

SeeBeyond ICAN Suite

eGate Integrator User's Guide

Release 5.0.1



The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

SeeBeyond, e*Gate, and e*Way are the registered trademarks of SeeBeyond Technology Corporation in the United States and select foreign countries; the SeeBeyond logo, e*Insight, and e*Xchange are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2003 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.

Version 20031121160704.

Contents

List of Figures	11
-----------------	----

List of Tables	20
----------------	----

Chapter 1

Introduction	22
Purpose and Scope	22
Intended Audience	22
Organization of Information	22
Writing Conventions	23
Supporting Documents	24
Online Documents	24
The SeeBeyond Web Site	24

Chapter 2

System Overview	25
Introduction	25
Integration Model	26
System Architecture	28
Repository	29
Environments	29
User Interfaces	30
Enterprise Designer	30
Enterprise Manager	32

Chapter 3

Enterprise Manager	33
Overview	33
Installing and Updating eGate	33

Monitoring and Managing eGate	33
Starting Enterprise Manager	34
The Enterprise Manager Interface	35
Home	36
Documentation	36
The Enterprise Monitor	37
Monitoring Collaborations	39
Viewing Alerts	45
Viewing Logs	48
Integration Server Level	48
Collaboration Level	49
Setting Log Levels	50
Editing XA Transactions	51

Chapter 4

Enterprise Designer	53
Overview	53
Starting Enterprise Designer	55
Interface Features	56
Menus	56
File Menu	56
Tools Menu	56
View Menu	57
Window Menu	57
Help Menu	58
Toolbar	58
Browser Buttons	59
Enterprise Explorer	60
Project Explorer	60
Environment Explorer	61
Enterprise Designer Editors	62
Connectivity Map Editor	63
OTD Editor	64
Java Collaboration Editor	65
XSLT Collaboration Editor	66
Environment Editor	67
Deployment Editor	68
Additional Tools and Features	69
Impact Analyzer	69
Version Control	71
Checking a Component In	71
Checking a Component Out	72
Viewing a Component's Version History	73
Repository Backup and Restoration	74
Backing Up a Repository	74
Restoring a Repository	74

Project Export and Import	76
Exporting a Project	76
Exporting a Project Using Enterprise Designer	76
Exporting a Project Using the Command Line	77
Importing a Project	78
Importing a Project Using Enterprise Designer	78
Importing a Project Using the Command Line	79

Chapter 5

eGate Projects	81
Overview	81
Project Components	81
The Project Explorer	82
Project Explorer Icons	83
Context Menus	84
Repository Menu	84
Project Menu	85
Connectivity Map Menu	86
Using the Connectivity Map Editor	88
Services	90
Collaborations	90
Message Destinations	91
Topics	91
Queues	91
External Applications	92
Schedulers	93
Component Connections	94
Configuring a Connection	95
Modifying a Configuration Property	97
Defining Constants and Variables	98

Chapter 6

Web Services	101
Overview	101
SeeBeyond Web Services	102
UDDI Repository	103
Building a Web Client	105
Building a Web Server	113

 Chapter 7

Object Type Definitions	120
Overview	120
The <i>Bean-like</i> Interface	120
OTD Types	121
Externally-Defined OTDs	121
User-Defined OTDs	121
Building OTDs	121
Using the OTD Wizard	122
Externally-Defined OTDs	124
Using the DTD Wizard	124
Using the WSDL Wizard	128
WSDL OTD Structure	131
WSDL Operation Elements	132
Using the XSD Wizard	133
User-Defined OTDs	137
Using the User-Defined OTD Wizard	137
Editing the OTD Properties	139
Node Properties	139
Element Properties	140
Field Properties	141
Specifying the Node Type	142
Specifying Delimiters	143
Delimiter Properties	144
Precedence	148
Creating a Delimiter List	149
Using the OTD Editor	152
Node Management	154
Using the OTD Tester	155

 Chapter 8

Java Collaboration Definitions	159
Overview	159
Using the Java Collaboration Wizard	160
Creating a Java Collaboration Definition	161
New Web Service	162
Existing Web Service	166
Using the Java Collaboration Editor	168
Java Toolbar Icons	169
Business Rules Editor	171
Commands	171
Business Rules Toolbar Icons	172
Business Rules Tree	173
Business Rules Designer	174

Method-Access Nodes	174
Business Rules Designer Toolbar Icons	175
Java Method Palette	179
Java Method Boxes	179
Java Source Editor	180
Java Methods	181
Boolean Methods	181
Comparison Methods	182
Math Methods	184
Object Methods	186
Object Method Dialog Boxes	187
String Methods	189
Array Operation Methods	192
Operator Methods	194
Java Encapsulation	196
Creating a Modified Java Collaboration Definition	196
Merging Two Versions of a Java Collaboration Definition	197
Setting Up a Java Collaboration Definition Variable	198
Creating a Variable	198
Invoking Variable Constructor	201
Displaying Method Classes	203
Using <i>Try-Catch</i>	204
Adding and Using Third-Party Java Classes	207
Adding Class Instances to a Collaboration	211
Validating Java Collaborations	213
Debugging Java Collaborations	214
Enabling the Debugger	214
Invoking the Java Debugger	216
Setting Breakpoints	218
Inspecting and Editing the Source Code	220
Stepping Into, Over, or Out	221
Inspecting Java Threads and Methods	222
Inspecting a Local Variable or Method	223
Saving and Resuming Debug Sessions	226
Creating Alerts	227
Creating Log Entries	230

Chapter 9

XSLT Collaboration Definitions	233
Overview	233
Using the XSLT Collaboration Wizard	234
Creating an XSLT Collaboration Definition	235
New Web Service	236
Existing Web Service	238

Using the XSLT Collaboration Editor	239
XSLT Toolbar Icons	240
XSLT Method Palette	241
XSLT Method Boxes	241
XSLT Methods	242
Operator Methods	242
String Methods	245
Number Methods	248
Boolean Methods	250
Nodes Methods	252
XSLT Encapsulation	255
Creating a Modified XSLT Collaboration Definition	255
Merging Two Versions of a XSLT Collaboration Definition	256

Chapter 10

Environments	257
Overview	257
Environment Explorer	257
Environment Editor	258
Environment Explorer Icons	259
Context Menus	260
Repository Menu	260
Environment Menu	261
Logical Host Menu	262
Defining Environmental Constants	264
Logical Hosts	266
Management Agent	267
Message Servers	267
Integration Servers	268
Configuring the Logical Host	269
Modifying the Logical Host Properties File	269
Installing the Logical Host as a Windows Service	272
Configuring the Base Port Number	274
Starting the Logical Host	275
Starting the Logical Host on a Windows System	276
Starting the Logical Host on a UNIX System	276
Starting the Logical Host on a Red Hat Linux System	276
Stopping the Logical Host	277

Chapter 11

Project Deployment	278
Deployment Profiles	278

Using the Deployment Editor	279
Creating a Deployment Profile	280
Activating and Deactivating Deployment Profiles	283
Mapping Variables	285
Deploying Projects to Third-Party Servers	286
BEA WebLogic	286
IBM WebSphere	289

Chapter 12

ICAN Security Features	292
Overview	292
Multiple Environments	293
User Management	294
Configuration User Management	294
Roles	294
Adding and Deleting Repository Users	294
Adding and Deleting User Roles	298
Creating User Roles	303
Using LDAP	307
Configuring LDAP Servers for Configuration User Management	309
Configuring the Active Directory Server	309
Configuring the SunONE Server	315
Configuring the ICAN Repository to Use LDAP	317
Configuring the ICAN Suite for ADS	318
Configuring the ICAN Suite for SunONE	318
Environment User Management	319
Creating and Configuring Users	319
Configuring for LDAP Servers in Environment User Management	320
ACL Management	321
ICAN Component Security	324
Message Server Security	324
JMS Client Security	324
Using SSL/HTTPS in ICAN	326
Overview	326
Certificates and Keys	326
The Keytool Utility	326
Installation and Configuration	327

Chapter 13

Logging	330
Overview	330
eGate Logs	331

Contents

Log File System	331
Log File Management	331
Logging Model	332
Loggers	332
Appenders	332
Layouts	332
Basic Log Files and Locations	333
Repository	333
ESR Installer Logs	334
Enterprise Designer	335
Enterprise Monitor	336
Run-Time Log Files and Locations	337
Logical Host	337
Integration Servers	338
Message Servers	338

Appendix A	
OTD Interfaces	339
The <i>Bean-like</i> Interface	339
The <i>Generic</i> Interface	340

Appendix B	
The Schema Runtime Environment	341
The SRE Monitor	341
Security	346
Glossary	347
e*Gate 4.x Terms in eGate 5.0	353
Index	355

List of Figures

Figure 1	SeeBeyond ICAN Suite	25
Figure 2	eGate Integrator	26
Figure 3	eGate Integrator Implementation Model	27
Figure 4	Typical eGate Integrator System	28
Figure 5	Enterprise Designer	30
Figure 6	Connectivity Map Editor	31
Figure 7	SeeBeyond Enterprise Manager Login	32
Figure 8	Enterprise Manager Login	34
Figure 9	Enterprise Manager GUI	35
Figure 10	Monitor Launch Window	36
Figure 11	Documentation Tab	36
Figure 12	Monitor Interface - Initial	37
Figure 13	Monitor Interface - Environment	37
Figure 14	Example Project Connectivity Map	39
Figure 15	Monitor - Details - List Collaborations	39
Figure 16	Starting and Stopping Collaborations - Details Panel	40
Figure 17	Pending and Processed Messages	41
Figure 18	Monitor - Project Explorer	42
Figure 19	Active/Non-active Collaborations	42
Figure 20	Pending and Processed Messages	43
Figure 21	Logging Details	43
Figure 22	Connectivity Map Details: Components	44
Figure 23	Connectivity Map View - Zoom In/Out	44
Figure 24	Collaboration Alerts	45
Figure 25	Alert Status	45
Figure 26	Alert Details	46
Figure 27	Integration Server Log Messages	48
Figure 28	Integration Server Log Messages - Filtered	48
Figure 29	Collaboration Log File	49
Figure 30	Search on Keyword	49
Figure 31	Logging Properties Page	50
Figure 32	Resetting Log Levels	50

List of Figures

Figure 33	Message Server Details - Controls Tab	51
Figure 34	Indoubt Transaction List	52
Figure 35	Indoubt Transaction List - Transaction Selected	52
Figure 36	SeeBeyond Enterprise Designer	53
Figure 37	Login Dialog Box	55
Figure 38	Enterprise Explorer: Project Explorer View	60
Figure 39	Enterprise Explorer: Environment Explorer View	61
Figure 40	Connectivity Map Editor	63
Figure 41	OTD Editor	64
Figure 42	Java Collaboration Editor	65
Figure 43	XSLT Collaboration Editor	66
Figure 44	Environment Editor	67
Figure 45	Deployment Editor	68
Figure 46	Impact Analyzer Dialog Box	69
Figure 47	Version Control - Check In Dialog Box	71
Figure 48	Version Control - Check Out Dialog Box	72
Figure 49	Version Control - History Dialog Box	73
Figure 50	Enter File Name Dialog Box	76
Figure 51	Message Dialog Box	77
Figure 52	Select File Dialog Box	78
Figure 53	File Destination Dialog Box	79
Figure 54	Message Dialog Box	79
Figure 55	Project Explorer	82
Figure 56	Repository Menu	84
Figure 57	Project Menu	85
Figure 58	Connectivity Map Menu	86
Figure 59	Connectivity Map Window	88
Figure 60	Linking JMS Topics	88
Figure 61	Service Component	90
Figure 62	External Application Drop-Down Menu	92
Figure 63	Selected External Applications in Toolbar	92
Figure 64	Scheduler Component	93
Figure 65	Connection Icons in a Connectivity Map	94
Figure 66	Default Configuration Dialog Box	95
Figure 67	Editing a Default Configuration Field	97
Figure 68	Project Variable Creation	98
Figure 69	Project Constant Creation	99
Figure 70	Variables and Constants Object Group	99

List of Figures

Figure 71	Connector Properties	100
Figure 72	SeeBeyond UDDI Repository	103
Figure 73	Example Web Service WSDL File	103
Figure 74	Microsoft Visual Studio Example	104
Figure 75	File Destination Dialog Box	105
Figure 76	Select WSDL Wizard	106
Figure 77	Select File Location	107
Figure 78	Select WSDL File	107
Figure 79	Select External Server	108
Figure 80	eInsight Business Process - Receive Rule	109
Figure 81	eInsight Business Process - Write Rule	110
Figure 82	Connectivity Map	111
Figure 83	Project Deployment	112
Figure 84	File Destination Dialog Box	113
Figure 85	Select WSDL Wizard	114
Figure 86	Select File Location	115
Figure 87	Select WSDL File	115
Figure 88	Select External Client	116
Figure 89	eInsight Business Process	117
Figure 90	Connectivity Map	118
Figure 91	Project Deployment	119
Figure 92	OTD Wizard Selection Dialog	122
Figure 93	OTD Wizard Selection: DTD Wizard	124
Figure 94	Select DTD File(s) Dialog Box	125
Figure 95	Select Document Elements Dialog Box	126
Figure 96	Select OTD Options Dialog Box	127
Figure 97	OTD Wizard Selection: WSDL Wizard	128
Figure 98	WSDL Wizard: Select WSDL Location	129
Figure 99	WSDL Wizard: Select WSDL File	130
Figure 100	WSDL Wizard: Select OTD Options	131
Figure 101	OTD Wizard Selection: XSD Wizard	133
Figure 102	XSD Wizard: Select XSD File(s)	134
Figure 103	Select Document Elements Dialog Box	135
Figure 104	Select OTD Options Dialog Box	136
Figure 105	OTD Wizard Selection: User-Defined OTD	137
Figure 106	Enter OTD Name	138
Figure 107	User-Defined OTD Node Properties	139
Figure 108	User-Defined OTD Element Properties	140

Figure 109	User-Defined OTD Field Properties	141
Figure 110	Node Type Property Options	142
Figure 111	Delimiter List Editor	144
Figure 112	Terminal Type Property Example	146
Figure 113	Optional Property (Example 1)	147
Figure 114	Optional Property (Example 2)	147
Figure 115	Example User-Defined OTD	149
Figure 116	Activate Field	149
Figure 117	Delimiter List Editor (1)	150
Figure 118	Context Menu	150
Figure 119	Delimiter List Editor (2)	150
Figure 120	Delimiter Specified	151
Figure 121	OTD Editor	152
Figure 122	OTD Tester	155
Figure 123	Test Panel Data Display	155
Figure 124	Select Data File	156
Figure 125	Object Elements and Values	156
Figure 126	Data Display: Refresh Icon	157
Figure 127	Status Data Display	157
Figure 128	Verbose Data Display	158
Figure 129	Initial Wizard Dialog	161
Figure 130	New Web Service: Operation Name	162
Figure 131	New Web Service: Input Message	163
Figure 132	New Web Service: Output Message	164
Figure 133	New Web Service: Auxiliary OTD	165
Figure 134	Existing Web Service: Select Operation	166
Figure 135	Existing Web Service: Select OTD	167
Figure 136	Java Collaboration Editor	168
Figure 137	Business Rules Editor	171
Figure 138	Business Rules Designer: Addition Method	174
Figure 139	Method-Access Nodes	174
Figure 140	Java Method Palette Dialog Box	179
Figure 141	Expanded Method Box	179
Figure 142	Collapsed Method Box	179
Figure 143	Java Source Editor	180
Figure 144	Method Palette: Boolean Methods	181
Figure 145	Method Palette: Comparison Methods	182
Figure 146	Method Palette: Math Methods	184

Figure 147	Method Palette: Object Methods	186
Figure 148	CAST Dialog Box	187
Figure 149	InstanceOf Dialog Box (1)	187
Figure 150	Find Class Dialog Box	188
Figure 151	InstanceOf Dialog Box (2)	188
Figure 152	Method Palette: String Methods	189
Figure 153	Method Palette: Array Operation Methods	192
Figure 154	Method Palette: Operator Methods	194
Figure 155	Java Collaboration Definition Context Menu	196
Figure 156	Version Control - Create Diff Dialog Box	196
Figure 157	Version Control - Merge Changes Dialog Box	197
Figure 158	Create a Variable Dialog Box	198
Figure 159	Find Class Dialog Box	199
Figure 160	Java Collaboration Definition with Variable	200
Figure 161	Call New Constructor Dialog Box	201
Figure 162	Java Collaboration Definition with Constructor	202
Figure 163	Variable Context Menu	203
Figure 164	Method Selection List Box	203
Figure 165	Try Icon	204
Figure 166	Catch Context Menu	204
Figure 167	Create a New Exception Variable Dialog Box	205
Figure 168	Find Class Dialog Box	205
Figure 169	Catch SQLException Rule	206
Figure 170	Exception Message	206
Figure 171	Project Context Menu: New File ...	207
Figure 172	File Selection Dialog Box	207
Figure 173	Import JAR File Icon	208
Figure 174	Add/Remove Jar File Dialog Box (1)	208
Figure 175	Add/Remove Jar File Dialog Box (2)	209
Figure 176	Call Java Method Dialog Box	209
Figure 177	Using the Third-Party Java Method	210
Figure 178	Call New Constructor Dialog Box	211
Figure 179	Constructor Example 1	212
Figure 180	Constructor Example 2	212
Figure 181	Validating a Collaboration Definition	213
Figure 182	Integration Server Properties Dialog Box	214
Figure 183	Logical Host Context Menu	215
Figure 184	Integration Server Context Menu	215

List of Figures

Figure 185	Java Debugger	216
Figure 186	File Menu	217
Figure 187	Attach to JVM Dialog Box	217
Figure 188	Collaboration Source Code Display	217
Figure 189	Breakpoint Example	218
Figure 190	Debug Menu	218
Figure 191	Stop in Method Dialog Box	219
Figure 192	Choose Exception Dialog Box	219
Figure 193	Break on Exception Dialog Box	219
Figure 194	Breakpoint Indicator	220
Figure 195	Stepping Into, Over, and Out Commands	221
Figure 196	Java Thread and Method Display	222
Figure 197	Local Variables Tab	223
Figure 198	Evaluate Local Variable	224
Figure 199	Evaluate Method	225
Figure 200	Save Debugger Session Dialog Box	226
Figure 201	Resume Debugger Session Dialog Box	226
Figure 202	Empty File Test	227
Figure 203	Alert Menu	228
Figure 204	Alert Severity Selection Window	228
Figure 205	Create Literal Dialog Box	228
Figure 206	Pass Alert Message to Object Argument	229
Figure 207	Logging Menu	230
Figure 208	Logging Level/Method Selection Window	231
Figure 209	Create Literal Dialog Box	231
Figure 210	Pass Log Message to Object Argument	232
Figure 211	XSLT Collaboration Wizard Dialog Box	235
Figure 212	New Web Service: Operation Name	236
Figure 213	New Web Service: Input Message	237
Figure 214	New Web Service: Output Message	237
Figure 215	Existing Web Service: Select Operation	238
Figure 216	XSLT Collaboration Editor	239
Figure 217	XSLT Method Palette Dialog Box	241
Figure 218	Expanded Method Box	241
Figure 219	Collapsed Method Box	241
Figure 220	Method Palette: Operator Methods	242
Figure 221	Method Palette: String Methods	245
Figure 222	Method Palette: Number Methods	248

Figure 223	Method Palette: Boolean Methods	250
Figure 224	Method Palette: Nodes Methods	252
Figure 225	XSLT Collaboration Definition Context Menu	255
Figure 226	Version Control - Create Diff Dialog Box	255
Figure 227	Version Control - Merge Changes Dialog Box	256
Figure 228	Enterprise Explorer: Environment Explorer View	257
Figure 229	Environment Editor	258
Figure 230	Repository Menu	260
Figure 231	Environment Menu	261
Figure 232	Logical Host Menu	262
Figure 233	Environmental Constants Editor	264
Figure 234	Logical Hosts	266
Figure 235	Integration Server (J2EE Compatible)	268
Figure 236	Example logical-host.properties File	269
Figure 237	Install as Service Script	272
Figure 238	Windows Services List	272
Figure 239	Uninstall as Service Script	273
Figure 240	Logical Host Properties Dialog Box	274
Figure 241	eGate Integrator Implementation Model	278
Figure 242	Deployment Editor Window	279
Figure 243	Example Deployment Profile (1)	280
Figure 244	Example Deployment Profile (2)	281
Figure 245	Example Deployment Profile (3)	282
Figure 246	Activate Dialog Box	283
Figure 247	Logical Host Context Menu - Apply	283
Figure 248	Activate Dialog Box	284
Figure 249	Deployment Profile Mappings	285
Figure 250	Project Variable Value Entry	285
Figure 251	WebLogic Deployment (1)	286
Figure 252	WebLogic Deployment (2)	287
Figure 253	WebLogic Deployment Verification	288
Figure 254	WebSphere Deployment (1)	289
Figure 255	WebSphere Deployment (2)	290
Figure 256	WebSphere Deployment Verification	291
Figure 257	User Management Dialog Box (1)	295
Figure 258	User Management - Add User	295
Figure 259	User Management Dialog Box (2)	296
Figure 260	User Management Dialog Box	297

Figure 261	Error Warning Box	297
Figure 262	User Management Dialog Box	298
Figure 263	User Management - Add Role (1)	299
Figure 264	Add Role Dialog Box	299
Figure 265	User Management - Add Role (2)	300
Figure 266	User Management Dialog Box	300
Figure 267	User Management Dialog Box	301
Figure 268	User Management - Delete Role	302
Figure 269	Error Warning Box	302
Figure 270	User Management Users List	303
Figure 271	User Management - Create Role (1)	304
Figure 272	Add Role Dialog Box (1)	304
Figure 273	Role Dialog Box	305
Figure 274	Add Role Dialog Box (2)	305
Figure 275	User Management - Create Role (2)	306
Figure 276	User Management Users List	306
Figure 277	LDAP Directory Tree (Traditional Naming)	308
Figure 278	LDAP Directory Tree (Internet Naming)	308
Figure 279	Active Directory Users and Computers Window	310
Figure 280	Active Directory - Create Organizational Unit	311
Figure 281	Active Directory - Create Roles Groups	312
Figure 282	Active Directory - ICANRoles Directory	313
Figure 283	Active Directory - Select Administrator	314
Figure 284	Sun ONE - Create New Role	315
Figure 285	Sun ONE - Roles	316
Figure 286	Environment Context Menu	319
Figure 287	Security Configuration Template Properties Dialog Box	320
Figure 288	ACL Management Dialog Box (1)	322
Figure 289	ACL Add Users Dialog Box	322
Figure 290	ACL Management Dialog Box (2)	323
Figure 291	ACL Management Dialog Box (3)	323
Figure 292	JMS IQ Manager Properties	324
Figure 293	JMS Client Security Properties	325
Figure 294	Recirculating Log File Stack	331
Figure 295	SRE Monitor Initial Page	343
Figure 296	Add Registry/Repository Dialog Box	343
Figure 297	Viewing SRE Components	344
Figure 298	FromExternal e*Way Status (Not Running)	345

List of Figures

Figure 299	FromExternal e*Way Status (Running)	345
Figure 300	Refreshing the Explorer Tree	346

List of Tables

Table 1	Writing Conventions	23
Table 2	Enterprise Manager - Tabs	35
Table 3	Enterprise Manager - Tabs	35
Table 4	Monitor Interface - Details Tabs	37
Table 5	Collaboration Status Types	39
Table 6	Basic Alert Types	46
Table 7	File Menu Options	56
Table 8	Tools Menu Options	56
Table 9	View Menu Options	57
Table 10	Window Menu Options	57
Table 11	Help Menu Options	58
Table 12	Enterprise Designer Toolbar Icons	58
Table 13	Browser Buttons	59
Table 14	Impact Analyzer Command Buttons	70
Table 15	Project Icons	83
Table 16	Repository Menu Options	84
Table 17	Project Menu Options	85
Table 18	Connectivity Map Menu Options	86
Table 19	Connectivity Map Toolbar Icons	89
Table 20	Configuration Dialog Box Toolbar Buttons	96
Table 21	OTD Wizard Navigation Buttons	123
Table 22	Node Properties	139
Table 23	Element Properties	140
Table 24	Field Properties	141
Table 25	Node Type Property Options	142
Table 26	Delimiter Properties and Value Options	144
Table 27	Escape Delimiters	145
Table 28	OTD Editor Toolbar Icons	153
Table 29	Wizard Navigation Buttons	160
Table 30	Java Toolbar Icons	169
Table 31	Business Rules Toolbar Buttons	172
Table 32	Rules for Placement of Subnodes	173

Table 33	Business Rules Designer Toolbar Icons	175
Table 34	Java Boolean Methods	181
Table 35	Java Comparison Methods	182
Table 36	Java Math Methods	184
Table 37	Java Object Methods	186
Table 38	Java String Methods	189
Table 39	Java Array Operation Methods	192
Table 40	Java Operator Methods	194
Table 41	Wizard Navigation Buttons	234
Table 42	XSLT Toolbar Icons	240
Table 43	XSLT Operator Methods	242
Table 44	XSLT String Methods	245
Table 45	XSLT Number Methods	248
Table 46	XSLT Boolean Methods	250
Table 47	XSLT Nodes Methods	252
Table 48	Environment Icons	259
Table 49	Repository Menu Options	260
Table 50	Environment Menu Options	261
Table 51	Logical Host Menu Options	262
Table 52	Environmental Constants Editor Icons	265
Table 53	User-Modifiable Logical Host Properties	271
Table 54	Startup Command Arguments	275
Table 55	Deployment Toolbar Buttons	279
Table 56	Realm Element Attributes	317
Table 57	Logging Levels	332
Table 58	Properties for the Master Repository Log	333
Table 59	Properties for the UDDI Repository Log	334
Table 60	Properties for the ESR Installer Log	334
Table 61	Properties for the Enterprise Designer Master Log	335
Table 62	Properties for Enterprise Monitor Log	336
Table 63	Properties for the Logical Host Log	337
Table 64	Properties for the Integration Server Logs	338
Table 65	Properties for Integration Server Logs	338
Table 66	SRE Monitor Explorer Icons	341
Table 67	eGate 5.0 Terms	353

Introduction

This chapter introduces you to this *eGate Integrator User's Guide*, its general purpose and scope, and its organization. It also provides sources of related documentation and information.

1.1 Purpose and Scope

The *eGate Integrator User's Guide* provides general information about the features and operation of SeeBeyond® eGate Integrator 5.0.

Note: *Any operational explanations provided in this document are generic, for reference purposes only, and do not necessarily address the specifics of setting up individual eGate Projects.*

1.2 Intended Audience

This guide is intended for experienced PC users who have the responsibility of helping to set up and maintain a fully functioning ICAN Suite system. This person must also understand any operating systems on which eGate will be installed (Windows and UNIX) must be thoroughly familiar with Windows-style GUI operations.

1.3 Organization of Information

This document provides information about eGate Integrator 5.0 and includes the following chapters:

- **Chapter 1 “Introduction”** describes the purpose of *eGate Integrator User's Guide* includes writing conventions and a list of related documents.
- **Chapter 2 “System Overview”** provides an overview of the general structure, architecture, and operation of the eGate system.
- **Chapter 3 “Enterprise Manager”** provides a detailed overview of the Enterprise Manager, including its structure and operation.

- **Chapter 4 “Enterprise Designer”** provides a detailed overview of the Enterprise Designer, including its structure and operation.
 - **Chapter 5 “eGate Projects”** explains how to create a Connectivity Map and use the Configuration Editor to modify eWay and JMS connections between Connectivity Map components.
 - **Chapter 6 “Web Services”** describes how to use the Web Services features of eGate Integrator.
 - **Chapter 7 “Object Type Definitions”** describes how to create Object Type Definitions (OTDs).
 - **Chapter 8 “Java Collaboration Definitions”** overviews the Java Collaboration Definition creation process.
 - **Chapter 9 “XSLT Collaboration Definitions”** looks at XSLT Collaboration Definitions.
 - **Chapter 10 “Environments”** explains how to create and populate eGate Environments, and how to configure and start Logical Hosts.
 - **Chapter 11 “Project Deployment”** explains how to create and activate deployment profiles.
 - **Chapter 12 “ICAN Security Features”** discussed the various security features in the ICAN Suite.
 - **Chapter 13 “Logging”** provides troubleshooting tips and describes the eGate logs
- In addition, refer to the [Glossary](#) on page 347 for a list of eGate-related terminology.

1.4 Writing Conventions

The writing conventions listed in this section are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Buttons, File, Icon, and Menu Names	Bold text	Click OK to save and close. From the File menu, select Exit . Select the logicalhost.exe file.
Command Line Code	Courier font	<code>java -jar EnterpriseDesigner.jar</code>
Hypertext Links	Blue text	For more information, see “Writing Conventions” on page 23 .
Notes	<i>Bold Italic</i> text	<i>Note: If a toolbar button is dimmed, you cannot use it with the selected component.</i>

1.5 Supporting Documents

The following SeeBeyond documents provide additional information about the eGate Integrator system as explained in this guide:

- *eGate Integrator JMS Reference Guide*
- *eGate Integrator Release Notes*
- *eGate Integrator Tutorial*
- *SeeBeyond ICAN Suite Deployment Guide*
- *SeeBeyond ICAN Suite Installation Guide*
- *SeeBeyond ICAN Suite Primer*

For information on a specific add-on product (for example, an eWay Intelligent Adapter), see the User's Guide for that product. A complete list of eGate-related documentation is included in the *SeeBeyond ICAN Suite Primer*.

1.6 Online Documents

The documentation for the SeeBeyond ICAN Suite is distributed as a collection of online documents. These documents are viewable with the Acrobat Reader application from Adobe Systems. Acrobat Reader can be downloaded from:

<http://www.adobe.com>

1.7 The SeeBeyond Web Site

The SeeBeyond Web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

<http://www.seebeyond.com>

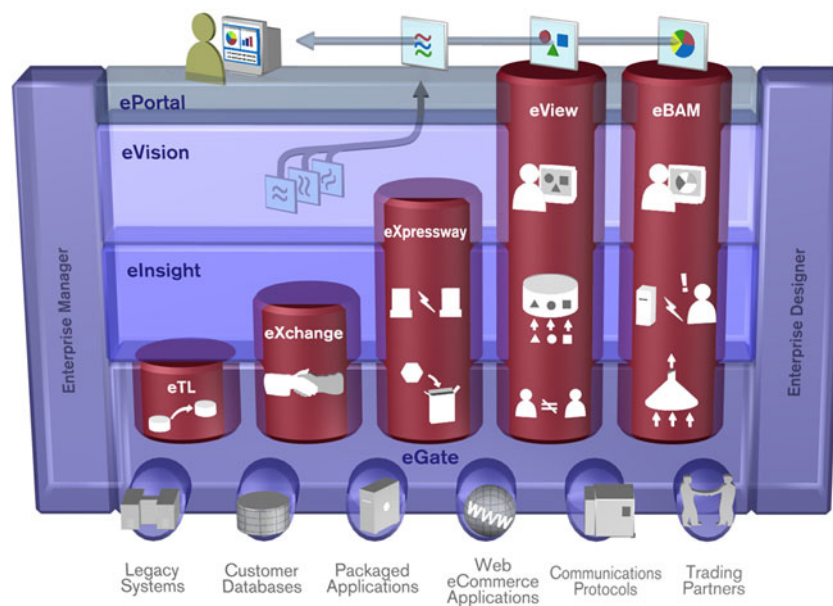
System Overview

This chapter provides an overview of the conceptual operation and general architecture of the eGate Integrator system.

2.1 Introduction

SeeBeyond's Integrated Composite Application Network (ICAN) Suite provides businesses with a comprehensive, unified eBusiness infrastructure to connect, integrate, and manage enterprise-wide software applications running on various computer systems. The full ICAN Suite is depicted in Figure 1.

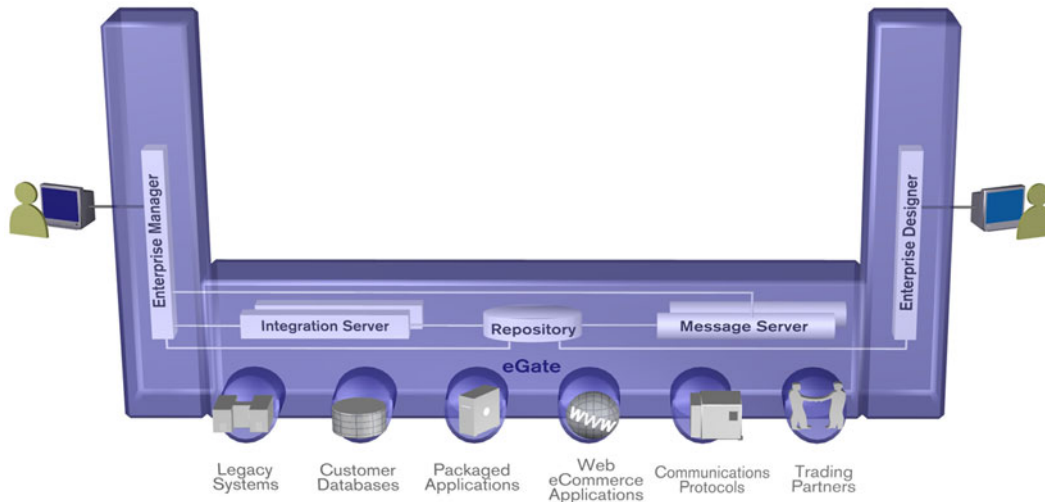
Figure 1 SeeBeyond ICAN Suite



SeeBeyond's eGate Integrator provides the "backbone" for the ICAN suite, integrating the various components of the Suite and all other connected components of the business enterprise. As shown in Figure 2, eGate Integrator includes the Enterprise Manager and Enterprise Designer, which provide graphical user interfaces for managing, configuring, and controlling the entire ICAN Suite and the business

processes running therein. See [Enterprise Manager](#) on page 32 and [Enterprise Designer](#) on page 30.

Figure 2 eGate Integrator



Other major constituents of eGate Integrator shown in Figure 2 are the Integration Server, the JMS IQ Manager, and the Repository, all of which will be described briefly later in this chapter. The flexibility of the eGate system allows the option of deploying it across a distributed network of hardware platforms, if desired, and running it on any combination of SeeBeyond, BEA WebLogic, and IBM WebSphere servers.

eGate Integrator can communicate with and link multiple applications and databases across a variety of different operating systems. eGate performs with a wide variety of hardware, message standards, operating systems, databases, and communication protocols in both real-time and batch (scheduled) integration modes.

2.2 Integration Model

SeeBeyond addresses application integration by means of an eGate Project, which contains the business logic required to solve the specific problem. The Project contains the various logical components and supporting information required to perform the routing, processing, and caching of messages containing the relevant data from one application to another. All Project information is stored in the Repository.

Projects are created using tools contained within the Enterprise Designer and, once deployed, can be run and monitored using Enterprise Manager. Projects can also be set up to be run from the business process level using the SeeBeyond eInsight Business Process Manager, if that product is also installed.

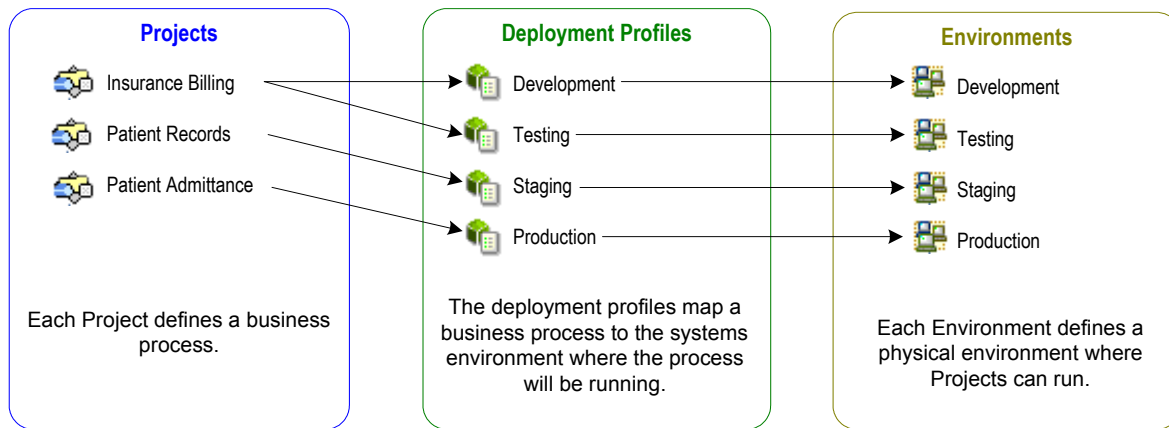
Projects are run within Logical Hosts, which are individual, runtime instances of eGate Integrator. Logical Hosts are defined within Environments, which represent the physical resources required to implement the Project. Projects are mapped to the individual Environments by means of Deployment Profiles, which are defined within

the Enterprise Designer and become part of the Project. Activating the Deployment Profile deploys the Project to the associated Environment.

This structure of Projects, Environments, and Deployment Profiles isolates each implementation into logical and physical components. This provides you with extensive flexibility and efficiency in designing eGate Integrator implementations. For example, once you build your Projects and Environments, you have the flexibility to change each component without having to make changes to the other component.

The finished Project, of course, will run in your production Environment; separate Environments, having the same structure as the production Environment, should be created for development and testing. You may also want some additional Environments, such as staging. The following figure illustrates the eGate Integrator implementation model using a healthcare-related example.

Figure 3 eGate Integrator Implementation Model



In the figure above, any of the Projects can be deployed to any of the Environments via the mapping defined in the deployment profiles. The example in the figure above shows that the patient admittance Project is already in the production phase and therefore was deployed using the production deployment profile. The patient records Project is in the staging phase and was therefore deployed to the staging Environment using the staging deployment profile. The insurance billing Project is still being developed and tested, and therefore it is deployed to development and testing via the development and testing profiles.

In broad outline, an eGate Integrator implementation includes the following steps:

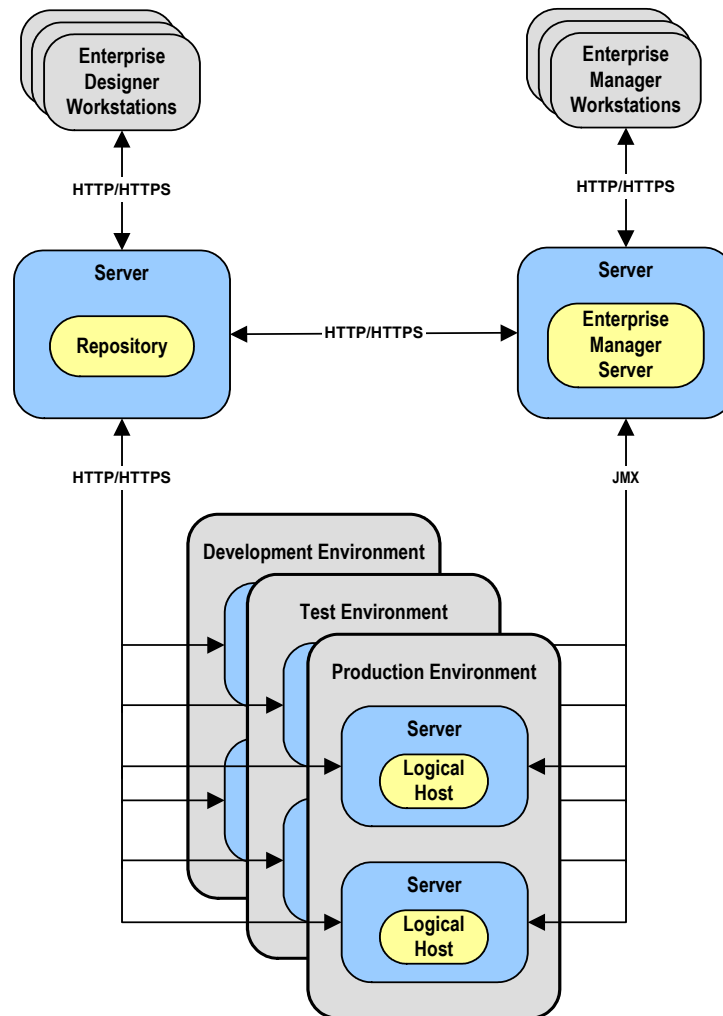
- 1 **Design your Project.**
- 2 **Define your Environments.**
- 3 **Create your Deployment Profiles.**
- 4 **Deploy the eGate Project.**

These implementation steps are all accomplished using the eGate Enterprise Designer, which is introduced in [Enterprise Designer](#) on page 30 and developed further in subsequent chapters.

2.3 System Architecture

eGate Integrator employs a flexible architecture that is ideal for distributed computing environments. As a result, the various components of an eGate Integrator system can reside on the same hardware platform (assuming adequate system resources), or be distributed across several different hardware platforms in the enterprise network. Figure 4 shows an example system implementation that is highly distributed.

Figure 4 Typical eGate Integrator System



2.3.1 Repository

The setup, components, and configuration information for the elements of a Project are stored in the Repository. The Repository also stores all of the product binary files that are required at runtime by the Logical Hosts. The components and configurations are downloaded to the Logical Host during the initial bootstrap process and as needed after design-time configuration changes are made.

As shown in Figure 4, a single Repository serves the entire enterprise. This common Repository is used for development, testing, and production purposes. Communication between the Repository and other eGate components can be configured to use either HTTP or HTTPS. The Enterprise Designer and Enterprise Manager clients can communicate with the Repository and Enterprise Manager servers through a firewall.

2.3.2 Environments

An eGate Environment represents the physical system required to implement a Project. It consists of a collection of Logical Hosts, capable of hosting components of the ICAN Suite, along with information about external systems involved in the implementation.

- **Logical Hosts**

Each Environment contains one or more Logical Hosts. A Logical Host contains one or more **integration servers**, which are the engines that run eGate Collaborations and eWays, and one or more **message servers** such as the SeeBeyond JMS IQ Manager, which manage JMS topics (publish-and-subscribe messaging) and queues (point-to-point messaging).

- **External Systems**

An external system is a representation of a real, physical system that exists within the specific Environment, with configuration properties for locating and accessing that system.

In the example system shown in Figure 4, the production environment is split across two hardware platforms, each running a single Logical Host. Separate environments for development and testing should duplicate the structure of the production environment. The test environment should be supported by hardware similar to that supporting the production environment, to allow performance and load testing to give representative throughput results. The hardware supporting the development environment, however, does not usually have the same performance requirements as that supporting the test and production environments.

An eGate Project is created within the development environment, then migrated to the test environment, and finally to the production environment. This migration path is a necessary and highly critical practice in implementing a working system.

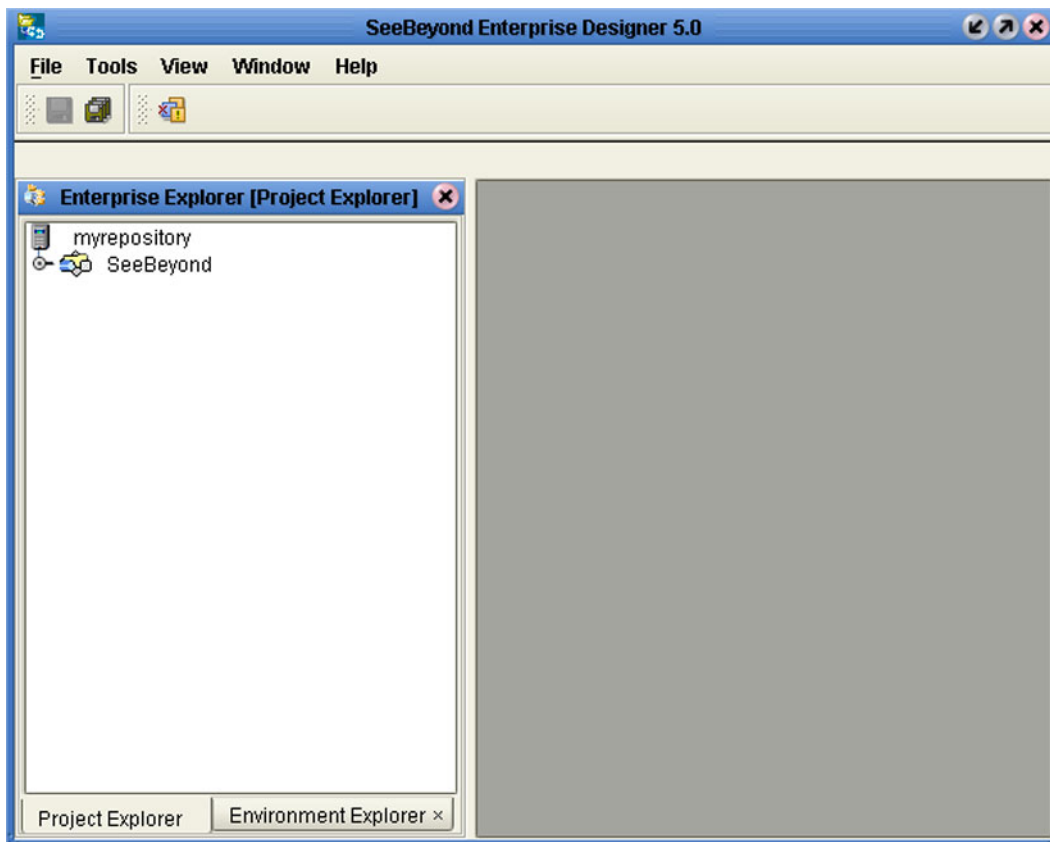
Note again that there is no requirement for the components shown in Figure 4 to run on separate systems; all could run on a single system, provided that resources (CPU, memory, and disk) are sufficient to support the concurrent usage.

2.4 User Interfaces

2.4.1 Enterprise Designer

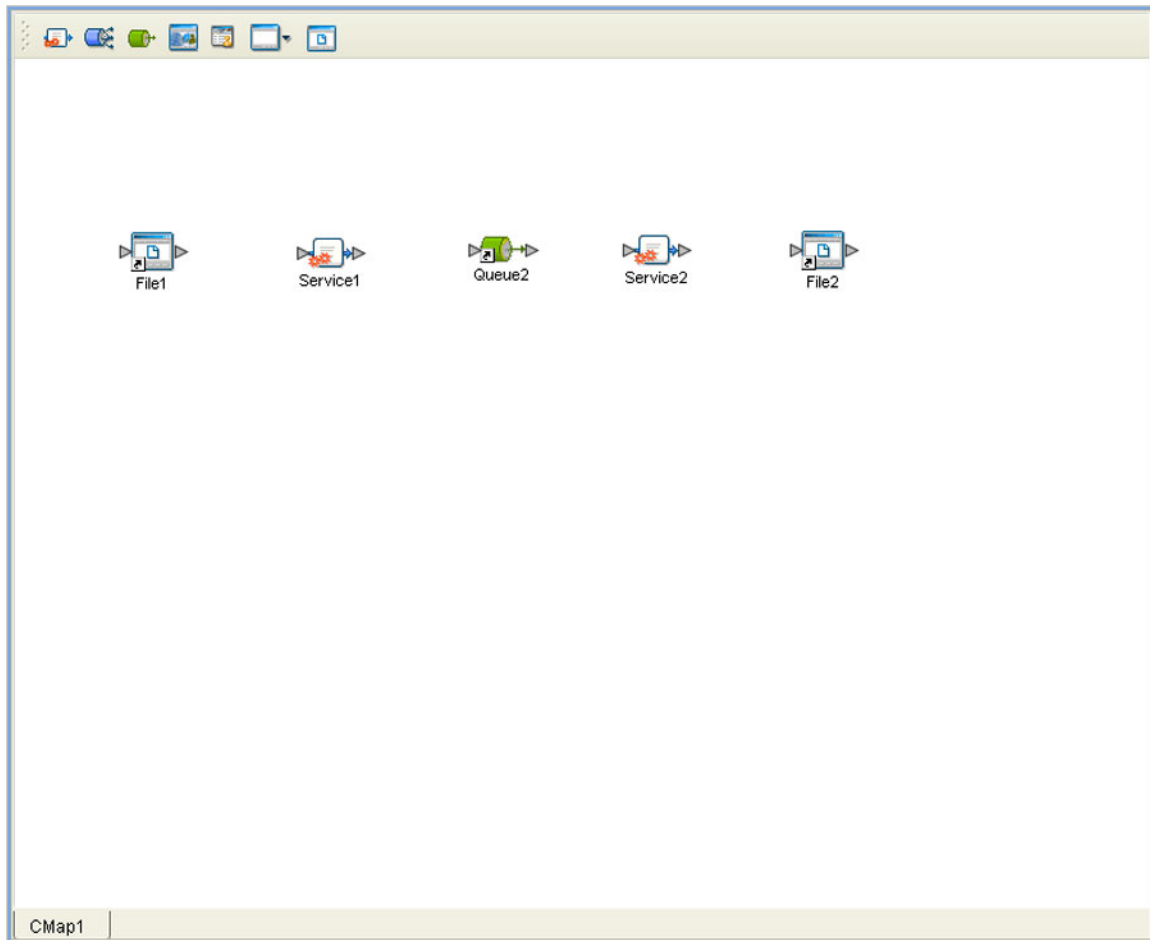
The Enterprise Designer graphical user interface (GUI) is used to create and configure the logical components and physical resources of an eGate Project. Through this GUI (see Figure 5), you can develop Projects to process and route data through an eGate Integrator system.

Figure 5 Enterprise Designer



The major features of the Enterprise Designer are the Enterprise Explorer on the left, and an editor panel on the right—which is initially blank. The Enterprise Explorer follows the familiar Windows Explorer format, displaying a tree structure. The editor panel displays a variety of editors, depending upon what component is selected in the Enterprise Explorer. The Connectivity Map Editor (see Figure 6) provides a graphic example of one of these, in which logical components of a Project are created and connected.

Figure 6 Connectivity Map Editor



The features and usage of the Connectivity Map Editor are described in [eGate Projects](#) on page 81. Other editors are displayed for creating and modifying Object Type Definitions, Collaboration Definitions, Deployment Profiles, and other Project components.

The Enterprise Designer also includes the design-time functionality for other ICAN products, such as eInsight and eXchange. For more information on using other ICAN products in the Enterprise Designer, see the product documentation for those products.

For more information on the Enterprise Designer, see [Enterprise Designer](#) on page 53 .

