installation is working fine.
2. Add the `TibcoCboInterface.jar` to the web server classpath.
3. Replace the `OrderSubmit.jsp` in the `eOrderJSP` folder of the eBusiness Framework installation with the `OrderSubmit.jsp` file given in the adapter package.
4. Replace the `reg_do_action_1.jsp` in the `PortalJSP` folder eBusiness Framework installation with the `reg_do_action_1.jsp` file given in the adapter package.
5. Test the creation of the order.

### To test the creation of the order:

1. Run the eOrder application and create an order.

Figure 67 Enter purchase order



The screenshot displays the eOrder application interface. At the top, there is a blue header bar with the 'Amdocs' logo on the left and 'eOrder' in the center. To the right of the header, there are navigation icons: a home icon, a magnifying glass, and a question mark. Below the header is a black bar with a shopping cart icon. The main content area is titled 'Enter Purchase Order' in a blue bar. Below this, the text 'The total cost for your order is US\$18.45 .' is displayed. Underneath, there is a label 'Apply to the following purchase order:' followed by a text input field. At the bottom of the form, there are two buttons: 'Purchase' and 'Reset'.

2. Fill all the required information.
3. Click the **Purchase** button. This will publish the purchase order along with some related objects on the TIBCO Rendezvous.
4. You can view the published message on the adapter console.

### To test the creation of a case:

1. Run the Customer Portal application and move to the B2B Admin User Registration page.

Figure 68 B2B admin user registration (view 1)

Amdocs Customer Portal ?

Amdocs -- B2B Admin User Registration

If you will not be managing user accounts, please go to [consumer registration](#).  
Please note that \* denotes a required field.

**Personal Information:**

First Name:  \*

Last Name:  \*

Title:  (e.g., Mr.)

Phone:  \*

Fax:  \*

Email:  \*

User Name:  \*

Password:  \*

Re-enter Password:  \*

Regional Setting:  \*

Time Zone:  \*

Figure 69 B2B admin user registration (view 2)

Mailing Address Line 1:  \*

Mailing Address Line 2:  \*

City:  \*

State, Territory, or Province:  \*

Zip/Postal Code:  \*

Country: USA ▼ \*

Site Shipping Method: Please Specify ▼ \*

SITE "BILL TO" ADDRESS

☒ Use Mailing Address

☐ Use "Ship To" Address

☐ Fill in below

Mailing Address Line 1:  \*

Mailing Address Line 2:  \*

City:  \*

State, Territory, or Province:  \*

Zip/Postal Code:  \*

Country: USA ▼ \*

\* Denotes Required Field

2. Fill all the required information.
3. Click **SubmitToCSR** to publish a new case along with its related objects on the TIBCO Rendezvous.
4. You can view the published message on the adapter console.

## Testing the Request-Response Functionality with Web Forms

A JSP page, `OrderSubmit.jsp` page, is provided along with the adapter installation to test the Request-Response functionality. Place the `ordersubmit.jsp` into the eBusiness Framework installation.

See `readme.txt` in the `samples/testPublisher` directory for details.

1. Make sure that your eBusiness framework installation is working fine.
2. Add the `RPCGlobalModule.jar` to the web server classpath.
3. Replace the `OrderSubmit.jsp` in the `eOrderJSP` folder of the eBusiness framework installation with the `OrderSubmit.jsp` file given in the Clarify adapter package.
4. Test the creation of the order.

### To test the creation of the order:

1. Run the eOrder application and create an order.
2. Fill all the required information and click the **Submit Order** button. This will publish the purchase order along with some related objects on the TIBCO environment.

Figure 70 Clarify eOrder cart

The screenshot displays the 'eOrder' application interface. At the top, there is a blue header with the 'Amdocs' logo and the text 'eOrder'. Below this is a black navigation bar with a shopping cart icon. The main content area is titled 'Checkout' and contains a table summarizing the order details.

| Description                   | Part Number | Unit Price | Qty. | Item Total     |
|-------------------------------|-------------|------------|------|----------------|
| NewPart                       | 10          | US\$10.00  | 1    | US\$10.00      |
| Sub Total:                    |             |            |      | US\$10.00      |
| Tax:                          |             |            |      | US\$1.00       |
| * (change) Shipping -- 2 Day: |             |            |      | US\$3.25       |
| Handling:                     |             |            |      | US\$4.20       |
| Total:                        |             |            |      | US\$18.45      |
| * Purchase Using:             |             |            |      | Purchase Order |

Below the table, there are two address sections:

| Billing Address (change)                             | Shipping Address (change)                            |
|------------------------------------------------------|------------------------------------------------------|
| Main Office<br>111 Main Street<br>Any Town, CA 00000 | Main Office<br>111 Main Street<br>Any Town, CA 00000 |
| * Attn: New                                          | * Attn: New                                          |

At the bottom of the form, there is a button labeled 'Submit Order -->'.