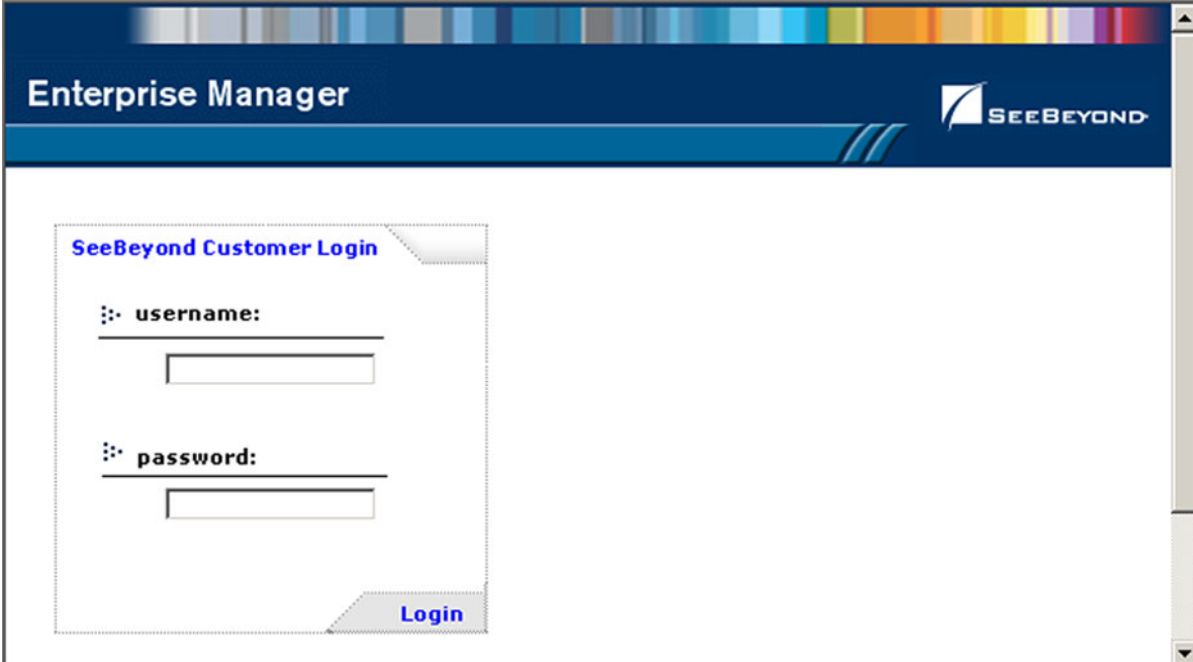
2.4.2 Enterprise Manager

The Enterprise Manager is a Web-based application you use for:

- Managing and monitoring eGate runtime components.
- Installing ICAN Suite products into the Repository.
- Downloading and installing products from the Repository.
- Accessing other Web-based ICAN Suite products.
- Accessing ICAN Suite product documentation.

The Enterprise Manager (see Figure 7) is accessed via Microsoft Internet Explorer.

Figure 7 SeeBeyond Enterprise Manager Login



The screenshot shows a web browser window displaying the SeeBeyond Enterprise Manager login page. The page features a dark blue header with the text "Enterprise Manager" on the left and the "SEE BEYOND" logo on the right. Below the header is a white box titled "SeeBeyond Customer Login". Inside this box, there are two input fields: "username:" and "password:", each with a small icon to its left. Below the password field is a "Login" button.

For more information on the Enterprise Manager, see [Enterprise Manager](#) on page 33.

For more information on Web Services capability, see [Web Services](#) on page 101.

Enterprise Manager

This chapter provides an introduction to the ICAN Suite Enterprise Manager.

3.1 Overview

Enterprise Manager is a Web-based interface with which you can install and update eGate Integrator, and monitor and manage deployed eGate components.

***Important:** Enterprise Manager works only with Microsoft Internet Explorer.*

3.1.1 Installing and Updating eGate

eGate Integrator components are uploaded from the installation media (CD-ROMs) to the Repository server via the Enterprise Manager. These products are then available to be downloaded and installed from the Repository server. For information on installing and updating eGate components, see the *eGate Integrator Installation Guide*.

3.1.2 Monitoring and Managing eGate

The Enterprise Manager allows you to monitor and manage deployed eGate components in real-time.

- **The Enterprise Monitor** on page 37 describes features of the Monitor interface itself.
- **Monitoring Collaborations** on page 39 describes the various ways you can monitor the Collaborations and control various components in your Project.
- **Viewing Alerts** on page 45 describes how to view and set the status of Alerts.
- **Viewing Logs** on page 48 briefly describes how to view, sort, search, and filter messages in the log files.
- **Setting Log Levels** on page 50 describes how to set logging levels in Enterprise Manager.
- The Enterprise Manager also contains a facility that allows you to monitor and manage schemas from e*Gate 4.5.x in eGate 5.0, using the Schema Runtime Environment. See **The SRE Monitor** on page 341.

3.2 Starting Enterprise Manager

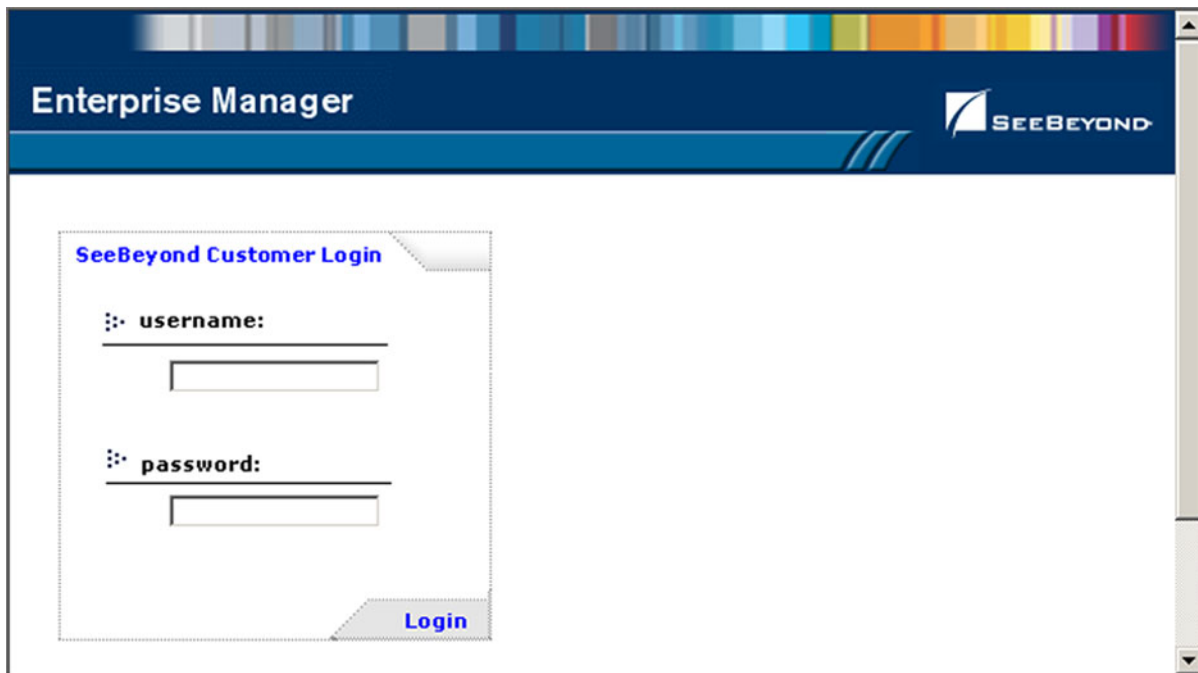
To start the Enterprise Manager

- 1 Launch Internet Explorer.
- 2 Enter **http://hostname:portnumber** in the **Address** box to display the SeeBeyond Customer Login window shown in Figure 8.

Note: The *hostname* is the TCP/IP host name of the server where you installed the Repository. The *portnumber* is the number of the port you entered during installation of the Repository. See the SeeBeyond ICAN Suite Installation Guide.

Important: The TCP/IP host name must be alphanumeric.

Figure 8 Enterprise Manager Login



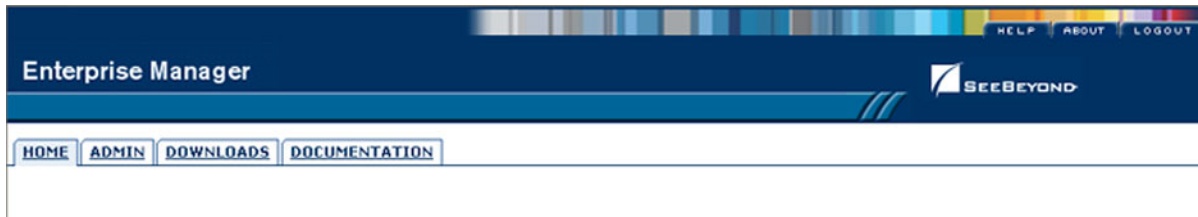
The screenshot shows a web browser window displaying the SeeBeyond Customer Login page. The page has a dark blue header with the text "Enterprise Manager" on the left and the "SEE BEYOND" logo on the right. Below the header is a white content area with a login form. The form is titled "SeeBeyond Customer Login" and contains two input fields: "username:" and "password:". Each field has a small icon to its left. Below the password field is a "Login" button.

- 3 Enter your login ID and password in the **Username** and **Password** boxes and click **Login**.

3.3 The Enterprise Manager Interface

Once you have logged in, you see the full Enterprise Manager user interface (see Figure 9).

Figure 9 Enterprise Manager GUI



The Enterprise Manager is organized into four pages represented by tabs, as described in the following table.

Table 2 Enterprise Manager - Tabs

Tab	Function
Home	The Home tab is used for accessing the eGate Monitor, which is the main subject of this chapter. See Home on page 36.
Administration	The Administration tab is used in installing and updating ICAN components. See the <i>SeeBeyond ICAN Suite Installation Guide</i> for information.
Downloads	The Downloads tab is used in installing and updating ICAN components. See the <i>SeeBeyond ICAN Suite Installation Guide</i> for information.
Documentation	The Documentation tab is used for accessing ICAN Suite documentation. See Documentation on page 36.

There are also three small tabs in the upper-right corner of the Enterprise Manager, which are described in the following table.

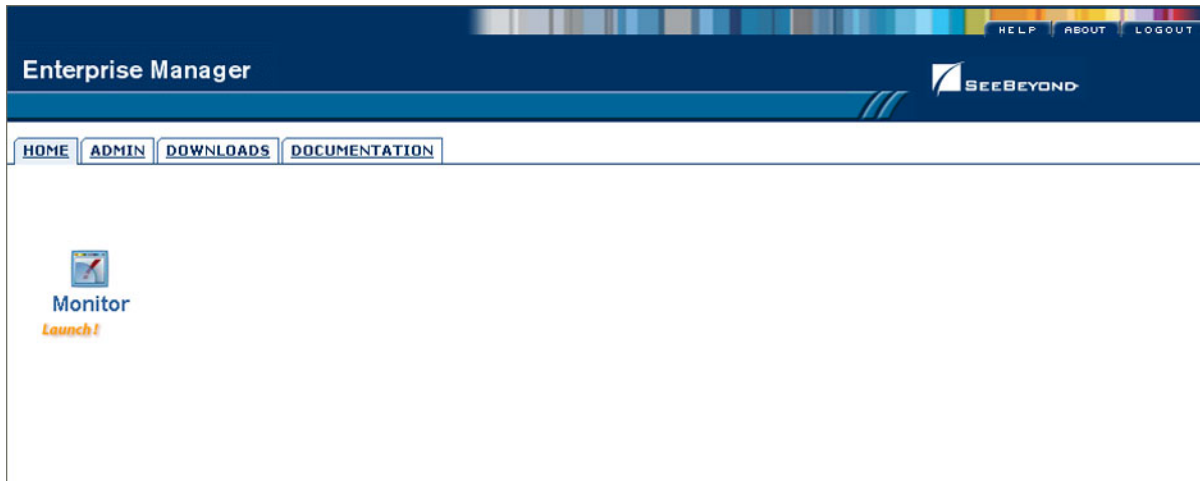
Table 3 Enterprise Manager - Tabs

Tab	Function
Help	The Help tab provides access to the online help system.
About	The About tab displays the installed version of the product. This tab is displayed only on the Home page.
Home	The Home tab returns you to the Home page. This tab is displayed on all pages other than the Home page.
Logout	The Logout tab logs you out of the Enterprise Manager and returns you to the Login page.

3.3.1 Home

The Enterprise Manager’s **Home** tab (see Figure 10) contains a link to the Enterprise Monitor. Click the **Monitor** icon to launch the Monitor (see [The Enterprise Monitor](#) on page 37).

Figure 10 Monitor Launch Window



3.3.2 Documentation

The **Documentation** tab (see Figure 11) contains links to the latest versions of the SeeBeyond ICAN documentation in PDF format. You must have Adobe Acrobat Reader installed on your computer to view or print these documents.

Figure 11 Documentation Tab

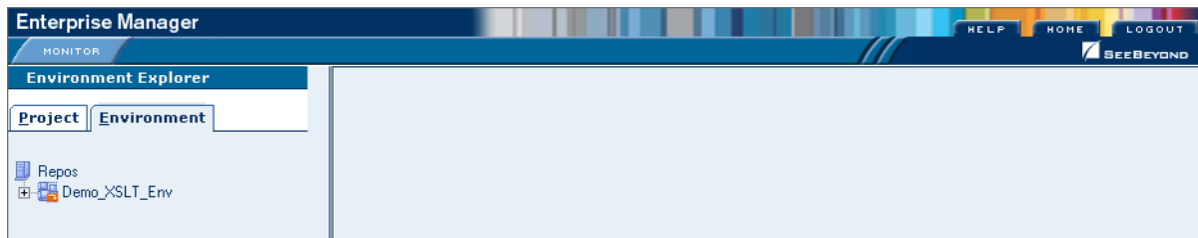


Note: You must download the documentation SAR files from the installation disk before you can access any documents from this page (see the ICAN Installation Guide).

3.4 The Enterprise Monitor

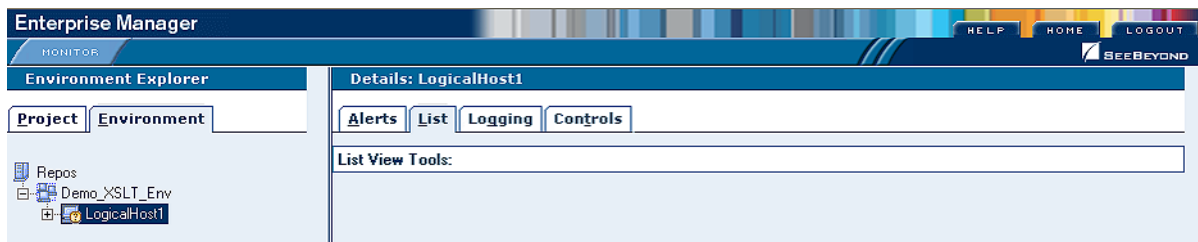
The Monitor has structure similar to that of the Enterprise Designer, with an Explorer panel on the left and a Details panel on the right. Initially, the Details panel is blank as shown in Figure 12.

Figure 12 Monitor Interface - Initial



Like the Enterprise Manager itself, the Monitor's **Details** area is organized into sections represented by tabs. Which tabs are present depends upon the component selected in the Explorer. For example, selecting the Logical Host displays the Monitor page shown in Figure 13.

Figure 13 Monitor Interface - Environment



At times, the Details panel will have two parts, as shown in [Figure 16 on page 40](#), to display an additional level of information. In this case, different tabs will be displayed in the upper and lower panels. The full set of tabs is described in Table 4.

Table 4 Monitor Interface - Details Tabs

Tab	Function
Alerts	Displays functionality-related information about the component selected in the Explorer. See Viewing Alerts on page 45 for an example.
List	Displays a list presenting information about the component selected in the Explorer. See Monitoring Collaborations on page 39 for an example.
Logging	Displays all log messages for the component selected in the Explorer. See Viewing Logs on page 48 for an example.
Controls	Displays controls that allow an Administrator to intervene in the run-time process and perform tasks such as rolling transactions forward or backward. See Editing XA Transactions on page 51 for an example.

Table 4 Monitor Interface - Details Tabs

Tab	Function
Summary	Displays a summary of information regarding the component selected in the upper Details panel. See Monitoring Collaborations on page 39 for an example.
Consumption	Displays the number of messages processed by the component selected in the upper Details panel, and the number of messages still pending. See Monitoring Collaborations on page 39 for an example.

Note: Not all listed tabs are always present, and tabs may be divided between an upper and lower Details panel.

The Monitor interface offers the following viewing controls:

- **ALT** and drag the cursor to scroll.
- **CTRL** and click to zoom out.
- **CTRL-SHIFT** and click to zoom in.

3.4.1 Monitoring Collaborations

Enterprise Manager has extensive monitoring capabilities for Projects, one of the most useful being for Collaborations. To demonstrate these capabilities, we shall take an example Project whose Connectivity Map is shown in Figure 14.

Figure 14 Example Project Connectivity Map



When you launch the Monitor Server, the Environment Explorer is displayed by default. If you expand the component tree and select **Collaborations**, the *List* tab in the *Details* panel will display all Collaborations deployed on that Integration Server (see Figure 15).

Figure 15 Monitor - Details - List Collaborations

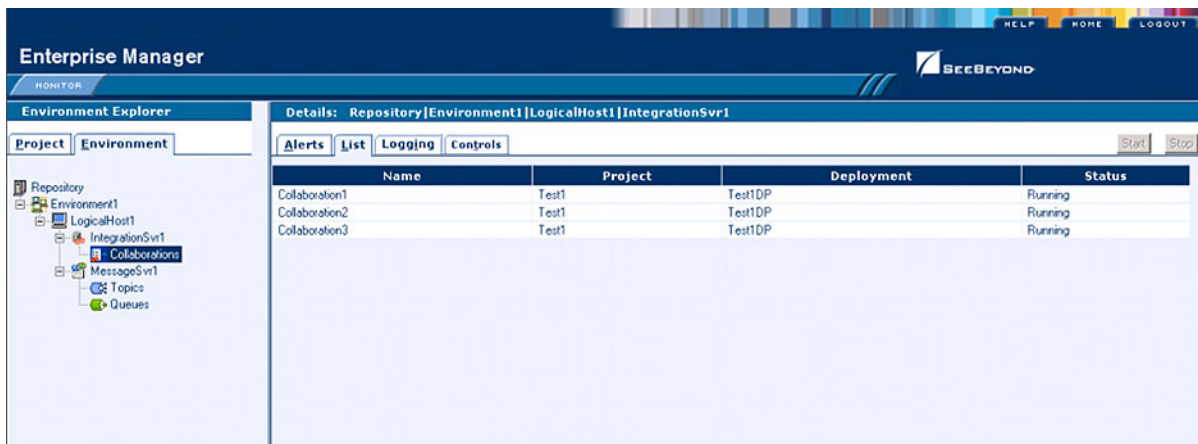
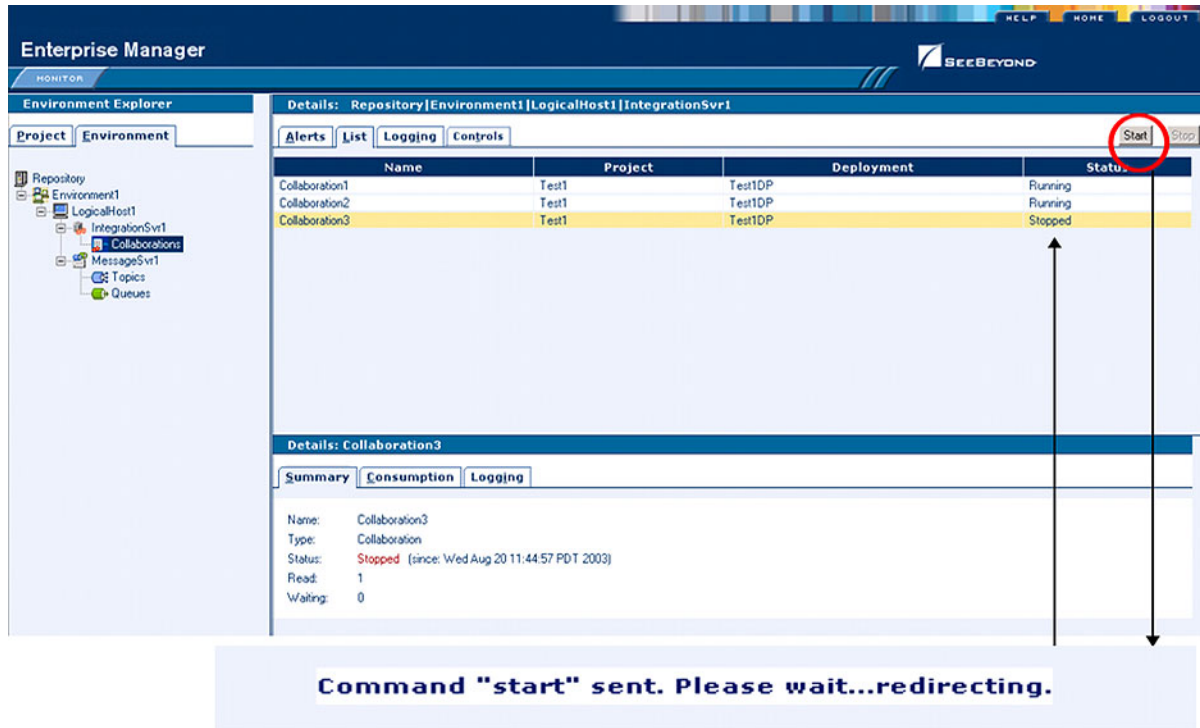


Table 5 Collaboration Status Types

Status	Description
Running	Running indicates that the Collaboration is up and running, and is either processing, or is ready to process, a message.
Stopped	Stopped indicates that the Collaboration is not accepting any further inbound messages.
Unknown	Unknown indicates that the monitor lost contact with the Collaboration. This status is shown if a fatal error occurs either with the Collaboration itself, or with the mbean that monitors that Collaboration.

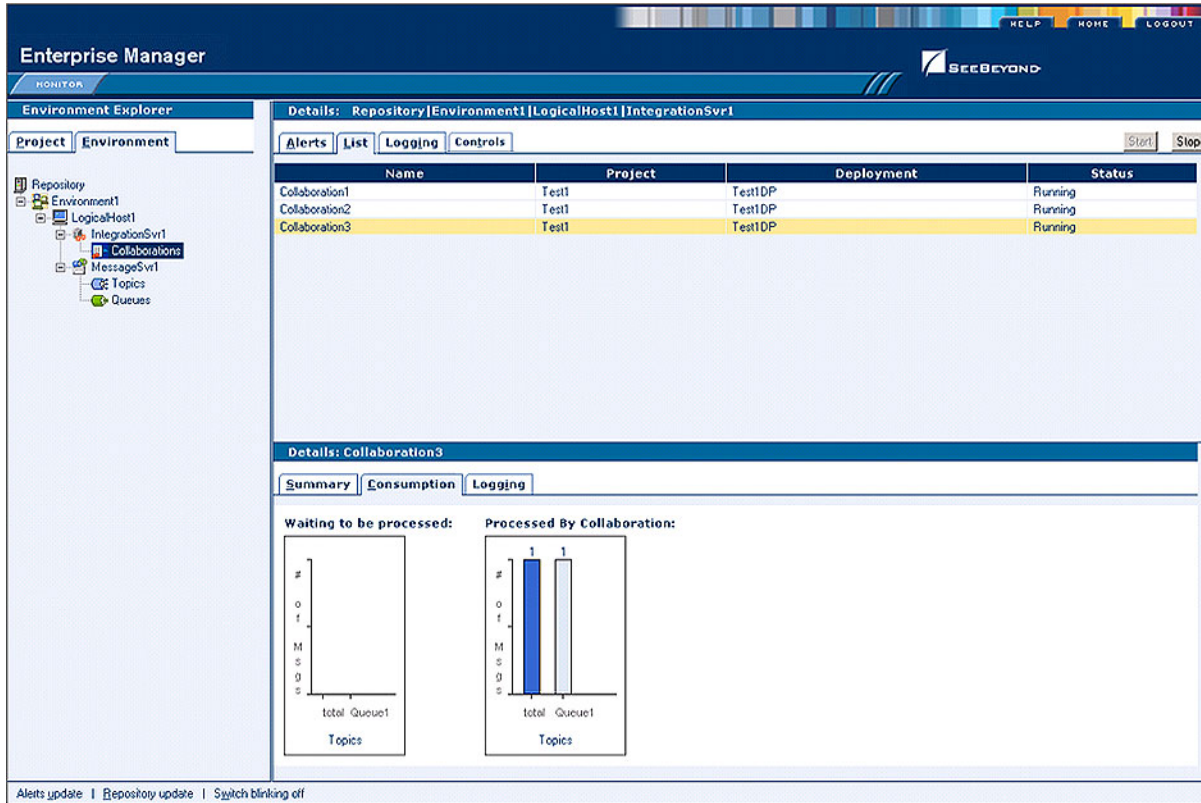
By selecting a specific Collaboration in the list, you can start or stop the Collaboration using the buttons above the list (see Figure 16). By default, a Collaboration summary is displayed in the lower *Details* panel.

Figure 16 Starting and Stopping Collaborations - Details Panel



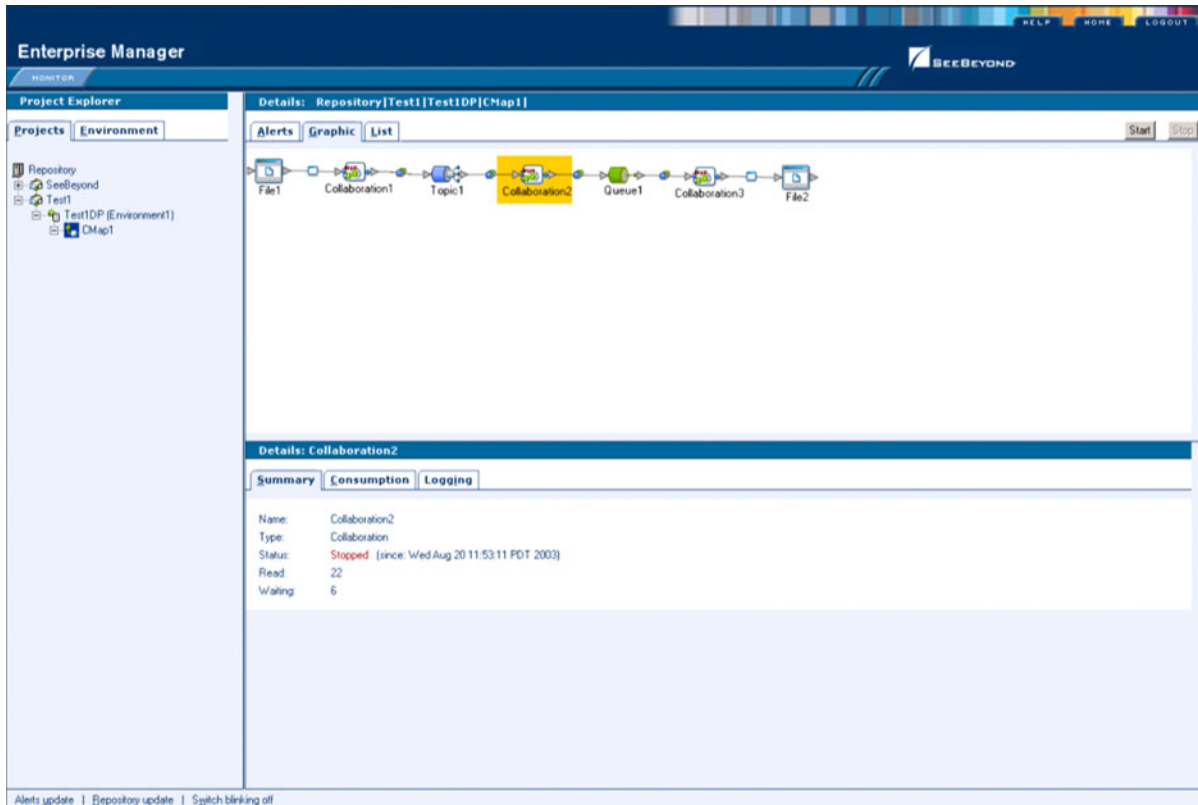
If you select the **Consumption** tab, you will see the number of pending and processed messages in graphical form (see Figure 17).

Figure 17 Pending and Processed Messages



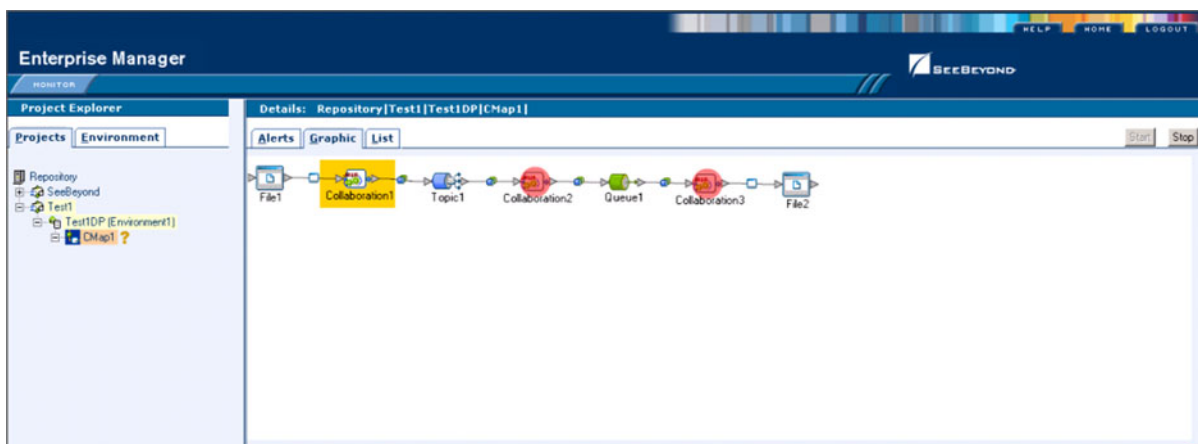
As an alternative, you can monitor Collaborations using the Projects Explorer, which displays all existing Deployment Profiles and Connectivity Maps. You will see the same Connectivity Map as in the Enterprise Designer, with the currently selected Collaboration highlighted in yellow (see Figure 18).

Figure 18 Monitor - Project Explorer



Additionally, non-active Collaborations are highlighted in red, making them immediately visible (see Figure 19).

Figure 19 Active/Non-active Collaborations



As with the monitoring option using the Environment Explorer (shown in [Figure 17 on page 41](#)), selecting a specific Collaboration in the Connectivity Map will display details in the lower panel (see [Figure 20](#) and [Figure 21](#)).

Figure 20 Pending and Processed Messages

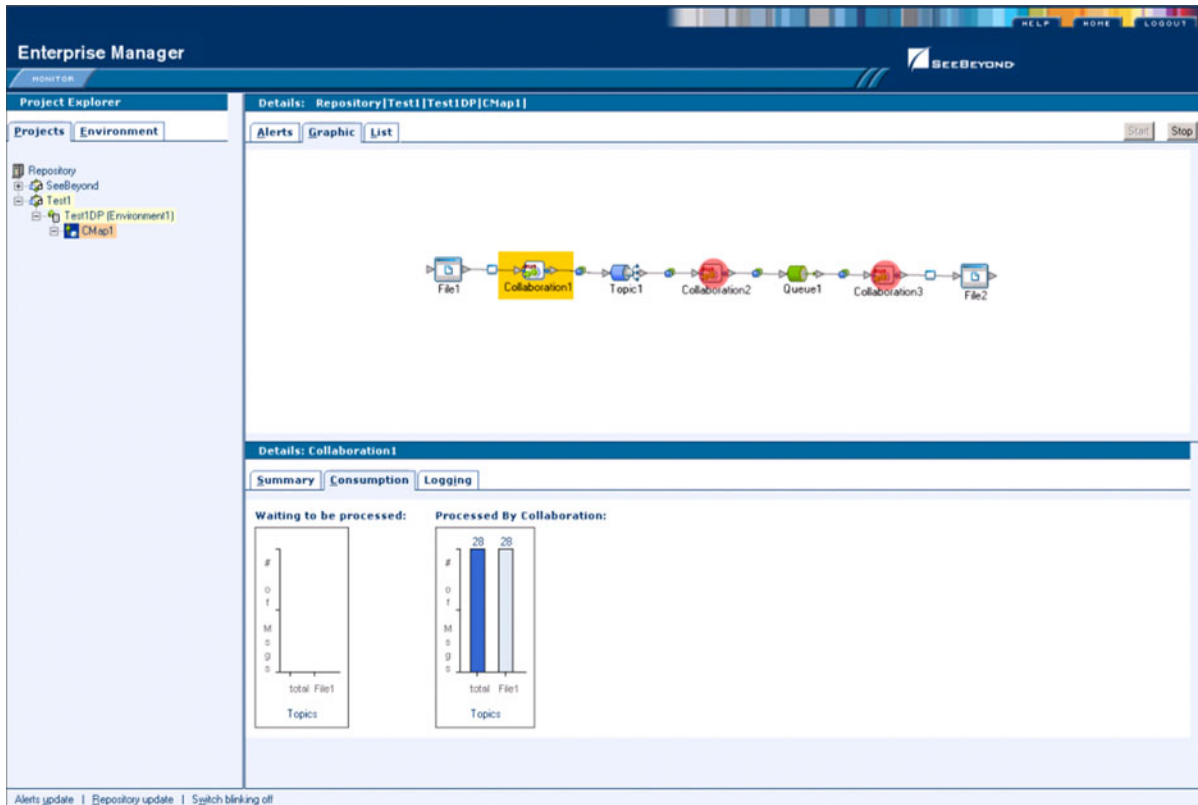
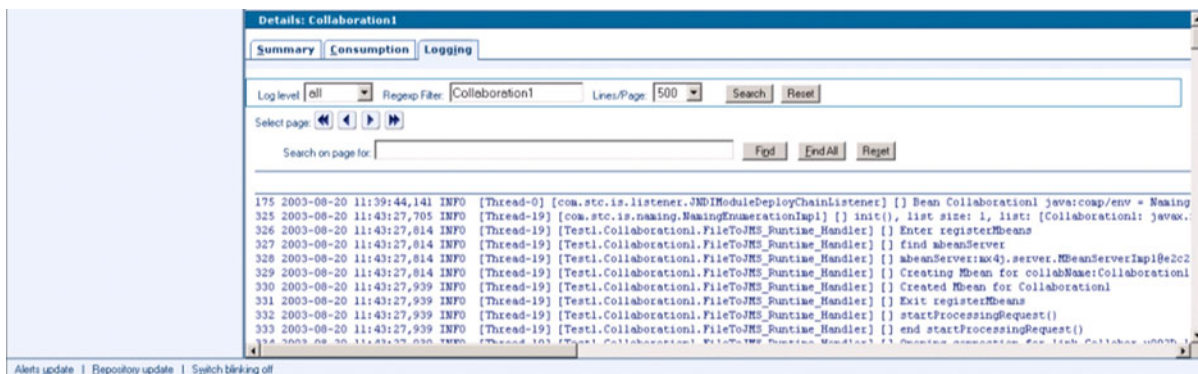


Figure 21 Logging Details



If you select a Connectivity Map in Project Explorer, you can see the status of all components of that Connectivity Map (see Figure 22).

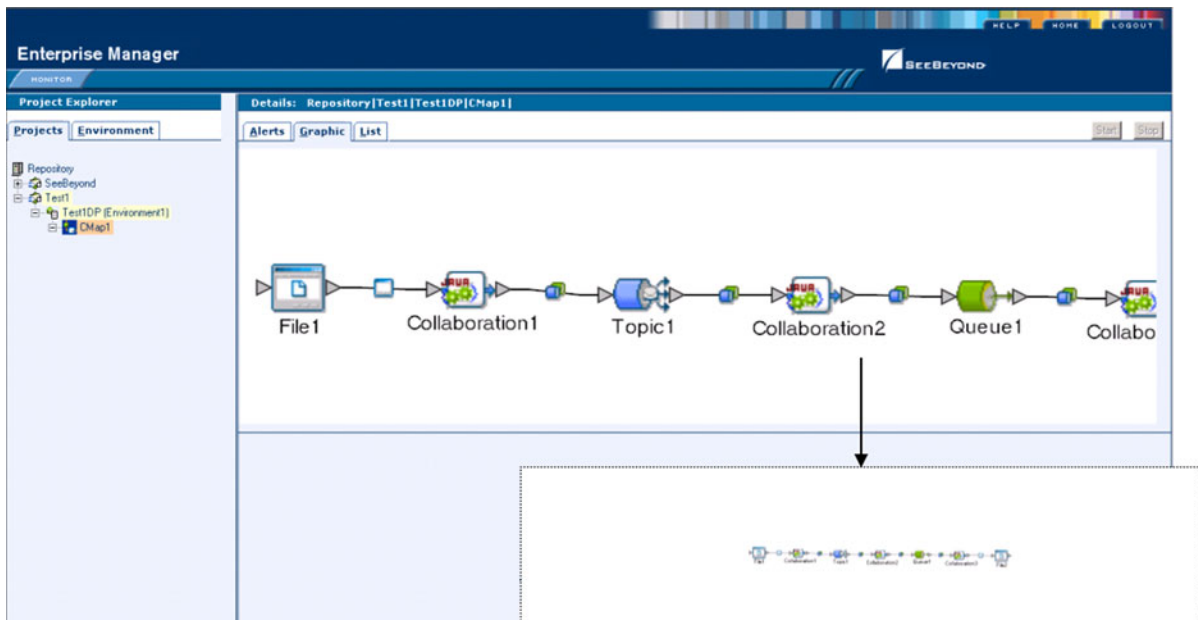
Figure 22 Connectivity Map Details: Components



As mentioned previously, you can scroll and zoom in or out on a Connectivity Map display using the following commands (see Figure 23):

- ALT and drag the cursor to scroll.
- CTRL and click on the Connectivity Map to zoom out.
- CTRL-SHIFT and click on the Connectivity Map to zoom in.

Figure 23 Connectivity Map View - Zoom In/Out



3.4.2 Viewing Alerts

Selecting a Collaboration and clicking the **Alerts** tab in the upper Details panel displays a list of all alerts for the selected Collaboration (see Figure 24). These can be sorted by different criteria, or marked as observed or resolved.

Figure 24 Collaboration Alerts

Date/Time	Environment	Logical Host	Server	Component	Physical Host	Severity	Type	Status	State	Description
8/20/03 11:45 AM	Environment1	LogicalHost1	IntegrationSrv1	Collaboration3	aloh_dell	INFO	ALERT	Unobserved	Running	Collaboration Collaboration3 under Project Test1 on IntegratorSrv1 on LogicalHost1 in Environment1 on aloh_dell is running.
8/20/03 11:44 AM	Environment1	LogicalHost1	IntegrationSrv1	Collaboration3	aloh_dell	INFO	ALERT	Unobserved	Stopped	Collaboration Collaboration3 under Project Test1 on IntegratorSrv1 on LogicalHost1 in Environment1 on aloh_dell is stopped.
8/20/03 11:43 AM	Environment1	LogicalHost1	IntegrationSrv1	Collaboration3	aloh_dell	INFO	ALERT	Unobserved	Running	Collaboration Collaboration3 under Project Test1 on IntegratorSrv1 on LogicalHost1 in Environment1 on aloh_dell is running.
8/20/03 11:43 AM	Environment1	LogicalHost1	IntegrationSrv1	Collaboration2	aloh_dell	INFO	ALERT	Unobserved	Running	Collaboration Collaboration2 under Project Test1 on IntegratorSrv1 on LogicalHost1 in Environment1 on aloh_dell is running.

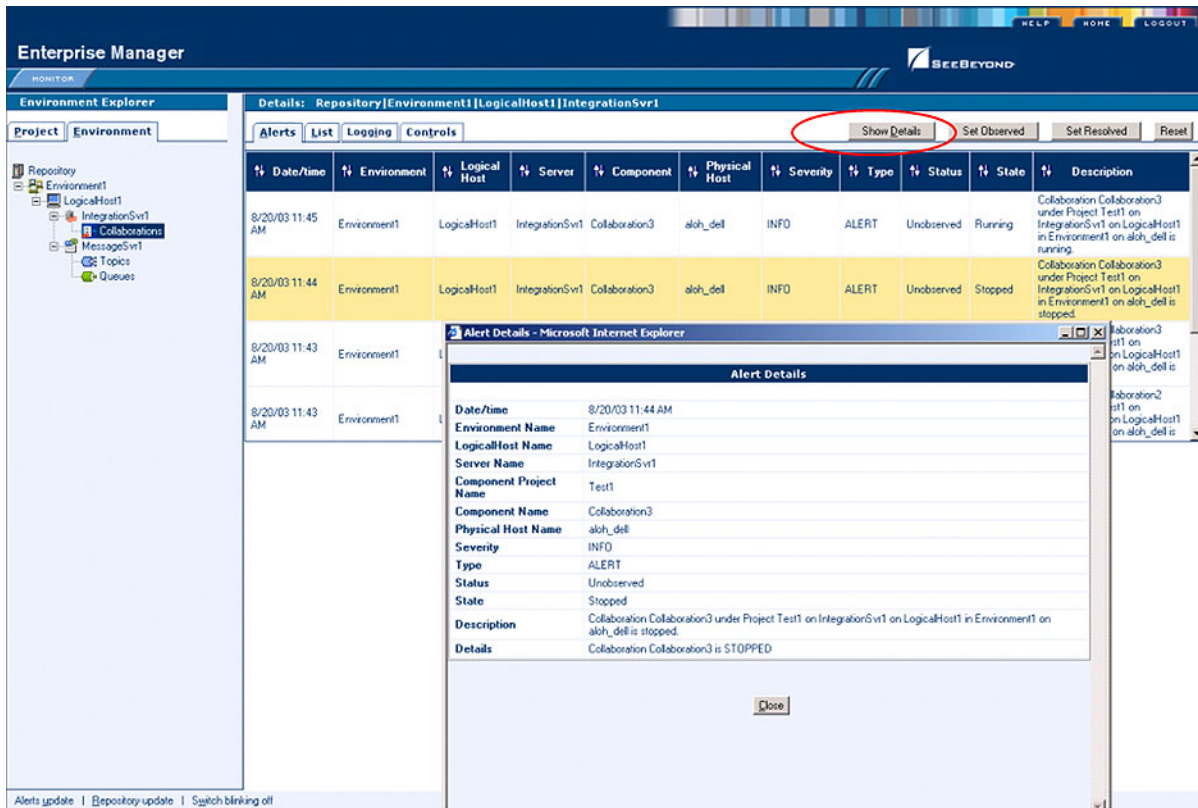
By clicking the **Set Resolved** button, you change the status of the selected Alert to *Resolved* (see Figure 25).

Figure 25 Alert Status

Date/Time	Environment	Logical Host	Server	Component	Physical Host	Severity	Type	Status	State	Description
8/20/03 11:45 AM	Environment1	LogicalHost1	IntegrationSrv1	Collaboration3	aloh_dell	INFO	ALERT	Unobserved	Running	Collaboration Collaboration3 under Project Test1 on IntegratorSrv1 on LogicalHost1 in Environment1 on aloh_dell is running.
8/20/03 11:44 AM	Environment1	LogicalHost1	IntegrationSrv1	Collaboration3	aloh_dell	INFO	ALERT	Resolved	Stopped	Collaboration Collaboration3 under Project Test1 on IntegratorSrv1 on LogicalHost1 in Environment1 on aloh_dell is stopped.
8/20/03 11:43 AM	Environment1	LogicalHost1	IntegrationSrv1	Collaboration3	aloh_dell	INFO	ALERT	Unobserved	Running	Collaboration Collaboration3 under Project Test1 on IntegratorSrv1 on LogicalHost1 in Environment1 on aloh_dell is running.
8/20/03 11:43 AM	Environment1	LogicalHost1	IntegrationSrv1	Collaboration2	aloh_dell	INFO	ALERT	Unobserved	Running	Collaboration Collaboration2 under Project Test1 on IntegratorSrv1 on LogicalHost1 in Environment1 on aloh_dell is running.

By clicking the **Show Details** button, you can see all details for the selected Alert (see Figure 26).

Figure 26 Alert Details



Basic default alert types are listed below. The format of the alert type is xxx-99999, where the first part of the code is the component designator, and the numeric part is a category code.

Table 6 Basic Alert Types

Type	Description
LH-00001	Logical Host {2} in {1} on {0} exited.
LH-00002	Logical Host {2} in {1} on {0} is already running.
LH-00003	Logical Host {2} in {1} on {0} started.
LH-00004	Logical Host {2} in {1} on {0} stopped.
LH-00005	Logical Host {2} in {1} on {0} killed.
LH-00006	Logical Host {2} in {1} on {0} is now running.
IS-00001	Integration Server {3} on {2} in {1} on {0} has exited.
IS-00002	Integration Server {3} on {2} in {1} on {0} is already running.
IS-00003	Integration Server {3} on {2} in {1} on {0} has stopped.
IS-00004	Integration Server {3} on {2} in {1} on {0} is not running (possibly crashed).

Table 6 Basic Alert Types

Type	Description
SNMP-00001	SNMP Agent has been configured.
SNMP-00002	SNMP Agent has not been configured.
COL-00001	Collaboration {5} under Project {4} on {3} on {2} in {1} on {0} is running.
COL-00002	Collaboration {5} under Project {4} on {3} on {2} in {1} on {0} is stopped.

3.4.3 Viewing Logs

Integration Server Level

In the Environment view, select an Integration Server and click the **Logging** tab in the Details panel to display all log messages for the selected component (see Figure 27). You can also filter the list for a specific log level (see Figure 28).

Figure 27 Integration Server Log Messages

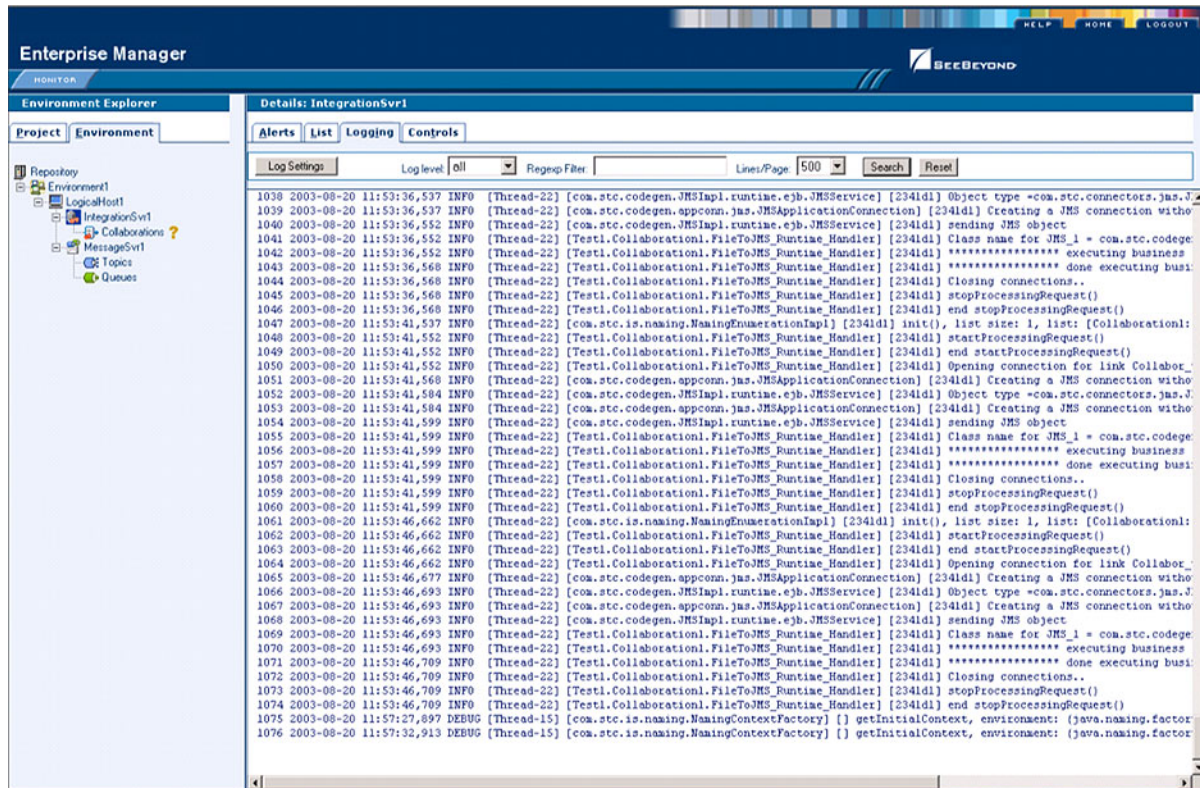
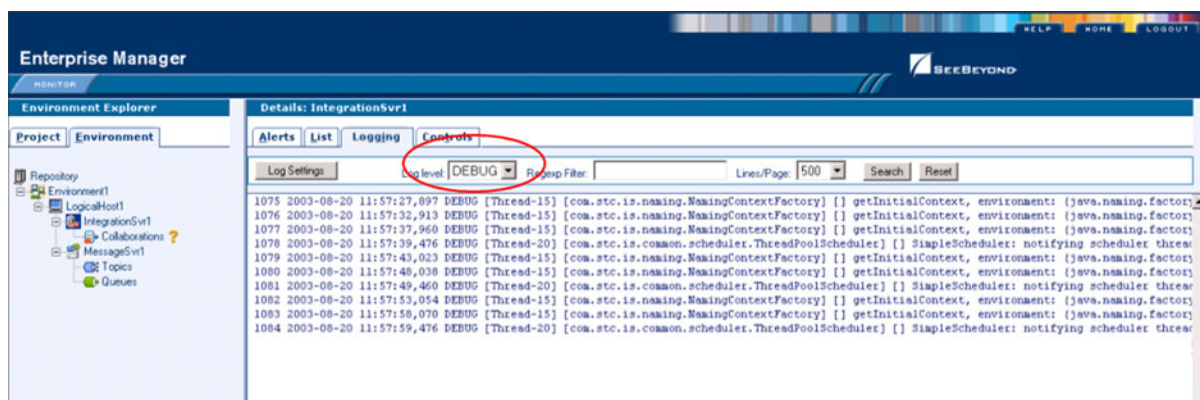


Figure 28 Integration Server Log Messages - Filtered



Collaboration Level

Select a Collaboration, click the **Log** tab in the upper Details panel, and click the **Logging** tab in the lower Details panel to display all log messages for the Collaboration (see Figure 29). This tab contains controls for viewing, sorting, and filtering of the log files on the server. By entering a keyword, you can search for a particular word in the log file (see Figure 30).

Figure 29 Collaboration Log File

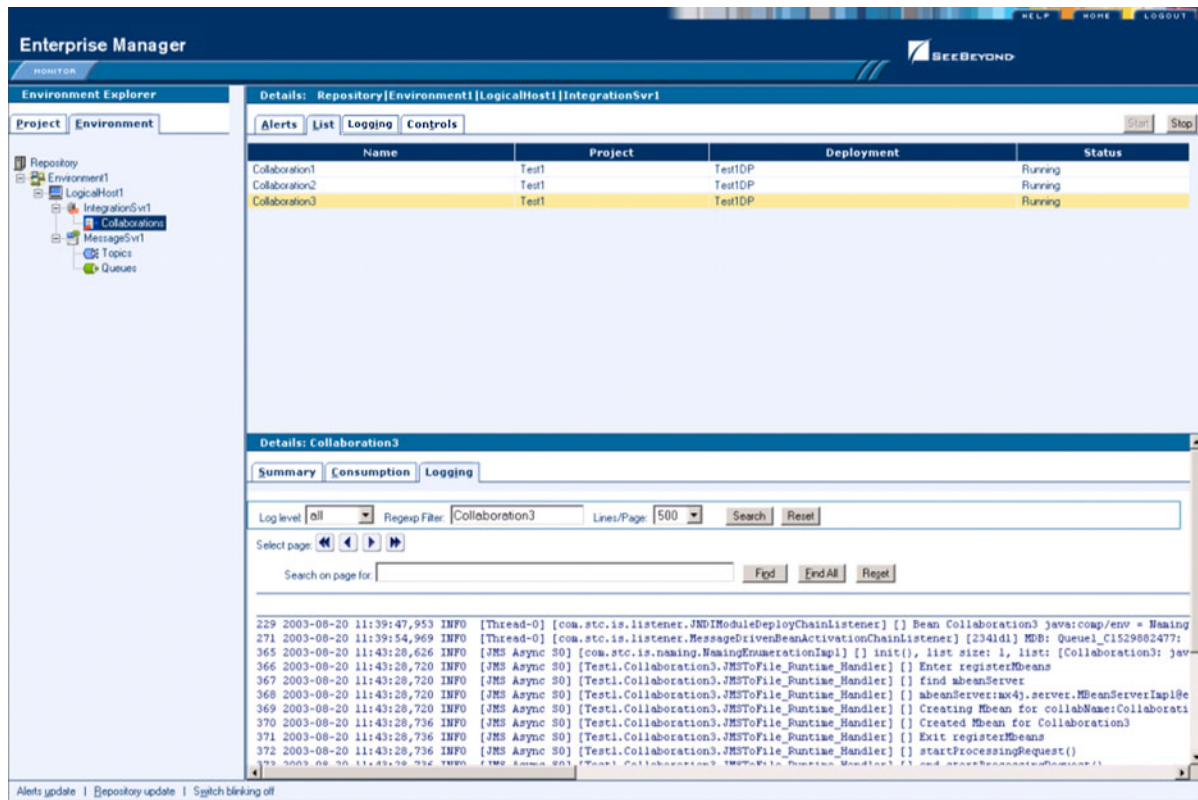
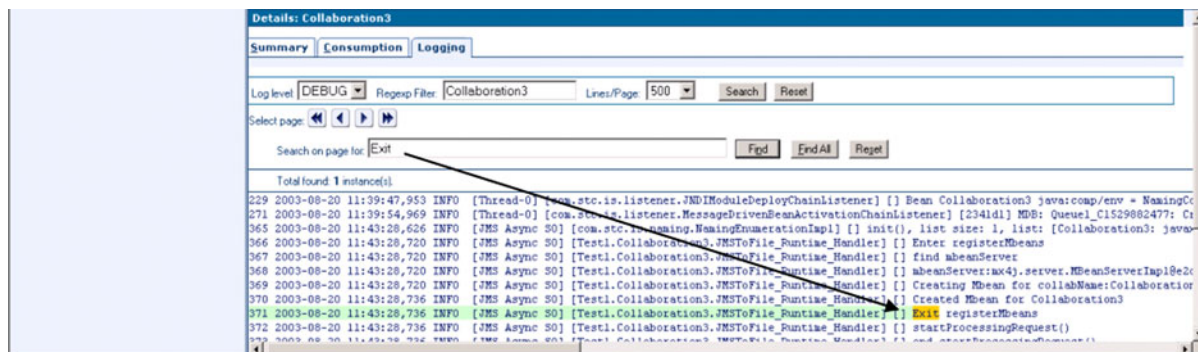


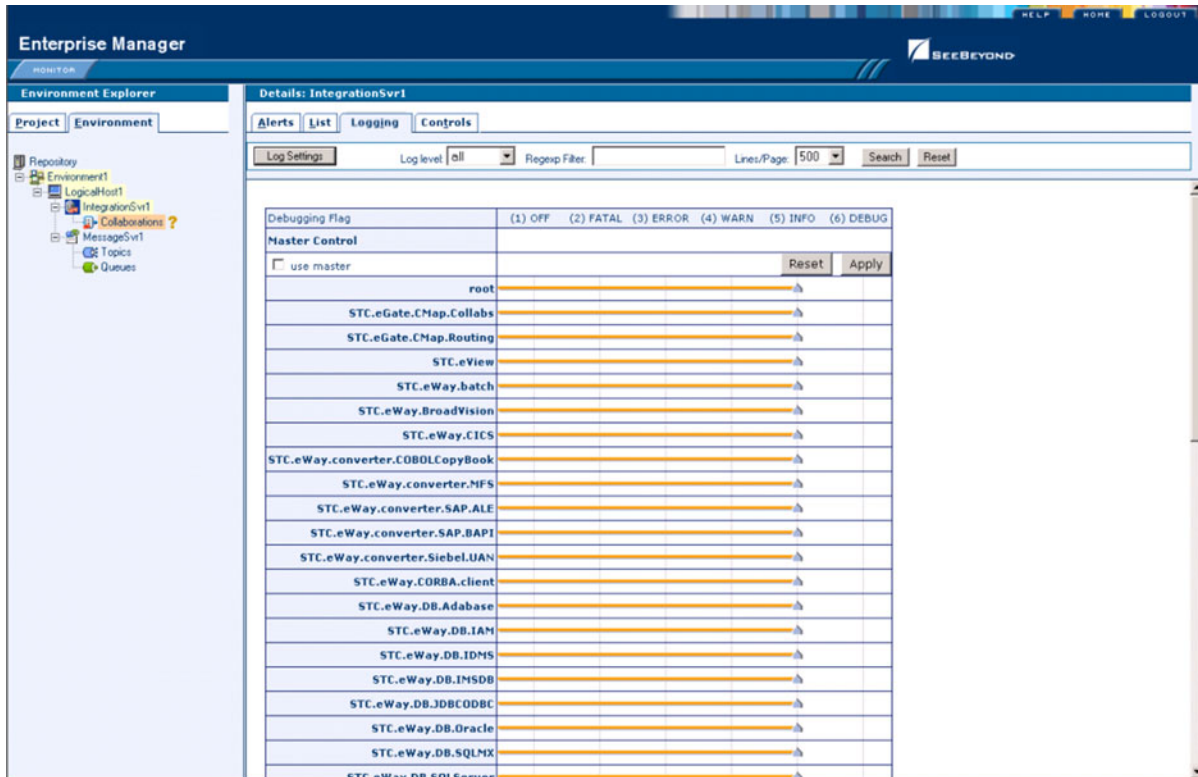
Figure 30 Search on Keyword



3.4.4 Setting Log Levels

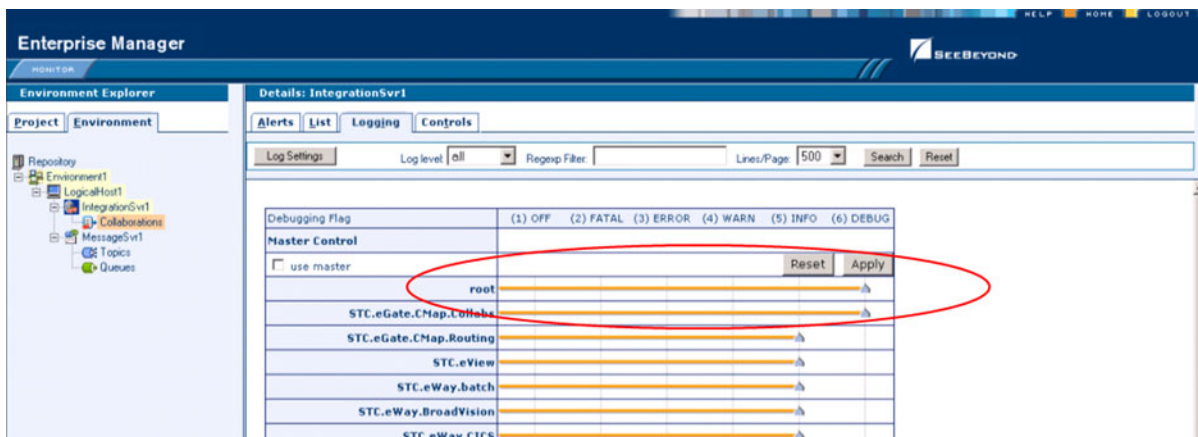
Select an Integration Server or Logical Host and click the **Log Settings** button to display the Logging Properties page (see Figure 31).

Figure 31 Logging Properties Page



To set a log level for a specific property, drag the line to the desired log level and click the **Apply** button. Figure 32 shows the first two properties reset to the **debug** level.

Figure 32 Resetting Log Levels



3.4.5 Editing XA Transactions

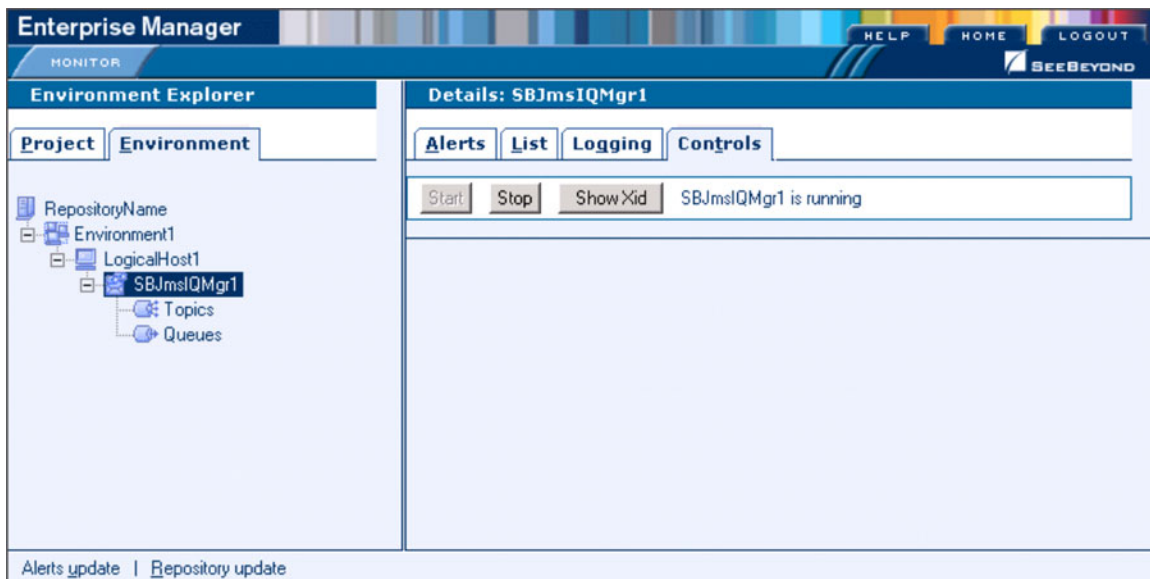
Occasionally, one of the Resource Managers (such as a database server or an external program) involved in an XA transaction will fail to commit. In that case, the transaction stays open until either the Resource Manager commits or rolls back, or the user intervenes.

The following feature is provided so that an Administrator can force those “indoubt” transactions to roll forward or backward. Typically, an external user will advise the Administrator of the problem, specifying the XID. The Administrator can then search for the indoubt transaction using this XID.

To force an Indoubt Transaction

- 1 Click the **Controls** tab in the Details panel for the appropriate message server to display the interface shown in Figure 33.

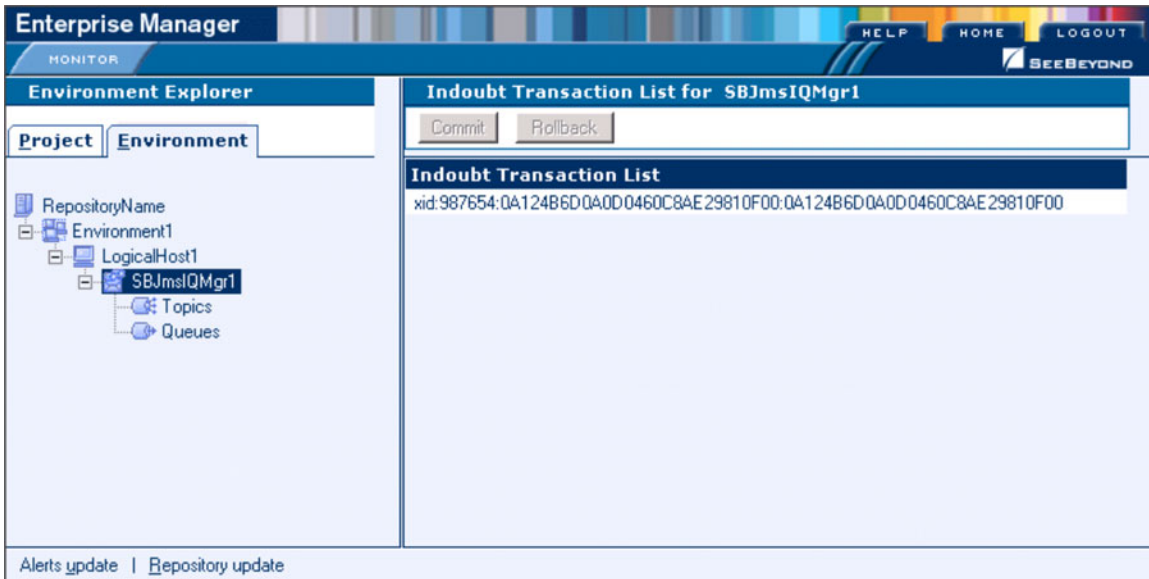
Figure 33 Message Server Details - Controls Tab



- 2 Click the **ShowXid** button to display the *Indoubt Transaction List* shown in Figure 34.

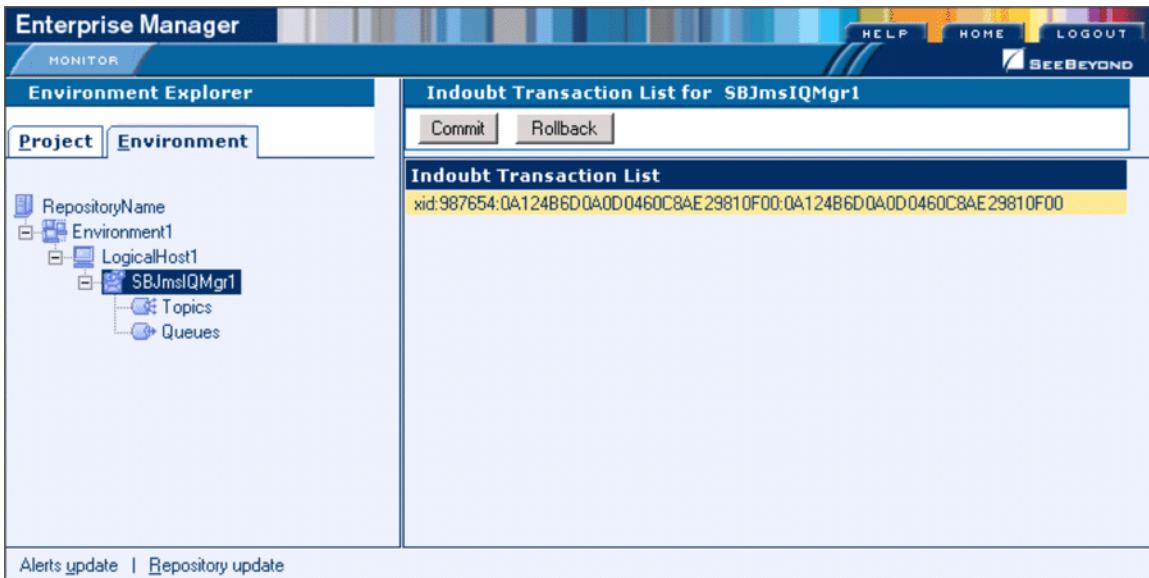
Note: For information regarding XA transactions, see the *eGate Integrator JMS Reference Guide*.

Figure 34 Indoubt Transaction List



- 3 Select the transaction having the specified XID, as shown in Figure 35, and click either **Commit** or **Rollback**.

Figure 35 Indoubt Transaction List - Transaction Selected



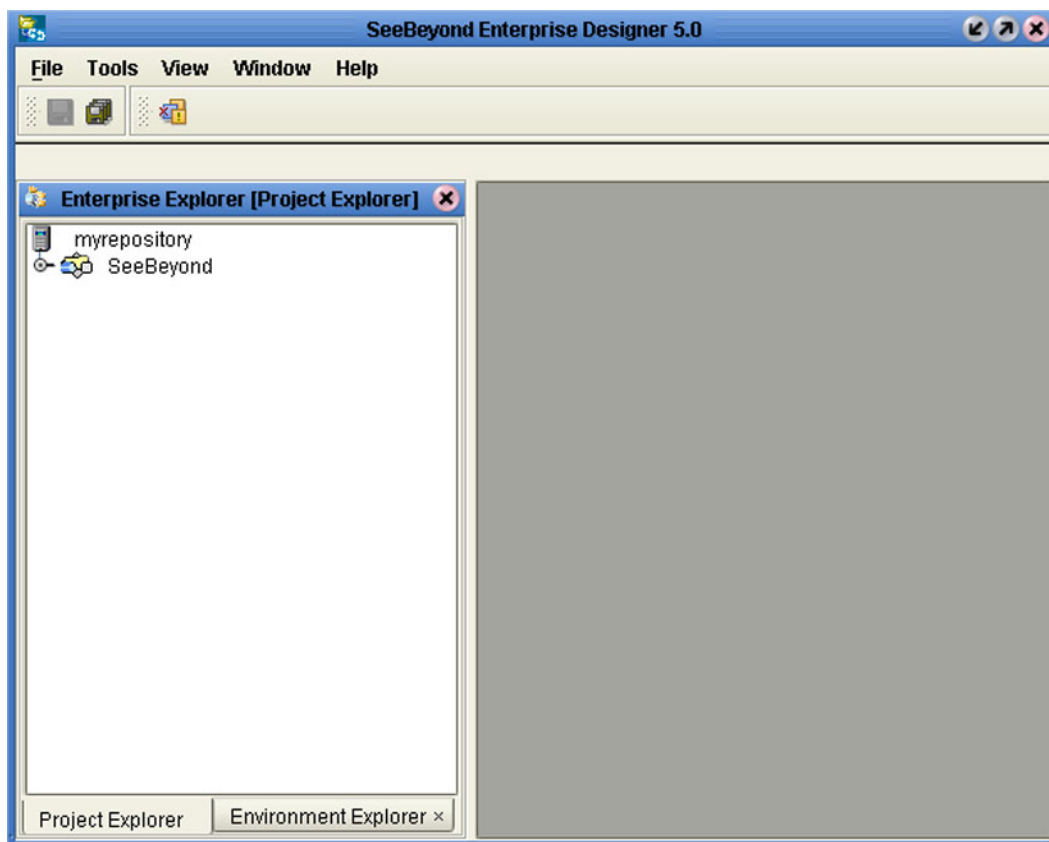
Enterprise Designer

This chapter describes the various features of the Enterprise Designer.

4.1 Overview

The Enterprise Designer graphical user interface (GUI) is used to create and configure the logical components and physical resources of an eGate Project. Through this GUI (see Figure 36), you can develop Projects to process and route data through an eGate Integrator system.

Figure 36 SeeBeyond Enterprise Designer



The major features of the Enterprise Designer are the Enterprise Explorer on the left, and an editor panel on the right—which is initially blank. The Enterprise Explorer follows the familiar Windows Explorer format, displaying a tree structure. The Enterprise Explorer provides two views of the ICAN system, which are described in the following sections of this chapter:

- **Project Explorer** on page 60
- **Environment Explorer** on page 61

The editor panel displays a variety of editors, depending upon what component is selected in the Enterprise Explorer. These editors are described in the following sections of this chapter:

- **Connectivity Map Editor** on page 63
- **OTD Editor** on page 64
- **Java Collaboration Editor** on page 65
- **XSLT Collaboration Editor** on page 66
- **Environment Editor** on page 67
- **Deployment Editor** on page 68

The Enterprise Designer includes several analysis and archiving tools, which are described in the following sections of this chapter:

- **Impact Analyzer** on page 69, which helps you visualize how a change to one part of a Project would affect the rest of the Project.
- **Version Control** on page 71, which allows you to maintain multiple versions of Project components.
- **Repository Backup and Restoration** on page 74, which allows you to back up the Repository to an external file and restore a Repository from an external file.
- **Project Export and Import** on page 76, which allows you to export a Project from Enterprise Designer to an external file, and import a Project into Enterprise Designer from an external file.

The Enterprise Designer also contains the customary graphical interface features, which are described in the following sections of this chapter:

- **Menus** on page 56 describes the options contained in the individual menus.
- **Toolbar** on page 58 describes the functionality of the toolbar icons.
- **Browser Buttons** on page 59 describes the browser buttons that appear throughout the Enterprise Designer, in various wizards and dialog boxes.

The procedure for invoking the Enterprise Designer is described in **Starting Enterprise Designer** on page 55.

4.2 Starting Enterprise Designer

To start the Enterprise Designer

- 1 Start the Enterprise Designer using one of these two methods:
 - If you are connecting to a Repository on an HP NonStop Server, open a command prompt and change directories to `C:\ICAN50\edesigner\bin`, and type the following command:

```
runed hostname port rep_name
```

where:

hostname is the TCP/IP host name of the server where you installed the Repository—not the name of the Repository itself,

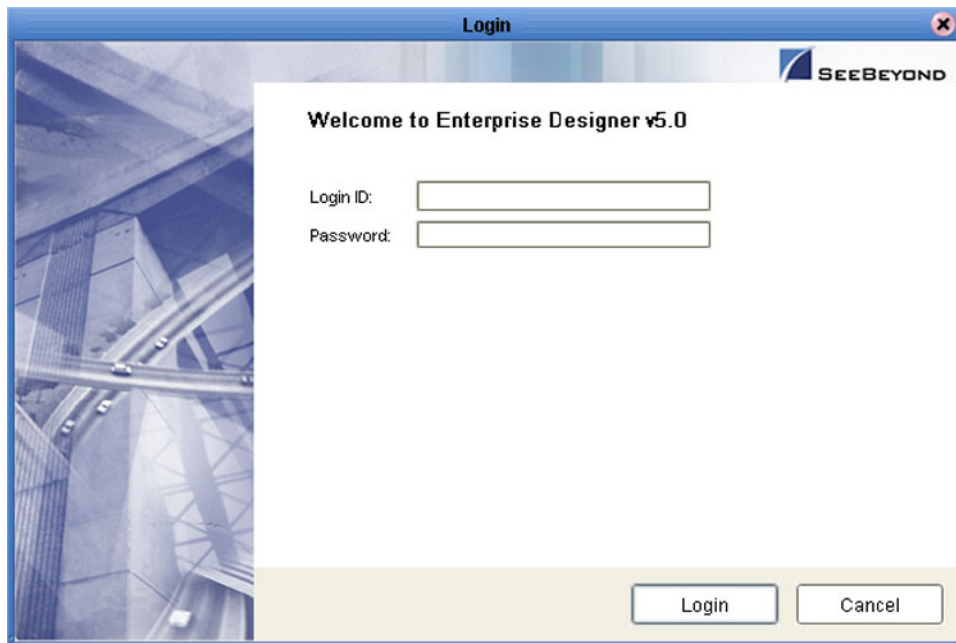
port is the port number of the Repository, and

rep_name is the name of the Repository.

This displays the dialog box shown in Figure 37.

- If you are connecting to a Repository on any other platform, run the batch file `C:\ICAN50\edesigner\bin\runed.bat` to display the *Login* dialog box shown in Figure 37.

Figure 37 Login Dialog Box



- 2 Click in the *Login ID* text box, and enter your login ID.
- 3 Tab to the *Password* text box, and enter your password.
- 4 Click **Login** to complete the login process and display the Enterprise Designer GUI shown in Figure 36.

4.3 Interface Features

4.3.1 Menus

The menu bar provides access to a variety of options for managing your Project. The individual menus are described in the following tables.

File Menu

Table 7 File Menu Options

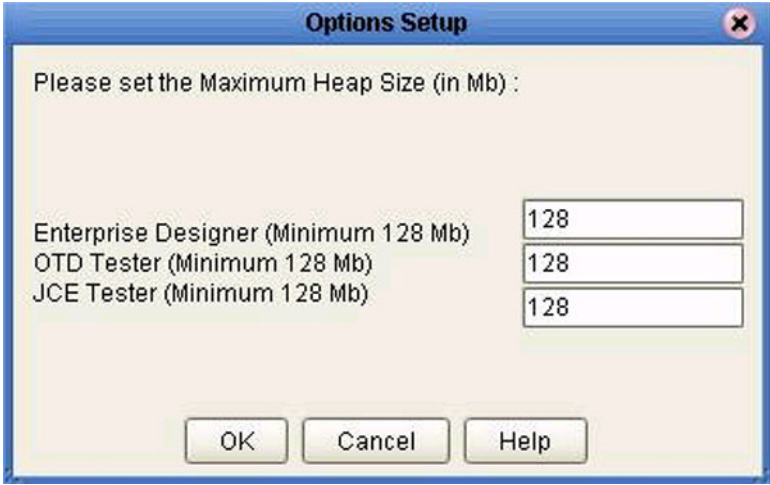
Option	Function
Restore	Displays a dialog box in which you can locate and select a Repository archive file to restore in the Enterprise Designer. See Restoring a Repository on page 74.
Backup	Displays a dialog box in which you can select a location on your computer to save a copy of the Repository archive. See Backing Up a Repository on page 74.
Save	Saves changes to the selected Project.
Save All	Saves changes to all Projects.
Exit	Closes the Enterprise Designer.

Tools Menu

Table 8 Tools Menu Options

Option	Function
Impact Analysis	Displays a dialog box in which you can view how one component of a Project impacts other components. See Impact Analyzer on page 69.
Update Center	Displays a series of dialog boxes in which you can check for program updates. See the <i>eGate Integrator Installation Guide</i> .

Table 8 Tools Menu Options

Option	Function
Options	<p>Displays the Options Setup dialog box, in which you can specify the maximum heap size for selected components:</p> 

View Menu

Table 9 View Menu Options

Option	Function
Environment Explorer	Activates the Environment Explorer tab on the Enterprise Explorer. See Environment Explorer on page 61.
Project Explorer	Activates the Project Explorer tab on the Enterprise Explorer. See Project Explorer on page 60.

Window Menu

Table 10 Window Menu Options

Option	Function
Cascade	Displays all open windows so that each window slightly overlaps the others in the Project Editor.
Tile	Displays all open windows in a stacked tile pattern.
Horizontal Layout	Displays all open windows from top to bottom.
Vertical Layout	Displays all open windows from left to right.
Minimize All	Minimizes all open windows so that only the title bar displays at the bottom of the Project Editor.
Restore All	Returns minimized windows to their original position on the Project Editor.
Close All	Closes all open windows.




Help Menu

Table 11 Help Menu Options

Option	Function
Contents	Displays the online help for eGate Integrator.
Help Set	Displays additional online help options.

4.3.2 Toolbar






Table 12 Enterprise Designer Toolbar Icons

Icon	Function
	Save saves changes to the selected Project (inactive if no changes have been made).
	Save All saves changes to all Projects (inactive if no changes have been made).
	Displays the Impact Analyzer dialog box, which allows you to view how one component of a Project impacts other components.

4.3.3 Browser Buttons

The following buttons are used throughout the Enterprise Designer, in wizards and file selection dialog boxes. They correspond to standard Windows browser buttons.

Table 13 Browser Buttons

Button	Function
	Up One Level returns you to the parent folder or directory.
	Home returns you to the root folder or directory.
	Create New Folder creates a new folder under the current folder.
	List displays folder/file names only.
	Details displays details of the folders or files (name, type, date last modified, etc.).

4.4 Enterprise Explorer

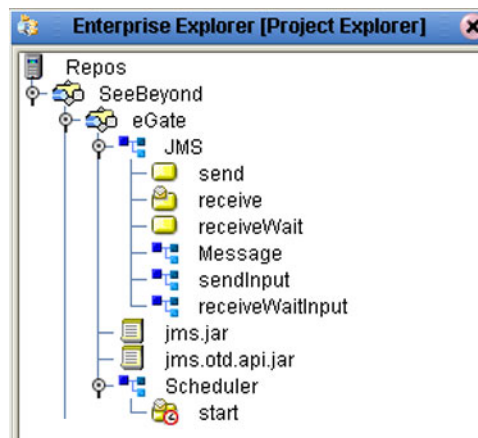
The Enterprise Explorer organizes the components of a Project into tabs that display different views of an eGate system.

- **Project Explorer** on page 60 deals with logical components.
- **Environment Explorer** on page 61 deals with physical resources, including the Logical Host and Integration Server.

4.4.1 Project Explorer

The **Project Explorer** tab includes folders and icons that represent the names and contents of Projects. Some example components of a Project are shown in Figure 38.

Figure 38 Enterprise Explorer: Project Explorer View

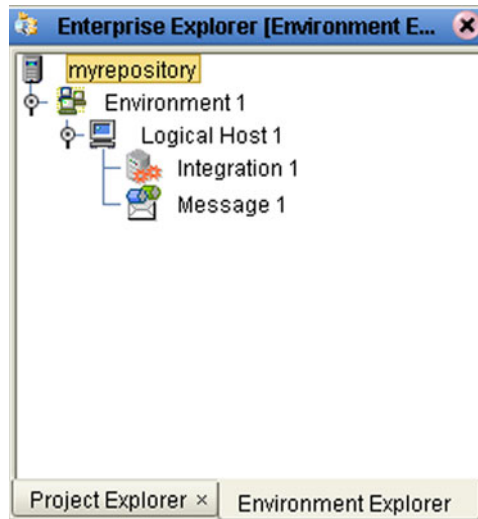


Details of the features and usage of the Project Explorer are found in [eGate Projects](#) on page 81.

4.4.2 Environment Explorer

An Environment consists of Logical Hosts capable of hosting eGate components and information about external systems which may be involved with an eGate configuration.

Figure 39 Enterprise Explorer: Environment Explorer View



Details of the features and usage of the Environment Explorer are found in [Environments](#) on page 257.

4.5 Enterprise Designer Editors

The editor panel—which is initially blank—displays a variety of editors, depending upon what component is selected in the Enterprise Explorer. These editors are briefly described in the following sections of this chapter.

- **Connectivity Map Editor** on page 63
- **OTD Editor** on page 64
- **Java Collaboration Editor** on page 65
- **XSLT Collaboration Editor** on page 66
- **Environment Editor** on page 67
- **Deployment Editor** on page 68

Additional facilities are also displayed here, such as the Java Debugger (see **Debugging Java Collaborations** on page 214).

Note: See the **eGate Tutorial** for detailed information and instructions for setting up a Project.

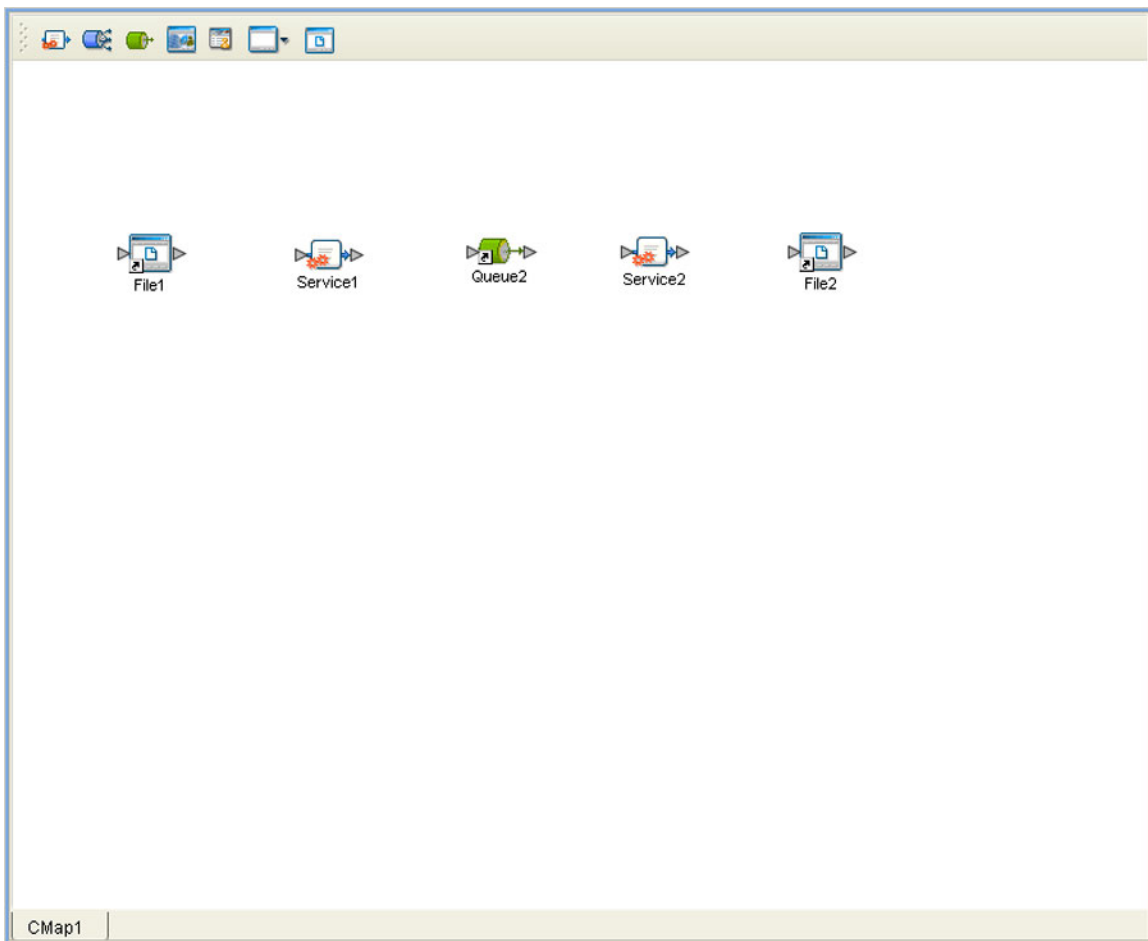
4.5.1 Connectivity Map Editor

A Connectivity Map is a graphical representation of your Project, containing the various logical components comprising the Project and the links between them. The Connectivity Map Editor, shown in Figure 40, allows you to create your Project by simply dragging and dropping icons onto a Project canvas and then connecting them to form data paths. You then can configure the components by means of dialog boxes that are displayed by clicking on the component icons.

Note: *It is best to create your Collaboration Definitions before using the Connectivity Map to connect components.*

See [Using the Connectivity Map Editor](#) on page 88 for detailed information.

Figure 40 Connectivity Map Editor

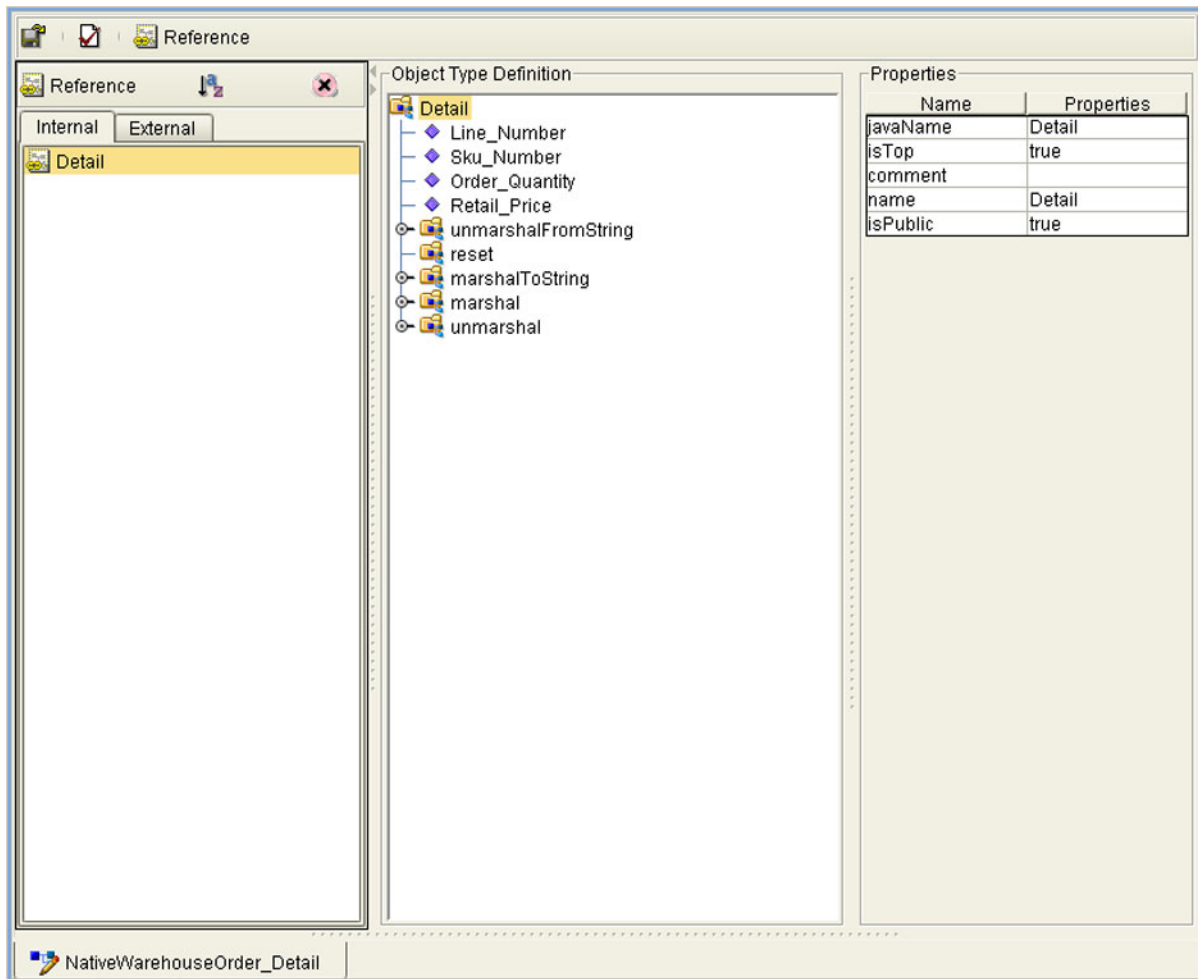


4.5.2 OTD Editor

The OTD Editor window, as shown in Figure 41, displays the source files used to create the Object Type Definitions (OTDs) to use with a Project. You use an OTD wizard tool to compile OTD files and add them to the **Project Explorer** tab.

See [Using the OTD Editor](#) on page 152 for detailed information.

Figure 41 OTD Editor

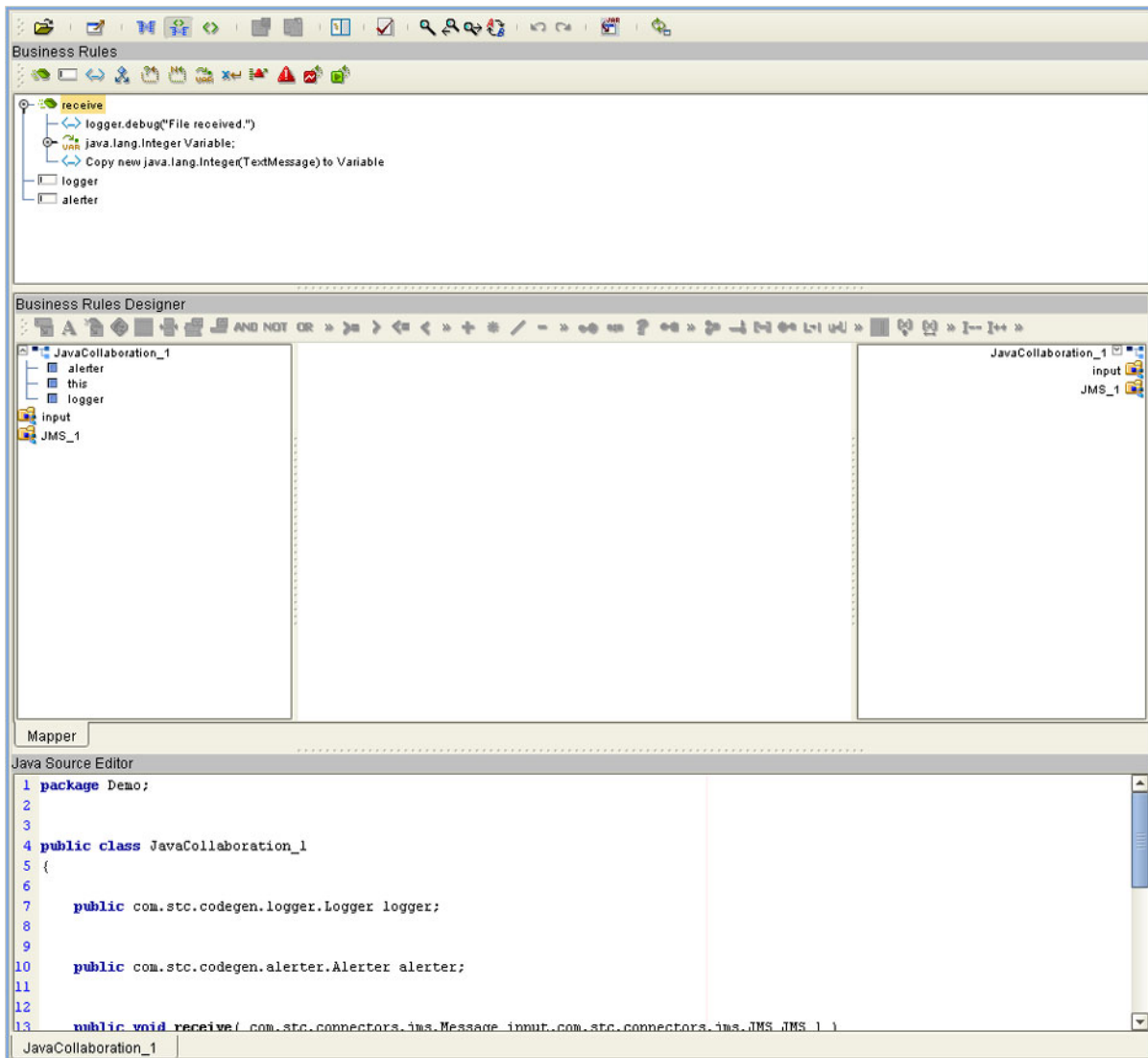


4.5.3 Java Collaboration Editor

The Java Collaboration Editor window, as shown in Figure 42, displays a Java-based Collaboration Definition that you want to include in a Project. You use a Java wizard tool to compile Collaboration Definition files and add them to the **Project Explorer** tab.

See [Using the Java Collaboration Editor](#) on page 168 for detailed information

Figure 42 Java Collaboration Editor

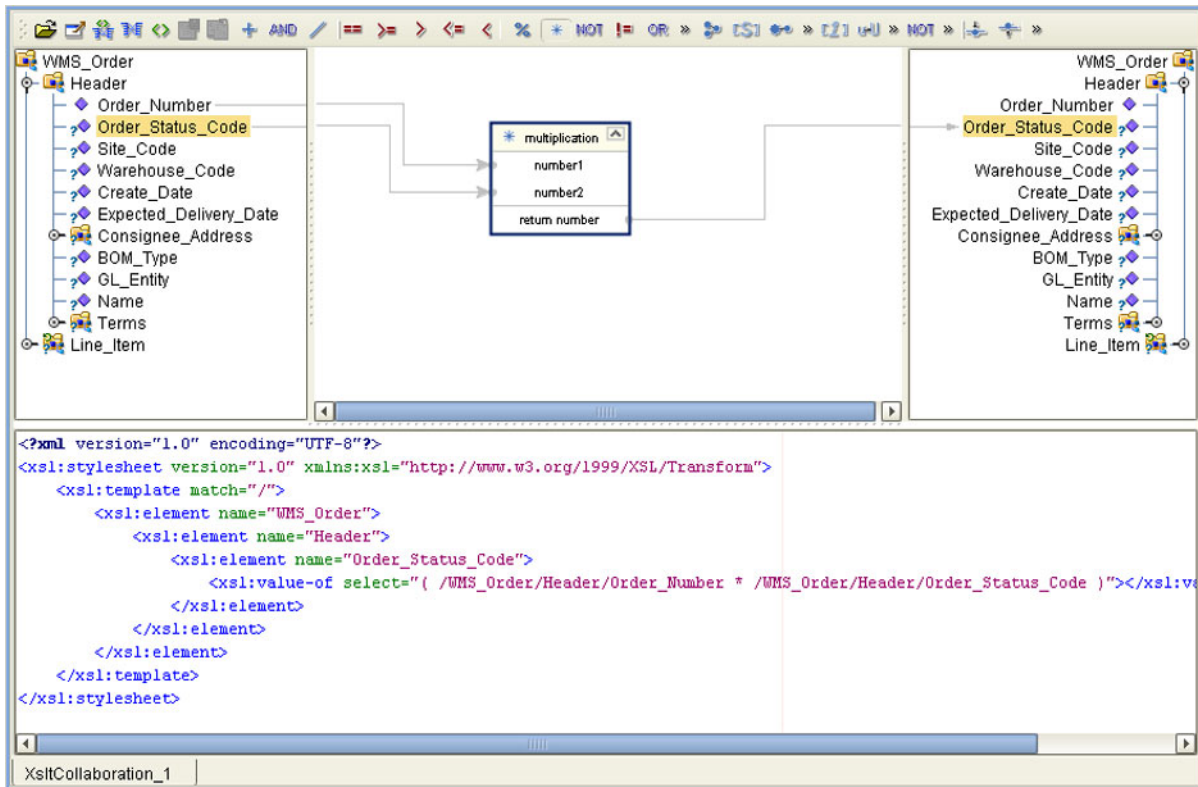


4.5.4 XSLT Collaboration Editor

The XSLT Collaboration Editor window, as shown in Figure 43, displays the XSLT Collaboration Definitions that you need to map together and include in the Project. You use a XSLT wizard tool to compile Collaboration Definition files and add them to the Project Explorer tab.

See [Using the XSLT Collaboration Editor](#) on page 239 for detailed information

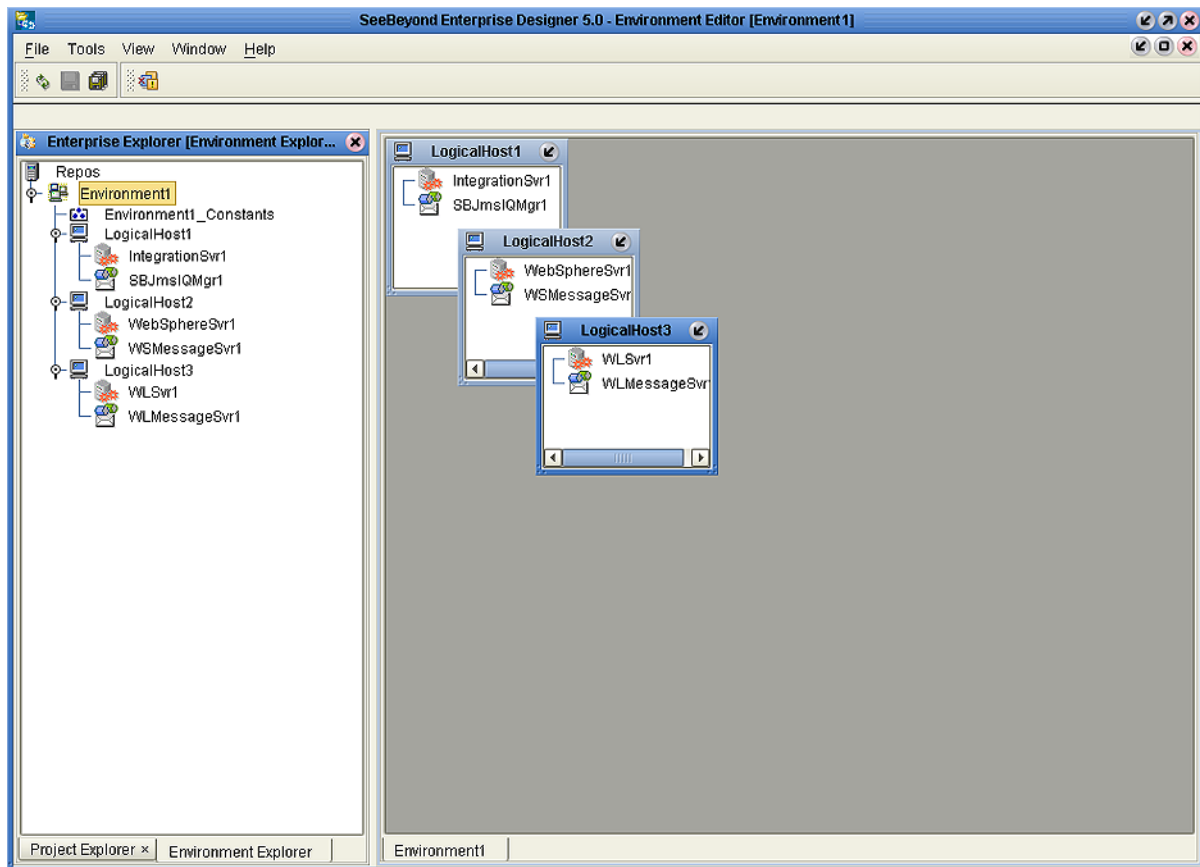
Figure 43 XSLT Collaboration Editor



4.5.5 Environment Editor

The Environment Editor provides a canvas in which you can create and customize an Environment. Here you can see the various components (Logical Hosts, servers, and external systems) included in the selected Environment. An environment containing example Logical Hosts is shown in Figure 44.