3. You can view the published message on the adapter console.

## Testing the Request-Response Functionality with ClearBasic Forms

---

A ClearBasic form, `RPCSampleSite.cbs`, is provided along with the adapter installation to test the Request-Response functionality. Using the example, set up one of the ClarifyCRM forms along the lines illustrated in the example. The location of the sample ClearBasic form is given next.

`TIBCO_HOME/adapter/adclycrm/<version_num>/samples/testRPCCClient`

1. Before running the test, ensure that you have completed certain prerequisites. See the following sections for details:
  - Prerequisites for Setting Up Request-Response Invocation Functionality using Clarify ClearBasic Forms in a Flexible Deployment on page 103
  - Prerequisites for Setting Up Request-Response Invocation Functionality Using Clarify ClearBasic Forms in a Non-Flexible Deployment on page 106
2. Make sure that your eBusiness framework installation is working fine.
3. Add the global RPC module to the ClarifyCRM Applications.
4. Add the code to send the TIBCO Rendezvous message in the form.

Test for the message you have configured to be published by listening (using `tibrvlisten` or `rvstream`) on that subject.

# TIBCO Software Inc. End User License Agreement

**READ THIS END USER LICENSE AGREEMENT CAREFULLY. BY DOWNLOADING OR INSTALLING THE SOFTWARE, YOU AGREE TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO THESE TERMS, DO NOT DOWNLOAD OR INSTALL THE SOFTWARE AND RETURN IT TO THE VENDOR FROM WHICH IT WAS PURCHASED.**

*Upon your acceptance as indicated above, the following shall govern your use of the Software except to the extent all or any portion of the Software (a) is subject to a separate written agreement, or (b) is provided by a third party under the terms set forth in an Addenda at the end of this Agreement, in which case the terms of such addenda shall control over inconsistent terms with regard to such portion(s).*

**License Grant.** The Software is the property of TIBCO or its licensors and is protected by copyright and other laws. While TIBCO continues to own the Software, TIBCO hereby grants to Customer a limited, non-transferable, non-exclusive, license to use the Number of Units set forth in the Ordering Document solely for Customer's internal business use.

**License Term.** The term of each license for the Software shall be either perpetual or on a term limited basis as set forth in the Ordering Document. If licensed on a term limited basis, the term shall be for a period of three years commencing on the Purchase Date, and on expiration, Customer must cease using and return or destroy all copies of the Software.

**Restrictions.** Customer agrees not to (a) make more copies than the Number of Units (except for a reasonable number of copies for archival and disaster recovery purposes) or use any unlicensed versions of the Software; (b) provide access to the Software to anyone other than employees, contractors, or consultants under written contract with Customer agreeing to be bound by terms at least as protective of TIBCO as those in this End User License Agreement ("Authorized Users"); (c) sublicense, transfer, assign, distribute to any third party, pledge, lease, rent, or commercially share the Software or any of Customer's rights under this Agreement (for the purposes of the foregoing a change in control of Customer is deemed to be an assignment); (d) use the Software for purposes of providing a service bureau, including, without limitation, providing third-party hosting, or third-party application integration or application service provider-type services, or any similar services; (e) use the Software in connection with ultrahazardous activities, or any activity for which failure of the Software might result in death or serious bodily injury to Customer or a third party; or (f) directly or indirectly, in whole or in part, modify, translate, reverse engineer, decrypt, decompile, disassemble, make error corrections to, create derivative works based on, or otherwise attempt to discover the source code or underlying ideas or algorithms of the Software. Customer may engage in such conduct as is necessary to ensure the interoperability of the Software as required by law, provided that prior to commencing any decompilation or reverse engineering of any Software, Customer agrees to it shall observe strict obligations of confidentiality and provide TIBCO reasonable advance written notice and the opportunity to assist with and/or conduct such activity on Customer's behalf and at Customer's expense; and (g) notwithstanding the method of delivery of the Software, the scope of the licenses granted under this Agreement are limited to the Software as set forth in an Ordering Document.

**Beta and Evaluation Licenses.** Notwithstanding the foregoing, if the Software is being provided for demonstration, beta testing, or evaluation purposes, then Customer agrees (a) to use the Software solely for such purposes, (b) that the Software will not be used or

deployed in a production or development environment, and (c) that such use shall automatically terminate upon the earlier of thirty days from the date Customer receives the right to install the Software, or Customer's receipt of notice of termination from TIBCO.