Figure 44 Environment Editor

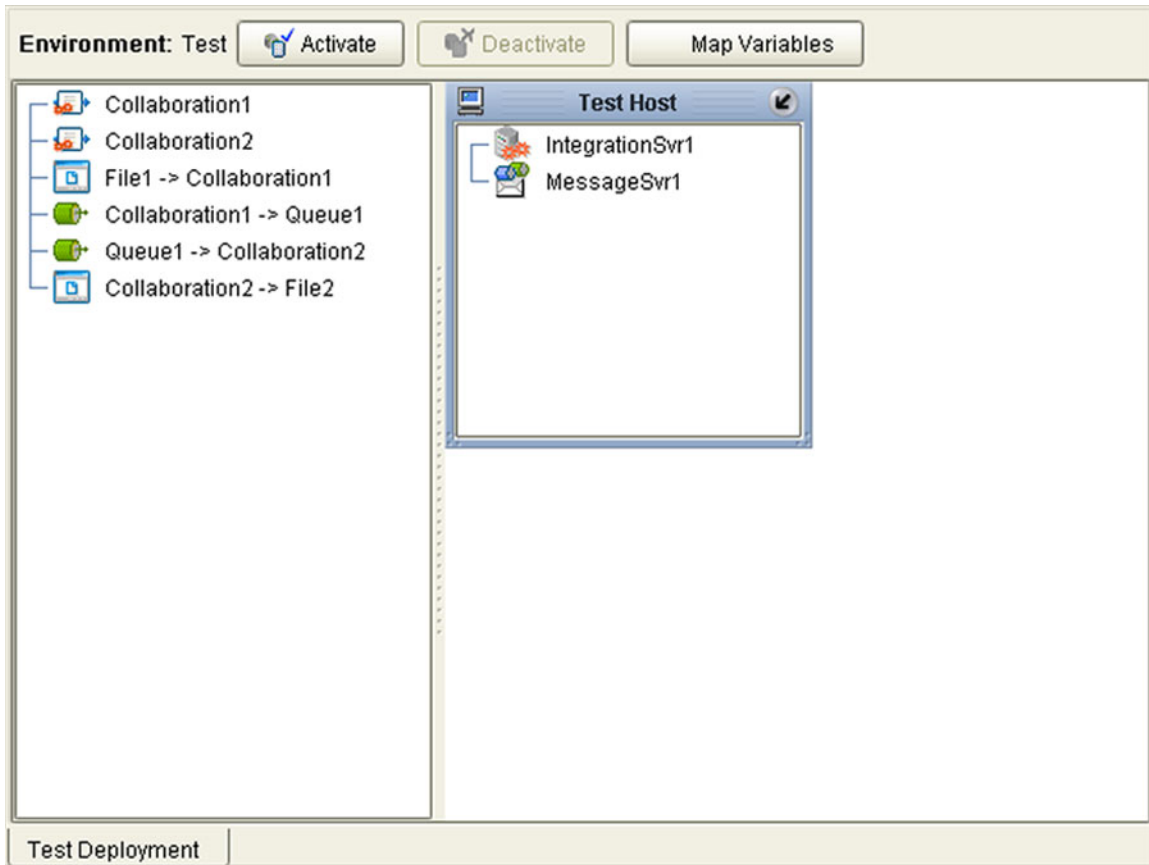


Note: Unlike changes to Project-related configuration properties, changes to Environment-related properties do not require redeployment.

4.5.6 Deployment Editor

The Deployment Editor, as shown in Figure 45, contains information about how Project components will be deployed in an Environment. See [Using the Deployment Editor](#) on page 279 for detailed information

Figure 45 Deployment Editor



4.6 Additional Tools and Features

4.6.1 Impact Analyzer

The Impact Analyzer helps you determine how a change to one component of a Project or Environment will affect other components in that Project or Environment.

To perform an Impact Analysis

- 1 Select a component in either the Project Explorer or Environment Explorer.
- 2 Click the **Impact Analyzer** button, or select **Impact Analyzer** from the Tools menu, to display the *Impact Analyzer* dialog box shown in Figure 46.
- 3 In the *Please show me* drop-down list, select items you would like to view.
- 4 From the list of objects that appears, select one for which you would like to perform an impact analysis.
 - ♦ You can print the object list by clicking **Print** to display the Windows *Print* dialog box.
- 5 Click **Impact** to see how that object would be affected by a change to the component you selected in step 1.

Figure 46 Impact Analyzer Dialog Box

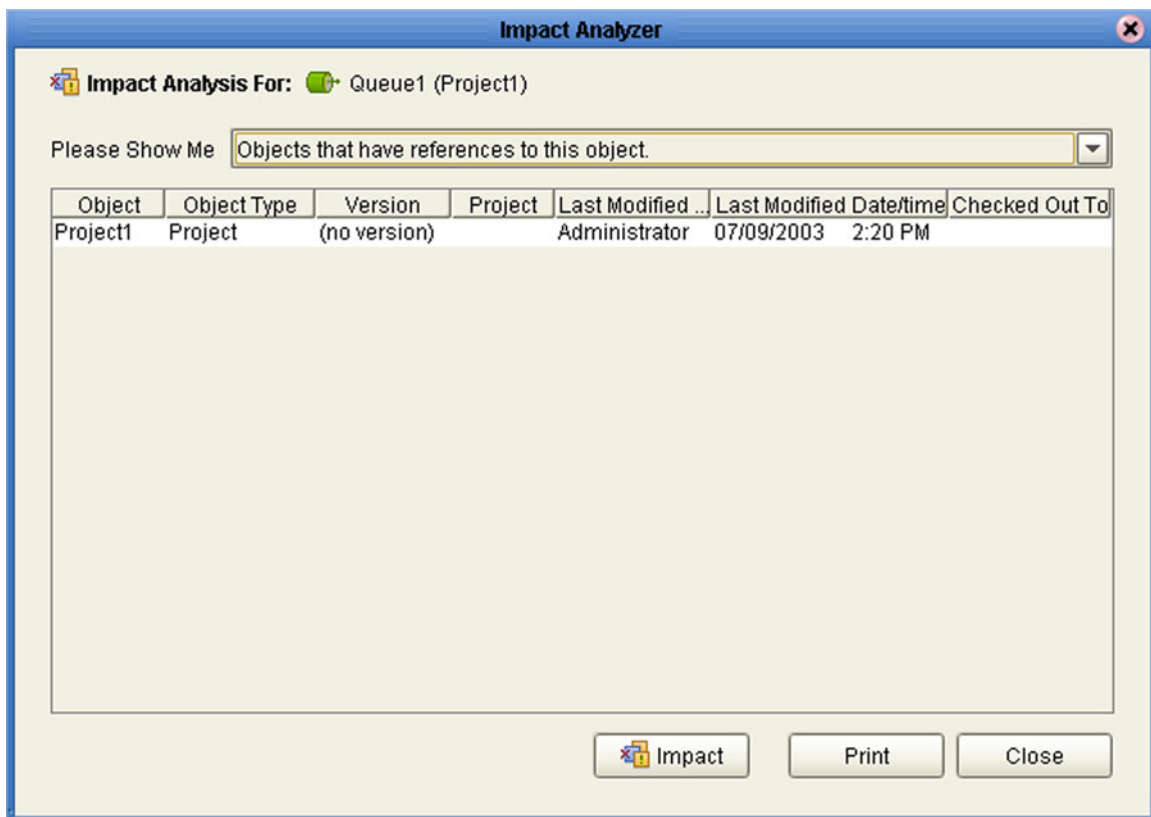



Table 14 Impact Analyzer Command Buttons

Button	Function
 Impact	Performs an impact analysis for the object selected from the object list.
Print	Displays the Windows Print dialog box, which you can use to print the object list.
Close	Closes the Impact Analyzer dialog box.

4.6.2 Version Control

Version control allows you to maintain multiple versions of a Project or Environment component. The version history of each component is recorded to a log file, and can be viewed by means of a menu option.

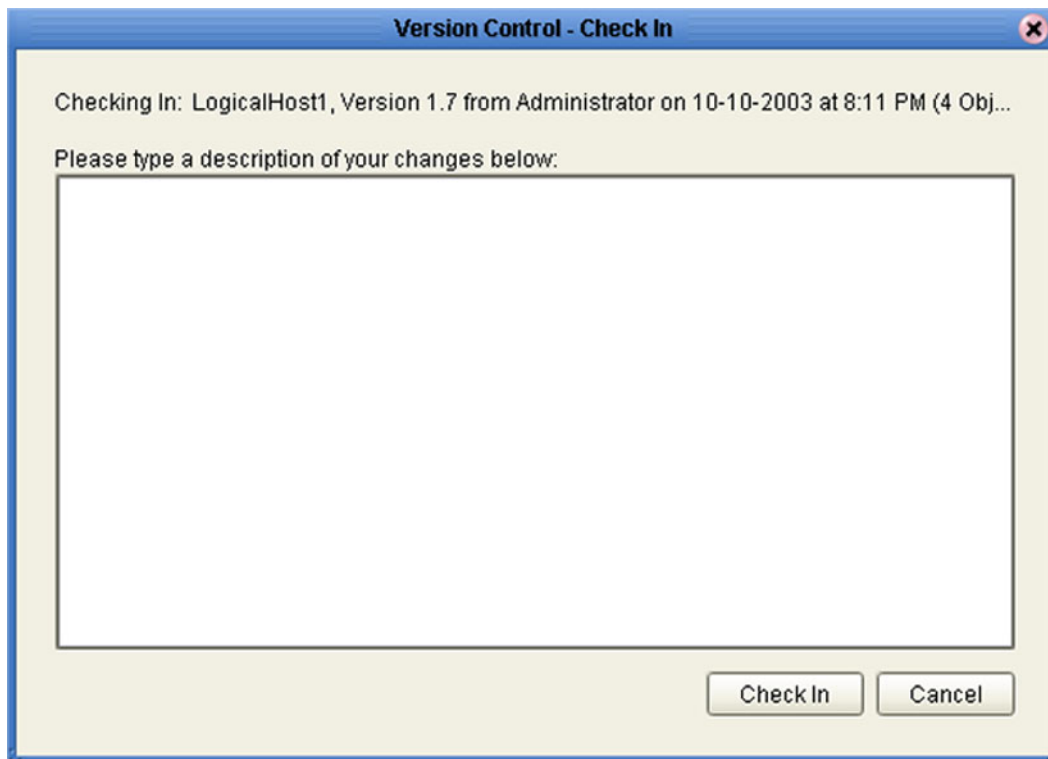
Checking a Component In

Once you have created and configured a Project or Environment component, you can check that object in by using the following procedure.

To check in a version of a Project/Environment component

- 1 Click the Project or Environment Explorer tab in the Enterprise Explorer.
- 2 Right-click on a component to display to display its context menu.
- 3 Select **Check In** to display the *Version Control - Check In* dialog box shown in Figure 47.

Figure 47 Version Control - Check In Dialog Box



- 4 Type in a description of the changes in the new version.
- 5 Click **Check In** to save your changes to a new version.

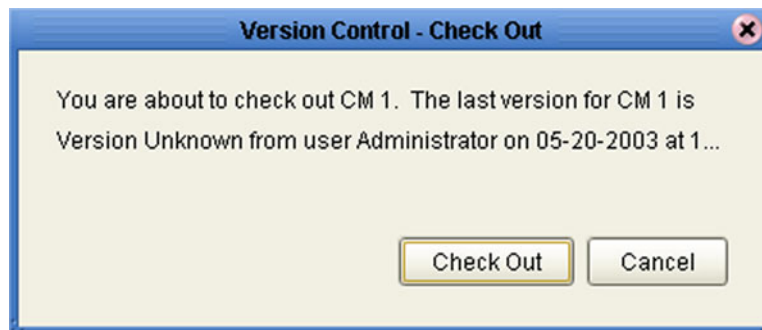
Checking a Component Out

Once an object has been checked in, you can check it out by using the following procedure.

To check out a version of a Project/Environment component

- 1 Click the Project or Environment Explorer tab from the Enterprise Explorer.
- 2 Right-click on a component to display to display its context menu.
- 3 Select **Check Out** to display the *Version Control - Check Out* dialog box shown in Figure 48.

Figure 48 Version Control - Check Out Dialog Box



- 4 Click **Check Out** to open the component.

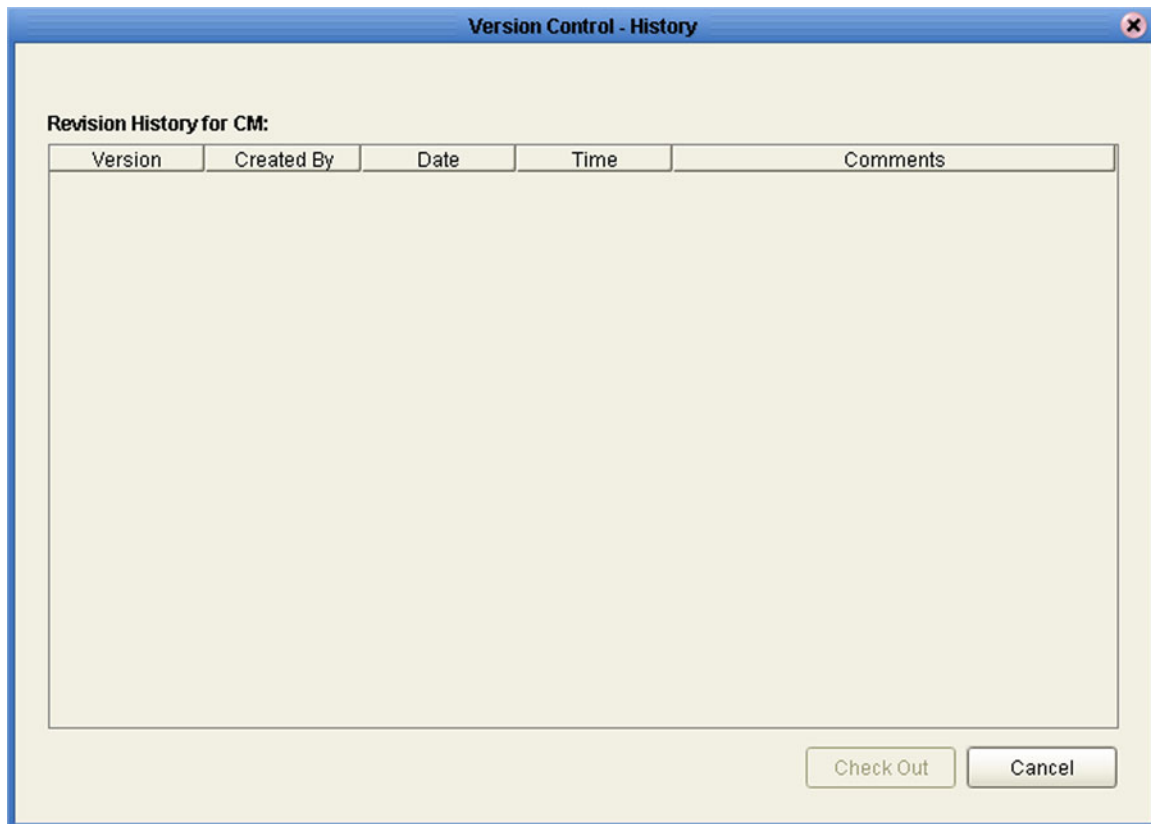
Note: *Only one user can have a file checked out for editing at a time. If another user attempts to check out the same file, they will receive a message indicating that the file is currently checked out.*

Viewing a Component's Version History

To view the version history for a component

- 1 Click the Project or Environment Explorer tab in the Enterprise Explorer.
- 2 Right-click on a component to display to display its context menu.
- 3 Select **Version History** to display the *Version Control - History* dialog box shown in Figure 49.

Figure 49 Version Control - History Dialog Box



- 4 Click **Check Out** to open the component.

4.6.3 Repository Backup and Restoration

Backing Up a Repository

The backup function allows you to back up an entire eGate Repository using a command-line script. The backup script creates a backup of all the repository objects and files in the **repository\data** directory including .jar, .nbn, and other binaries, workspaces, users and locks.

During the process of backup the server is locked, so that users are not able to change objects while a backup is in progress. The backup files are .zip files and can be viewed using a decompression utility such as WinZip.

Location of script file:

```
repository\util\backup.bat (or backup.sh)
```

Command Syntax:

```
backup username password filename
```

Important: *The Repository backup produces a complete snapshot of the Repository, including all installed products—the resulting file, even though compressed, is very large.*

To back up a Repository using the backup script

- 1 Open a command prompt and change directory to *source-repository\util*.
- 2 Type (for example): **backup Administrator stc c:\mybackup.zip**.

Restoring a Repository

The restore function allows you to restore an entire eGate Repository using a command-line script. The restore script restores from a backup file. It will wipe out any existing objects/files in the repository and overwrite them with the values from the backup file.

In effect, it will restore the complete snapshot of the repository contained in the backup file including the workspaces, users and locks (checkouts). You can restore the backup to the same repository or a different repository.

Before restoring, the repository server must be running. During the restore process the repository is locked. You **must** restart the repository server after restoring.

Location of script file:

```
repository\util\restore.bat (or backup.sh)
```

Command Syntax:

```
restore username password filename
```

When restoring a Repository, note that:

- Restoring overwrites the contents of the target Repository.
- The restored Repository will have the same name as the Repository it replaced.
- After restoring a Repository, you must:
 - A Restart the Repository.
 - B Reactivate all deployments.

To restore a Repository using the restore script

- 1 Open a command prompt and change directory to *target-repository\util*.
- 2 Type (for example): **restore Administrator stc c:\mybackup.zip**.
- 3 Restart the Repository.
- 4 Stop Enterprise Designer.
- 5 Restart Enterprise Designer.

4.6.4 Project Export and Import

Exporting a Project

The export function allows you to export an eGate Project to an external file using either the Enterprise Designer or a command-line script.

When exporting a Project, note that:

- The exported Project may have references to elements that are in other Projects. A list of such references is generated during the export process.
- Project deployment objects are not exported, because they have references to both Project and Environment elements that are not required at the Project level.

Note: Logs of these processes can be found in:

```
ICAN-root\repository\logs\repository.log
```

Exporting a Project Using Enterprise Designer

To export a Project using Enterprise Designer

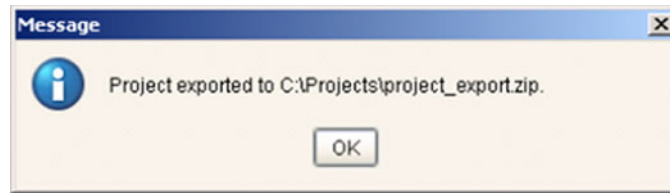
- 1 From the Project context menu, select **Export Project ...** to display the *Enter file name to export to* dialog box shown in Figure 50.

Figure 50 Enter File Name Dialog Box



- 2 Select a location to save the Project file to using the **Look In** drop-down list.
- 3 Enter a name for the Project file in the **File Name** text box.
- 4 Click **Save** to save the Project file. The Message dialog box shown in Figure 51 is displayed when the Project file has been created. (This process takes a few minutes.)

Figure 51 Message Dialog Box



- 5 Click **OK** to close the Message dialog box.

Exporting a Project Using the Command Line

You can also export a Project using the following command-line script.

Location of script file:

```
repository\util\exportProject.bat (or exportProject.sh)
```

Command Syntax:

```
exportProject repositoryURL username password filename projectname
```

To export a Project using the export script

- 1 Open a command prompt and change directory to *export-repository\util*.
- 2 Type (for example): **exportProject http://localhost:12345/myrepository Administrator stc c:\myprojectExport.zip myProject1.**

Importing a Project

The import function allows you to import an eGate Project file using the Enterprise Designer.

Important: APIs installed in the source Repository must be installed in the Repository into which the Project is imported.

When importing a Project, note that:

- Existing Projects are not affected by the imported Project.
- During import, if another Project having the same name exists in the target Repository, you will receive an error message and the existing file will not be overwritten.
- You can specify a new Project name and location (in Project Explorer) during import.
- References are validated during import.

Note: Logs of these processes can be found in:

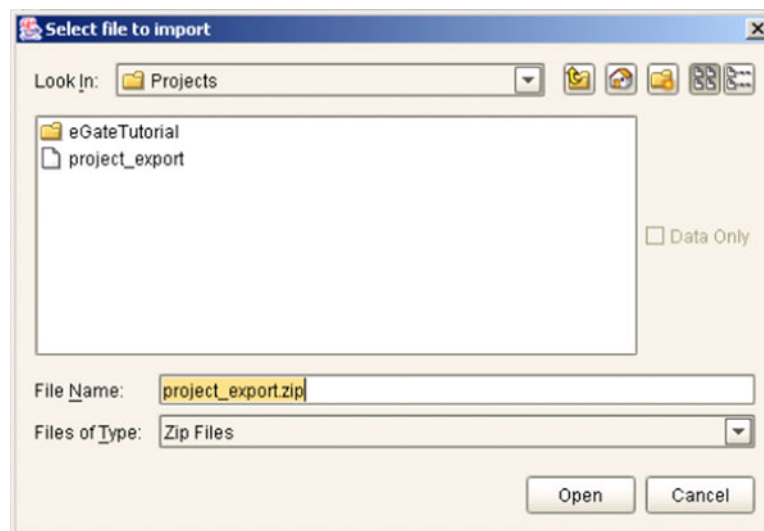
`ICAN-root\repository\logs\repository.log`

Importing a Project Using Enterprise Designer

To import a Project using Enterprise Designer

- 1 Create a new Project folder in Project Explorer.
- 2 From the Project context menu, select **Import Project ...** to display the *Select file to import* dialog box shown in Figure 52.

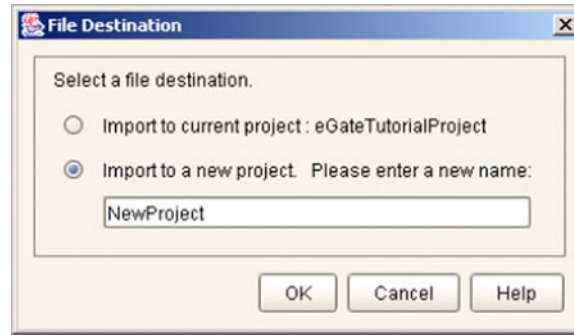
Figure 52 Select File Dialog Box



- 3 Locate and select the Project file that you want to import.

- 4 Click **Open** to import the file.
- 5 A File Destination dialog box will appear as shown in Figure 53, in which you choose whether to import to the same Project name or to a different name.

Figure 53 File Destination Dialog Box



- 6 If you choose to import to a new Project, enter the name for the Project.
- 7 Click **OK** to import the file.
- 8 The message box shown in Figure 54 appears after the file has been imported.

Figure 54 Message Dialog Box



- 9 Click **OK** to close the Message dialog box.

Importing a Project Using the Command Line

You can also import a Project using the following command-line script.

Location of script file:

```
repository\util\importProject.bat (or importProject.sh)
```

Command Syntax:

```
"importProject repositoryURL username password filename  
projectname ParentProjectName"
```

Note: If root is the parent, replace *ParentProjectName* with "".

To import a Project using the import script

- 1 Open a command prompt and change directory to *import-repository\util*.
- 2 Type (for example): **"importProject http://localhost:12345/myrepository Administrator stc c:/myprojectExport.zip myNewProject1 myExistingProject1"**.

This will take the Project that exists in the export file, rename it to myNewProject1 and attach it as a subproject of myExistingProject1. If the Project is to be attached to the root, then leave the last parameter as an empty string. Make sure you include the quotes (""") around the entire command.

eGate Projects

This chapter describes components of an eGate Project, and the use of the Enterprise Designer in defining your Project.

5.1 Overview

An eGate Project represents the logical system designed to solve either all or part of a business problem. Projects are created using tools contained within the Enterprise Designer, and are deployed to specific Logical Hosts in specific Environments by means of Deployment Profiles (see [Environments](#) on page 257). An end-to-end example of Project implementation is given in the *eGate Tutorial*.

If you are also using eInsight Business Process Manager, an eGate Project is related to an Activity in an eInsight business process. Components developed for use in one Project can be used in another, and a Project can internally reference another Project.

5.1.1 Project Components

The components found in a typical Project are described in the following sections of this chapter:

- [Services](#) on page 90
- [External Applications](#) on page 92
- [Schedulers](#) on page 93
- [Component Connections](#) on page 94
- [Message Destinations](#) on page 91

Behind the scenes, and not explicitly shown in a Connectivity Map, are other Project components, such as:

- **Collaboration Definitions**

A Collaboration Definition defines the logical operation taking place in the related Collaboration. It is created in either the Java Configuration Editor or the XSLT Configuration Editor, and is based on an Object Type Definition.

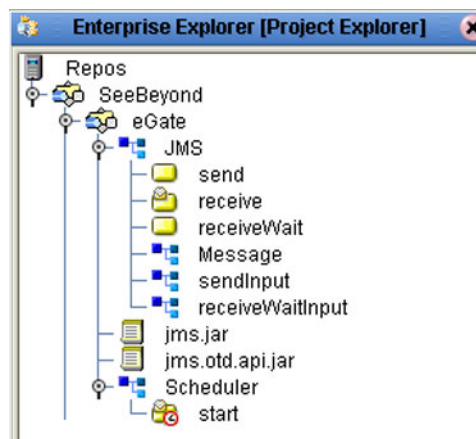
- **Object Type Definitions**

Object Type Definitions (OTDs) are sets of rules that define the encoding of an object. They describe messages that are propagated through eGate, and the methods available for operating on them, and also interactions with external APIs.

5.2 The Project Explorer

A Project consists of logical constructs and configurations designed to solve some or all of a business problem. The **Project Explorer** displays the contents of the Repository that belong to the selected Project (see Figure 55).

Figure 55 Project Explorer












The Project Explorer is used in conjunction with the Connectivity Map Editor (see [Using the Connectivity Map Editor](#) on page 88) to create and configure a Project.

Note: Select *Refresh All from Repository* before you *Open* any component (such as a Collaboration) to ensure that you open the latest version of that component.

5.2.1 Project Explorer Icons

The icons described in Table 15 appear in the Project Explorer.

Table 15 Project Icons

Icon	Description
	Represents the Repository , which is the database where all Projects and contents are saved.
	Represents the Project or subproject.
	Represents a Connectivity Map , which contains the business logic and information about the data transmission. .
	Represents a Project variable or constant .
 	Represents an Object Type Definition (OTD) file. A lock displayed in the lower-left corner indicates that the OTD is currently checked into the version control system.
	Represents a Java Collaboration Definition file. A lock is displayed in the lower-left corner for Collaboration Definitions that are currently checked into the version control system.
	Represents an XSLT Collaboration Definition file. A lock is displayed in the lower-left corner for Collaboration Definitions that are currently checked into the version control system.
	Represents a Deployment Profile , which specifies how Project components will be deployed in an Environment.

5.2.2 Context Menus

Right-clicking on a component in the Project Explorer displays a context menu for that component. Included here are descriptions of options for the following component context menus:

- [Repository Menu](#) on page 84
- [Project Menu](#) on page 85
- [Connectivity Map Menu](#) on page 86

Repository Menu

Figure 56 Repository Menu

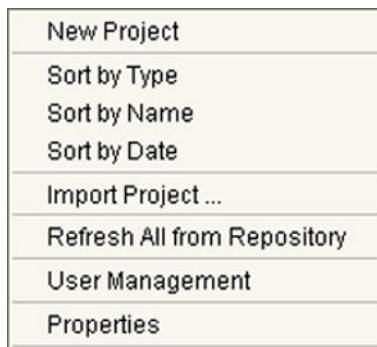


Table 16 Repository Menu Options

Option	Function
Project	Adds a new Project to the Project Explorer tab.
Sort by Type	Places Project components in order by grouping component types.
Sort by Name	Places Projects and Project components in alphabetical order.
Sort by Date	Places Projects in order by creation date from oldest Project to newest.
Import Project	Displays a dialog box, which you can use to locate and select a Project archive file to import into the Enterprise Designer.
Refresh All from Repository	Updates Enterprise Designer with Project/Environment configurations stored in the Repository. (Open editors are not refreshed.)
User Management	Displays the User Management dialog box, where you can manage user access to the Repository with options for adding, modifying, and deleting users. See Configuration User Management on page 294.
Properties	Displays a dialog box (the exact name of the dialog box mirrors the name of your Repository). Use this dialog box to manage Enterprise Designer users and their access to the Repository. See Configuration User Management on page 294.

Note: Select **Refresh All from Repository** before you **Open** any other component (such as a Collaboration) to ensure that you open the latest version of the component.

Project Menu

Figure 57 Project Menu

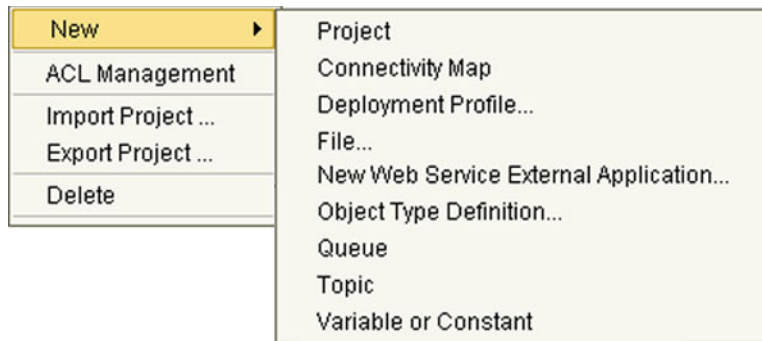


Table 17 Project Menu Options

Option	Option	Function
New	Project	Adds a Subproject folder to the selected Project.
	Connectivity Map	Adds a Connectivity Map to the Project. See Using the Connectivity Map Editor on page 88.
	Deployment Profile	Displays a dialog box with which you can assign a Deployment Profile to the selected Project. See Using the Deployment Editor on page 279.
	File	Displays a dialog box with which you can create an external file to use with the Project.
	Java Collaboration Definition	Displays the Java Collaboration Definition Wizard , with which you can create a Java Collaboration Definition. See Using the Java Collaboration Wizard on page 160.
	New Web Service ...	Adds a third-party Web service application to the Project Explorer. See SeeBeyond Web Services on page 102.
	Object Type Definition	Displays the OTD Wizard , with which you can create an Object Type Definition (OTD) file. See Using the OTD Wizard on page 122 for more information.
	Queue	Adds a queue to your Project.
	Topic	Adds a topic to your Project.
	Variable or Constant	Displays a dialog box with which you can add a constant or variable icon to your Project.
	XSLT Collaboration Definition	Displays the XSLT Collaboration Definition Wizard , with which you can create a Java Collaboration Definition. See Using the XSLT Collaboration Wizard on page 234.

Table 17 Project Menu Options

Option	Option	Function
ACL Management		Displays the ACL Properties dialog box, with which you can assign read and/or write privileges to users for the selected Project. See ACL Management on page 321.
Import Project		Displays a dialog box with which you can locate and select a Project file to import into the Enterprise Designer.
Export Project		Displays a dialog box with which you can select a location on your computer to save a copy of the Project.
Delete		Deletes the selected Project.

Connectivity Map Menu

Figure 58 Connectivity Map Menu

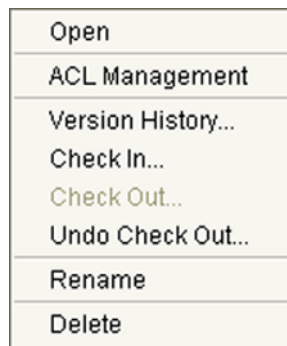


Table 18 Connectivity Map Menu Options

Command	Function
Open	Opens the Connectivity Map Editor for the selected Connectivity Map. See Using the Connectivity Map Editor on page 88.
ACL Management	Displays the ACL Properties dialog box, with which you can assign read and/or write privileges to users for the selected Project. See ACL Management on page 321.
Version History	Displays a dialog box with which you can track the version history for OTDs and Collaboration Definitions. Version control allows users to maintain multiple versions of the same OTD and Collaboration Definition files. See Viewing a Component's Version History on page 73 for more information.
Check In	Displays a dialog box, with which you can check in the current version of a Project. Refer to Checking a Component In on page 71 for more details.
Check Out	Displays a dialog box with which you can check out a version of a Project. See Checking a Component Out on page 72 for more information.
Undo Check Out	Reverses the Check Out command, returns you to the previous state.

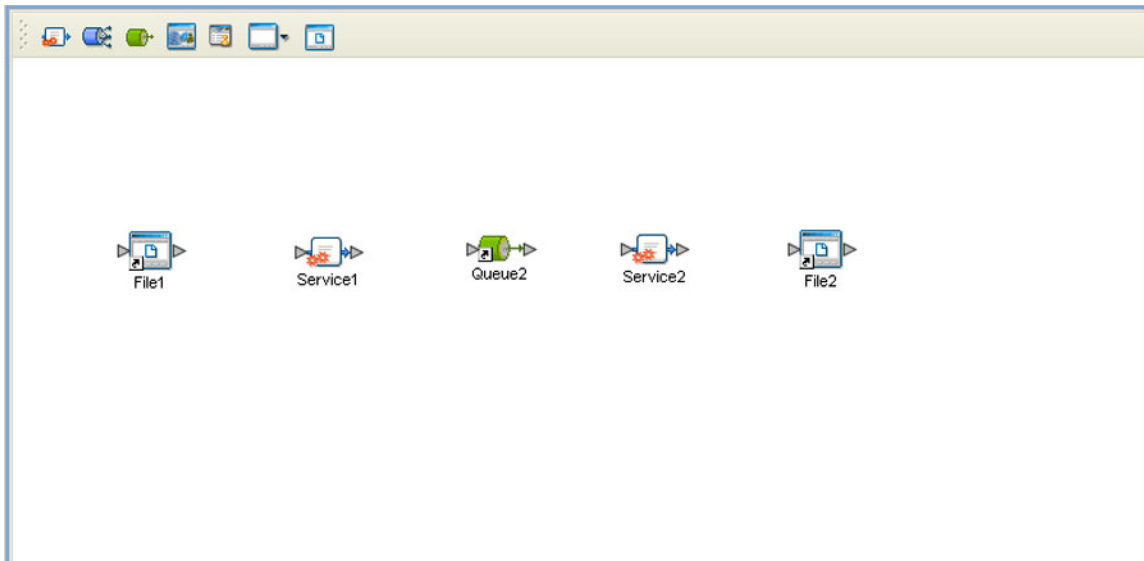
Table 18 Connectivity Map Menu Options

Command	Function
Rename	Allows you to rename the selected Connectivity Map.
Delete	Displays a dialog box in which you confirm that you want to delete the selected Connectivity Map. Clicking Yes then deletes the Connectivity Map.

5.3 Using the Connectivity Map Editor

When you create a new Connectivity Map in the Enterprise Explorer, the editor panel displays the Connectivity Map Editor (see Figure 59). To define your Project, you simply drag icons from the toolbar to the workspace, or canvas, to populate the Connectivity Map with the necessary components. You subsequently link the components by dragging the cursor from one to the other.

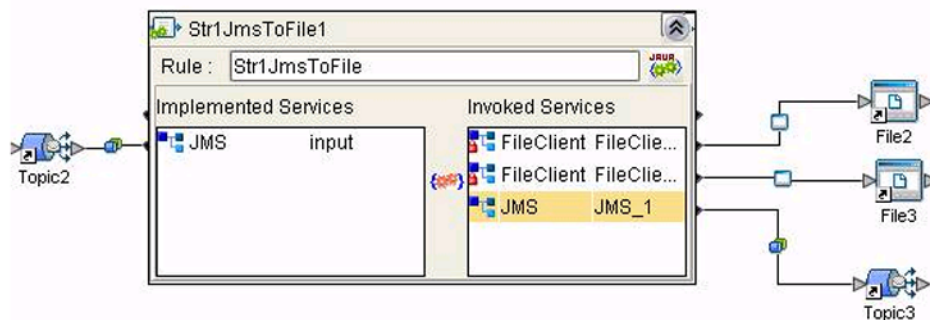
Figure 59 Connectivity Map Window



The drag-and-drop components include Services, queues, topics, schedulers, and external applications. Additional components, such as eWays and JMS Clients, are placed automatically when you link the components you have placed manually.







When there are multiple destinations, as with a JMS topic, the Connectivity Map Editor cannot resolve which output port to connect to which destination. Because of this, the Collaboration definition must be created first, and the connections must be drawn by opening the Collaboration Binding box in Connectivity Map (see Figure 60).

Figure 60 Linking JMS Topics



The Connectivity Map Editor toolbar contains the icons listed in Table 19, plus additional icons representing eGate add-ons and other ICAN components that you may have installed.

Table 19 Connectivity Map Toolbar Icons

Icon	Component	Function
	Service	A logical component that provides the framework for a process or Collaboration. See Service Component on page 90.
	Queue	A Message Destination that conforms to the point-to-point messaging paradigm, having one sender and one receiver. See the <i>eGate Integrator JMS Reference Guide</i> for information.
	Topic	A Message Destination that conforms to the publish/subscribe messaging paradigm, having one sender (publisher) and multiple receivers (subscribers). See the <i>eGate Integrator JMS Reference Guide</i> for information.
	Web Service External Application	Represents a third-party Web service application external to eGate. See SeeBeyond Web Services on page 102.
	External Applications	Represents an application external to eGate. Click the arrow beside the icon to view a list of specific applications to which you can connect. See External Application Drop-Down Menu on page 92.
	Scheduler	Represents a scheduling component of the Connectivity Map. Use this component to set data transfer to occur at set intervals. See Scheduler Component on page 93.

It is important to understand that the logical components appearing in the Connectivity Map are essentially *placeholders* that refer to the “actual” components that exist in the Repository and appear in the Project Explorer. Renaming or deleting a queue or topic in the Connectivity Map only affects the placeholder, not the object in the Repository.

Also, renaming or deleting a queue or topic in the Repository will not affect the existence or name of the associated placeholder in the Connectivity Map. The change will, however, be reflected in the *tooltips* for the placeholder. This allows you to re-assign the placeholder without disrupting the continuity of the Connectivity Map.

5.4 Services

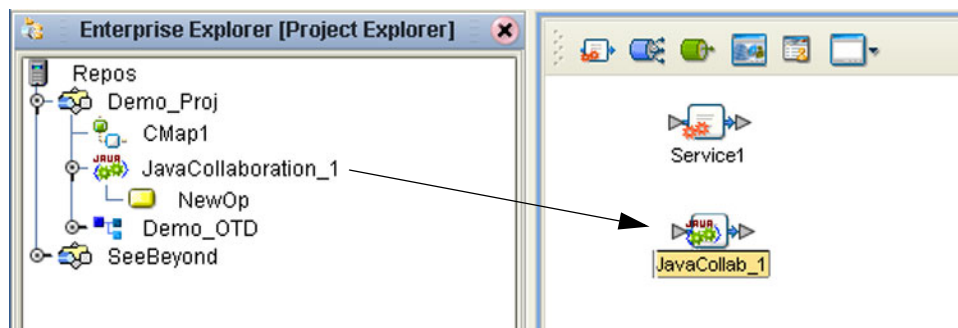
A service provides a framework for a process or a Collaboration, which contains the information required to execute a set of business rules.

5.4.1 Collaborations

A Collaboration is a logical operation performed between some combination of message destinations and external applications. The operation is defined by a Collaboration Definition, which can be encoded in either Java or XSLT.

The Collaboration acts as a service having a publication or subscription relationship with each linked entity. The link is provided by a JMS Client connection (see [Component Connections](#) on page 94. Dragging a Collaboration from the Project Explorer to the Service icon in the Connectivity Map defines the service as a Collaboration (see Figure 61).

Figure 61 Service Component



Note: Any changes made to the names of the Collaborations should be done when the Collaboration is created. If the name is changed later, the Collaboration should be opened, regenerated if applicable, and saved again. For safe measure, this should also be done before creating the Connectivity Map and Deployment Profile.

Connection-related properties for the Collaboration (or other service) are configured in the adjoining JMS Client. These properties include:

- Concurrent or serial processing
- Transaction mode (transacted or XA)
- Security

All properties, and the procedures for configuring them, are detailed in the *eGate Integrator JMS Reference Guide*.

5.5 Message Destinations

A Message Destination is a container for stored data, and can follow either the topic or queue JMS model.

5.5.1 Topics

A *topic* is a message destination that conforms to the publish-and-subscribe messaging paradigm.

5.5.2 Queues

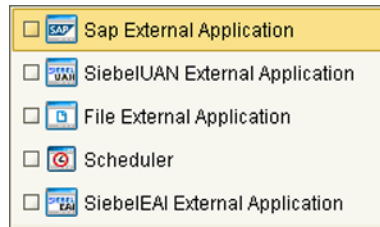
A *queue* is a message destination that conforms to the point-to-point messaging paradigm.

5.6 External Applications

The basic purpose of eGate Integrator is to facilitate the interchange of data between external business applications. These business applications are collectively referred to as external applications, and are represented in the Project by logical proxies for the specific applications involved. An external application can be identified with an ERP application such as SAP or PeopleSoft, a DBMS such as Oracle or SQL, or with a particular communications protocol, such as TCP/IP or HTTPS.

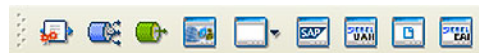
External applications are logical representations of external software applications that are being integrated by the eGate system. These are linked to a Service by means of an eWay. Clicking the drop-down arrow beside the external application icon displays a menu showing those applications corresponding to eWays that have been purchased and installed, plus the Scheduler. An example is shown in Figure 62.

Figure 62 External Application Drop-Down Menu



Checking the individual external applications adds the selected icon to the toolbar (see Figure 63).

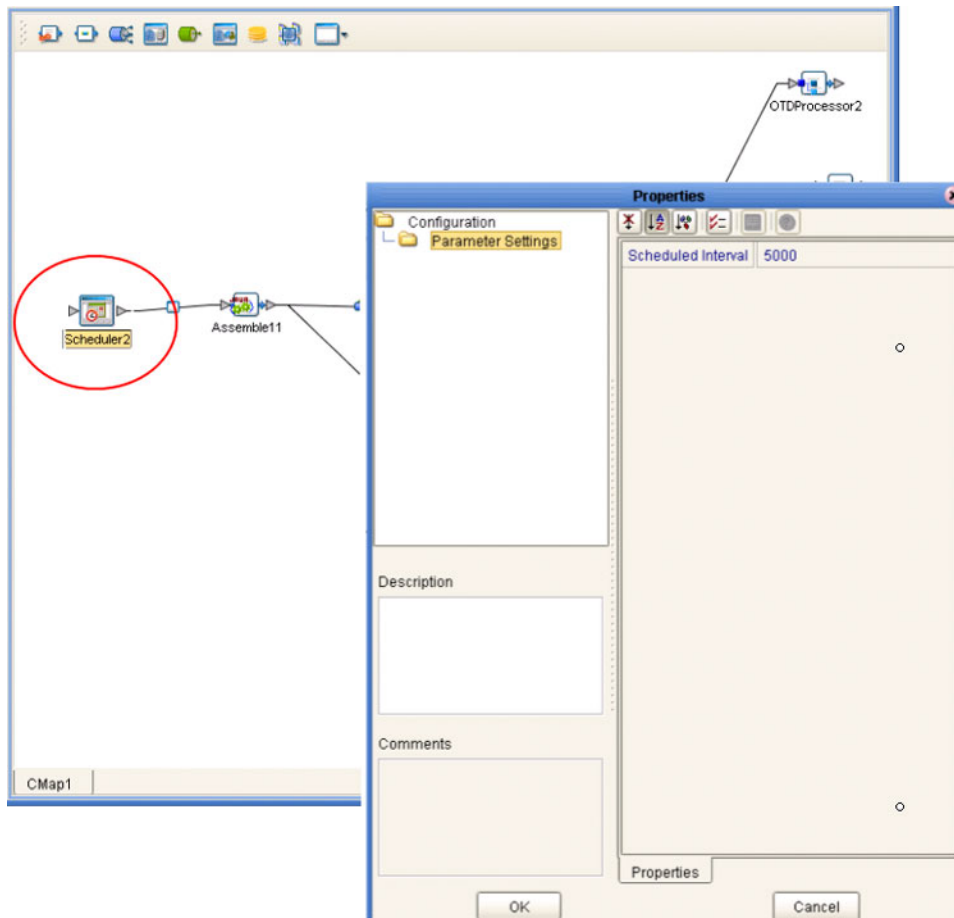
Figure 63 Selected External Applications in Toolbar



5.7 Schedulers

A Scheduler allows a service to be performed at a prescribed interval. The interval can be static, or can be made dynamic by using a Project variable for the interval value.

Figure 64 Scheduler Component

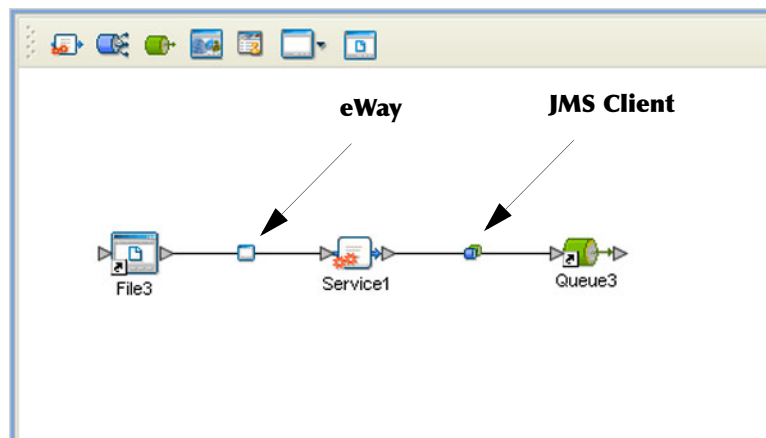


As an example, depicted in Figure 64, a scheduler can be used to invoke a Collaboration with an interval of 5 seconds (5000 milliseconds).

5.8 Component Connections

When you link two components on a Connectivity Map, the Enterprise Designer places either an eWay or JMS Client connection icon on the link, depending upon the type of components you are linking (see Figure 65).

Figure 65 Connection Icons in a Connectivity Map



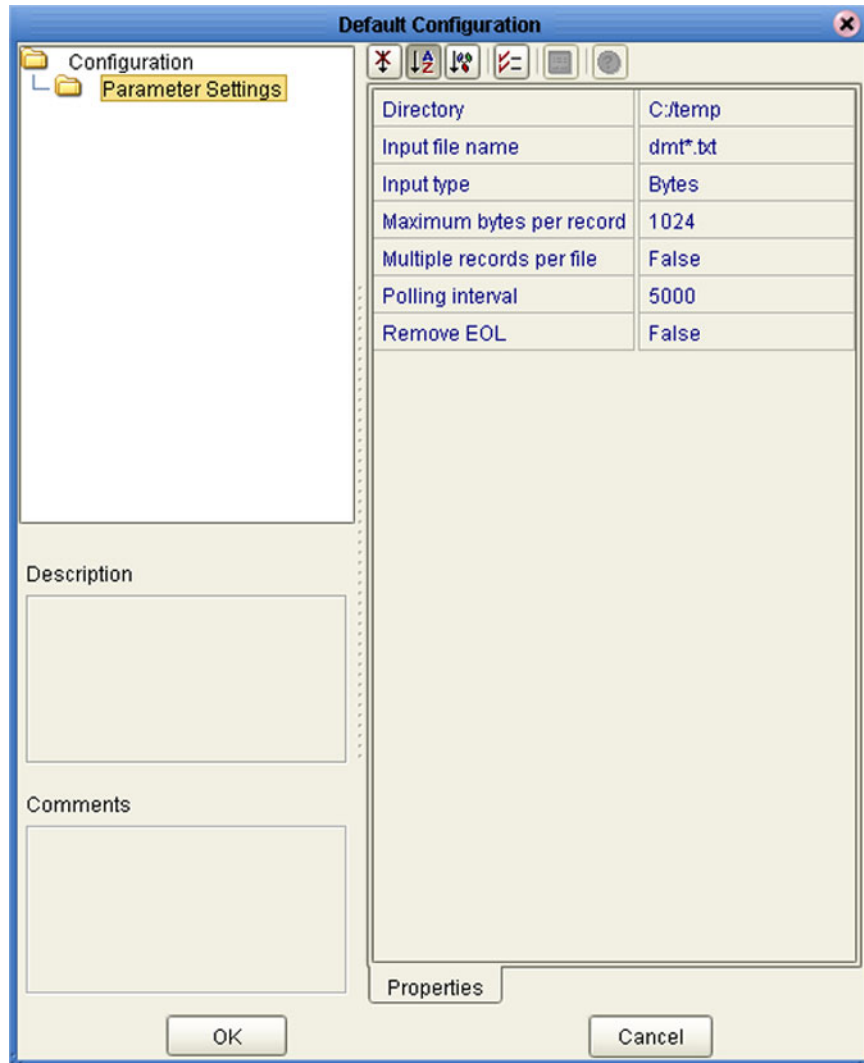
- When you link an external application with a Collaboration, the Enterprise Designer automatically adds an eWay Connection icon to the link. The eWay enables communication and movement of data between the external application and the eGate system. The eWay configuration specifies the logical connection properties for the link. See the individual eWay Intelligent Adapter User's Guides for specific information.
- When you link a Service with a Message Destination (queue or topic), the Enterprise Designer adds a JMS Client Connection icon. The JMS Client configuration specifies the logical connection properties for the linked Service. See the *eGate Integrator JMS Reference Guide* for information.

Note: You *must* configure the JMS properties, since no default values are assigned. Failure to do so results in an error at a later time.

5.8.1 Configuring a Connection

Double-clicking an eWay or JMS Client connection icon in the Connectivity Map displays the Default Configuration dialog box. As an example, Figure 66 shows a dialog box that lists the configuration properties for a File eWay.

Figure 66 Default Configuration Dialog Box









Note: The first time you double-click an eWay or JMS Client icon, you will see a Templates dialog box. Here, you must designate an eWay to be inbound or outbound, or a Service to be a publisher or subscriber. Clicking **OK** will then display the Default Configuration dialog box.

The constituent parts of the Default Configuration dialog box are:

- The **Configuration Tree** includes folders that contain configuration and connection properties for the selected eWay or message destination.
- The **Toolbar** contains a series of buttons used to sort and modify the information listed in the Properties folder, as described in Table 20.
- The **Properties** folder lists the default properties for the selected eWay or JMS Client. See, respectively, the individual eWay Intelligent Adapter User’s Guides and the *eGate Integrator JMS Reference Guide* for information.
- The **Description** box contains a brief description of the contents of the item currently selected in the Configuration Tree.
- The **Comments** box lists additional information about the item selected in the Configuration Tree.

Table 20 Configuration Dialog Box Toolbar Buttons

Button	Command	Function
	Unsorted	Displays configuration properties in their default order.
	Sort by Name	Sorts configuration properties alphabetically by name.
	Sort by Type	Displays configuration properties by property type.
	Show Editable Properties Only	Displays only the properties of an eWay or message destination that can be modified.
	Customizer	Displays the Customizer dialog box, which you can use to customize the selected eWay or message destination.
	Help	Displays the online help documentation for the Configuration Editor.

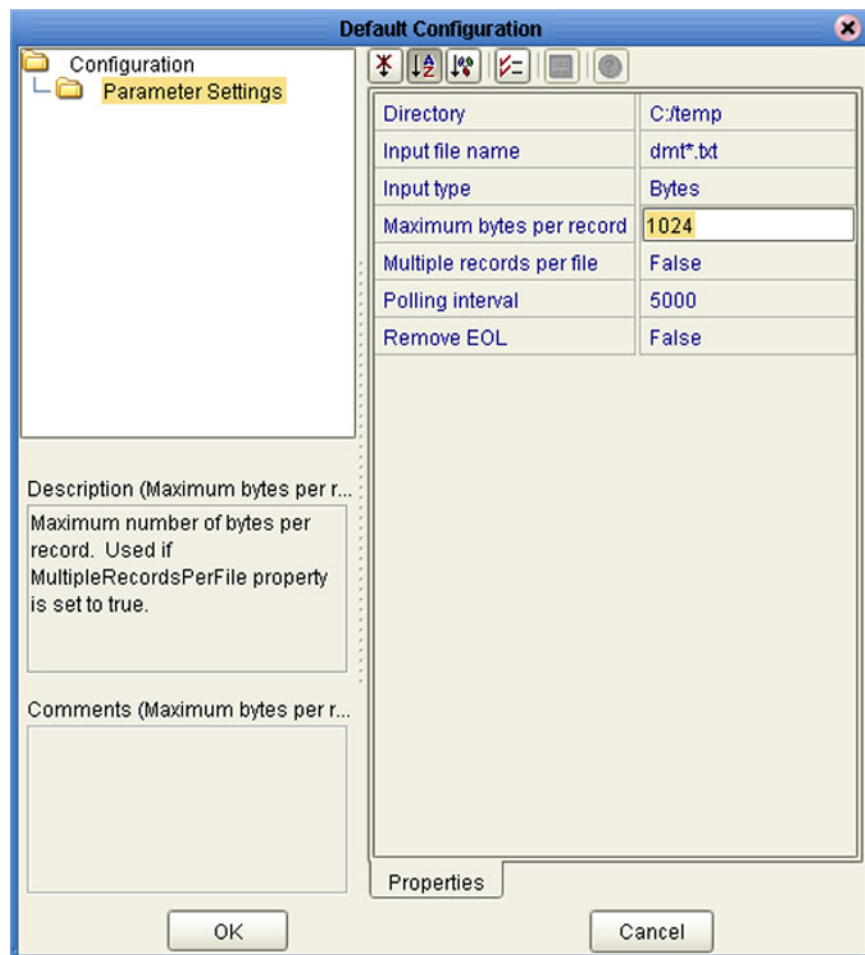
5.8.2 Modifying a Configuration Property

The properties initially listed in the Configuration Properties folder are the default settings for the selected eWay or JMS Client. This section describes how to change the default configuration values, using an inbound eWay as an example.

To change a configuration property

- 1 Double-click on an eWay or JMS Client properties handle icon in a Connectivity Map to display the Default Configuration dialog box.
- 2 Click on the right column of a row to place the row in edit mode. For example, click on the *Maximum bytes per record* field as shown in Figure 67, which highlights the current value.

Figure 67 Editing a Default Configuration Field



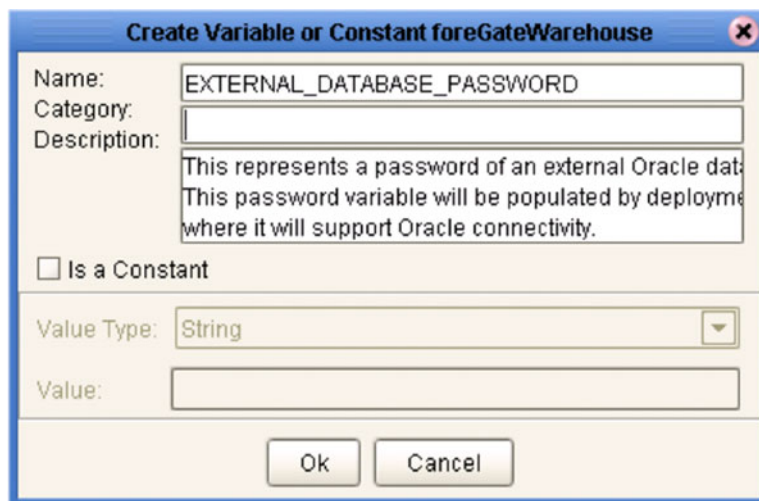
- 3 Enter a new value in the field. (Note that in this example, the parameter is not used unless you allow multiple records per file, as explained in the Description box).
- 4 Click **OK** to save the new settings.

5.9 Defining Constants and Variables

You can define variables and constants for a specific Project. Variables function as placeholders, having values that are determined when you create a specific Deployment Profile (see [Mapping Variables](#) on page 285). Project variable values can be literals or Environmental constants.

For example, a project variable is defined to represent a password of a database user in a target environment (see Figure 68). System managers will assign an actual value to this variable in the deployment profile editor. The value of the assigned project variable—an Environment constant—is then used to connect the database in the target environment.

Figure 68 Project Variable Creation

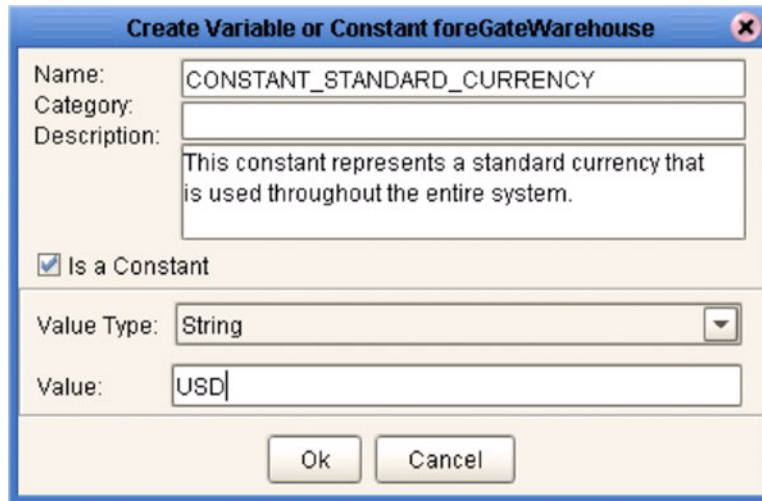


The screenshot shows a dialog box titled "Create Variable or Constant foreGateWarehouse". The dialog contains the following fields and controls:

- Name:** A text box containing "EXTERNAL_DATABASE_PASSWORD".
- Category:** An empty text box.
- Description:** A text box containing two lines of text: "This represents a password of an external Oracle dat..." and "This password variable will be populated by deployme... where it will support Oracle connectivity."
- Is a Constant:** A checkbox that is currently unchecked.
- Value Type:** A dropdown menu currently set to "String".
- Value:** An empty text box.
- Buttons:** "Ok" and "Cancel" buttons at the bottom.

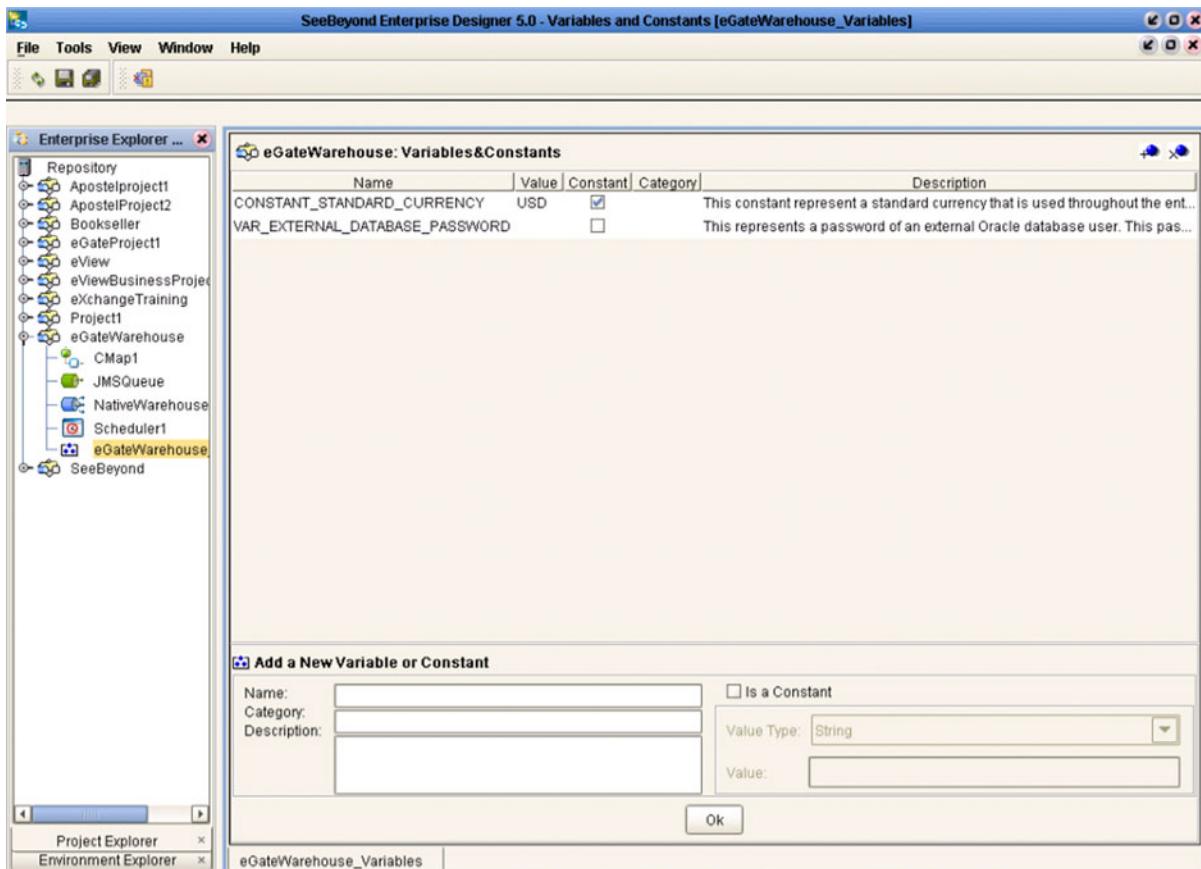
A constant is a name-value pair; when you create a constant you assign a permanent value to it, which cannot be overridden. An example of this would be a standard currency used globally throughout the system (see Figure 69).

Figure 69 Project Constant Creation



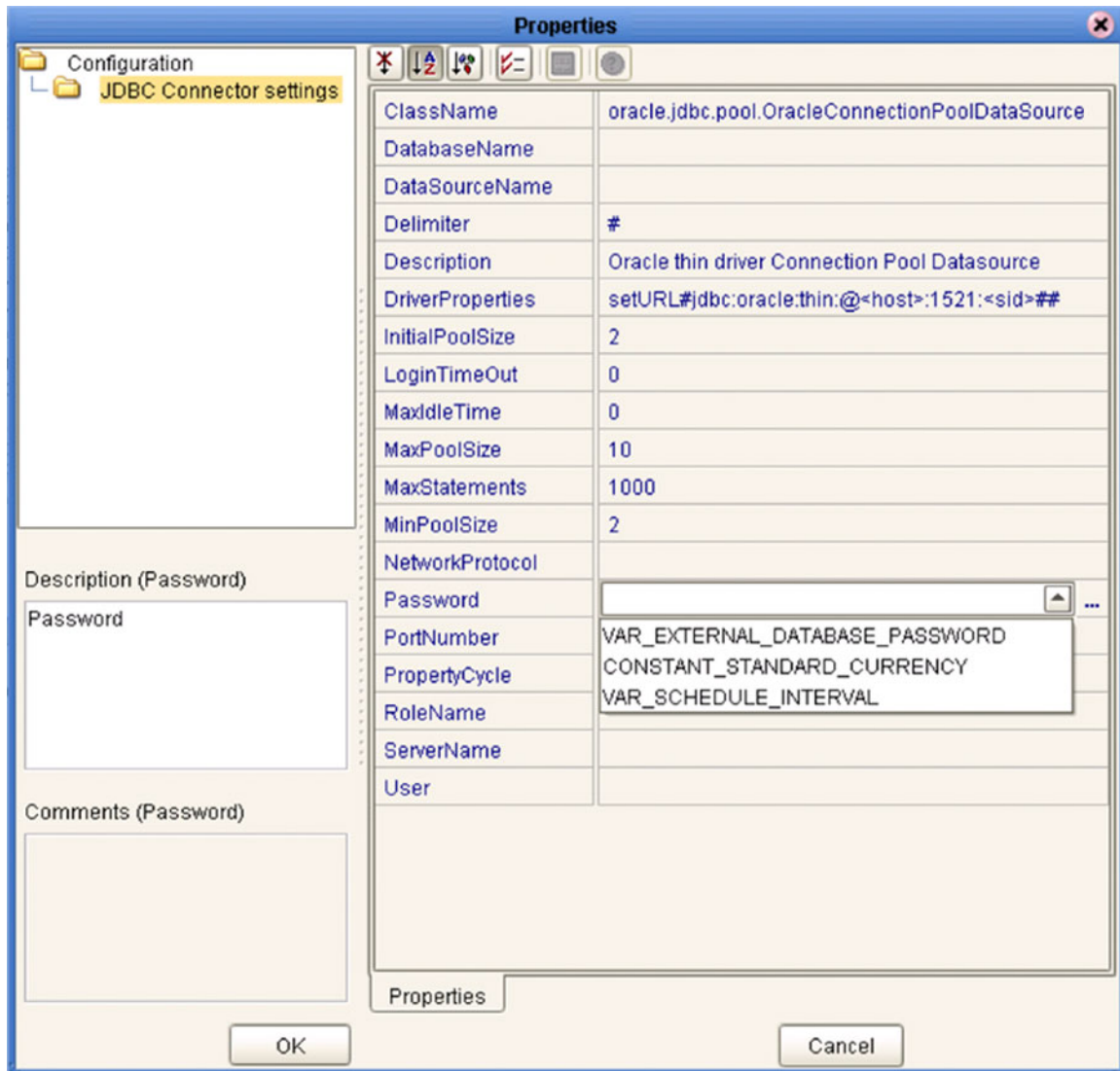
These constants and variables are automatically added to a Variables and Constants object group within the Project (see Figure 70).

Figure 70 Variables and Constants Object Group



The Project variables and constants can be referenced as properties within the Project. For example, the password variable described previously can be selected from the Variables and Constants object group to provide the Password property in the JDBC Connector settings dialog box (see Figure 71).

Figure 71 Connector Properties



See [Mapping Variables](#) on page 285 for information on setting the values for Project variables.

Web Services

This chapter describes the use of the Web Services capability of eGate Integrator, acting with other components of the ICAN Suite.

6.1 Overview

Basically, Web Services enables communication and data transfer between diverse applications using the Internet. In doing so, it provides a means for implementing EAI (Enterprise Application Integration) within an organization, or B2B (Business-to-Business) integration between partner organizations. This capability is achieved by wrapping back-end systems to present a common, standardized interface to the connecting network.

Four related technologies are used to transform and transport data within Web Services:

- **XML** (Extensible Markup Language)
Provides a language for defining both the data itself and the way to process it.
- **WSDL** (Web Services Description Language)
Defines the interfaces, data types, interactions, and mappings used in the Web Services. WSDL files are used to invoke and operate Web services on the Internet and to access and invoke remote applications and databases.
- **SOAP** (Simple Object Access Protocol)
Defines a communications envelope that is mappable to HTTP and provides a format for transmitting XML documents over a network.
- **UDDI** (Universal Description, Discovery, and Integration)
Provides a mechanism for storing and categorizing information that allows publication of services and discovery of external services.

6.2 SeeBeyond Web Services

eGate Integrator provides the capability to create either a client or a server to receive WSDL file from a remote server, or send WSDL files to a remote client. eGate works in conjunction with eInsight Business Process Manager, in which the associated business processes are developed. See [Building a Web Client](#) on page 105 and [Building a Web Server](#) on page 113.

The ICAN Suite contains the following components that implement the Web Services capability:

- **WSDL Wizard**

The WSDL Wizard creates an OTD from a WSDL file. See [Using the WSDL Wizard](#) on page 128.

- **WSDL Editor**

See the *eInsight Business Process Manager User's Guide*.

- **WSDL Interface Designer**

See the *eInsight Business Process Manager User's Guide*.

- **WSDL Viewer**

See the *eInsight Business Process Manager User's Guide*.

- **UDDI Repository**

All ICAN objects that can be accessed as Web services are presented in the SeeBeyond UDDI Repository. See [UDDI Repository](#) on page 103.

6.3 UDDI Repository

In general, all ICAN objects that expose themselves as a Web service (such as an eInsight business process) are presented in the SeeBeyond UDDI Repository (see Figure 72). The URL of this repository is:

```
http://ICAN_Suite_host_name:enterprise_manager_installation_port/  
stcuddi
```

Figure 72 SeeBeyond UDDI Repository



Environment	Service Name	WSDL
Environment2	BusinessProcess1	http://art2k:10000/repository/MyRepository/data/uddocs/Environment2/BusinessProcess1/BusinessProcess1.wsdl
Environment2	BusinessProcess1	http://art2k:10000/repository/MyRepository/data/uddocs/Environment2/BusinessProcess1/BusinessProcess2.wsdl

Each entry in the UDDI Repository includes:

- The ICAN environment name.
- The actual (Web) Services name.
- The location of the Web Service’s WSDL file.

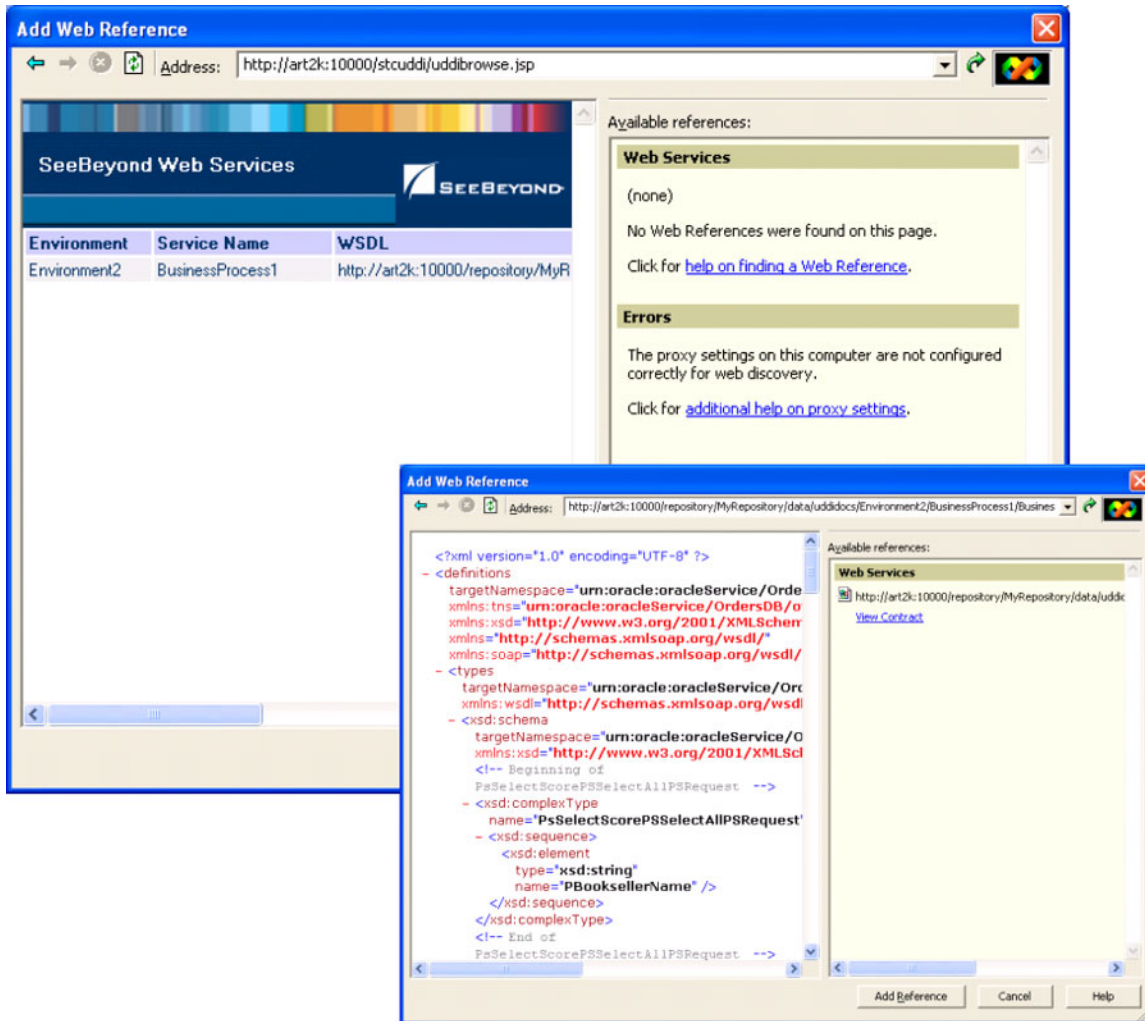
By selecting an entry its WSDL file is displayed, as shown in Figure 73.

Figure 73 Example Web Service WSDL File

```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions targetNamespace="urn:oracle:oracleService/OrdersDB/otdGetCreditScore" xmlns:tns="urn:oracle:oracleService/OrdersDB/otdGetCreditScore"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/wsdl/" xmlns:soap11="http://schemas.xmlsoap.org/wsdl/soap11/" />
- <types targetNamespace="urn:oracle:oracleService/OrdersDB/otdGetCreditScore" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" />
- <xsd:schema targetNamespace="urn:oracle:oracleService/OrdersDB/otdGetCreditScore" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!-- Beginning of PsSelectScorePSSelectAllPSRequest -->
  - <xsd:complexType name="PsSelectScorePSSelectAllPSRequest">
    - <xsd:sequence>
      <xsd:element type="xsd:string" name="PBooksellerName" />
    </xsd:sequence>
  </xsd:complexType>
  <!-- End of PsSelectScorePSSelectAllPSRequest -->
  <!-- Beginning of PsSelectScorePSSelectOnePSResponseType -->
  - <xsd:complexType name="PsSelectScorePSSelectOnePSResponseType">
    - <xsd:sequence>
      <xsd:element type="xsd:decimal" name="CREDIT_SCORE" />
    </xsd:sequence>
  </xsd:complexType>
  <!-- End of PsSelectScorePSSelectOnePSResponseType -->
  <!-- Beginning of PsSelectScorePSSelectMultiplePSResponseType -->
  - <xsd:complexType name="PsSelectScorePSSelectMultiplePSResponseType">
    - <xsd:sequence>
      <xsd:element type="xsd:long" name="rowCount" />
      <xsd:element type="xsd:decimal" name="CREDIT_SCORE" />
    </xsd:sequence>
  </xsd:complexType>
  <!-- End of PsSelectScorePSSelectMultiplePSResponseType -->
  <!-- Beginning of PsSelectScorePSSelectAllPSResponseType -->
  - <xsd:complexType name="PsSelectScorePSSelectAllPSResponseType">
    - <xsd:sequence>
      <xsd:element type="xsd:long" name="rowCount" />
    </xsd:sequence>
  </xsd:complexType>
  </xsd:schema>
```

The SeeBeyond UDDI Repository can be used in a third party tool, for example Microsoft Visual Studio (see Figure 74). In this example, a so-called *Web reference* (to the UDDI repository) is added to a C# project.

Figure 74 Microsoft Visual Studio Example



eGate Integrator can exchange data with Internet and Web Services applications using the Web Services Description Language (WSDL). This language is XML-based and is used to define Web services and describe how to access them. The WSDL OTD Wizard is used to build OTDs that are used in the Project Collaborations (see [Using the WSDL Wizard](#) on page 128).

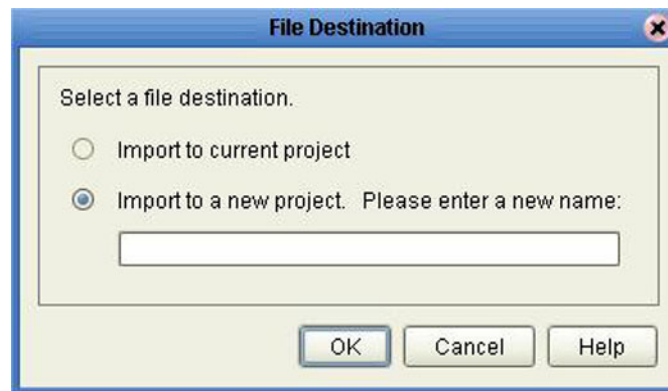
6.4 Building a Web Client

Here we briefly demonstrate the procedure for building a Web client. The Project used in the following example is available as **webclient.zip**, contained in the eGate User Guide Sample file included with this User's Guide.

To import the sample project

- 1 The sample files are uploaded with the User's Guide .sar file and downloaded from the Enterprise Manager's Documentation tab. Extract the samples from the Enterprise Manager to a local file.
- 2 From the Enterprise Designer's Project Explorer pane, right-click the Repository and click **Import Project** from the selection menu. The **Select File to Import** dialog box appears.
- 3 Browse to the directory that contains the sample project zip file. Select the sample file (**webclient.zip**) and click **Open**.
- 4 From the *File Destination* dialog box (see Figure 75), select **Import to a new Project**, enter the name of the Project, and click **OK**.

Figure 75 File Destination Dialog Box

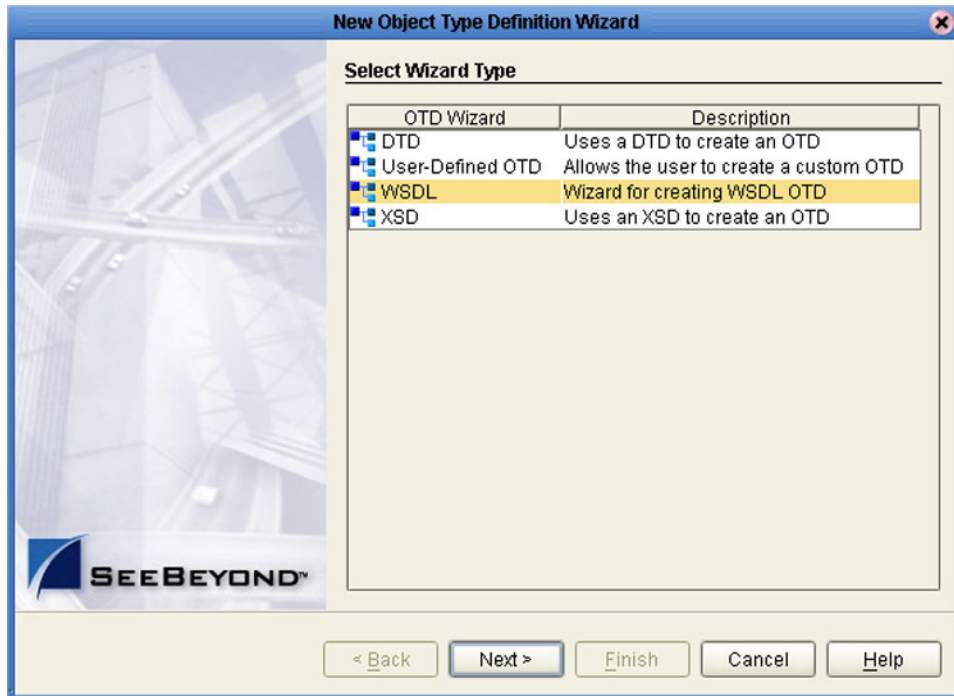


- 5 After the import has successfully completed, select the **Repository** in the Project Explorer and click **Refresh All from Repository**.

To build a Web Client Using the ICAN Suite

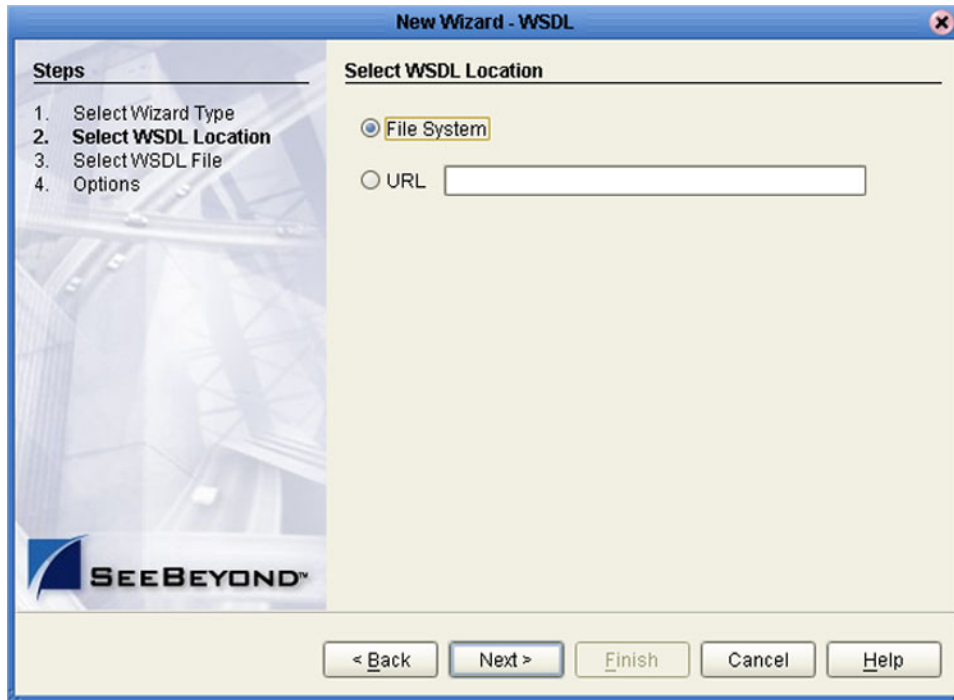
- 1 In Project Explorer, create a new OTD.
- 2 Select the WSDL OTD Wizard (see Figure 76).

Figure 76 Select WSDL Wizard



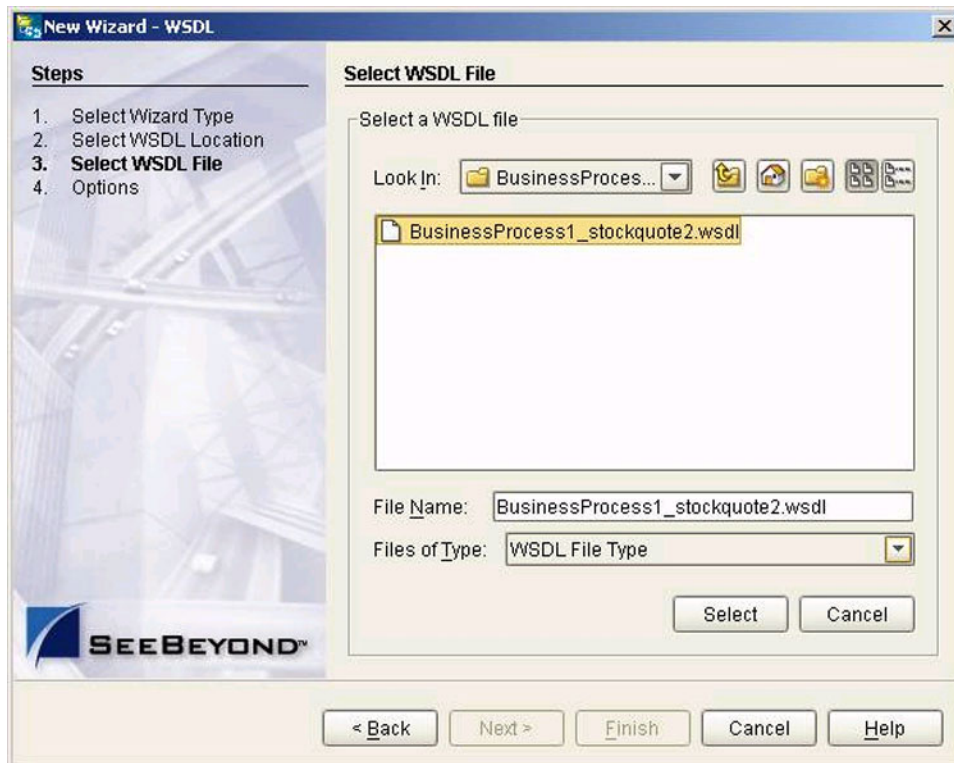
- 3 Select the WSDL file location (see Figure 77).

Figure 77 Select File Location



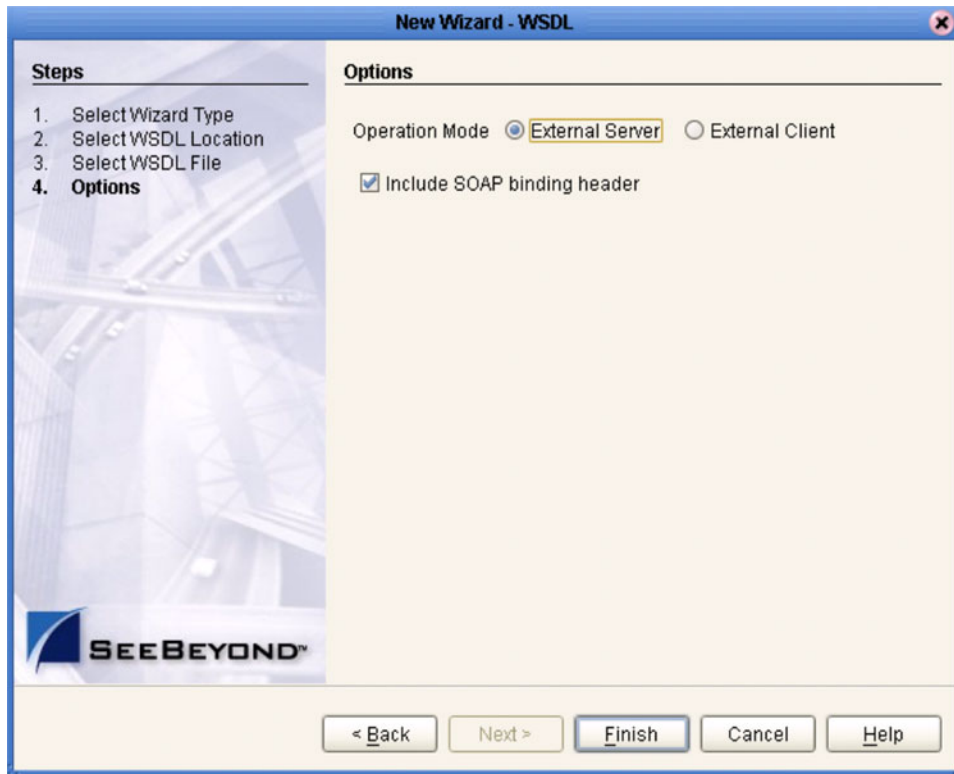
- 4 Select the WSDL file you want to use for the OTD (see Figure 78).

Figure 78 Select WSDL File



- 5 For a Web Client, select External Server (see Figure 79).

Figure 79 Select External Server



- 6 Build your business process using eInsight. The *receive* business rule is shown in Figure 80, and the *write* business rule is shown in Figure 81.

Figure 80 eInsight Business Process - Receive Rule

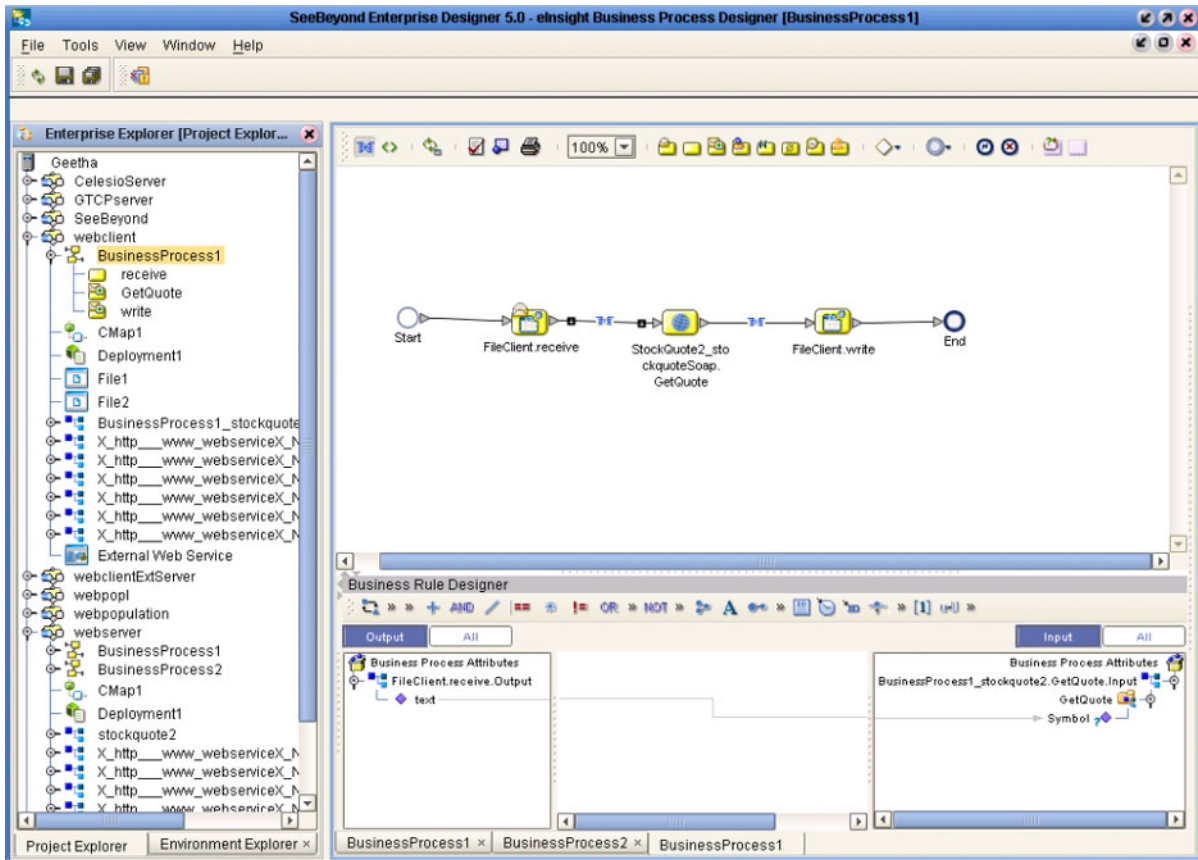
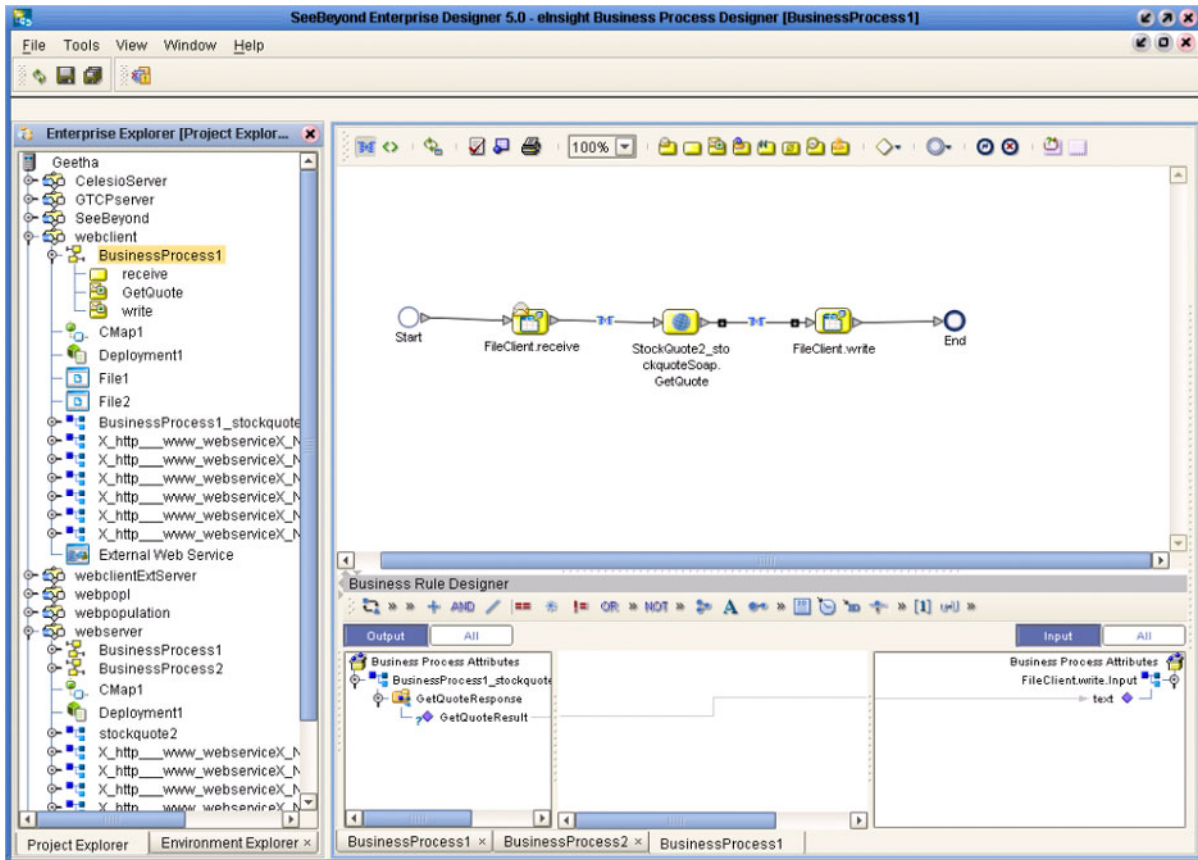
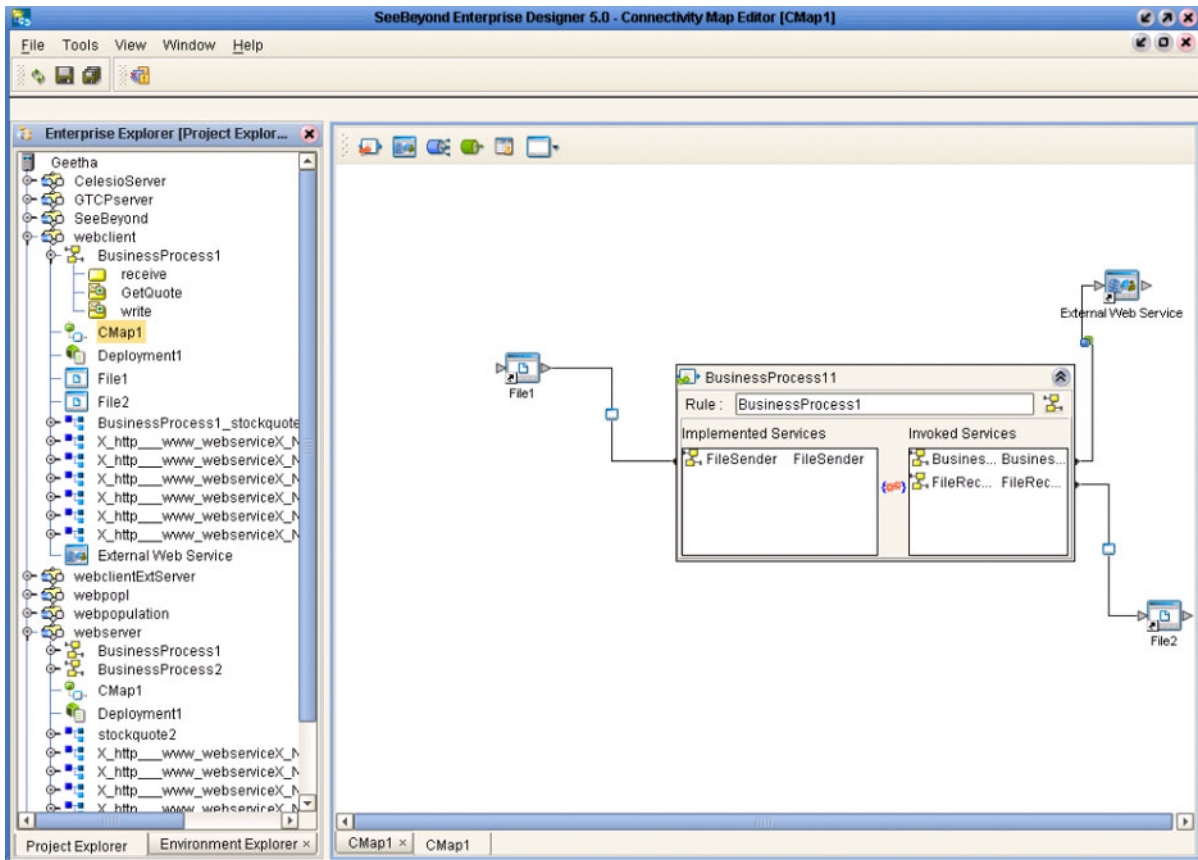


Figure 81 eInsight Business Process - Write Rule



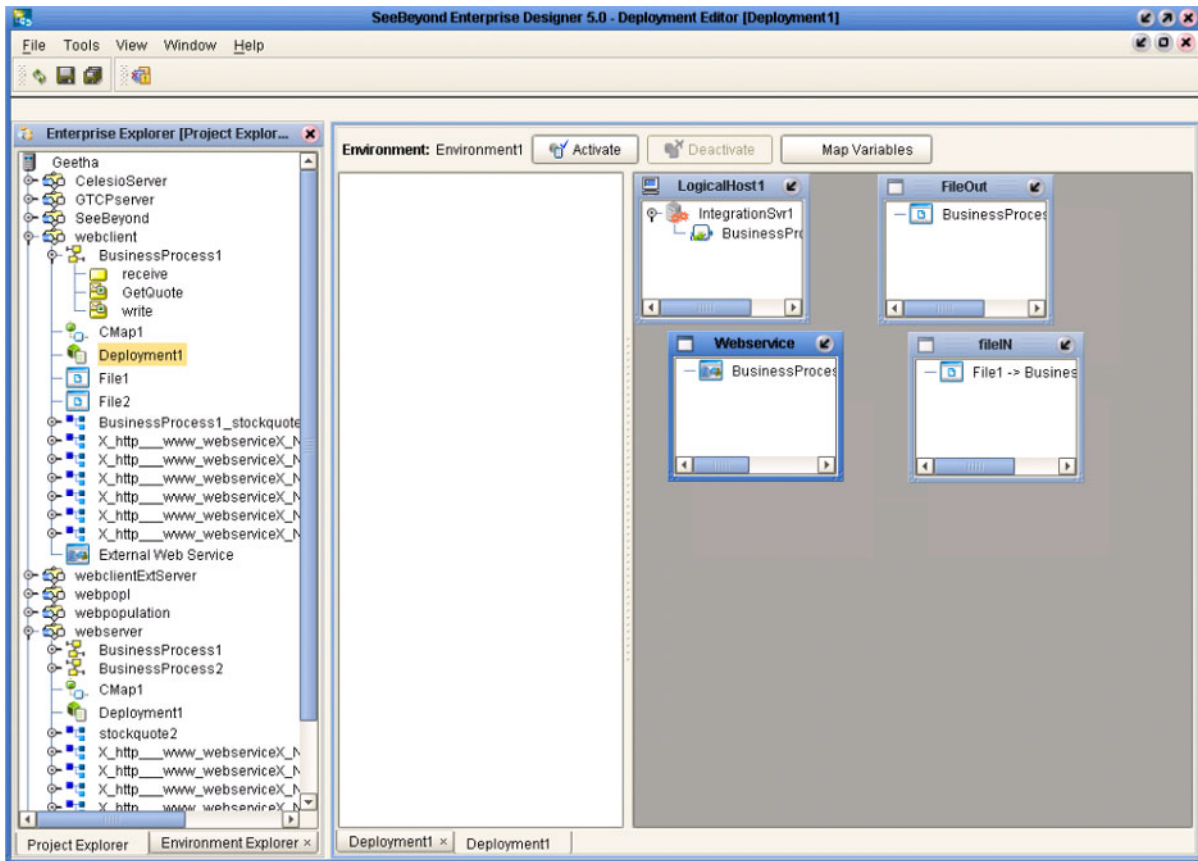
- 7 Create and map the Project components in the Enterprise Designer Connectivity Map (see Figure 82).

Figure 82 Connectivity Map



8 Deploy your Project as shown in Figure 83.

Figure 83 Project Deployment



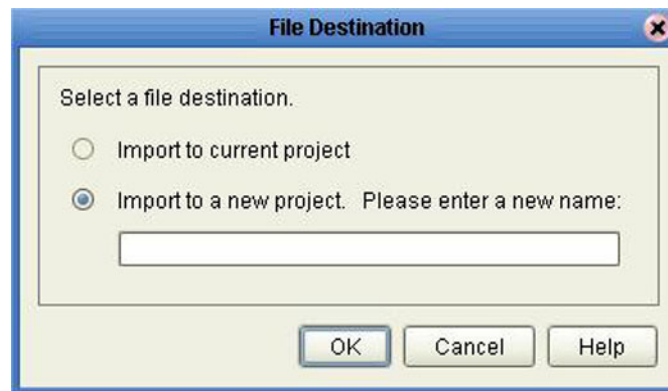
6.5 Building a Web Server

Here we briefly demonstrate the procedure for building a Web server. The Project used in the following example is available as **webservice.zip**, contained in the eGate User Guide Sample file included with this User's Guide.

To import the sample project

- 1 The sample files are uploaded with the User's Guide .sar file and downloaded from the Enterprise Manager's Documentation tab. Extract the samples from the Enterprise Manager to a local file.
- 2 From the Enterprise Designer's Project Explorer pane, right-click the Repository and click **Import Project** from the selection menu. The **Select File to Import** dialog box appears.
- 3 Browse to the directory that contains the sample project zip file. Select the sample file (**webservice.zip**) and click **Open**.
- 4 From the *File Destination* dialog box (see Figure 75), select **Import to a new Project**, enter the name of the Project, and click **OK**.