**Maintenance.** Maintenance, if ordered, including first year and all subsequent years, is provided under the policies set forth in the Maintenance Program Guide in effect at the time Maintenance services are provided. The policies set forth in the Maintenance Program Guide, incorporated in this Agreement, are subject to change at TIBCO's discretion; however the level of Maintenance service provided by TIBCO will not be materially reduced during the period for which Maintenance fees have been paid by Customer. The current version of the Maintenance Program Guide can be accessed at <http://www.tibco.com/services/support/default.jsp>. To receive Maintenance, all Software must be properly licensed and annual Maintenance fees paid. TIBCO is not obligated to continue providing Maintenance if annual Maintenance fees have not been paid.

**Services.** Customer may request additional services ("Services") either in an Ordering Document, or by a separate mutually executed work order, statement of work or other work-request document incorporating the term of this End User License Agreement (each, a "Work Order"). Unless otherwise expressly agreed to in a Work Order, all Services and any work product therefrom shall be (a) performed on a time and materials basis, plus meals, lodging, travel, and other expenses reasonably incurred in connection therewith, (b) deemed accepted upon delivery, and (c) exclusively owned by TIBCO (except for Confidential Information of Customer), including all right, title and intellectual property or other right or interest therein. Each Work Order is intended to constitute an independent and distinct agreement of the parties, notwithstanding that each shall be construed to incorporate all applicable provisions of this End User License Agreement.

**Limited Warranty.** If Customer obtained the Software directly from TIBCO, then TIBCO warrants that for a period of thirty (30) days from the Purchase Date: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software will substantially conform to its Documentation. This limited warranty extends only to the original Customer hereunder. Customer's sole and exclusive remedy and the entire liability of TIBCO and its licensors under this limited warranty will be, at TIBCO's option, repair, replacement, or refund of the Software and applicable Maintenance fees, in which event this End User License Agreement shall terminate upon refund thereof.

This warranty does not apply to any Software which (a) is licensed for beta, evaluation, testing or demonstration purposes for which TIBCO does not receive a license fee, (b) has been altered or modified, except by TIBCO, (c) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by TIBCO, (d) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (e) is used in violation of any other term of this End User License Agreement. Customer agrees to pay TIBCO for any Maintenance or Services provided by TIBCO related to a breach of the foregoing on a time, materials, travel, lodging and other reasonable expenses basis. If Customer obtained the Software from a TIBCO reseller or distributor, the terms of any warranty shall be as provided by such reseller or distributor, and TIBCO provides Customer no warranty with respect to such Software.

EXCEPT AS SPECIFIED IN THIS LIMITED WARRANTY, THE SOFTWARE, MAINTENANCE AND SERVICES ARE PROVIDED "AS IS", ALL EXPRESS OR IMPLIED CONDITIONS,

REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OR CONDITION OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, SATISFACTORY QUALITY OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW. CERTAIN THIRD PARTY SOFTWARE MAY BE PROVIDED TO CUSTOMER ALONG WITH CERTAIN TIBCO SOFTWARE AS AN ACCOMMODATION TO CUSTOMER. THIS THIRD PARTY SOFTWARE IS PROVIDED "AS IS". CUSTOMER MAY CHOOSE NOT TO USE THIRD PARTY SOFTWARE PROVIDED AS AN ACCOMMODATION BY TIBCO. NO WARRANTY IS MADE REGARDING THE RESULTS OF ANY SOFTWARE, MAINTENANCE OR SERVICES OR THAT THE SOFTWARE WILL OPERATE WITHOUT ERRORS, PROBLEMS OR INTERRUPTIONS, OR THAT ERRORS OR BUGS IN THE SOFTWARE WILL BE CORRECTED, OR THAT THE SOFTWARE'S FUNCTIONALITY, MAINTENANCE OR SERVICES WILL MEET CUSTOMER'S REQUIREMENTS. NO TIBCO DEALER, DISTRIBUTOR, AGENT OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATIONS, EXTENSIONS OR ADDITIONS TO THIS WARRANTY.

**Indemnity.** If Customer obtained the Software from TIBCO directly, then TIBCO agrees at its own expense to defend or, at its option, to settle, any claim or action brought against Customer to the extent it is based on a claim that the unmodified Software infringes any patent issued by the United States, Canada, Australia, Japan, or any member of the European Union, or any copyright, or any trade secret of a third party; and TIBCO will indemnify and hold Customer harmless from and against any damages, costs and fees reasonably incurred (including reasonable attorneys' fees) that are attributable to such claim or action and which are assessed against Customer in a final judgment; provided that TIBCO is promptly notified in writing of such claim, TIBCO has the exclusive right to control such defense and/or settlement, and Customer shall provide reasonable assistance (at TIBCO's expense) in the defense thereof. In no event shall Customer settle any claim, action or proceeding without TIBCO's prior written approval. In the event of any such claim, litigation or threat thereof, TIBCO, at its sole option and expense, shall (a) procure for Customer the right to continue to use the Software or (b) replace or modify the Software with functionally equivalent software. If such settlement or modification is not commercially reasonable (in the reasonable opinion of TIBCO), TIBCO may cancel this End User License Agreement upon sixty days prior written notice to Customer, and refund to Customer the unamortized portion of the license fees paid to TIBCO by Customer based on a five-year straight-line depreciation. This Section states the entire liability of TIBCO with respect to the infringement of any intellectual property rights, and Customer hereby expressly waives any other liabilities or obligations of TIBCO with respect thereto. The foregoing indemnity shall not apply to the extent any infringement could have been avoided by use of the then-current release.

**Limitation of Liability.** EXCEPT AS PROVIDED UNDER INDEMNITY OR RESULTING FROM A BREACH OF CONFIDENTIALITY (THE "EXCLUDED MATTERS"), IN NO EVENT WILL EITHER PARTY OR TIBCO'S LICENSORS BE LIABLE FOR ANY LOST DATA, LOST REVENUE, LOST PROFITS, DAMAGE TO REPUTATION, BUSINESS INTERRUPTION, OR ANY OTHER INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, PUNITIVE, EXEMPLARY OR ANY SIMILAR TYPE DAMAGES ARISING OUT OF THIS AGREEMENT, THE USE OR THE INABILITY TO USE THE SOFTWARE, OR THE PROVISION OF ANY MAINTENANCE OR SERVICES, EVEN IF A PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT FOR THE EXCLUDED MATTERS, IN NO EVENT SHALL A PARTY BE LIABLE TO THE OTHER, WHETHER IN CONTRACT, TORT (INCLUDING ACTIVE

OR PASSIVE NEGLIGENCE), BREACH OF WARRANTY, CLAIMS BY THIRD PARTIES OR OTHERWISE, EXCEED THE PRICE PAID BY CUSTOMER UNDER THE APPLICABLE ORDERING DOCUMENT.

THE FOREGOING LIMITATIONS SHALL APPLY EVEN IF THE ABOVE-STATED REMEDY OR LIMITED WARRANTY FAILS OF ITS ESSENTIAL PURPOSE. BECAUSE SOME STATES OR JURISDICTIONS DO NOT ALLOW LIMITATION OR EXCLUSION OF CONSEQUENTIAL OR INCIDENTAL DAMAGES, THE ABOVE LIMITATION MAY NOT APPLY TO CUSTOMER.

**Confidentiality.** "Confidential Information" means the terms of this End User License Agreement; all information marked by the disclosing party as proprietary or confidential; any provided software, related documentation or related performance test results derived by Customer; and any methods, concepts or processes utilized in provided software or related documentation. Confidential Information shall remain the sole property of the disclosing party and shall not be disclosed to any non-Authorized User of either TIBCO or Customer without the prior written consent of the disclosing party. If Confidential Information is communicated orally, such communication shall be confirmed as "Confidential" in writing within thirty days of such disclosure. The parties agree to protect the Confidential Information of the other in the same manner it protects the confidentiality of similar information and data of its own (and at all times exercising at least a reasonable degree of care). Except with respect to the Software, items will not be deemed Confidential Information if (i) available to the public other than by a breach of an agreement with TIBCO, (ii) rightfully received from a third party not in breach of any obligation of confidentiality, (iii) independently developed by one party without use of the Confidential Information of the other; (iv) known to the recipient at the time of disclosure (other than under a separate confidentiality obligation); or (v) produced in compliance with applicable law or court order, provided the other party is given reasonable notice of the same. Both parties agree to indemnify the other for any damages the other may sustain resulting from their unauthorized use and/or disclosure of the other's Confidential Information. Such damages shall include reasonable expenses incurred in seeking both legal and equitable remedies. To the extent required by law, at Customer's request, TIBCO shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of TIBCO's applicable fee. Customer agrees to observe obligations of confidentiality with respect to such information.

To the extent TIBCO is exposed to individual personal data owned or otherwise held by Customer during the provision of Services, which is subject to various data protection laws and/or regulations ("Protected Data"), TIBCO agrees to treat such Protected Data in accordance with the Customer Privacy and Security Statement set forth at [http://www.tibco.com/customer\\_privacy\\_security\\_statement.jsp](http://www.tibco.com/customer_privacy_security_statement.jsp) (the "Statement"). The policies and procedures set forth in the Statement as well as those set forth in the Data Protection Policy Statement at [http://www.tibco.com/resources/data\\_protection\\_statement.pdf](http://www.tibco.com/resources/data_protection_statement.pdf) are in place to meet TIBCO's obligations for the protection, integrity and confidentiality of any Protected Data which exceed TIBCO's standard obligations to safeguard Confidential Information.