Figure 84 File Destination Dialog Box

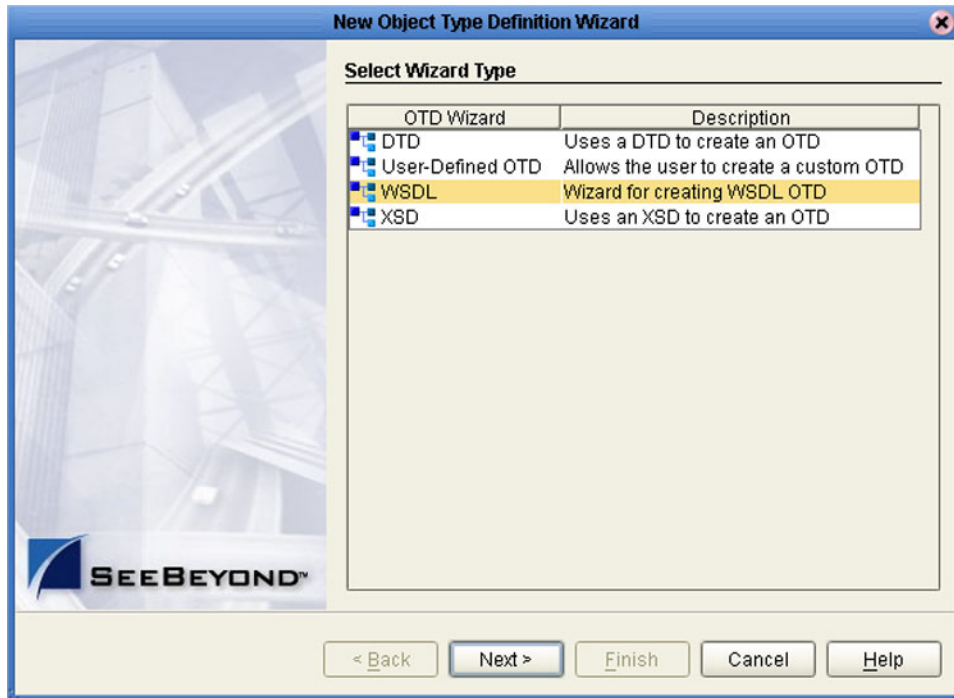


- 5 After the import has successfully completed, select the **Repository** in the Project Explorer and click **Refresh All from Repository**.

To build a Web Server Using the ICAN Suite

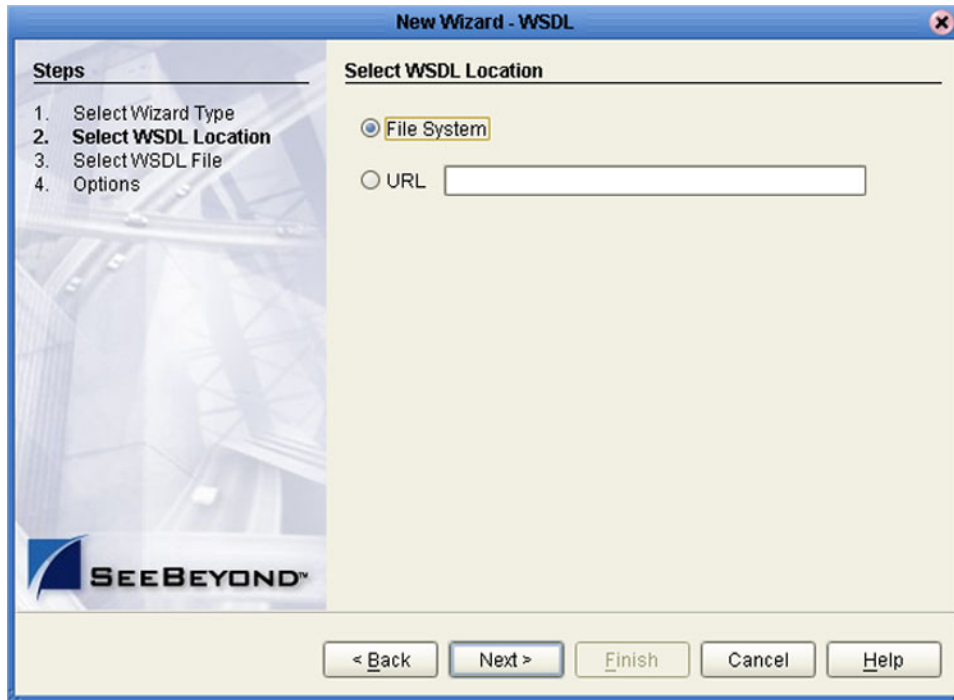
- 1 In Project Explorer, create a new OTD.
- 2 Select the WSDL OTD Wizard (see Figure 76).

Figure 85 Select WSDL Wizard



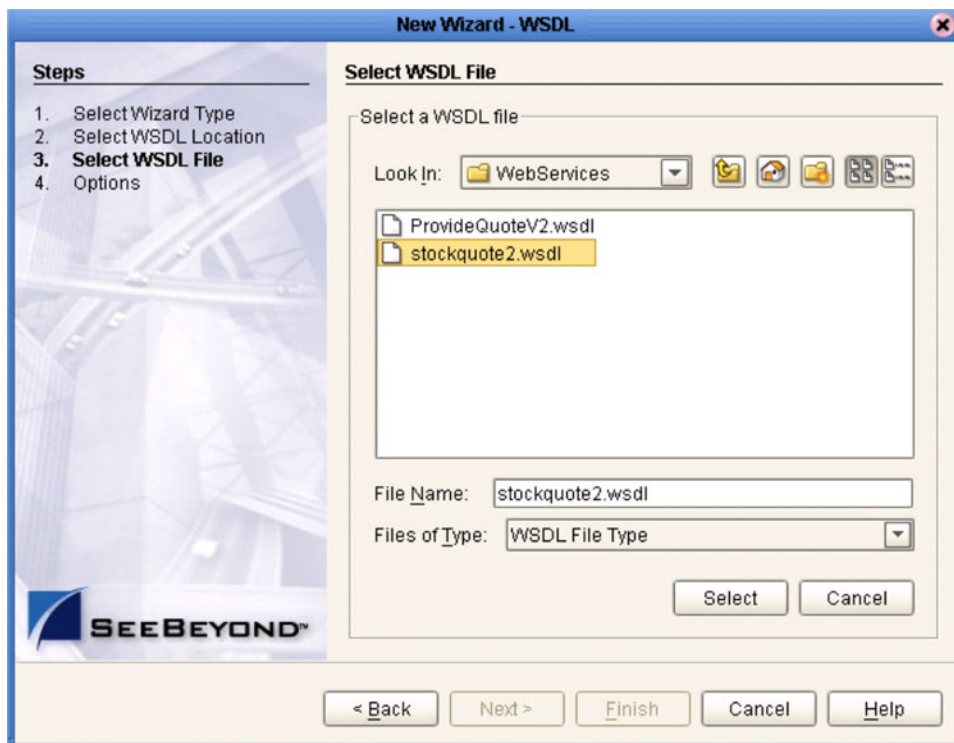
- 3 Select the WSDL file location (see Figure 86).

Figure 86 Select File Location



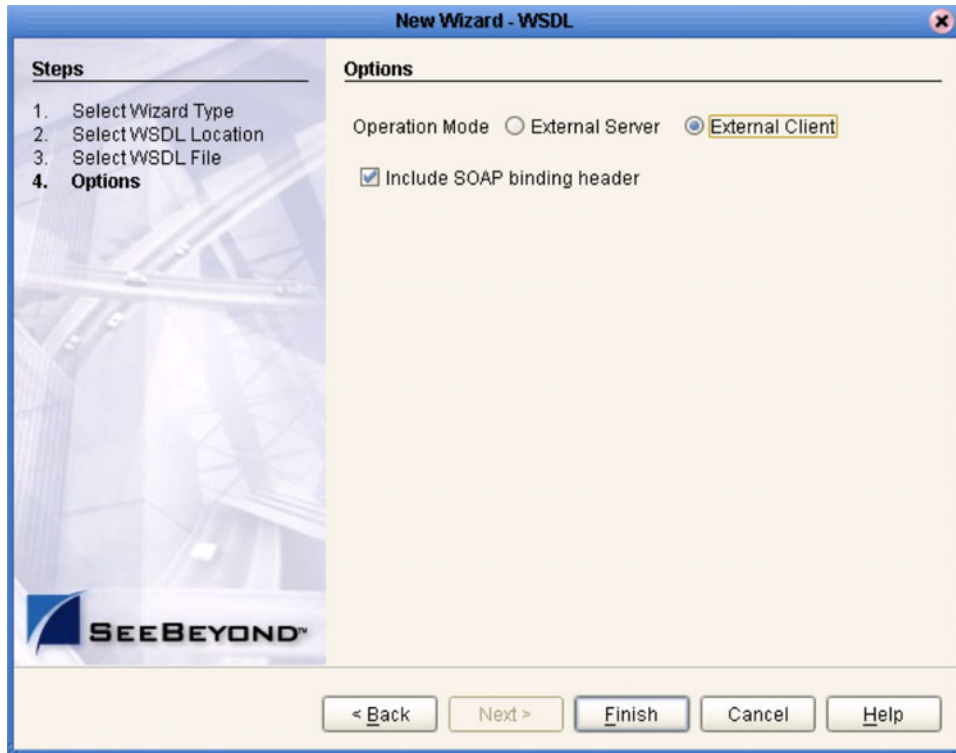
- 4 Select the WSDL file you want to use for the OTD (see Figure 87).

Figure 87 Select WSDL File



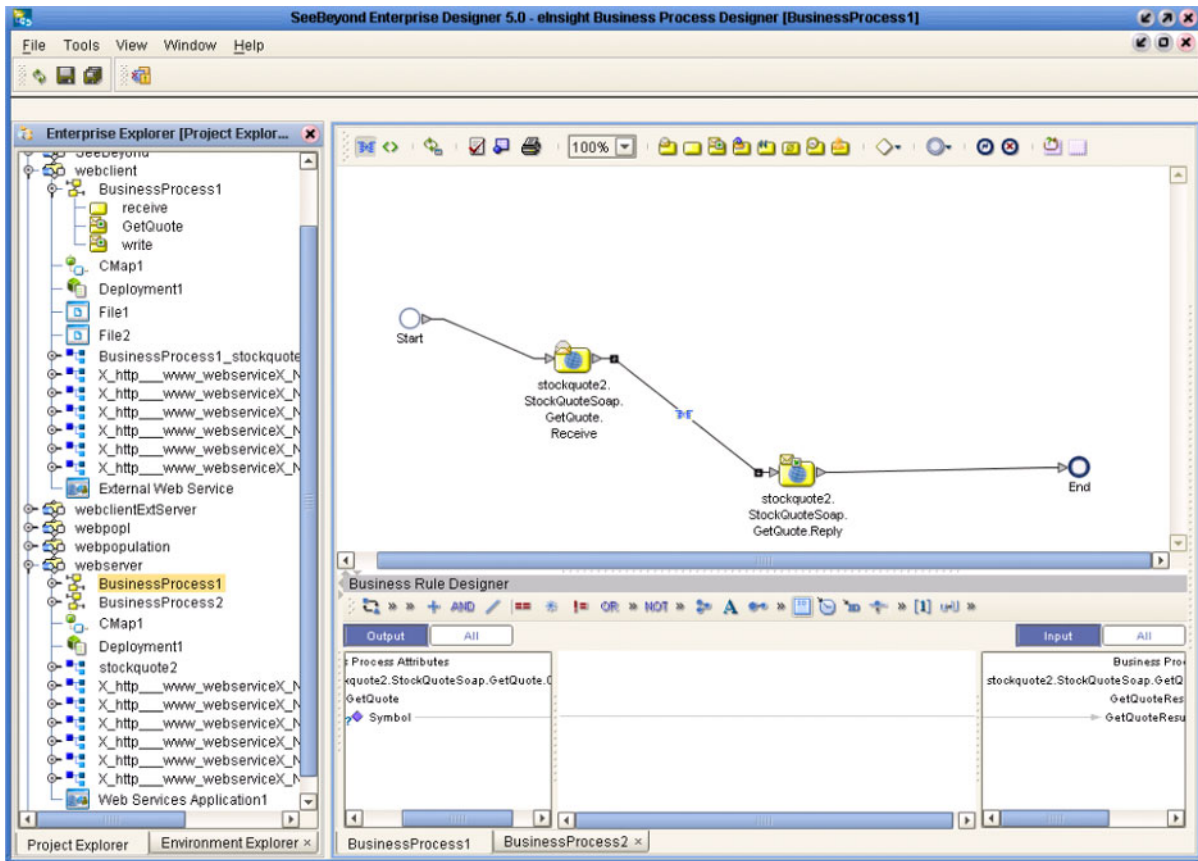
- 5 For a Web server, select External Client (see Figure 88).

Figure 88 Select External Client



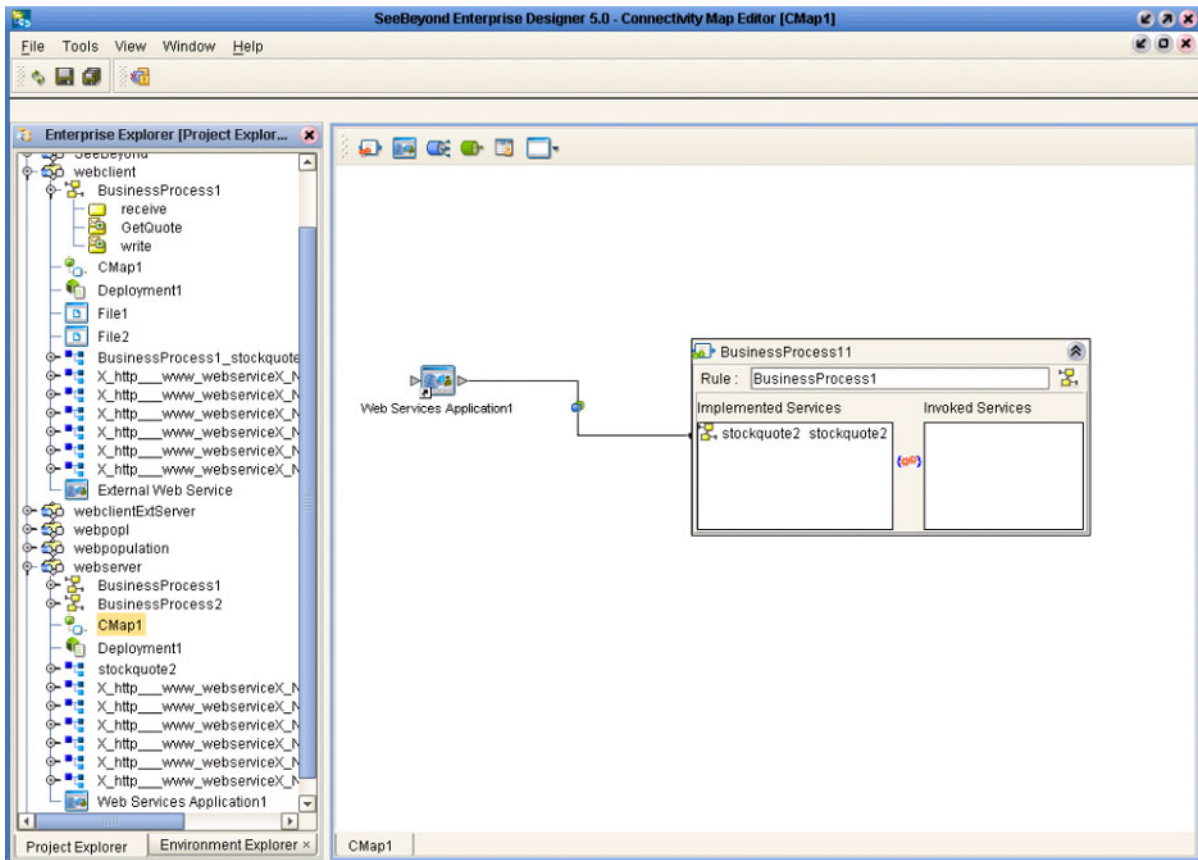
- 6 Build your business process using eInsight, as shown in Figure 89.

Figure 89 eInsight Business Process



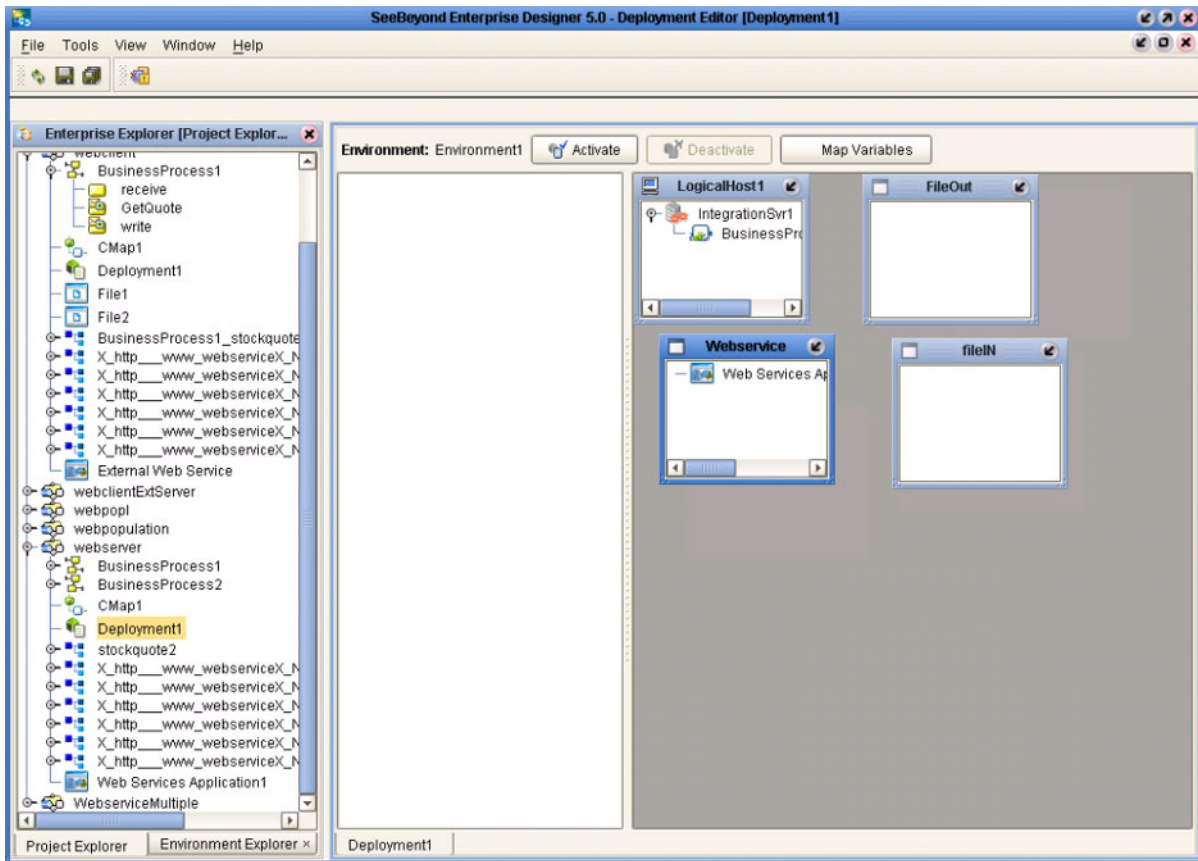
- 7 Create and map the Project components in the Enterprise Designer Connectivity Map (see Figure 82).

Figure 90 Connectivity Map



8 Deploy your Project as shown in Figure 91.

Figure 91 Project Deployment



Object Type Definitions

This chapter describes the OTD creation process. The Enterprise Designer includes two tools, the OTD Wizard and OTD Editor, to help you create and customize OTDs.

7.1 Overview

An Object Type Definition (OTD) is a description of a complex hierarchical data structure that can be accessed and manipulated by your code in a Collaboration. OTDs typically have a specific external representation format that is used to store and transport the OTD contents through the parts of a eGate Project. The OTD defines both the run-time structure and the external representation.

At run time, an OTD instance is accessed either directly from Java in a Java Collaboration, using the so-called *bean-like* accessors, or from BPEL using XPath expressions. In Java, the nodes comprising the hierarchy of the data structure have an interface similar to Java beans: each node has a set of properties with *get* and *set* methods to manipulate them. There is also a so-called generic interface (see [Appendix A](#)).

Typically, a collaboration will receive a message containing the external representation of a particular OTD. It will use the *unmarshal* method of an instance of that OTD to parse the data and make it accessible through the hierarchical data structure. Then it will perform some operation: for example, copying parts of the data to another OTD instance. Finally, it will invoke the *marshal* method on the other OTD instance to render the contents of its data structure as a single, serialized data stream for further transport.

7.1.1 The *Bean-like* Interface

The hierarchical data structure of an OTD is represented at run time by a set of generated Java classes. These classes follow the Java bean rule. They have a set of zero or more properties, each of which has a specific type and a given name, and may be optional and/or repeating. In contrast to regular bean properties, a OTD node property has two distinct names: a display name, that can be a virtually-arbitrary string, and a Java name that is the accessor basename.

For example, if a node has a property with Java name "X", then the implementing class for that node will have a method "getX". The Java name is normally derived from the display name, modified to suit the restrictions on Java identifiers, and supplied automatically by eGate.

7.1.2 OTD Types

Externally-Defined OTDs

Externally-defined OTDs are based on formats or standards external to eGate Integrator, such as Document Type Definition (DTD), Web Services Definition Language (WSDL), XML Schema Definition (XSD), and various proprietary formats such as SAP BAPI. Some of these OTDs are *messagable*, others are API-based. Externally defined OTDs are read-only.

User-Defined OTDs

User-defined OTD are native to eGate Integrator. You can create a User-defined OTD from scratch using the User-Defined OTD Wizard. User-defined OTDs are read/write—you can add or delete nodes or edit their properties, and you can work with internal and external templates.

7.1.3 Building OTDs

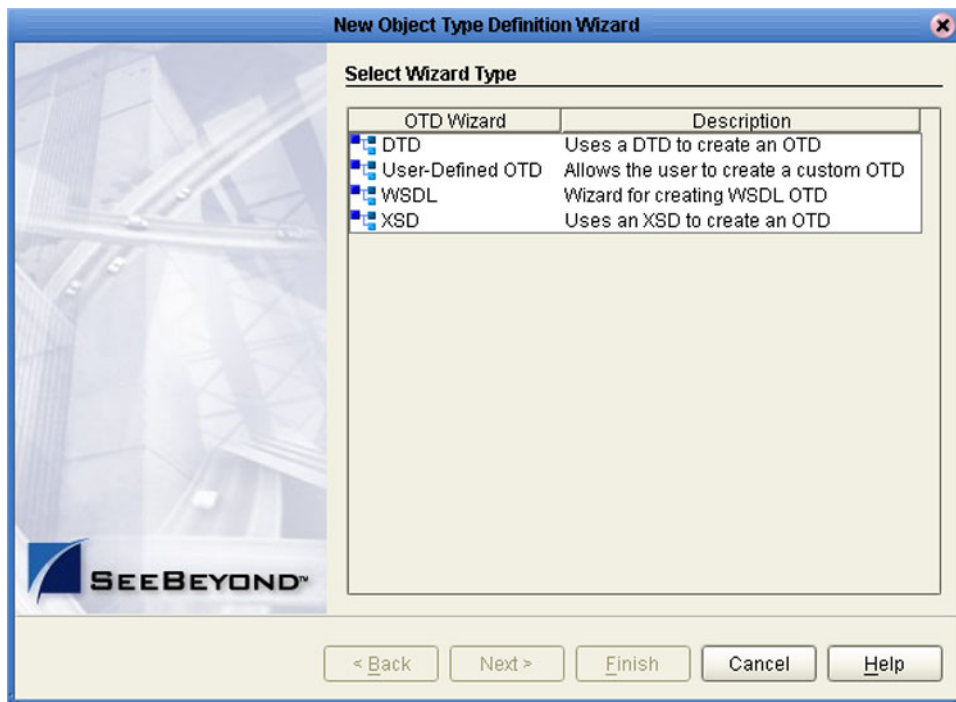
Wizards are provided in the Enterprise Designer GUI to guide you through the OTD building process. These Wizards call back-end builders that actually implement the building of the code, based on the provided information.

7.2 Using the OTD Wizard

Right-click on a Project in the Enterprise Explorer to display the Project context menu, then select **New Object Type Definition** to display the OTD Wizard, shown in Figure 92. The initial dialog allows you to select a specific OTD Wizard. The basic Wizards supplied with eGate Integrator are described in:

- [Using the DTD Wizard](#) on page 124
- [Using the User-Defined OTD Wizard](#) on page 137
- [Using the WSDL Wizard](#) on page 128
- [Using the XSD Wizard](#) on page 133

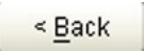

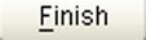
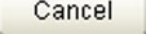

Figure 92 OTD Wizard Selection Dialog



Additional OTD Wizards are supplied with eGate Add-ons, and are described in the User's Guides for the specific products. When these products are installed, the OTD Wizards are added to the list shown in Figure 92.

The OTD Wizards guide you through the initial phases of creating an Object Type Definition, and then invoke the OTD Editor. The user interface is highly self-explanatory, but details of the navigation buttons are listed in Table 21 for your reference.

Table 21 OTD Wizard Navigation Buttons

Button	Function
	Returns to the previous step in the wizard. This button is disabled on the first step.
	Goes to the next step in the wizard. This button is disabled on the last step.
	Saves all OTD settings and closes the wizard. This button is only enabled on the last step.
	Closes the wizard without saving the OTD.
	Displays the online help documentation for the OTD Wizard dialog box.

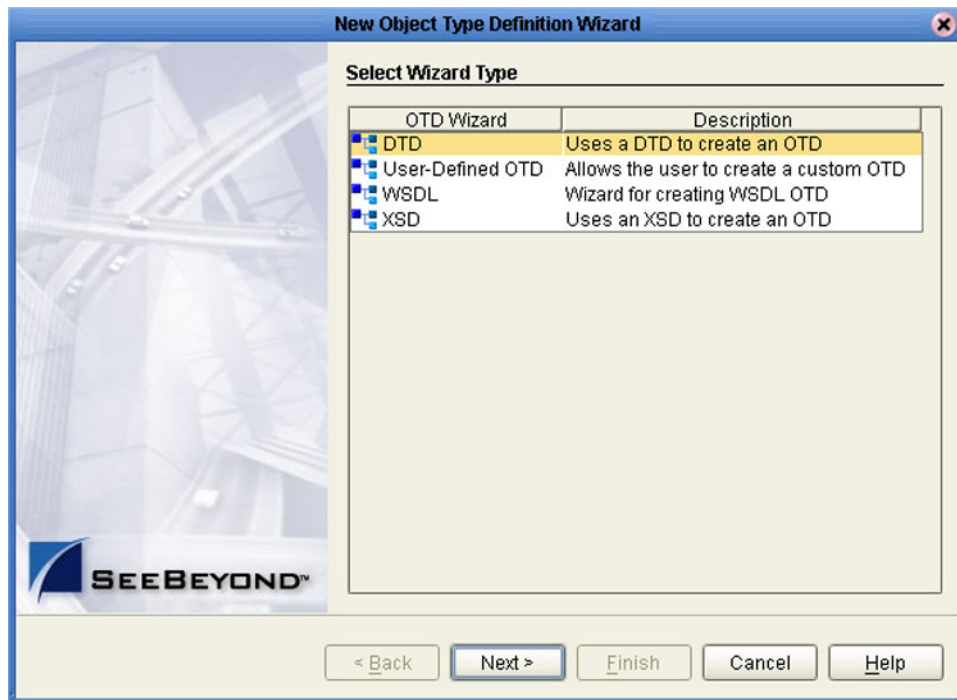
7.3 Externally-Defined OTDs

7.3.1 Using the DTD Wizard

To create an OTD file from a DTD file

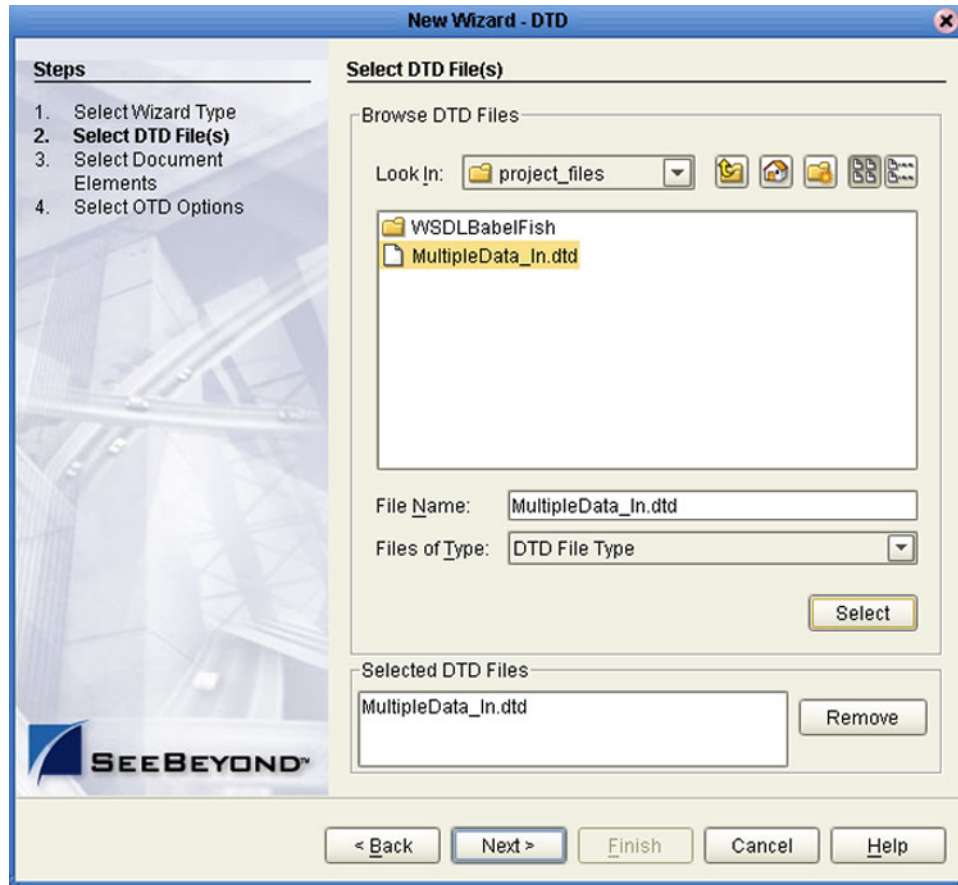
- 1 In the *Select Wizard Type* dialog, select **DTD** from the *OTD Wizard* list (see Figure 93) to create an OTD file from a Data Type Definition (DTD) file.

Figure 93 OTD Wizard Selection: DTD Wizard



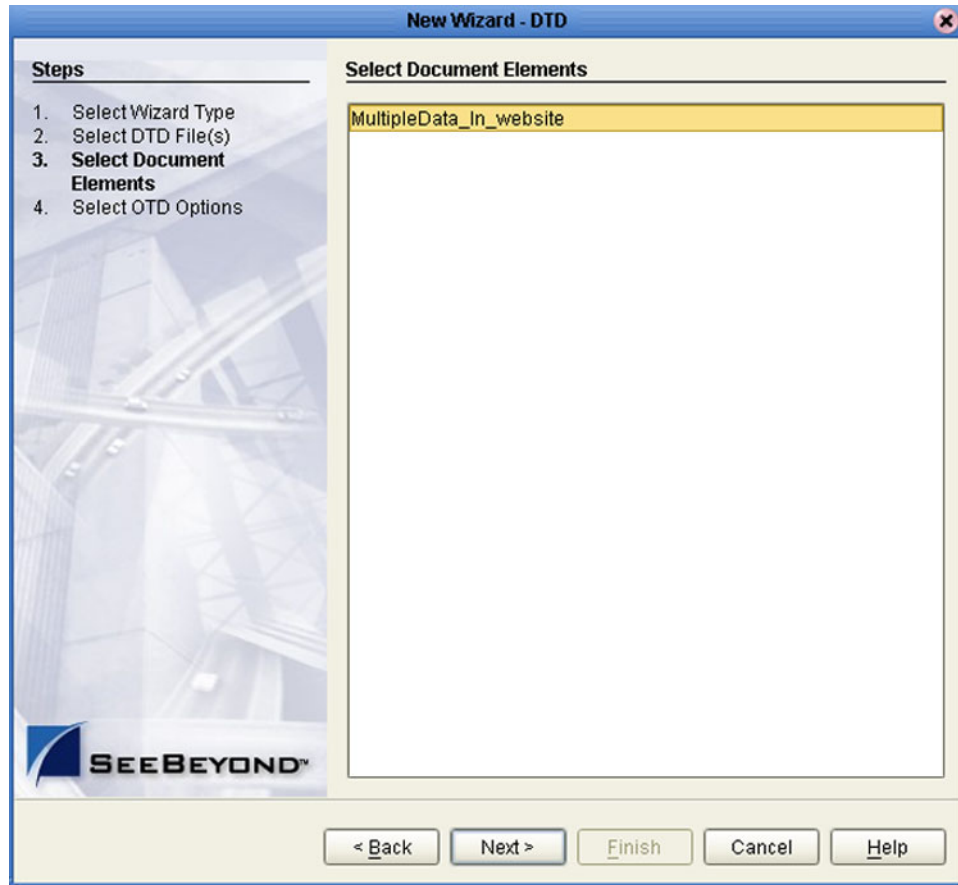
- 2 Click **Next** to display the *Select DTD File(s)* dialog box, shown in Figure 94.

Figure 94 Select DTD File(s) Dialog Box



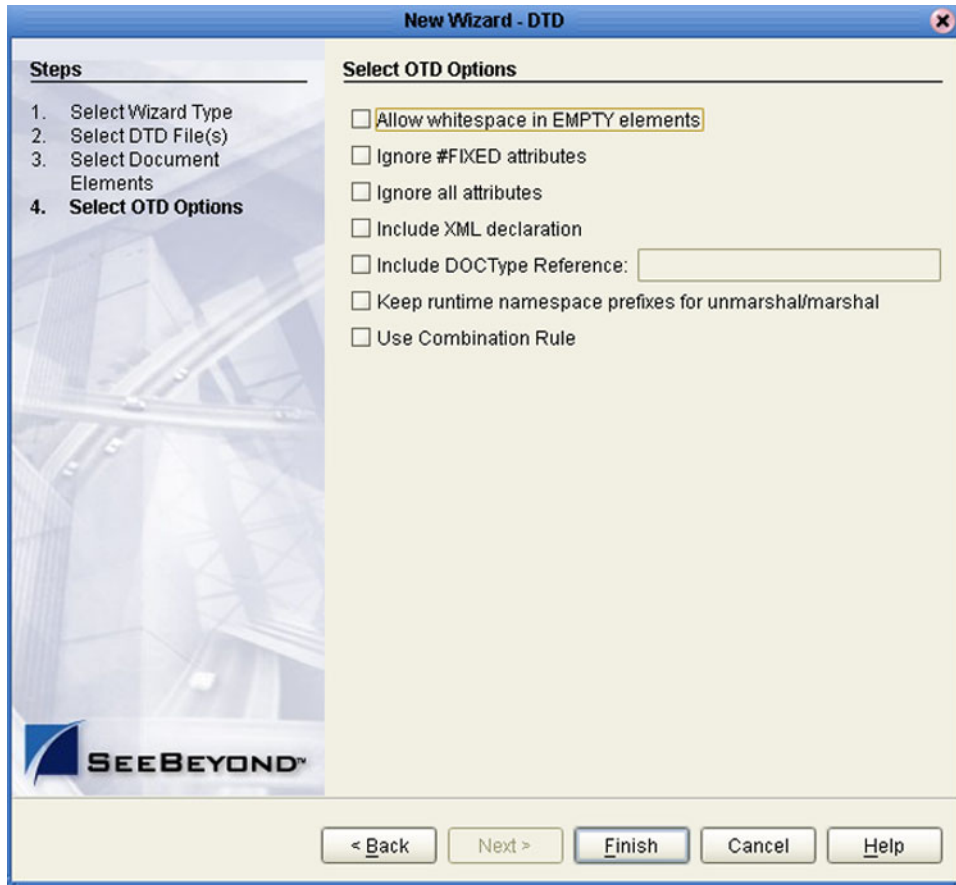
- 3 In the *Look In* drop-down list, navigate to the DTD file or files that you want to use to create the OTD. Click **Select** to add the files to the *List of Selected DTDs*.
- 4 Click **Next** to display the *Select Document Elements* dialog box, shown in Figure 95.

Figure 95 Select Document Elements Dialog Box



- 5 Select the elements of the document that you want to include in the OTD.
- 6 Click **Next** to display the *Select OTD Options* dialog box, shown in Figure 96.

Figure 96 Select OTD Options Dialog Box



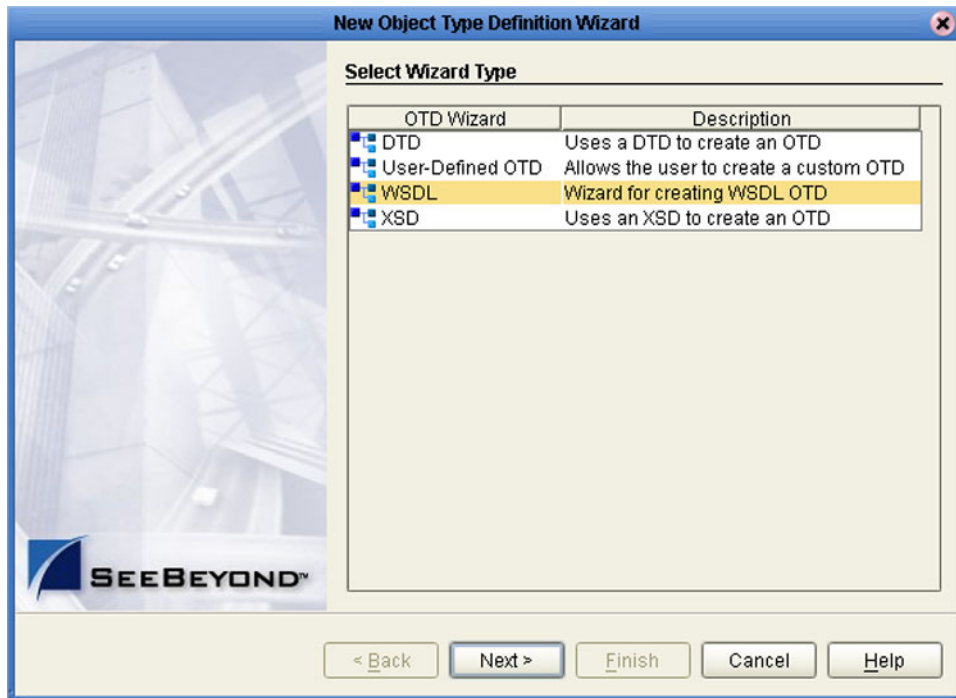
- 7 Select the check boxes next to the OTD options you want to enable, and click **Finish** to add the OTD to the Enterprise Designer with the selected OTD options.

7.3.2 Using the WSDL Wizard

To create an OTD file from a WSDL file

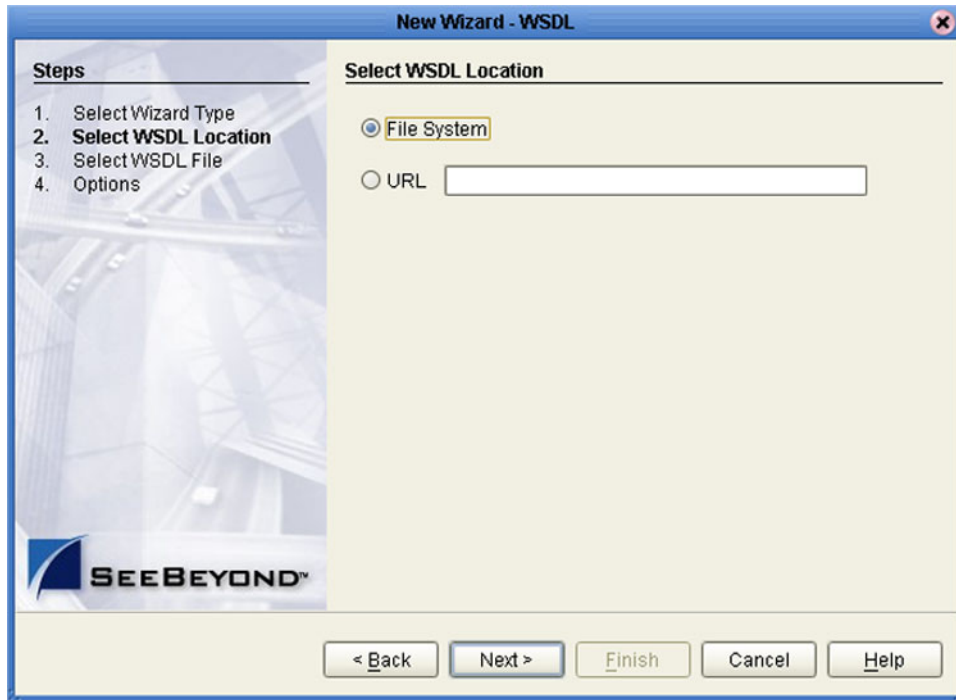
- 1 In the *Select Wizard Type* dialog, select **WSDL** from the *OTD Wizard* list (see Figure 97) to create an OTD file from an WSFL file.

Figure 97 OTD Wizard Selection: WSDL Wizard



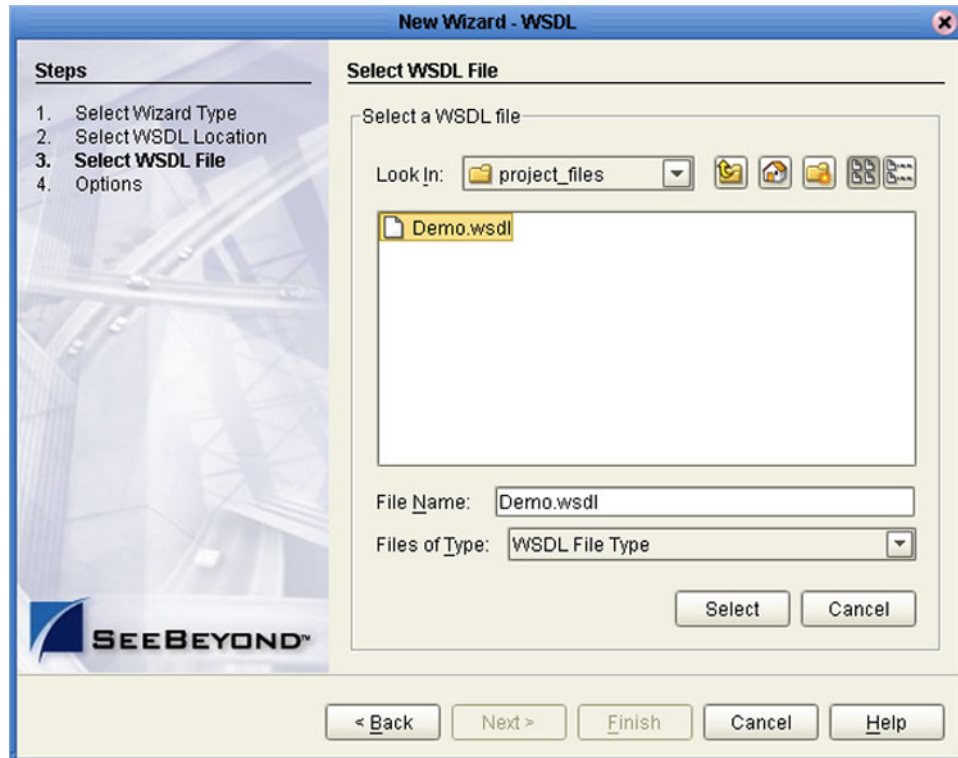
- 2 Click **Next** to display the *Select WSDL File Location* dialog, shown in Figure 98

Figure 98 WSDL Wizard: Select WSDL Location



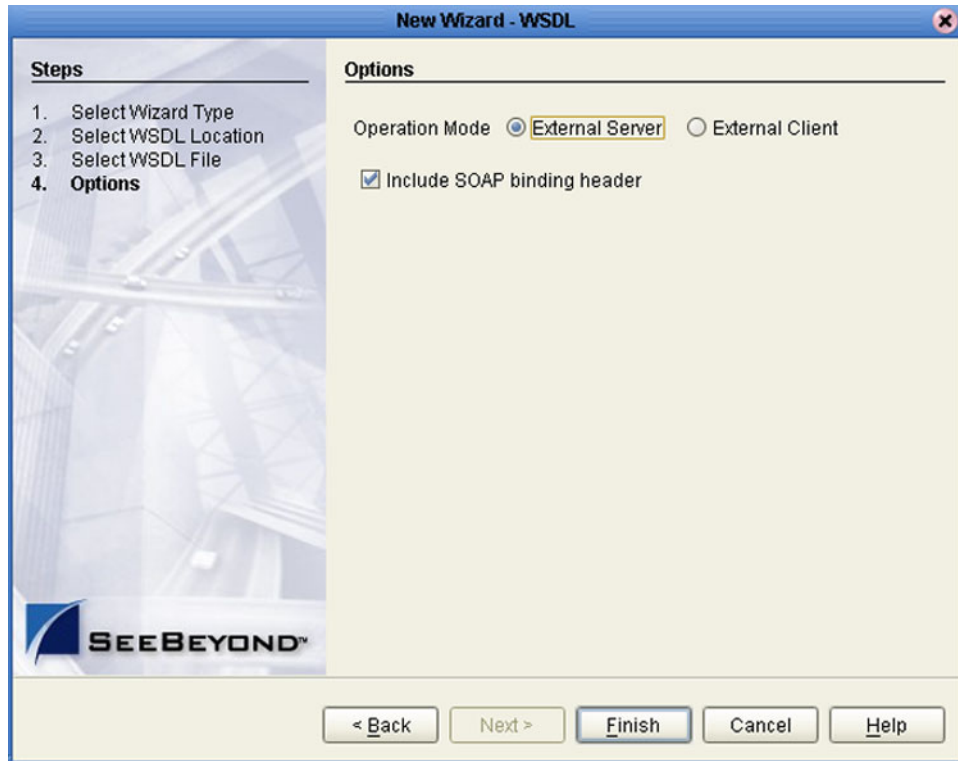
- 3 In the *Select WSDL Location* dialog, select **File System** or enter a **URL**, depending upon where your WSDL file is located.
- 4 Click **Next** to display the *Select WSDL File* dialog, shown in Figure 99.

Figure 99 WSDL Wizard: Select WSDL File



- 5 In the *Look In* drop-down list, navigate to the WSDL file or files that you want to use to create the OTD. Click **Select** to add the files to the *List of Selected WSDLs*.
- 6 Click **Next** to display the *Options* dialog, shown in Figure 100.

Figure 100 WSDL Wizard: Select OTD Options



- 7 Select the check boxes next to the OTD options you want to enable:
 - ◆ If you are using a Web client, select **External Server**.
 - ◆ If you are using a Web server, select **External Client**.
 - ◆ To include the SOAP binding header in the WSDL file, select the check box.
- 8 Click **Finish** to add the OTD to the Enterprise Designer with the selected OTD options.

WSDL OTD Structure

The WSDL OTD has the following basic structure:

```
Root Node
  PortType_XXX
    Operation_XXX
      Input_XXX
      Output_XXX
  PortType_XXX
    Operation_XXX
      Input_XXX
      Output_XXX (and so on)
```

Where **XXX** is the name for each element given in the original WSDL file.

WSDL Operation Elements

To tie your messages together as a request-response pair corresponding to a method call, you must define operations using the WSDL **<operation>** element. A WSDL operation specifies which message is the *input* and which message is the *output*.

Inside the WSDL file's **<operation>** element, you specify your **<input>** and **<output>** elements. Each element refers to the corresponding message by its fully qualified name. The collection of all WSDL operations (that is, methods) exposed by your service is called a **portType** and is defined using the WSDL **<portType>** element.

The **<operation>** element is a child of **<portType>**. You can name the **<portType>** whatever you want. The port type **name** attribute provides a unique name among all the PortTypes defined within the enclosing WSDL file. Each WSDL operation is named via the **name** attribute.

Each operation within a WSDL OTD (like its WSDL file counterpart) uses one of the following operation modes for communication:

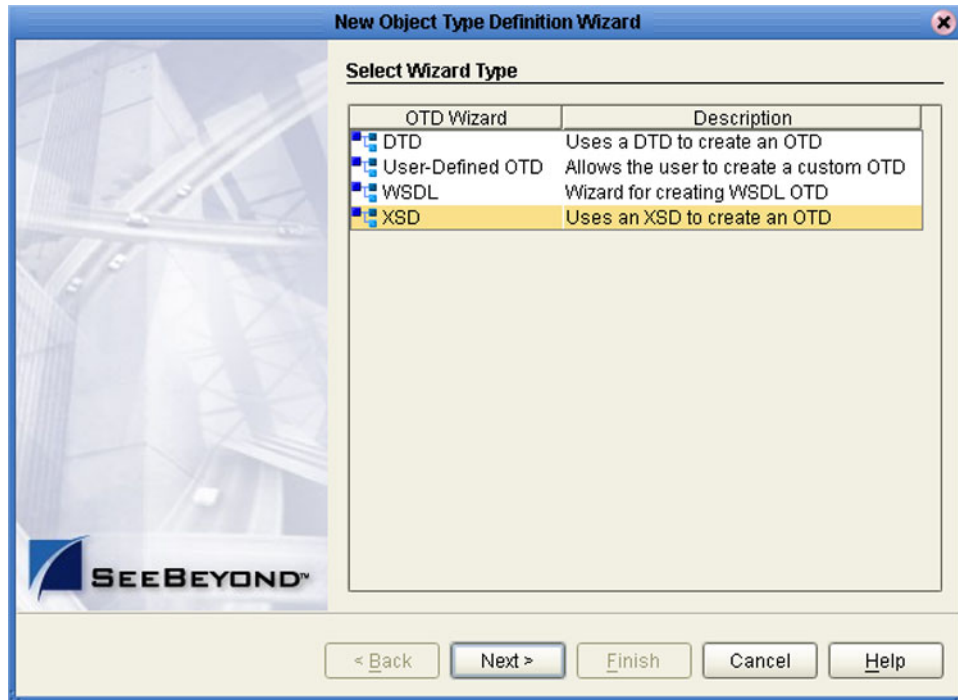
- **One-way:** The server receives a message from the client; also referred to as “fire and forget.”
- **Request-response:** The server receives a message from the client and sends a correlated message back

7.3.3 Using the XSD Wizard

To create an OTD file from an XSD file

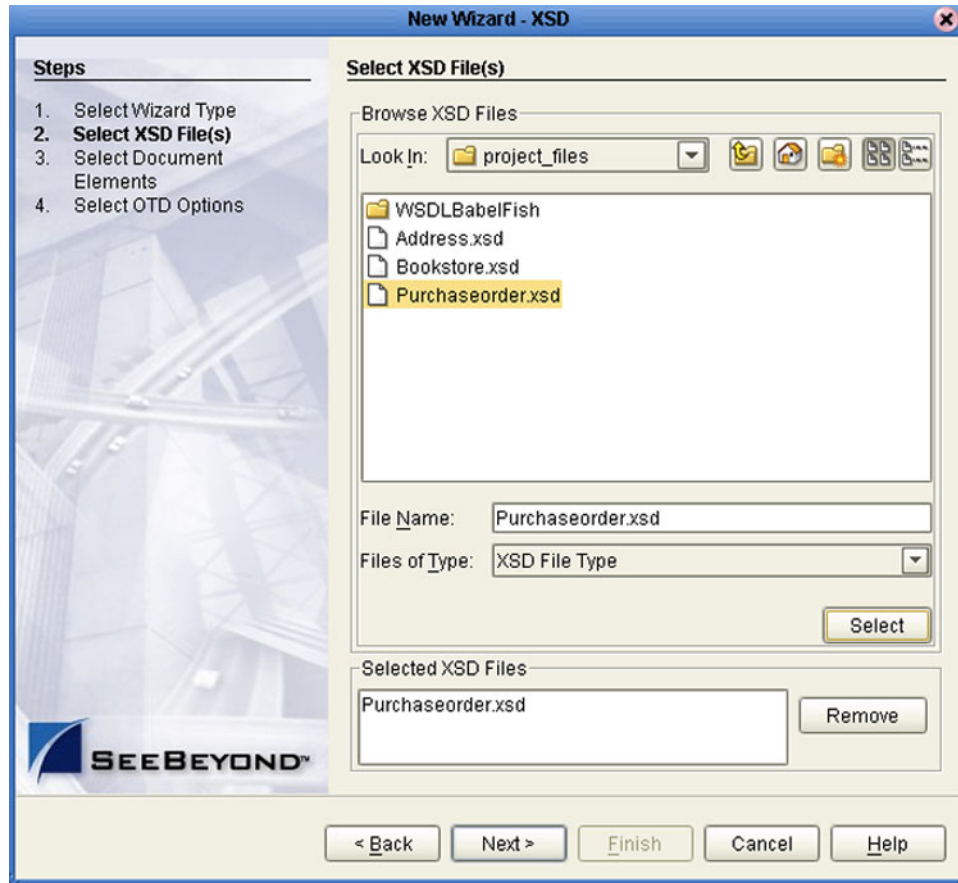
- 1 In the *Select Wizard Type* dialog, select **XSD** from the *OTD Wizard* list (see Figure 101) to create an OTD file from an XSD file.