**Export.** Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Customer agrees to comply strictly with all such regulations and agrees to obtain all necessary licenses to export, re-export, or import Software.



**Government Use.** If the Software, Maintenance or Services are being or have been acquired with U.S. Federal Government funds, or Customer is an agency, department, or other entity of the United States Government ("**Government**"), the use, duplication, reproduction, release, modification, disclosure or transfer of the Software, Maintenance or Services, or any related documentation of any kind, including technical data, or manuals, is restricted in accordance with Federal Acquisition Regulation 12.212 for civilian agencies and Defense Federal Acquisition Regulation Supplement 227.7202 for military agencies. The Software, Maintenance and Services are COMMERCIAL ITEMS AS DEFINED BY THE FEDERAL ACQUISITION REGULATION. Use of the Software, Maintenance and Services by the Government is further restricted according to the terms of this Agreement and any amendment hereto.

**Orders.** An Ordering Document means any purchase order, similar document or agreement requesting Software, Maintenance or Services, which shall be deemed accepted only by issuance of a TIBCO invoice and solely for purposes of administrative convenience. None of the terms of the Ordering Document (other than the Software product name, Number of Units, level of Maintenance, description of Services, and fees due in connection therewith) shall apply for any reason or purpose whatsoever, regardless of any statement on any Ordering Document to the contrary, unless countersigned by an officer of TIBCO. This Agreement constitutes the entire agreement between the parties with respect to the use of the Software, Maintenance and Services, and supersedes all proposals, oral or written, and all other representations, statements, negotiations and undertakings relating to the subject matter hereof. All orders of Software, Maintenance or Services by Customer to TIBCO shall be deemed to occur with or without reference to, under the terms of this End User License Agreement, unless expressly superseded by a signed written agreement between the parties. Neither the license to use the Software granted in this Agreement nor the obligation to pay the license fees set forth above are dependent upon the performance by any party of any Services or the supply of any other software program or product. Software shall be delivered electronically, and delivery deemed complete when the Software is made available for download by Customer.

**Termination.** Maintenance or Services may be terminated: (a) by either party upon a default of the other, such default remaining uncured for fifteen days from written notice from the non-defaulting party; (b) upon the filing for bankruptcy or insolvency of the other party; (c) by either party upon prior written notice at least sixty (60) days prior to the end of any annual Maintenance term; or (d) by Customer (for Services), upon ten days prior written notice or 30 days prior written notice by TIBCO. Termination of Maintenance or Services shall not terminate this End User License Agreement. Customer may terminate this End User License Agreement in its entirety at any time by destroying all copies of the Software. Upon termination or expiration of this End User License Agreement in its entirety, for any reason, Customer must cease using and return or destroy all copies of the Software. Customer's obligation to pay accrued charges and any fees due as of the date of termination, as well as the sections entitled "Confidentiality", "Limited Warranty" and "Limitation of Liability" shall survive any such termination.

**Authority.** You hereby represent and warrant that you have full power and authority to accept the terms of this End User License Agreement on behalf of Customer, and that Customer agrees to be bound by this End User License Agreement.

**Definitions.** In connection with this End User License Agreement Software product definitions are as set forth at <http://tibco.com/software/productdefinitions.jsp> which are hereby incorporated into the End User License Agreement and the following

capitalized terms shall have the following meaning: "**Customer**" means the original purchaser or licensee of the Software and any permitted successors and assigns; "**Documentation**" means text material that accompanies the Software on delivery, which describes how to make use of that software. "**Enterprise**" means an unlimited Number of Units of the Software set forth in an Ordering Document as Enterprise, to be deployed by Customer during a period of one year (or such other period of time set forth in the Ordering Document) from the Purchase Date (the "**Enterprise Term**"), at which time, the Number of Units then deployed in Production and Non-Production use by Customer becomes fixed and Customer may not thereafter deploy additional Units. During the Enterprise Term, Customer's right to deploy an unlimited Number of Units does not extend to any entity which acquires, is acquired by, merged into, or otherwise combined with Customer. Customer hereby agrees to provide TIBCO, within sixty (60) days after the end of the Enterprise Term, with written notice of the Number of Units deployed at the end of the Enterprise Term by Unit and License Type. In the event Customer elects to renew Maintenance (subject to the section entitled "Termination"), then during the Enterprise Term and for the first annual renewal thereafter, TIBCO agrees the annual Maintenance fee for the Software licensed on an Enterprise basis shall not increase by more than the percentage rate change in the United States Department of Labor - Consumer Price Index for All Urban Consumers (CPI-U) for the twelve month period immediately preceding the anniversary date of Maintenance. "**Purchase Date**" means the date the Ordering Document is accepted by TIBCO. "**Project**" means an unlimited Number of Units of the Software set forth in an Ordering Document as Project, to be deployed by Customer solely in connection with the description and scope of the Customer project set forth in the Ordering Document, during a period of one year (or such other time period set forth in the Ordering Document) from the Purchase Date (the "**Project Term**"), at which time, the Number of Units then deployed in Production and Non-Production use by Customer becomes fixed and Customer may not thereafter deploy additional Units. During the Project Term, Customer's right to deploy an unlimited Number of Units does not extend beyond the scope of the Project as set forth herein or to any entity which acquires, is acquired by, merged into, or otherwise combined with Customer. Customer hereby agrees to provide TIBCO, within sixty (60) days after the end of the Project Term, with written notice of the Number of Units deployed at the end of the Project Term by Unit and License Type. In the event Customer elects to renew Maintenance (subject to the section entitled "Termination"), then during the Project Term and for the first annual renewal thereafter, TIBCO agrees the annual Maintenance fee for the Software licensed on a Project basis shall not increase by more than the percentage rate change in the United States Department of Labor - Consumer Price Index for All Urban Consumers (CPI-U) for the twelve month period immediately preceding the anniversary date of Maintenance. "**Severity**" means the impact levels of an error, defect or malfunction, which under Maintenance are determined by TIBCO as follows: "**Severity 1**" is an emergency production situation where the Software is totally inoperable or fails catastrophically and there is no workaround; "**Severity 2**" is a detrimental situation (and there is no workaround) where (a) performance degrades substantially under reasonable loads causing a severe impact on use, (b) the Software is usable but materially incomplete; or (c) one or more mainline functions or commands is inoperable; "**Severity 3**" is where the Software is usable, but does not provide a function in the most convenient manner; and "**Severity 4**" is a minor problem or documentation error. "**Software**" means the most current, generally available object code version on all Platforms then currently available. Software does not include multiple Platforms if the software product licensed on a Platform specific basis as designated in the Software product name or as otherwise listed in an Ordering Document, including its Documentation and any subsequent Updates provided under Maintenance. For the purposes of the product definitions where

"TIBCO Software" or "Licensor Software" is used it shall have the same meaning as "Software". **"TIBCO"** means TIBCO Software Inc. and any entities, regardless of corporate status, controlled by, controlling, or under common control with TIBCO Software Inc.

**"Updates"** means Software bug fixes, enhancements, and updates, if and when made generally available by TIBCO. Updates provided under Maintenance, warranty or which are provided for any other reason by TIBCO, or TIBCO's authorized resellers or distributors (if applicable), are subject to the license rights, limitations and restrictions of the End User License Agreement.

**Special Product Provision. Embedded/Bundled Products.** Some Software embeds or bundles other Software. Use of such embedded or bundled Software is solely to enable the functionality of the Software licensed under this Agreement, and may not be used or accessed by any other Software, or for any other purpose. **Bundle.** A collection of Software listed in TIBCO's then current price book to be sold together under a collective name such as "XXX Bundle" which consists of X, Y and Z. The Software which comprises a Bundle must be used in accordance with any specific license restrictions imposed in this Agreement and solely in conjunction with the components of the Bundle; provided that a Bundle component may be accessed by or communicate with other Software separately licensed by Customer. In no event may the Software which comprises a Bundle be used on a standalone basis. A Bundle is sold at a discount to the cost of licensing the individual components due to the restrictions imposed on the use of the Bundle by this section and any specific license restrictions imposed by this Agreement. If the terms of this Agreement with regard to a Bundle are breached, and such breach is not cured within ten (10) days of TIBCO notifying Customer in writing of the breach, TIBCO shall be entitled to invoice Customer for the list price of the individual components of the Bundle based upon the manner in which Customer has deployed and is making use of the components. **Open Source Software.** If Customer uses open source software or any other third party software not supplied by TIBCO in conjunction with the Software, Customer must ensure that its use does not: (i) create, or purport to create, obligations of use with respect to the Software; or (ii) grant, or purport to grant, to any third party any rights to or immunities under TIBCO's intellectual property or proprietary rights in the Software. Customer also may not combine Software with programs licensed under the GNU General Public License ("GPL") in any manner that could cause, or could be interpreted or asserted to cause, the Software or any modifications thereto to become subject to the terms of the GPL. **Third Party Software.** Third Party Software means third-party software identified by its company and/or product name, the provision of which by TIBCO is made solely as an accommodation and in lieu of Customer purchasing a license to Third Party Software directly from the third party vendor.