Figure 101 OTD Wizard Selection: XSD Wizard



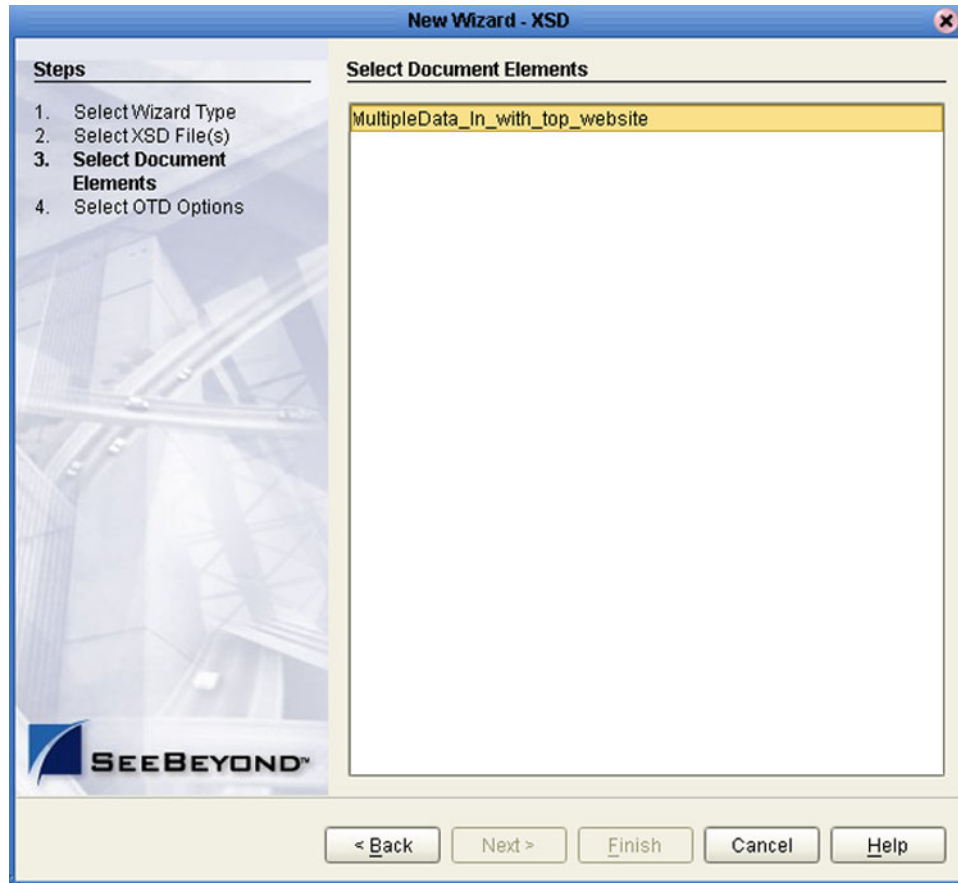
- 2 Click **Next** to display the Select XSD File(s) dialog box, shown in Figure 102.

Figure 102 XSD Wizard: Select XSD File(s)



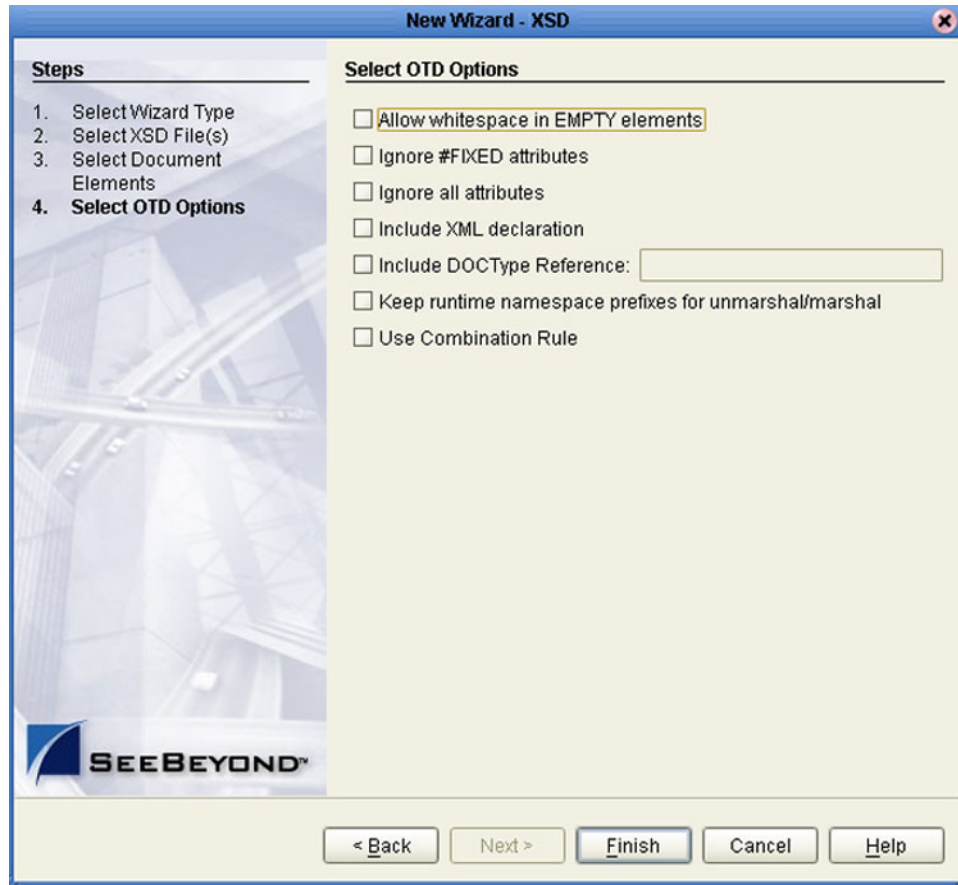
- 3 In the *Look In* drop-down list, navigate to the XSD file or files that you want to use to create the OTD. Click **Select** to add the files to the *List of Selected XSDs*.
- 4 Click **Next** to display the *Select Document Elements* dialog box, shown in Figure 95.

Figure 103 Select Document Elements Dialog Box



- 5 Select the elements of the document that you want to include in the OTD.
- 6 Click **Next** to display the *Select OTD Options* dialog box, shown in Figure 96.

Figure 104 Select OTD Options Dialog Box



- 7 Select the check boxes next to the OTD options you want to enable, and click **Finish** to add the OTD to the Enterprise Designer with the selected OTD options.

7.4 User-Defined OTDs

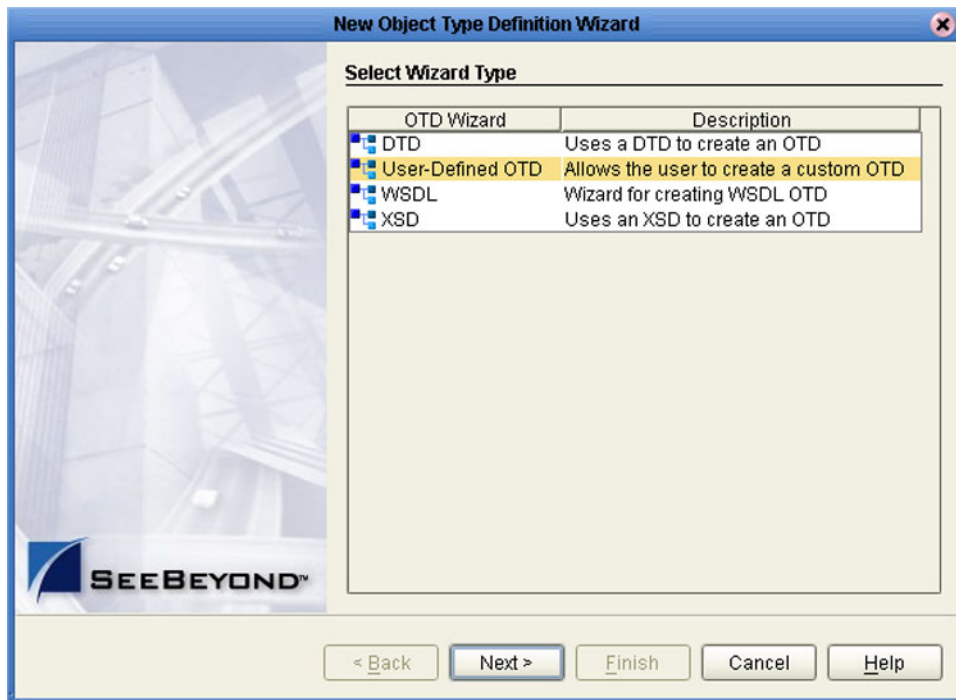
The User-Defined OTD Wizard creates an OTD from your specifications.

7.4.1 Using the User-Defined OTD Wizard

To create a User-Defined OTD

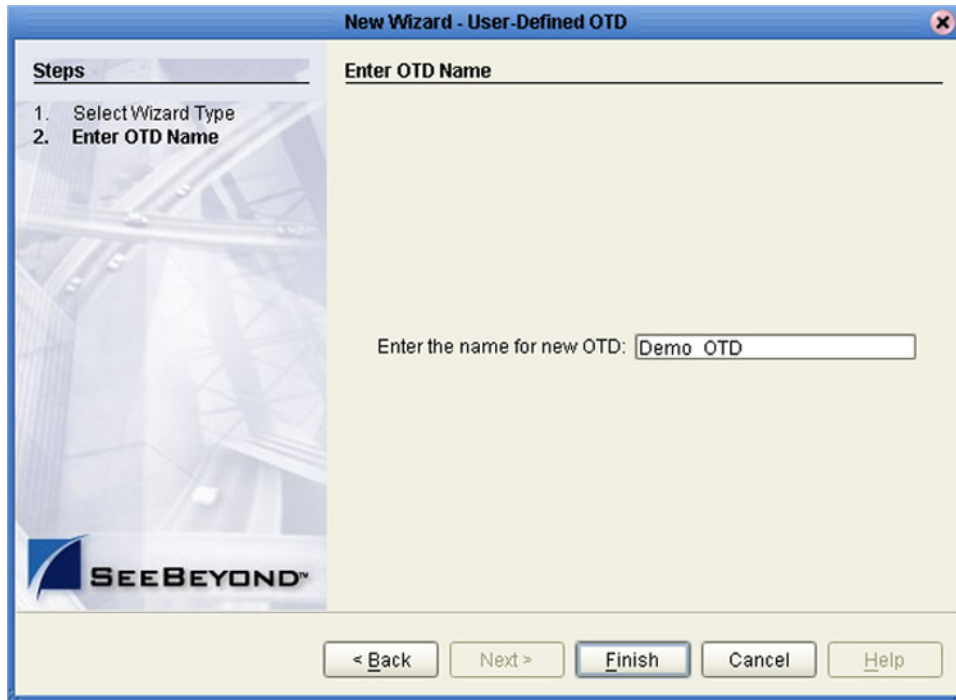
- 1 In the *Select Wizard Type* dialog, select **User-Defined OTD** from the *OTD Wizard* list (see Figure 105) to create an OTD file without using a source file.

Figure 105 OTD Wizard Selection: User-Defined OTD



- 2 Click **Next** to display the *Enter OTD Name* dialog box, shown in Figure 106.

Figure 106 Enter OTD Name



- 3 Enter a name for the OTD into the text box provided, then click **Finish** to add the OTD to the Enterprise Designer.

7.4.2 Editing the OTD Properties

Node Properties

Once a new User-Defined OTD has been created, the properties of the root node will be as shown in Figure 107.

Figure 107 User-Defined OTD Node Properties

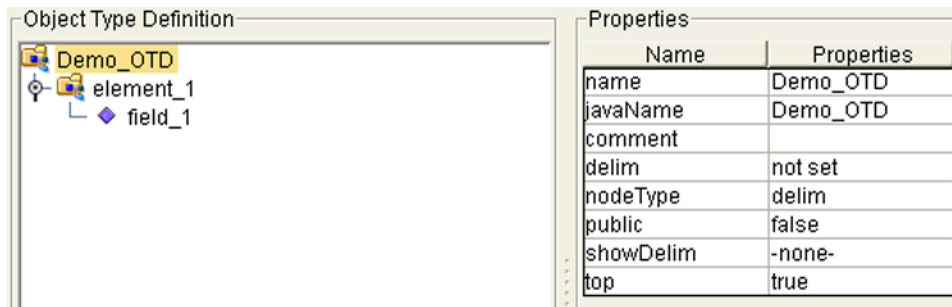


Table 22 Node Properties

Name	Description
name	Node display name (see The Bean-like Interface on page 120).
javaName	Property accessor basename (see The Bean-like Interface on page 120).
comment	Free-form text (no run-time effect).
delim	Delimiter specification (see Specifying Delimiters on page 143).
nodeType	Governs the marshal/unmarshal format (see Specifying the Node Type on page 142).
public	(for future use, not currently active)
showDelim	If nodeType is delimited,
top	Flag on root node: support marshal/unmarshal (T/F).

Important: Do not modify the javaName property.

Element Properties

The set of properties associated with the element level is shown in Figure 108.

Figure 108 User-Defined OTD Element Properties

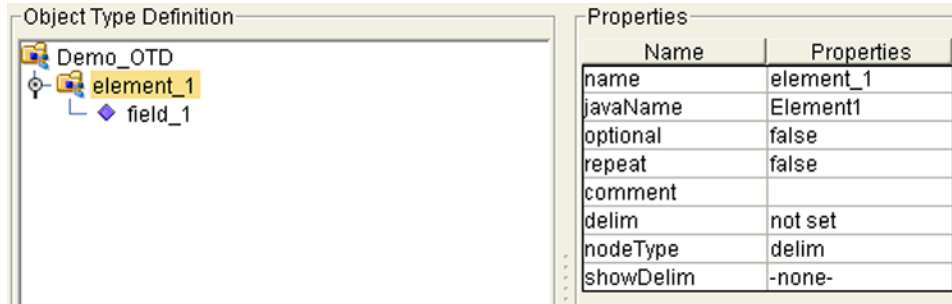


Table 23 Element Properties

Name	Description
name	Element display name.
javaName	Property accessor basename.
optional	Flag: Can the element be absent? (T/F) Not applicable to root, or child of a choice Node.
repeat	Flag: Can the node appear multiple times? (T/F) Not applicable to root, or child of a choice Node.
comment	Free-form text (no run-time effect).
delim	Delimiter specification (see Specifying Delimiters on page 143).
nodeType	Governs the marshal/unmarshal format.
showDelim	If nodeType is delimited,

Important: Do not modify the *javaName* property.

Field Properties

The set of properties associated with the field level is shown in Figure 109.

Figure 109 User-Defined OTD Field Properties

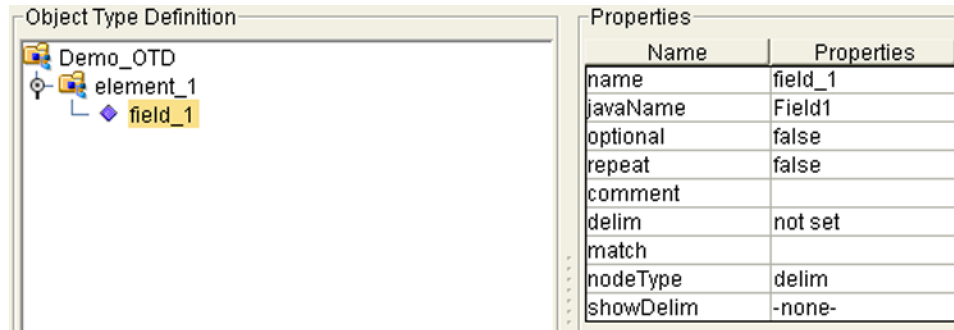


Table 24 Field Properties

Name	Description
name	Field display name.
javaName	Property accessor basename.
optional	Flag: Can the element be absent? (T/F) Not applicable to root, or child of a choice Element.
repeat	Flag: Can the node appear multiple times? (T/F) Not applicable to root, or child of a choice Element.
comment	Free-form text (no run-time effect).
delim	Delimiter specification (see Specifying Delimiters on page 143).
match	If nodeType is delimited, performs exact match to the data.
nodeType	Governs the marshal/unmarshal format.
showDelim	If nodeType is delimited,

Important: Do not modify the *javaName* property.

7.4.3 Specifying the Node Type

Double-clicking in the nodeType properties field activates the field for editing. Click the button to display the drop-down menu (see Figure 110). Descriptions of the property options are listed in Table 25.

Figure 110 Node Type Property Options

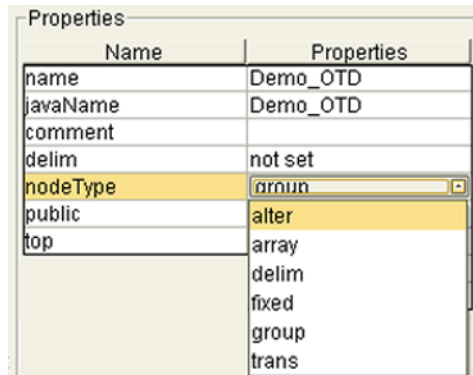


Table 25 Node Type Property Options

Option	Description	Element	Field	Internal
alter	Alter (<i>alternate</i>) selects one child or the other. One child is always present after the unmarshal operation. Applies only to elements.	Yes	No	choice
array	Array is a delimited structure. If repeated, occurrences are separated by the <i>repeat</i> delimiter. The last occurrence may be terminated by a <i>normal</i> delimiter.	Yes	Yes	simple or group
delim	Delim (<i>delimited</i>) structure. If repeated, occurrences are separated by a <i>normal</i> delimiter.	Yes	Yes	simple or group
fixed	Fixed indicates a fixed length, which is specified by non-negative integer (or zero to indicate end of parent node data).	Yes	Yes	simple or group
group	Group provides organizational grouping for purposes such as repetition. Applies only to elements.	Yes	No	group
trans	Trans (<i>transient</i>) appears only in an internal tree as a scratch pad field. It does not appear in external data representation, and can only have <i>trans</i> nodeTypes as children.	Yes	Yes	choice, simple, or group

7.4.4 Specifying Delimiters

Any node can define a set of delimiters to be used in the external data representation for itself and its descendents in the hierarchical data structure. If a node defines a delimiter list, this negates any effect of any ancestor's delimiter list on itself and its descendents. We typically specify the list on the root node.

For example, if you want to parse the following data:

```
a^b|c^d|e
```

you might create a User-Defined OTD as follows:

- demo-otd
 - ♦ element1
 - ♦ field1
 - ♦ field2
 - ♦ element2
 - ♦ field3
 - ♦ field4
 - ♦ field5

The delimiter list for this OTD will be specified on the *demo-otd* element, so that it applies to the entire OTD, and will have two levels:

- Level 1
 - ♦ Delimiter |
- Level 2
 - ♦ Delimiter ^

Level 1's delimiter applies to the two elements and field5, and level 2's delimiter applies to fields 1 through 4.

Delimiter lists can be much more complex than this very simple example. For instance, you can create multiple delimiters of different types at any given level, and you can specify a delimiter list on any node within the OTD—not only the root node as shown in the example. See [Creating a Delimiter List](#) on page 149 for a description of the procedure for creating a Delimiter List.

Delimiter Properties

Delimiters are defined using the Delimiter List Editor (see Figure 111).

Figure 111 Delimiter List Editor

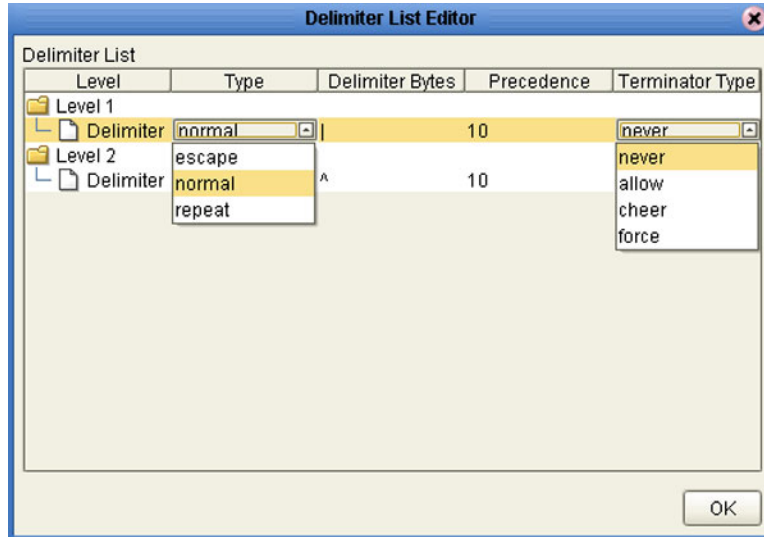


Table 26 Delimiter Properties and Value Options

Property	Option	Description
Level		Child level beneath defining node.
Type	escape	Escape sequence.
	repeat	Array delimiter/separator.
	normal	Terminator.
Delimiter Bytes		Delimiter (single or multiple characters).
Precedence		See Precedence on page 148.
Terminator Type	never	Do not allow on input, do not emit on output (pure separator).
	allow	Allow on input, do not emit on output.
	cheer	Allow on input, always emit on output.
	force	Require on input, always emit on output (pure terminator).

Table 26 Delimiter Properties and Value Options

Property	Option	Description
Optional	never	Do not allow on input, do not emit on output (empty field between delimiters implies zero length data field).
	allow	Skip empty field if present; if absent, do not delimit on output.
	cheer	Skip empty field if present; if absent, do delimit on output.
	force	Require empty, delimited field on input; always delimit on output.

Type Property - Escape Option

An *escape* delimiter is simply a sequence that will be recognized *and ignored* during parsing. Its purpose is to allow the use of escape sequences to embed byte sequences in data that would otherwise be seen as delimiter occurrences.

For example, if there is a normal delimiter "+" at a given level, and we define an escape delimiter "\+", then **aaa+b\+c+ddd** will parse as three fields: **aaa**, **b\+c**, and **ddd**. If the escape delimiter were not defined, the sequence would then parse as four fields: **aaa**, **b**, **c**, and **ddd**.

If there is *only* an escape delimiter on a given level, however, it presents a *no delimiter defined* situation for **delim** and **array** nodes.

The following escape delimiters are allowed.

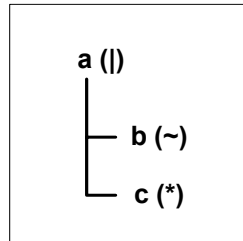
Table 27 Escape Delimiters

Delimiter	Description
\n	Newline
\r	Carriage return
\t	Tab
\\	Backslash

Terminator Type Property

Consider the tree structure shown in Figure 112, where the node **a** has a pipe (|) as its delimiter, the sub-node **b** has a tilde (~) as its delimiter, and sub-node **c** has an asterisk (*) as its delimiter.

Figure 112 Terminal Type Property Example



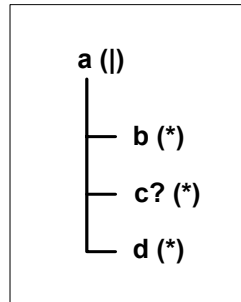
Option	Input	Output
never	c	c
allow	c or c*	c
cheer	c or c*	c*
force	c*	c*

Optional Property

Consider the tree structures shown in Figure 113 and Figure 114, where the node **a** has a pipe (|) as its delimiter, and the sub-nodes **b**, **c**, and **d** all have asterisks (*) as their delimiters.

- **Example 1:** Sub-node **c** is *optional*. (Sub-node **c** and sub-node **d** must have different values for the *match* parameter.)

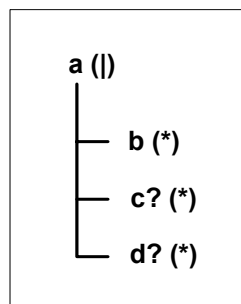
Figure 113 Optional Property (Example 1)



Option	Input	Output
never	b*d 	b*d
allow	b**d 	b*d
cheer	b**d 	b**d
force	b**d 	b**d

- **Example 2:** Both sub-node **c** and sub-node **d** are *optional*.

Figure 114 Optional Property (Example 2)



Option	Input	Output
never	b 	b
allow	b , b* , or b** 	b
cheer	b , b* , or b** 	b**
force	b** 	b**

Precedence

Precedence (see [Figure 119 on page 150](#)) indicates the priority of a certain delimiter, relative to the other delimiters. By default, all delimiters are at precedence 10, which means they are all considered the same; fixed fields are hard-coded at precedence 10. Delimiters on parent nodes are not considered when parsing the child fields; only the child's delimiter (or if it is a fixed field, its length).

Changing the precedence of a delimiter will cause them to be applied to the input data-stream in different ways. For example:

- *root node*
 - ♦ element (type delim, delimiter = "^", repeat)
 - ♦ field1 (type fixed, length = 5)
 - ♦ field2 (type fixed, length = 8, optional)

Although this will parse 'abcde12345678^zyxvuABCDEFGH', it will *not* parse the text 'abcde^zyxvuABCDEFGH' even though the second fixed field is optional. The reason is that the element's delimiter is ignored within the fixed field because they have the same precedence. If you want the element's delimiter to be examined within the fixed field data, you must change its precedence, for example:

- *root node*
 - ♦ element (type delim, delimiter = "^", repeat, **precedence = 11**)
 - ♦ field1 (type fixed, length = 5)
 - ♦ field2 (type fixed, length = 8, optional)

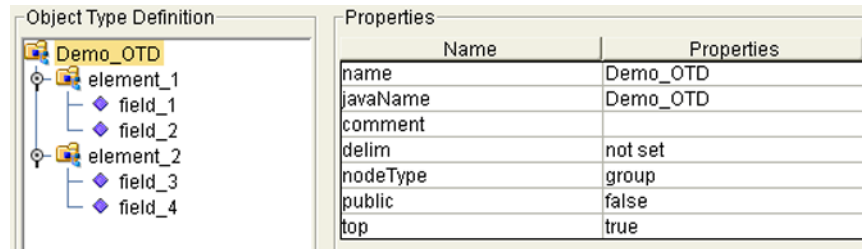
This will successfully parse the text 'abcde^zyxvuABCDEFGH'.

7.4.5 Creating a Delimiter List

To create a delimiter list

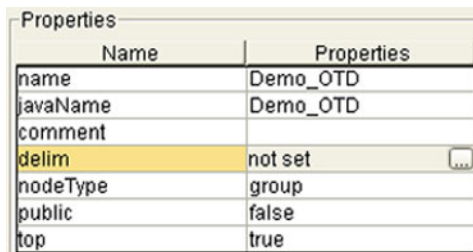
- 1 Create the structure of this example OTD in the User-Defined OTD editor as shown in Figure 115.

Figure 115 Example User-Defined OTD



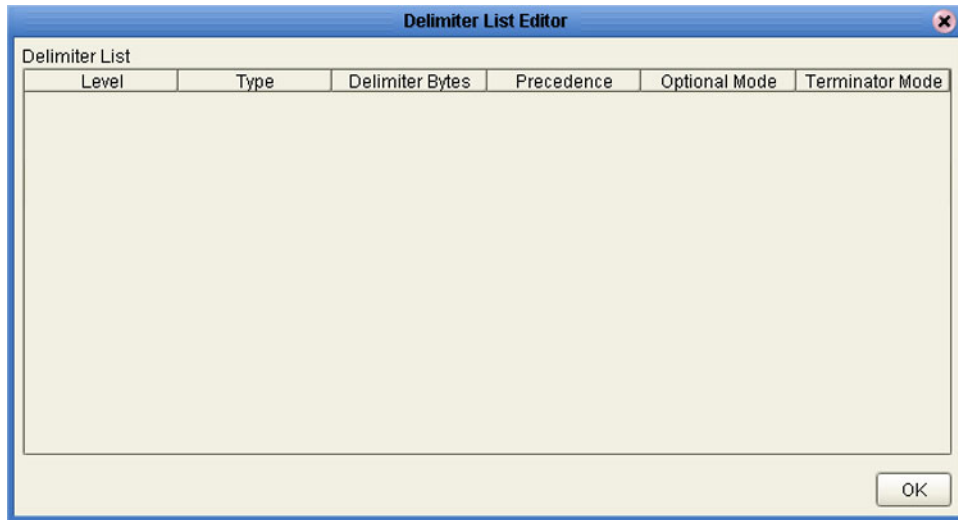
- 2 Select the node to which the delimiter list will apply.
- 3 Double-click on the value of the **delim** property to activate the property field for editing (see Figure 116). Initially it appears as *not set*.

Figure 116 Activate Field



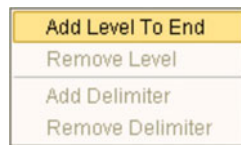
- 4 Click on the ellipsis (...) button to display the Delimiter List Editor, which is blank when the delimiter list is *not set* (see Figure 117).

Figure 117 Delimiter List Editor (1)



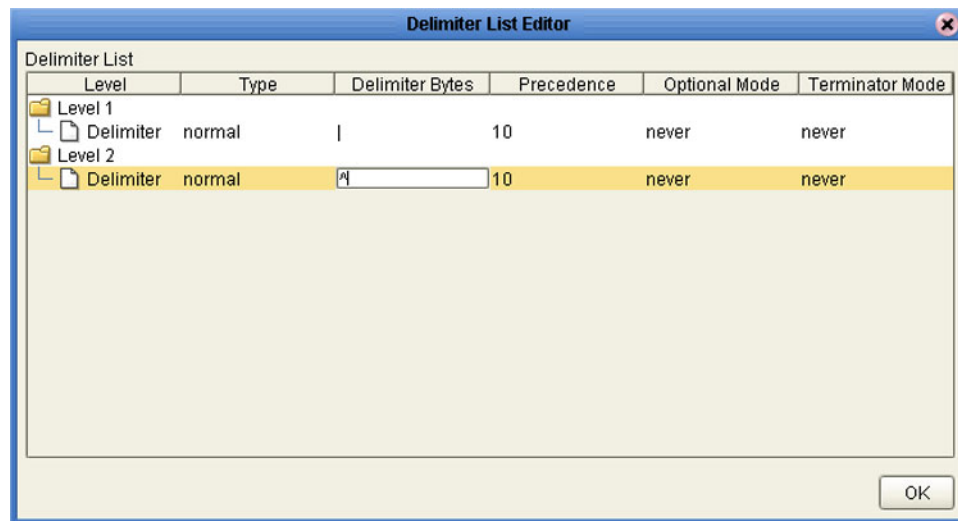
- 5 Right-click anywhere within the table area to invoke the context menu (see Figure 118).

Figure 118 Context Menu



- 6 Levels can be added and removed in the delimiter list, and delimiters can be added or removed from under a given level, using the context menu options. Using the **Add Level To End** option, you can create the delimiter list shown in Figure 119.

Figure 119 Delimiter List Editor (2)



- Click the **OK** button to complete the edit. The **delim** property's value will now appear as *specified*, as shown in Figure 120.

Figure 120 Delimiter Specified

repeat	raise
comment	
delim	specified
nodeType	group
nonRootType	

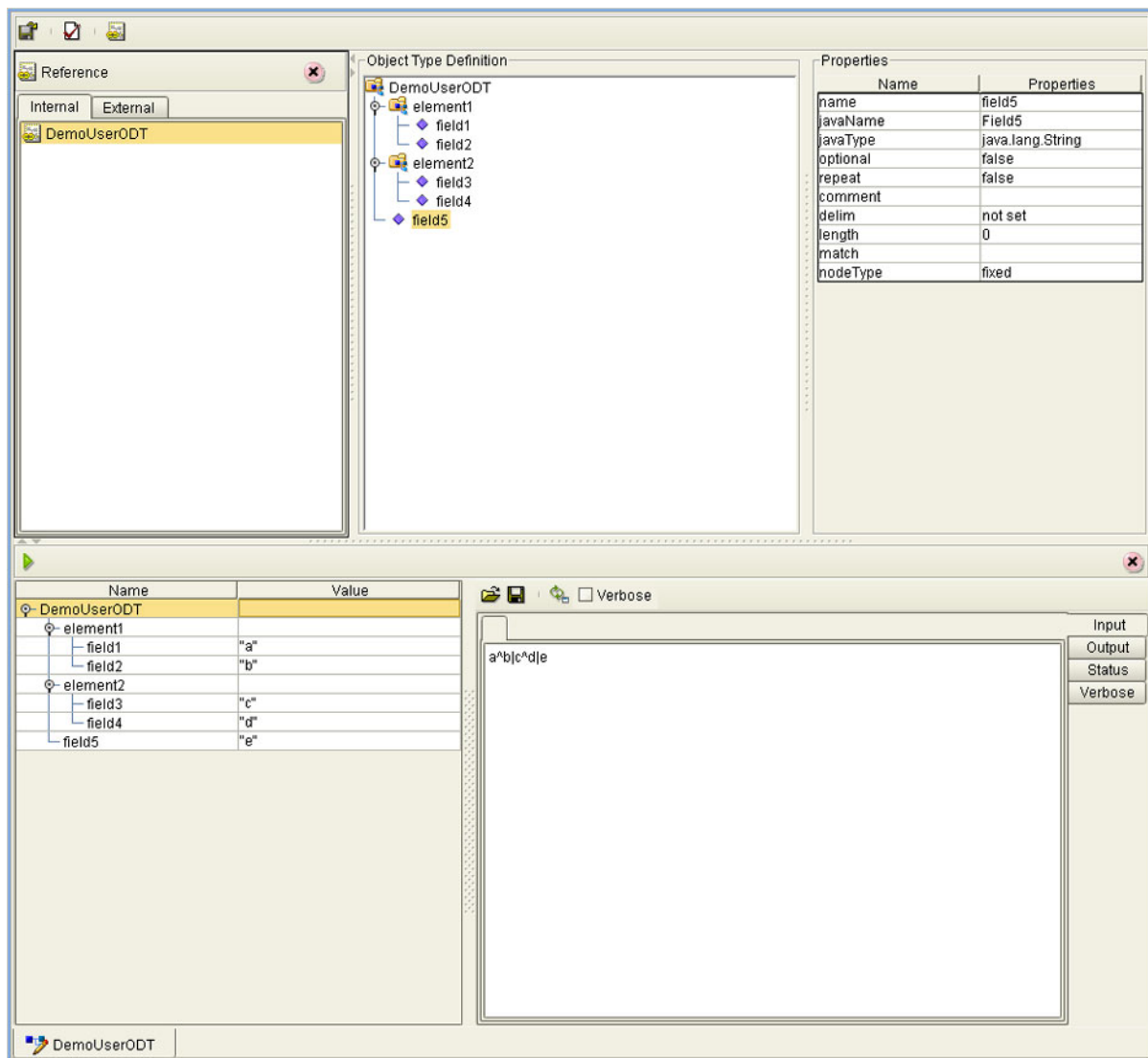
- Test the OTD by invoking the OTD Tester, as described in [Using the OTD Tester](#) on page 155. Entering the data `a^b | c^d | e` will result in a successful parse, as shown in Figure 121.

7.5 Using the OTD Editor

After you create an OTD file using the OTD Wizard, the OTD Editor appears in the editor panel of the Enterprise Designer, as shown in Figure 121. You can also invoke the OTD Editor by selecting **Open** in the context menu for an existing OTD in the Project Explorer. OTDs are saved to the Project automatically.

Important: If you delete an OTD in the Project Explorer, any Collaboration Definitions that have been built using that OTD will be affected. It is recommended that you run the Impact Analyzer before attempting to delete any OTDs (see [Impact Analyzer](#) on page 69).









Figure 121 OTD Editor



Major features of the OTD Editor interface are:

- **Reference**
This area contains internal and external templates for the OTD file.
- **Object Type Definition**
This area displays each field and element included in the OTD file.
- **Properties**
This area displays details about the OTD file or field selected in the *Object Type Definition* list.
- **Tester**
This area displays in the bottom part of the window when you click **Tester**. Use this area to perform tests on the contents of the OTD.
- **Toolbars**
Several toolbars appear in the OTD Editor, containing icons as described in Table 28.

Table 28 OTD Editor Toolbar Icons

Icon	Command	Function
	Save as New Name in Repository	Saves current OTD under a new name in the Repository.
	Tester	Displays/refreshes the Tester area.
	Toggle Reference Tab Panel	Displays/hides the Reference area.
	Sort by Name	Sorts list alphabetically by name.
	Run Tester	Runs the tester with the entered values.
	Open	Displays file browser.
	Save	Saves displayed file.
	Refresh	Repopulates the OTD object elements with the values from the data display panel.

7.5.1 Node Management

The OTD Editor allows you to:

- **Add** nodes and elements to an OTD.
- **Delete** nodes and elements from an OTD.

When a node is *deleted*, both the node and its associated 'children' (data elements) are deleted.

- **Prune** nodes in an OTD.

When a node is *pruned*, only its associated 'children' (data elements) are deleted, while the node itself is preserved. Pruning can only be performed on nodes.

These commands are accessed from the node context menu.

7.6 Using the OTD Tester

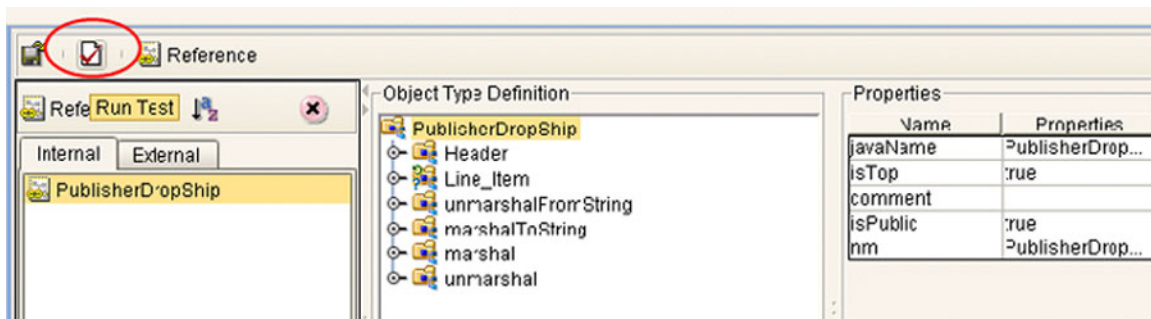
The OTD tester provides a facility to verify the correctness of OTDs, for example to:

- Prevent data errors at runtime.
- Verify that all required data elements are available.
- Verify that all used data formats are correct.

To use the OTD tester

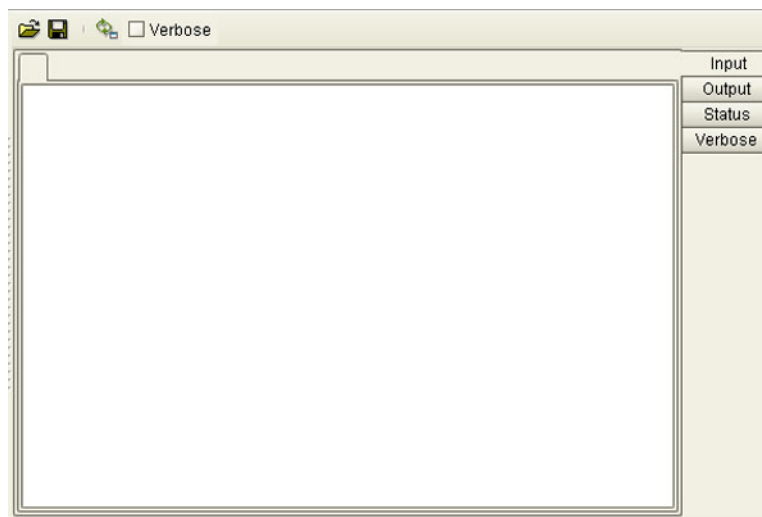
- 1 Open or create an OTD.
- 2 Click the **Tester** icon (see Figure 122).

Figure 122 OTD Tester



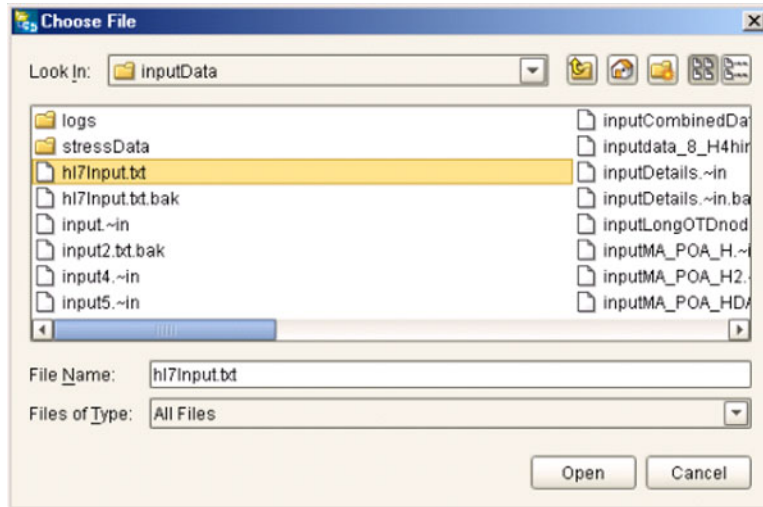
A test panel will appear below the OTD detail area of the editor. Note that there are four data display modes, selectable by tabs (see Figure 123). The Input tab is selected by default.

Figure 123 Test Panel Data Display



- 3 You can provide the input test data either by selecting a data file (see Figure 124), or by entering the data manually.

Figure 124 Select Data File



- 4 Click the **Run Tester** icon (green arrow) to test the selected OTD.
- 5 Verify the output by checking the values for each element for correctness (see Figure 125).

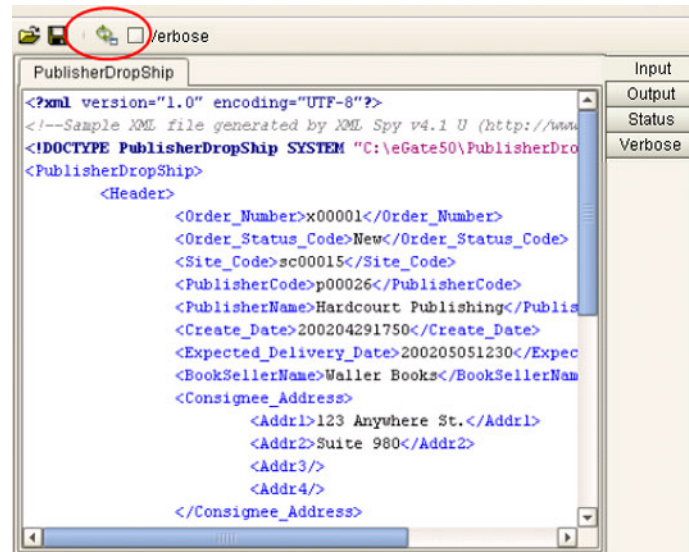
Figure 125 Object Elements and Values

Name	Value
☐ PublisherDropShip	
☐ header	
- name	""
- order_Number	"x00001"
- order_Status_Code	"New"
- site_Code	"sc00015"
- publisherCode	"p00026"
- publisherName	"Hardcourt Publi..."
- create_Date	"200204291750"
- expected_Delivery	"200205051230"
- bookSellerName	"Waller Books"
☐ consignee_Address	
- bom_type	""
- gl_entity	"GLN"
☐ terms	
☐ line_Item	
- length	1
☐ [0]	
- value	"500"
- counter	"0"
- itemCode	"ISBN000139298"
- itemDescription	"King James Bib..."
- qty	"100"
- cost	"5.00"

- 6 You can save your input test data to a file for re-use by selecting the **Input** data display and clicking the **Save** icon.

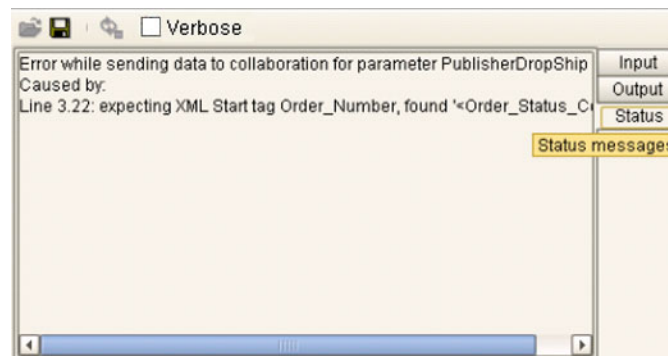
- 7 You can also change your test data in the Input data display, then re-test the OTD by clicking the **Refresh** icon (see Figure 126) to repopulate your OTD object elements with the new values.

Figure 126 Data Display: Refresh Icon



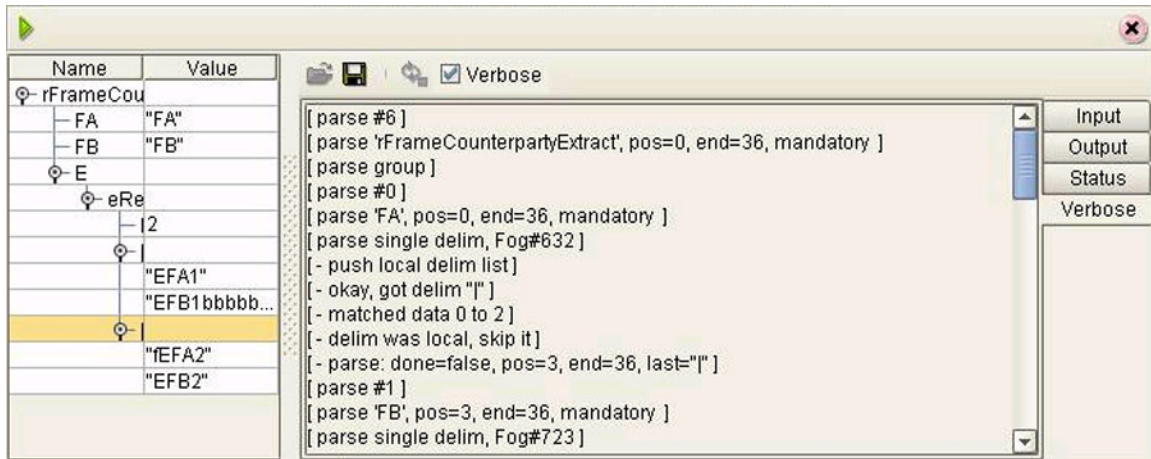
- 8 If there are errors in your input data, the **Status** data display is automatically invoked, showing the appropriate error messages (see Figure 127).

Figure 127 Status Data Display



- 9 For selected User-Defined OTDs, the **Verbose** option provides a trace of parsing actions during the *unmarshal* process to aid in debugging the OTD structure. When this option is available, the OTD will activate the **Verbose** check box. Selecting the **Verbose** check box causes parsing information to appear on the **Verbose** data display (see Figure 128). The format and content of the data display are OTD-specific.

Figure 128 Verbose Data Display



Java Collaboration Definitions

This chapter describes the process for creating Java Collaboration Definitions.

8.1 Overview

Collaborations use Collaboration Definitions to define how data should be routed between Project components. Collaborations also define how databases should be queried in response to requests and how APIs to one or more applications should be invoked. Collaborations are used when data translation is required.

The Enterprise Designer includes two tools, the Java Collaboration Wizard and Java Collaboration Editor, that are used to create and customize your Java Collaboration Definitions. You must have OTDs available to use as the foundation for creating a Java Collaboration Definition. See [Object Type Definitions](#) on page 120 for more details.

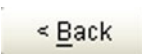






Important: *If you delete an OTD in the Project Explorer, any Java Collaboration Definitions that have been built using that OTD will be affected. It is recommended that you run the Impact Analyzer before attempting to delete any OTDs (see [Impact Analyzer](#) on page 69).*

Note: *Any changes made to the names of the Collaborations should be done when the Collaboration is created. If the name is changed later, the Collaboration should be opened, regenerated if applicable, and saved again. For safe measure, this should also be done before creating the Connectivity Map and Deployment Profile.*

8.2 Using the Java Collaboration Wizard

The Java Collaboration Wizard guides you through the initial phases of creating a Java Collaboration Definition, and then invokes the Java Collaboration Editor. The user interface is highly self-explanatory, but details of the navigation buttons are listed in Table 29 for your reference.