**General.** All payments of fees due shall be made in U.S. dollars, net 30 from Purchase Date, or, for any other amounts coming due hereafter, net 30 from TIBCO's invoice. Fees do not include sales, use, withholding, value-added or similar taxes, and Customer agrees to pay all sales, use, value-added, goods and services, consumption, withholding, excise and any other similar taxes or government charges, exclusive of TIBCO's income tax. Customer agree to pay all reasonable costs incurred (including reasonable attorneys' fees) in collecting past due amounts. Except as set forth in the sections entitled "Limited Warranty" and "Indemnity" all fees paid under or in connection with this End User License Agreement are non-refundable and no right of set-off exists. A service charge of one and one-half percent per month will be applied to all invoices that are not paid on time. No delay in the performance of any obligation by either party, excepting all obligations to make payment, shall constitute a breach of this End User License Agreement to the extent caused by force majeure. Customer hereby grants TIBCO and its independent auditors the right to audit Customer's compliance with this End User License

Agreement. If any portion of this End User License Agreement is found to be void or unenforceable, the remaining provisions shall remain in full force and effect.

**Governing Law.** This End User License Agreement shall be governed by and construed in accordance with the laws of the State of California, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. The United Nations Convention on Contracts for the International Sale of Goods is excluded from application hereto.

Version 5.6, April 2008

# Index

## A

Adapter Instance Fields 136  
 Adapter Services 5  
 Adapter Services and Components 11  
 Adapter Termination Criteria 140  
 Add a Form to Resource Configuration 334  
 Adding Adapter Services 151  
 Adding TIBCO Staging Tables to the ClarifyCRM Database 53  
 AddUsers to Resource Configuration 333  
 agents 214  
 alerts 214  
 Auto-Discovery Process, TIBCO Hawk 216  
 Available Microagents 220

## B

B2B Admin User Registration (View 1) 337  
 B2B Admin User Registration (View 2) 338  
 broadcast messaging 8

## C

Certified Messaging, RVCN 8  
 Changing the Polling Interval and the Update Interval 141  
 Clarify Adapter Configuration 136  
 ClarifyCRM Business Objects, CBOs 2  
 Class Microagent Name field, adapter 149  
 ClearBasic 12  
 ClearBasic Forms 102  
 ClearBasic Functions 56  
 ClearBasic Global File 56  
 ClearBasic script 57

ClearBasic Script Coding Conventions 59  
 command line arguments to start the adapter 231  
 commands  
     cbex 54, 334  
     dataex 53  
     -dir 54, 334  
 configuration properties about TIBCO Rendezvous session, retrieving through TIBCO Hawk 239  
 Configure the Adapter 118  
 Configuring a Remote Adapter 211  
 Configuring Multiple Publication Service Endpoints 154  
 Connection Pooling 9  
 Console Mode 24  
 Create Customized Form 333  
 Custom JavaBean Invocation 81  
 customer support xxvi  
 Customized Operations 95

## D

Database Host Name 139  
 Database Name 138  
 Database Port 139  
 Database Server Name 137, 139  
 Database Vendor 137  
 Default Microagent Session field, adapter 149  
 Defining a TIBCO Hawk Session 202  
 Description 137  
 Directive Window 334  
 Distributed Queue 8  
 DTA Invocation Timeout (milliseconds) 138  
 DTA Transport 138

**E**

- Enter Purchase Order 336
- Enterprise Archive file, EAR 190
- Error Message Listing 264
- Error Publisher 85
- Example of an Input Schema - Query Operation 90
- examples
  - user exit for the Subscriber 309

**F**

- Flexible Deployment 102
- Frequently Asked Questions 260
- Function
  - fetchSite() 323
  - insert2CaseHistory 330
  - performAccept() 323
  - performAssign() 324
  - performDispatch() 325
  - performForward() 326
  - performMove() 327
  - performReject() 328
  - performYank() 329

**G**

- General Tab 142
- generic configuration properties, retrieving through
  - TIBCO Hawk 231
- Getting Started 115
- Global Java Class 61
- global variables 203
- GUI Mode 24

**H**

- How the Adapter Implements User Exits 307
- How the Publication Service Works 52

- How the Publication Service Works for ClarifyCRM
  - Web Client 61

- How the Publication Service Works for Web Forms 60

- How the Request-Response Invocation Service
  - Works 102

- How the Request-Response Service Work 89

- How the Subscription Service Works 80

**I**

- IclarifyUserExit Interface 308

- Import ClearBasic Global Files 54

- Importing ClearBasic Files 332

- Incoming Message Formats for Query Operations 90

- Installation
  - Microsoft Windows 33
  - UNIX System 36

- Installation Package Names 36

- Installer Account
  - UNIX 38
  - Windows 32

- Installing Adapter Components 29

- Instance Name 136

- Interval Between Reconnect Attempts
  - (milliseconds) 140

- Invoking Microagent Methods 217

**J**

- JDBC Driver 139

- JDBC URL 140

- JSP Coding Conventions 64

**L**

- ledger files, retrieving information through TIBCO
  - Hawk 249

- Limitations of the Publication Service 75

- Log File field, adapter 146

Log to Standard field, adapter 146  
 Log trace messages 146

## M

Maximum Number of Reconnect Attempts 140  
 Message Filter 137  
 microagent methods supported 220  
 Modifying the Design-time Adapter Rendezvous Connection Properties 201  
 Modifying the Properties File 201  
 Multithreading 9

## N

Non-Flexible Deployment 105  
 Number of Reconnect Attempts Before Suspending Impacted Service(s) 140  
 Number of Threads 143

## O

Outgoing Message Formats for Customized Operations 96  
 Overview  
   Adapter Instance 134  
   Product 4  
   TIBCO Hawk 214  
   User exits 306  
   Wrapper classes 312

## P

Password 138, 139  
 password property 192  
 point-to-point 8  
 Polling Batch Size 141

Post-Installation Tasks  
   Microsoft Windows 34  
   UNIX 40  
 Post-Subscription 143  
 Pre-Publication 142  
 Pre-requisites for Setting Up Request-Response Invocation Functionality  
   using Clarify ClearBasic Forms in a Flexible Deployment 103  
   using Clarify ClearBasic Forms in a Non-Flexible Deployment 106  
   using Clarify Web Forms in a Web Client Deployment 110  
 Pre-Subscription 143  
 Product Overview 4  
 Publication Service 12  
   Features 48  
   Fields 152  
   Overview 48  
   typical flow diagram 12  
 Publisher  
   functions 12

## Q

Query Operations 89  
 Queue 8

## R

Reconnection Mechanism 10  
 Remember Password 138  
 Reply subject 85  
 Request-Response Limitations, workflow operation 94  
 Request-Response Invocation Service  
   Fields 179  
   Overview 100  
   typical flow 15  
   Using Flexible Deployment 100  
   Using the Web Client Deployment 102

## Request-Response Service 14

Features 88

Fields 170

Overview 88

## Request-Response Service Features 88

## Request-Response Service Transport Tab with TIBCO

Rendezvous 174

resetThreadStatistics() 248

Retain Published Messages 141

## RPC Server

function 14

## Runtime Adapter Components 4

## S

## Sample Code

Used for a Flexible Deployment 104

Used for a Non-Flexible Deployment over JMS 108

Used for a Non-Flexible Deployment over TIBCO

Rendezvous 106

Used for a Web Client Deployment 112

## Sample Configuration Screen

Request-Response Service Configuration tab 171

Subscription Service Configuration tab 163

Show All Tabs 137

Silent Mode 24

Standard Microagent Name field, adapter 148

Starting TIBCO Hawk Software 215

## Subscription Service 13

Features 78

Fields 162

Functionality 78

Overview 78

typical flow diagram 13

## Subscription Service Transport Tab with TIBCO

Rendezvous 166

substitution 203

support, contacting xxvi

Supported Request-Response Operations 88

## T

table\_tibco\_message\_fields table 257

table\_tibco\_message\_queue table 256

technical support xxvi

Termination Subject or Topic 142

Test CBO Connection 140

Test JDBC Connection 140

Testing the Publication Service

with ClearBasic Forms 332

with Web Forms 336

Testing the Request-Response Functionality

with ClearBasic Forms 340

with Web Forms 339

testPublisher 332, 336

TIBCO Hawk

enterprise monitor components 214

TIBCO Hawk methods

getConfig 231

getRvConfig 239

getStatus 240

reviewLedger 249

TIBCO Rendezvous, retrieving configuration through

TIBCO Hawk 239

TIBCO Run-time Agent 4

TIBCO Staging Tables 48

Topic 8

Typical Flow

Publication Service 12

Request-Response Invocation Service 15

Request-Response Service 14

Subscription Service 13

typographical conventions xxiii

## U

Uninstalling the Adapter 25

Upgrading an Adapter 24

Use Advanced Logging 144

Use Design-time Connection for Run-time 138

user exits for pre-processing of Subscription service

event message 307

User Name 138, 139

username property 192

## V

variable substitution 203

variables 203

Version 137

## W

Web Client Deployment 109

Web Forms 115

Windows Windows

    installation 19

workflow operation incoming message formats 90

Workflow Operations 91

Wrapper Class and TibcoCBOInterface 320

Writing Wrapper Class for Case Business Object 321