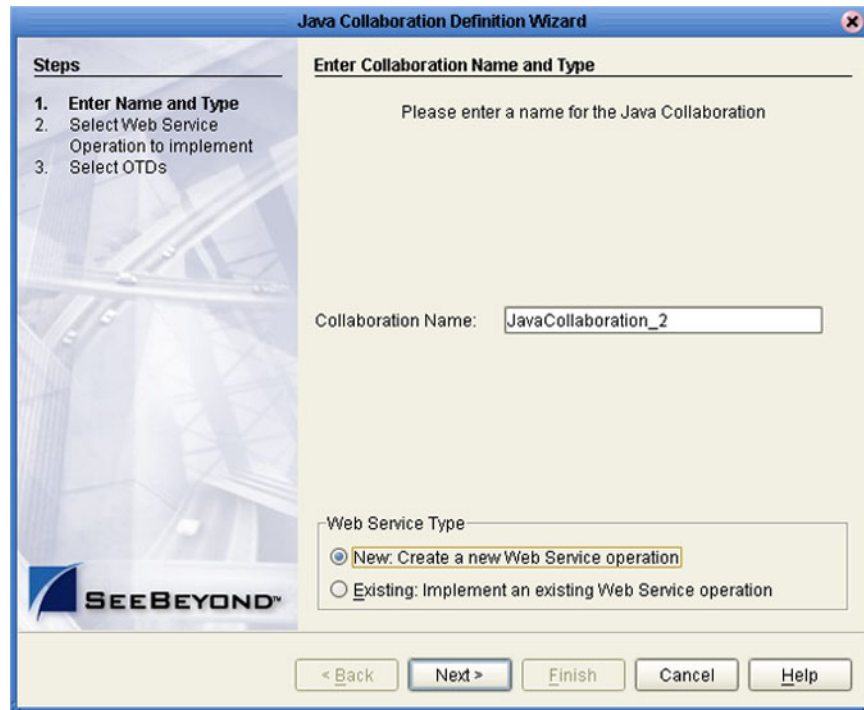
Table 29 Wizard Navigation Buttons

Button	Function
	Returns to the previous step in the wizard. This button is disabled on the first step.
	Goes to the next step in the wizard. This button is disabled on the last step.
	Saves all Collaboration Definition settings and closes the wizard. This button is only enabled on the last step.
	Closes the wizard without saving the Collaboration Definition.
	Displays the online help documentation for the Collaboration Definition Wizard dialog box.
	Adds a selected Object Type Definition to the Collaboration Definition.
	Removes a selected Object Type Definition from the Collaboration Definition.

8.2.1 Creating a Java Collaboration Definition

- 1 Right-click on a Project in the Enterprise Explorer to display the Project context menu.
- 2 Select **New Java Collaboration Definition** to invoke the Java Collaboration Wizard.
- 3 Enter a **Name** for your Collaboration, as shown in Figure 129.

Figure 129 Initial Wizard Dialog



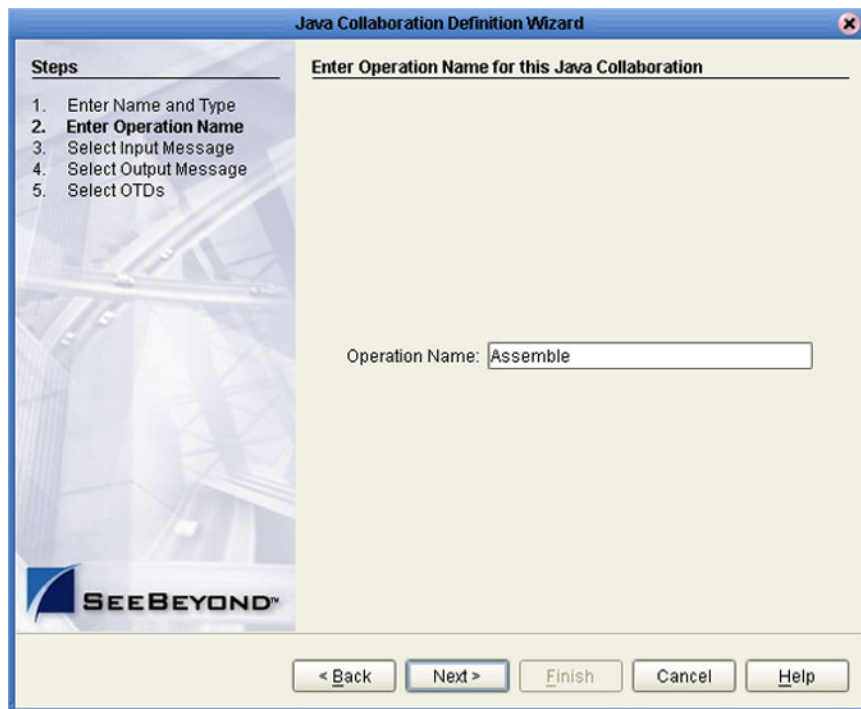
- 4 Select a Web service, which can be either:
 - A New Web Service.
 - An Existing Web Service (for example, an eInsight process or an OTD).
- 5 Click **Next** to proceed to the next Wizard dialog, which is dependent upon your Web Service selection.

New Web Service

If you selected a New Web Service, you will be presented with the following set of Wizard dialogs.

- 1 Enter an operation name, as shown in Figure 130. This will become the *method* that can be used to invoke the Java Collaboration as a Web service.

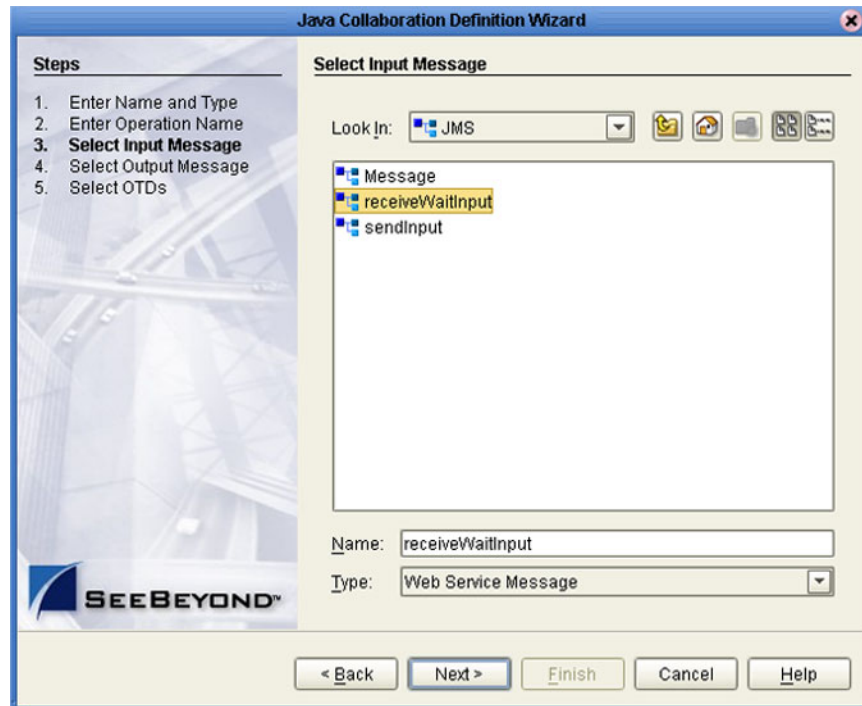
Figure 130 New Web Service: Operation Name



- 2 Click **Next** to proceed to the next Wizard dialog.

- 3 Select the input Web service message, as shown in Figure 131.

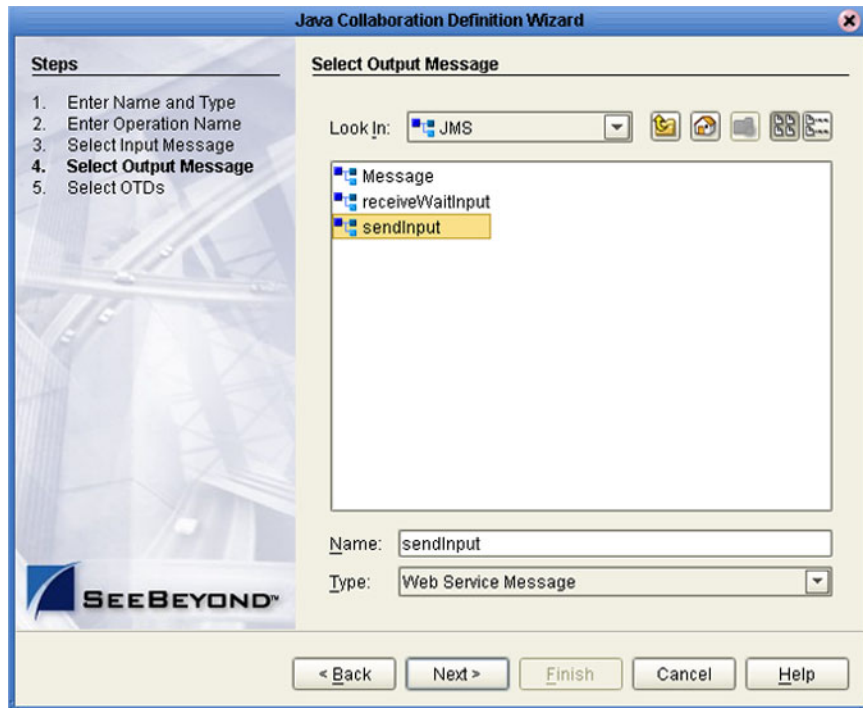
Figure 131 New Web Service: Input Message



- 4 Click Next to proceed to the next Wizard dialog.

- 5 Select the output Web service message, as shown in Figure 132.

Figure 132 New Web Service: Output Message

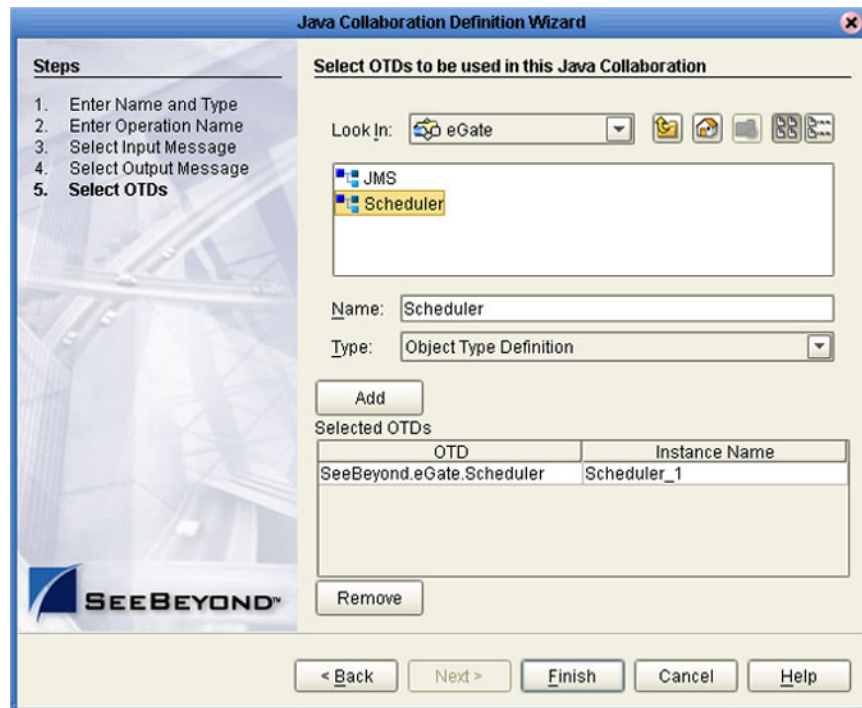


- 6 Click Next to proceed to the next Wizard dialog.

- 7 Select an auxiliary OTD, as shown in Figure 133. This step is optional, and is intended to support additional functionality such as a database lookup.

Note: Use caution here, since you may already have OTDs selected in the preceding steps.

Figure 133 New Web Service: Auxiliary OTD



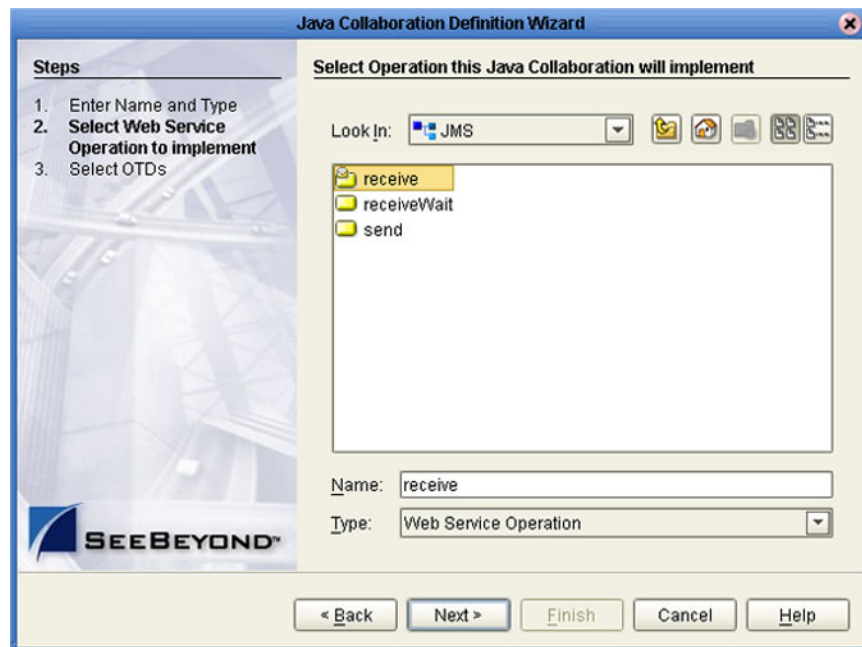
- 8 Click **Add** to add the OTD to the Collaboration Definition.
- 9 Click **Next** to proceed to the Java Collaboration Editor.

Existing Web Service

If you selected an Existing Web Service, you will be presented with the following set of Wizard dialogs.

- 1 Select a Web service operation, which can be either:
 - An installed ICAN Web Service (for example, a JMS *receive* Web service - see Figure 134).
 - A custom Web Service (for example, something that has been created in an eGate Project).

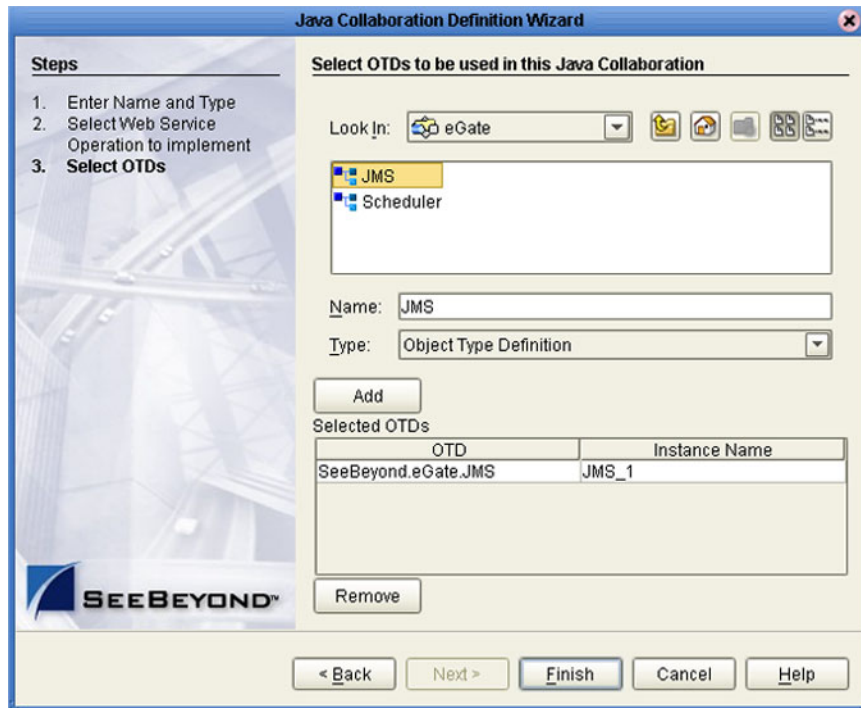
Figure 134 Existing Web Service: Select Operation



- 2 Click **Next** to proceed to the next Wizard dialog.

- 3 Select an OTD, as shown in Figure 135.

Figure 135 Existing Web Service: Select OTD

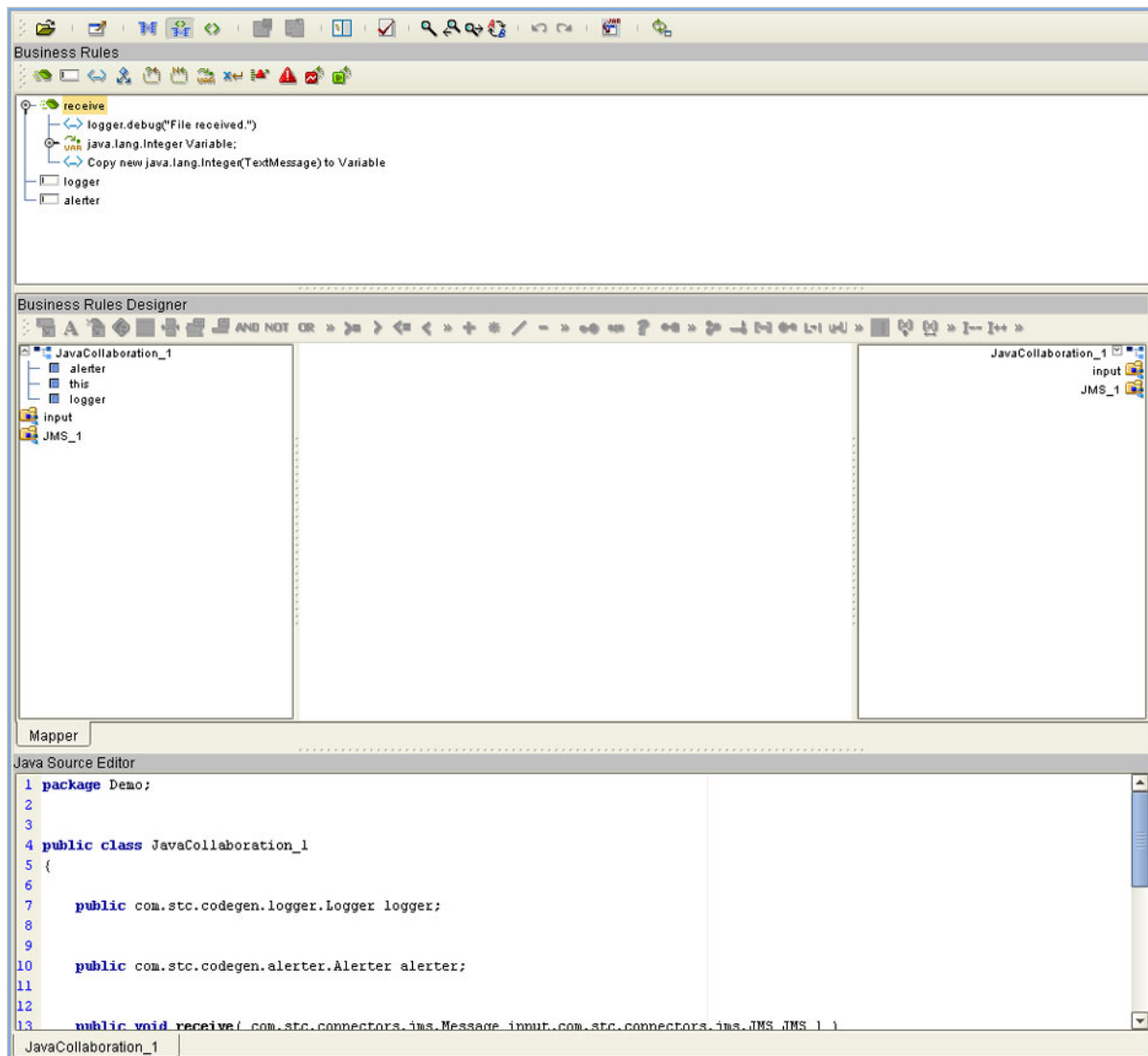


- 4 Click **Add** to add the OTD to the Collaboration Definition.
- 5 Click **Next** to proceed to the Java Collaboration Editor.

8.3 Using the Java Collaboration Editor

After you create a Java Collaboration Definition file using the Java Collaboration Definition Wizard, the Java Collaboration Editor appears in the editor panel of the Enterprise Designer (see Figure 136).

Figure 136 Java Collaboration Editor



The features of the Editor are described in the following sections:

- [Java Toolbar Icons](#) on page 169.
- [Business Rules Editor](#) on page 171.
- [Business Rules Designer](#) on page 174.
- [Java Source Editor](#) on page 180.

8.3.1 Java Toolbar Icons

Table 30 Java Toolbar Icons












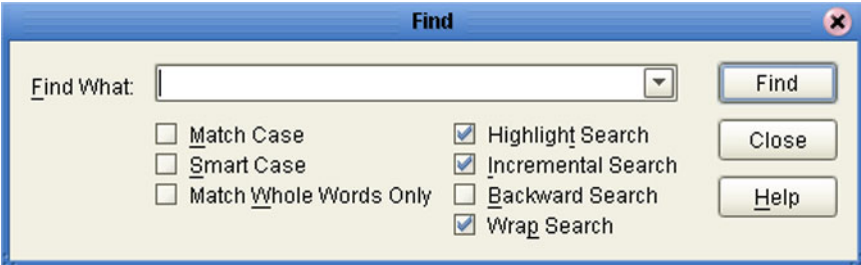








Icon	Command	Function
	Import a Local File	Displays the Open dialog box, which you can use to locate and select a Java Collaboration Definition to import. When you import a file, any previously generated code or rules are deleted. The imported code does not get appended to the existing Collaboration Rules.
	Export Java Rule to a Local File	Displays the Save dialog box, which you can use to save the selected Java Collaboration Definition to a file.
	Standard Mode	Displays the Business Rules Editor and Business Rules Designer only (default setting).
	Advanced Mode	Displays the Business Rules Editor, Business Rules Designer, and Java Source Editor.
	Source Code Mode	Displays the Business Rules and Java Source Editors only.
	Commit Changes	Saves changes made in the Business Rules or Java Source Editors. Re-enables the Business Rules Editor and Business Rules Designer after using the Java Source Editor.
	Roll Back Changes	Cancels changes made in the Business Rules or Java Source Editors and returns the Java Collaboration Definition to the last saved state.
	Business Rules on Left	Changes the editor layout to display the Business Rules panel to the left of the Business Rules Designer. Toggles with <i>Business Rules on Top</i> .
	Business Rules on Top	Changes the editor layout to display the Business Rules panel above the Business Rules Designer (default setting). Toggles with <i>Business Rules on Left</i> .
	Validate	Verifies that changes made to the Java code are valid.

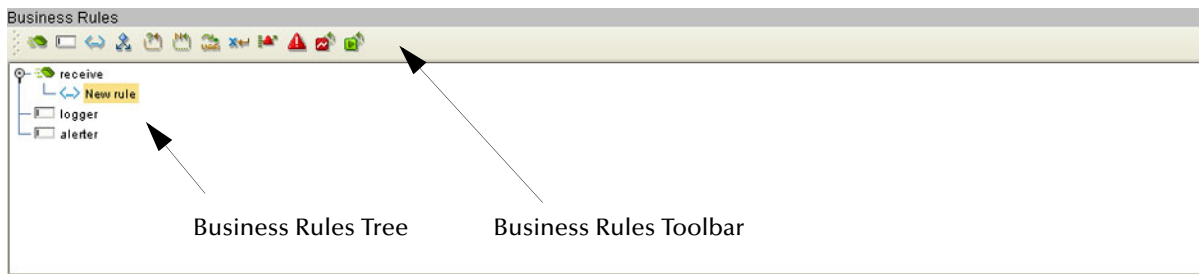
Table 30 Java Toolbar Icons

Icon	Command	Function
	Find	<p>Opens the Find dialog box, where you can enter text to search for in the Java Source Editor.</p> 
	Find Previous	Searches backward for a previous instance of the text entered in the Find dialog box.
	Find Next	Searches forward for the next instance of the text entered in the Find dialog box.
	Replace	<p>Opens the Replace dialog box, where you can enter text to search for, and to replace with, in the Java Source Editor.</p> 
	Undo	Undoes the last operation.
	Redo	Restores the last operation, if undone.
	Import JAR File	Displays the Add/Remove JAR Files dialog box, which you can use to import a .jar file.
	Refresh Collaboration	Refreshes the Collaborations from the Repository.

8.3.2 Business Rules Editor

The Business Rules Editor lists each business rule included with the Java Collaboration Definition. This editor has its own toolbar which you use to add business rules to the Collaboration Definition (see Figure 137). You add rules simply by dragging and dropping the rules into the Business Rules tree. Click the **Commit Changes** button when you are finished.

Figure 137 Business Rules Editor



Note: *The Enterprise Designer automatically disables the Business Rules Editor when you are actively using the Java Source Editor. This is a safety feature that prevents you from inadvertently making contradictory changes. To re-enable the Business Rules Editor, click the **Commit Changes** button in the main toolbar.*

Commands













The **Method** command is used when you want to create a reusable set of instructions inside a Java Collaboration for a specific purpose. Methods are implemented as Java methods, and can be enhanced by means of parameters, return values, and access types (such as public, which means they are also available to other Collaborations).

The **Field** command is used when you want to create a field within a Java Collaboration for some specific purpose, such as storing a temporary variable. As soon as the field has been created, all other rules in the Collaboration are able to read or write from it. A field can be used with the Collaboration (access type = local) or by any other Collaboration (access type = public).

The remaining commands, as represented in the toolbar, are explained in Table 31.

Business Rules Toolbar Icons

Table 31 Business Rules Toolbar Buttons

Icon	Command	Function
	Method	Displays the <i>Add Method</i> dialog box, which you can use to add a new business rule method to the Business Rules tree. The Business Rules tree includes a default method called executeBusinessRules .
	Field	Displays the <i>Create a Parameter</i> dialog box, which you can use to add a new field to the Business Rules tree.
	Rule	Adds a new (empty) rule statement to the Business Rules tree. Use this command as a safeguard regarding positioning.
	If-Then	Adds an if-then statement to the Business Rules tree.
	While	Adds a while statement to the Business Rules tree, starting a specific iteration (repetition) of business rules. You can configure the condition using drag-and-drop when the while statement is selected.
	For	Adds a for statement to the Business Rules tree, starting a specific iteration (repetition) of business rules.
	Local Variable	Displays the <i>Create a Variable</i> dialog box, which you can use to add a local variable to the Business Rules tree.
	Return	Adds a return statement to the Business Rules tree.
	Throw	Adds a throw statement for throwing exceptions.
	Try	Adds a try statement to the Business Rules tree, initiating a number of programming statements that are monitored to see whether they succeed or fail. A finally statement is added automatically; you must right-click on the rule to add a catch statement.
	Break	Breaks out of a business rule while , for , or switch loop.
	Continue	Continues the execution of business rules in a for or while loop by starting the next iteration.

Business Rules Tree

The business rules tree consists of method and field nodes. These are the top level nodes on the tree and cannot be moved. The rule, if-then, while, for, local variable, return, and try statements can be added to a method as subnodes. Statements cannot be added to fields.

The rules for placing subnodes in a method are described in Table 32.

Table 32 Rules for Placement of Subnodes

Subnode	Description
if	Place as a sibling after the last rule in the if-then-else block.
then	Place as a child after the last rule in the then code block.
else	Place as a child after the last rule in the else code block.
while	Place as a sibling after the last rule in the while node, just after the closing parenthesis.
while (code block)	The target node is represented by a code block node under the while node. The source rule becomes a child as the last rule in the block.
for	Place as a sibling after the last rule in the for node, just after the closing parenthesis.
for (code block)	The target node is represented by a code block node under the for node. The source rule becomes a child as the last rule in the block.
local variable	Place as a sibling after the last local variable block.
return	Place as a sibling after the last return block.
try	Place as a sibling after the last rule in the try-catch-finally block.
catch	Place as a child after the last rule in the catch code block.
finally	Place as a child after the last rule in the finally code block.

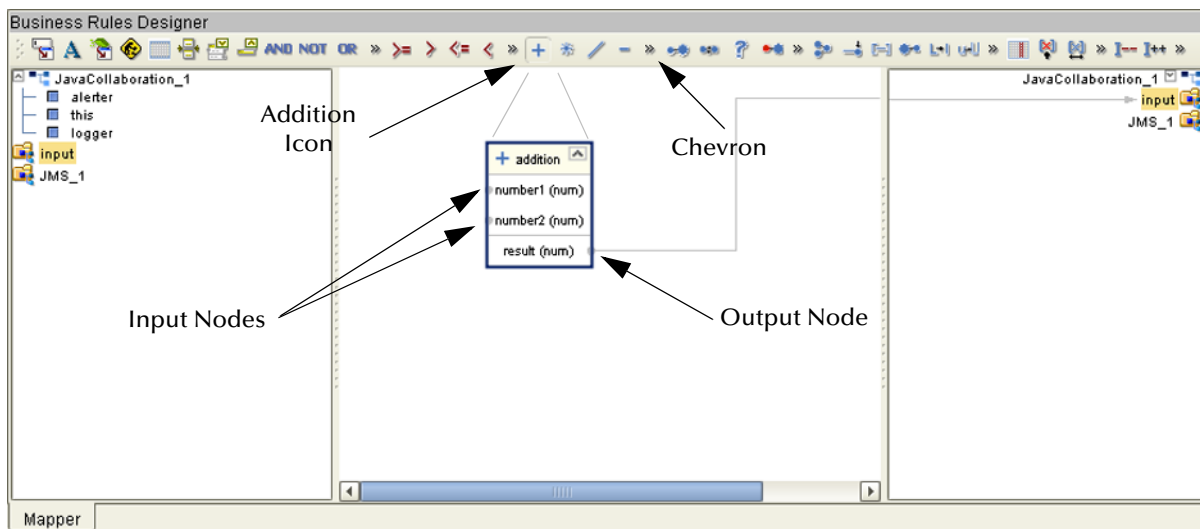
Note: When a subnode is moved to another method, or node, it becomes the first subnode listed under that node.

8.3.3 Business Rules Designer

The Business Rules Designer (see Figure 138) lists each field included in the Java Collaboration Definition. This area has its own toolbar and method palette which you use to map fields and add methods.

Note: The Enterprise Designer automatically disables the Business Rules Designer when you are actively using the Java Source Editor. This is a safety feature that prevents you from inadvertently making contradictory changes. To re-enable the Business Rules Designer, click the **Commit Changes** button in the main toolbar.

Figure 138 Business Rules Designer: Addition Method

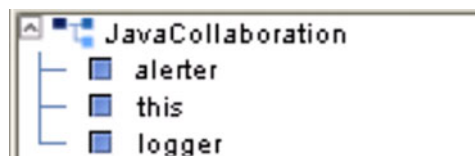


Note: If you encounter an error while using the addition, subtraction, multiplication, or division operators, try manually converting the string to integers.

Method-Access Nodes

When you create a Java Collaboration, three method-access nodes are added automatically in the Business Rules Designer: *alerter*, *this*, and *logger* (see Figure 138 and Figure 139).

Figure 139 Method-Access Nodes



The *alerter* and *logger* nodes allow access to alert and logging methods, as described in [Creating Alerts](#) on page 227 and [Creating Log Entries](#) on page 230, respectively. The *this* node allows you to call other methods defined for *this* Collaboration without having to edit the Java code. To do so, right-click on the node to display the context

menu and click *Select a method to call ...*. The procedure is similar to that used for the other two nodes.

Business Rules Designer Toolbar Icons

Table 33 Business Rules Designer Toolbar Icons


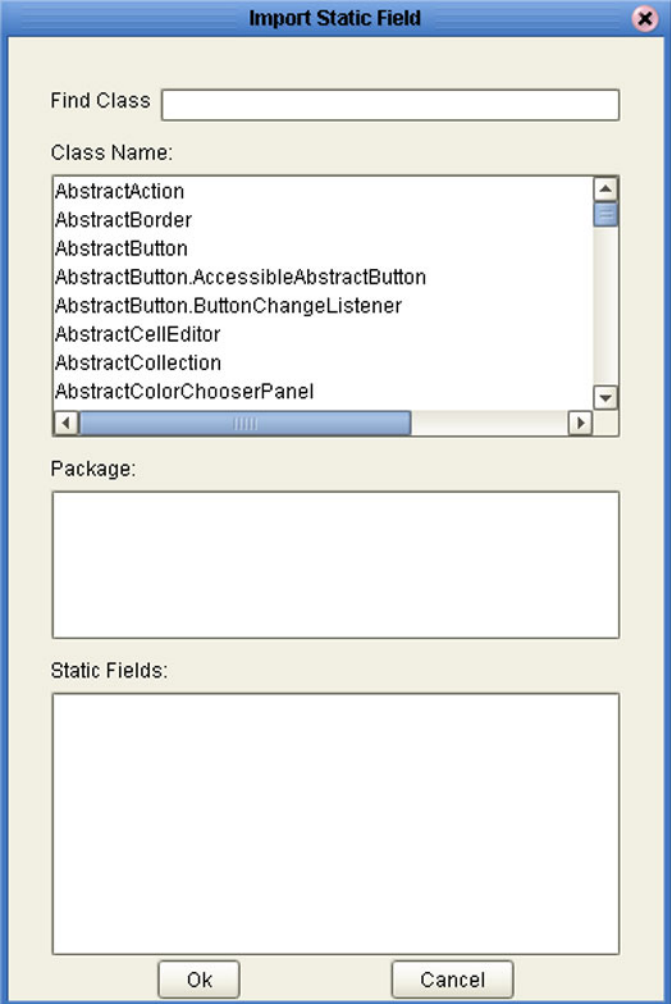
Icon	Command	Function
	Import Static Field	<p>Displays the <i>Import Static Field</i> dialog box, which you can use to select a field to add to the Collaboration Definition.</p> 

Table 33 Business Rules Designer Toolbar Icons


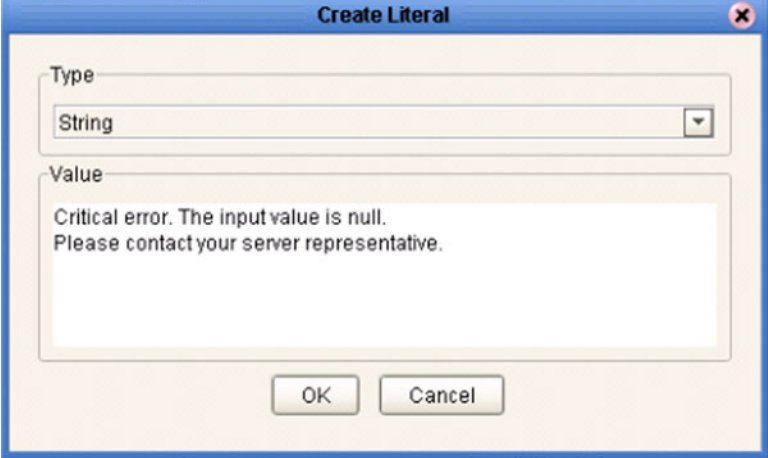

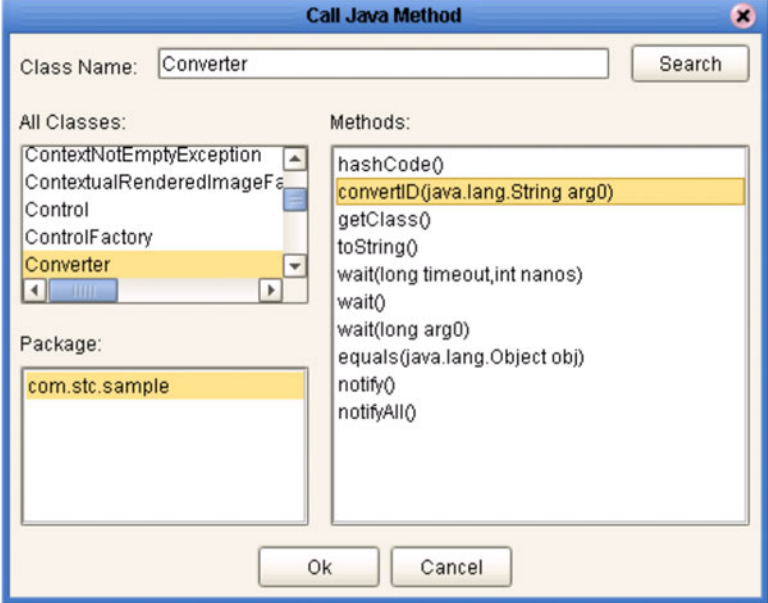
Icon	Command	Function
	<p>Create Literal</p>	<p>Displays the <i>Create Literal</i> dialog box, which you can use to select a literal and enter a value to add to the Collaboration Definition.</p> 
	<p>Call Java Method</p>	<p>Displays the <i>Call Java Method</i> dialog box, which you can use to select a Java method to add to the Collaboration Definition.</p> 

Table 33 Business Rules Designer Toolbar Icons


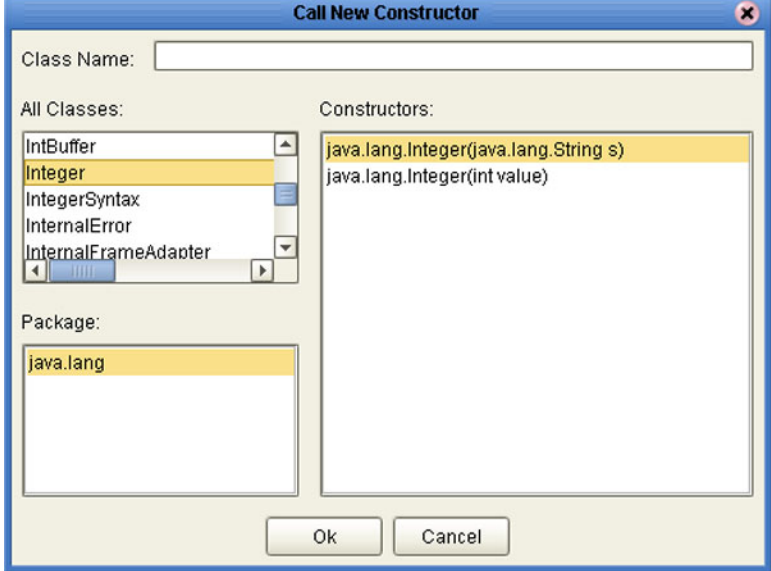

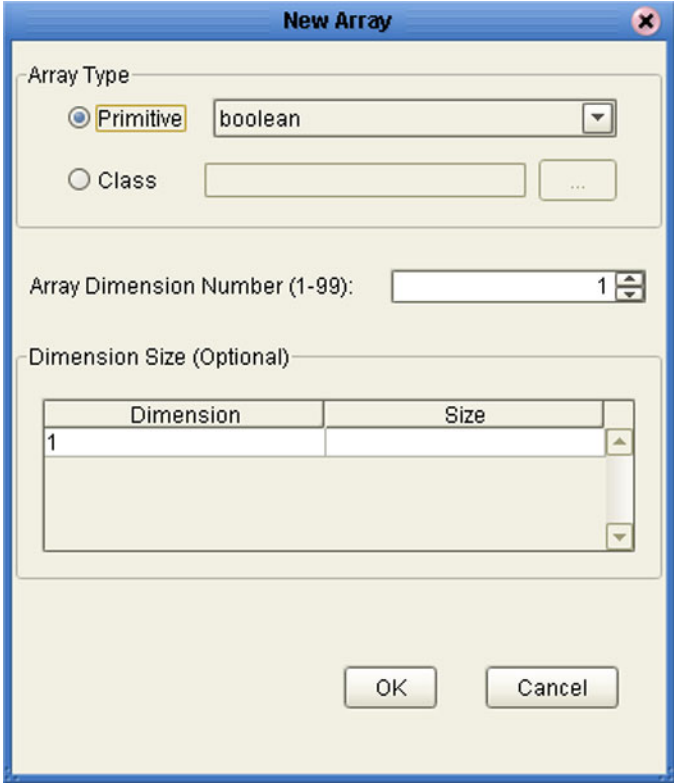



Icon	Command	Function
	<p>Call New Constructor</p>	<p>Displays the <i>Call New Constructor</i> dialog box, which you can use to select a Java class to the Collaboration Definition.</p> 
	<p>New Array</p>	<p>Displays the <i>New Array</i> dialog box, which you can use to add an array to the Collaboration Definition.</p> 

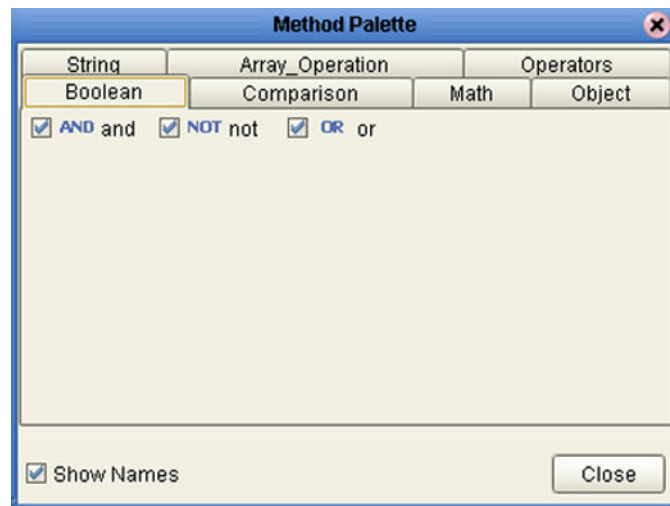
Table 33 Business Rules Designer Toolbar Icons

Icon	Command	Function
	Auto Layout	Updates the layout of fields, literals, and methods that you have added to the middle column of the Business Rules Designer and vertically aligns method boxes.
	Expand All Methods	Displays the title and contents of fields, literals, and methods.
	Collapse All Methods	Displays the title of fields, literals, and methods only.

Java Method Palette

The Java Method Palette includes a series of method icons that you can drag onto the Business Rules Designer mapping area. Click the Chevron (>>) to the right of the method groups to display the dialog box shown in Figure 140. Select a check box to add the method to the toolbar; clear a check box to remove the method from the toolbar. The methods are described in [Java Methods](#) on page 181.

Figure 140 Java Method Palette Dialog Box



Java Method Boxes

The method boxes are placed in the mapping panel of the Business Rules Designer by dragging the corresponding icon from the method palette toolbar. As shown in [Figure 138 on page 174](#), the method boxes typically have input and output nodes that you link to fields in the left and right panels, respectively. The method boxes are expanded by default (see Figure 141); you can collapse them (see Figure 142) by clicking the caret icon in the upper right corner of the box, which can be useful when the mapping area becomes crowded. Some boxes expand further as needed to provide additional arguments.

Figure 141 Expanded Method Box

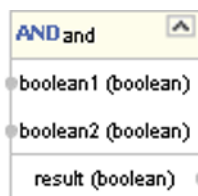


Figure 142 Collapsed Method Box



8.3.4 Java Source Editor

The Java Source Editor (see Figure 143) displays the actual Java code as you map the Collaboration Definition. At your option, you can enter and edit the raw Java code for the Collaboration Definition. Click the **Commit Changes** button in the main toolbar when you are finished.

Note: The Java Source Editor is not displayed by default—you must click the Source Code Mode or Advanced Mode toolbar buttons to display it.

Figure 143 Java Source Editor



Note: The Enterprise Designer automatically disables the Business Rules Editor and the Business Rules Designer when you are actively using the Java Source Editor. This is a safety feature that prevents you from inadvertently making contradictory changes. You must click the **Commit Changes** button in the main toolbar to re-enable the Business Rules Editor and the Business Rules Designer.

8.4 Java Methods

8.4.1 Boolean Methods

Figure 144 Method Palette: Boolean Methods

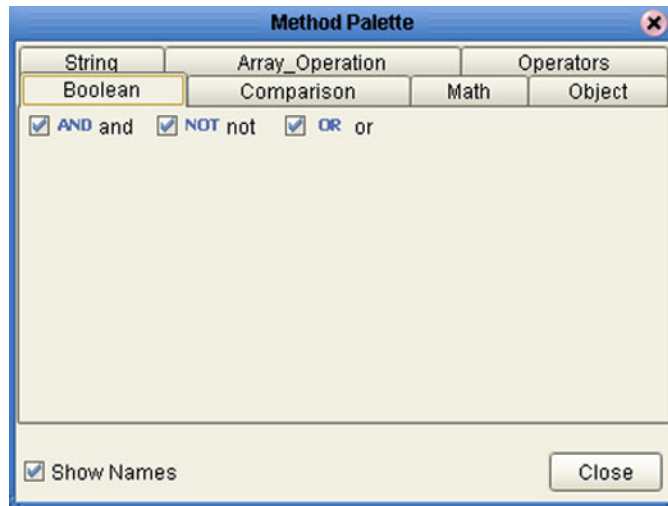


Table 34 Java Boolean Methods

Method Box	Description/Usage
	The AND method returns Boolean true if both <i>boolean1</i> and <i>boolean2</i> are true; otherwise, returns Boolean false.
	The OR method returns Boolean false if both <i>boolean1</i> and <i>boolean2</i> are false; otherwise, returns Boolean true.
	The NOT method returns the inverse of <i>boolean1</i> .

8.4.2 Comparison Methods

Figure 145 Method Palette: Comparison Methods

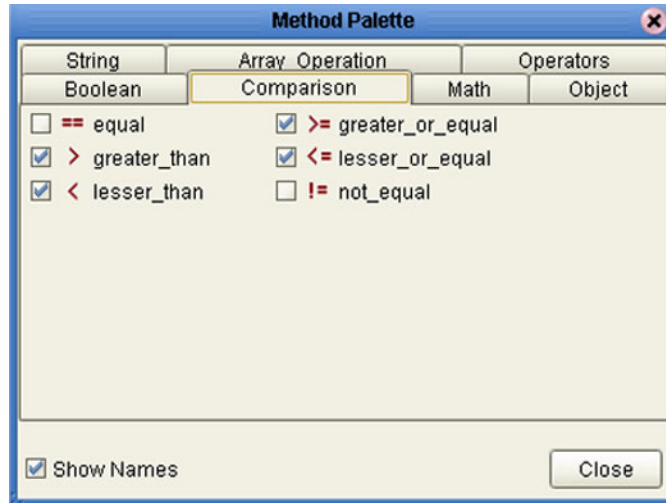
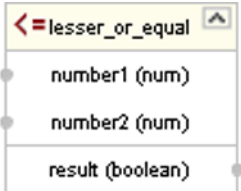
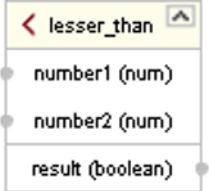
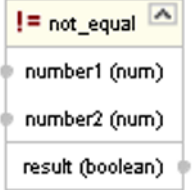


Table 35 Java Comparison Methods

Method Box	Description/Usage
	The equal method returns Boolean true if <i>number1</i> is equal to <i>number2</i> ; otherwise, returns Boolean false.
	The greater or equal method returns Boolean true if <i>number1</i> is greater than or equal to <i>number2</i> ; otherwise, returns Boolean false.
	The greater than method returns Boolean true if <i>number1</i> is greater than <i>number2</i> ; otherwise, returns Boolean false.

Table 35 Java Comparison Methods

Method Box	Description/Usage
 <p>The image shows a method box for the <code>lesser_or_equal</code> method. It features a yellow header with a left-pointing arrow and the text "<code><=</code> lesser_or_equal". Below the header are three rows: "number1 (num)", "number2 (num)", and "result (boolean)". Each row has a small grey dot on its left side, and the "result" row has a small grey dot on its right side.</p>	<p>The lesser or equal method returns Boolean true if <i>number1</i> is less than or equal to <i>number2</i>; otherwise, returns Boolean false.</p>
 <p>The image shows a method box for the <code>lesser_than</code> method. It features a yellow header with a left-pointing arrow and the text "<code><</code> lesser_than". Below the header are three rows: "number1 (num)", "number2 (num)", and "result (boolean)". Each row has a small grey dot on its left side, and the "result" row has a small grey dot on its right side.</p>	<p>The lesser than method returns Boolean true if <i>number1</i> is less than <i>number2</i>; otherwise, returns Boolean false.</p>
 <p>The image shows a method box for the <code>not_equal</code> method. It features a yellow header with a left-pointing arrow and the text "<code>!=</code> not_equal". Below the header are three rows: "number1 (num)", "number2 (num)", and "result (boolean)". Each row has a small grey dot on its left side, and the "result" row has a small grey dot on its right side.</p>	<p>The not equal method returns Boolean true if <i>number1</i> is not equal to <i>number2</i>; otherwise, returns Boolean false.</p>

8.4.3 Math Methods

Figure 146 Method Palette: Math Methods

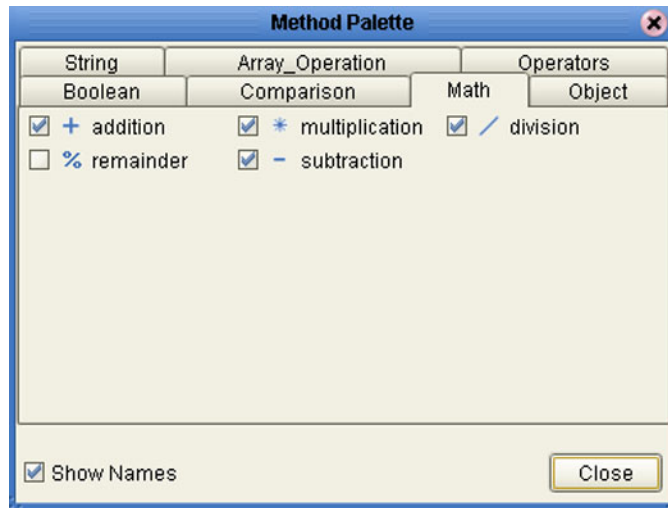
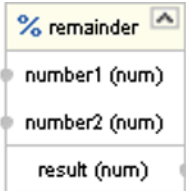
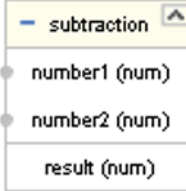


Table 36 Java Math Methods

Method Box	Description/Usage
	<p>The addition method adds the value of <i>number1</i> to the value of <i>number2</i>, returns the sum.</p>
	<p>The division method divides the value of <i>number1</i> by the value of <i>number2</i>, returns the quotient.</p>
	<p>The multiplication method multiplies the value of <i>number1</i> by the value of <i>number2</i>, returns the product.</p>

Table 36 Java Math Methods

Method Box	Description/Usage
 <p>The image shows a method box for the remainder operation. It has a title bar with a blue percent sign icon and the text '% remainder' followed by a small upward-pointing arrow icon. Below the title bar, there are two input fields: 'number1 (num)' and 'number2 (num)', each with a grey dot on its left side. At the bottom, there is an output field labeled 'result (num)' with a grey dot on its right side.</p>	<p>The remainder method divides the numerical value of <i>number1</i> by the numerical value of <i>number2</i>, returns the remainder.</p>
 <p>The image shows a method box for the subtraction operation. It has a title bar with a blue minus sign icon and the text '- subtraction' followed by a small upward-pointing arrow icon. Below the title bar, there are two input fields: 'number1 (num)' and 'number2 (num)', each with a grey dot on its left side. At the bottom, there is an output field labeled 'result (num)' with a grey dot on its right side.</p>	<p>The subtraction method subtracts the numerical value of <i>number2</i> from the numerical value of <i>number1</i>, returns the difference.</p>

8.4.4 Object Methods

Figure 147 Method Palette: Object Methods

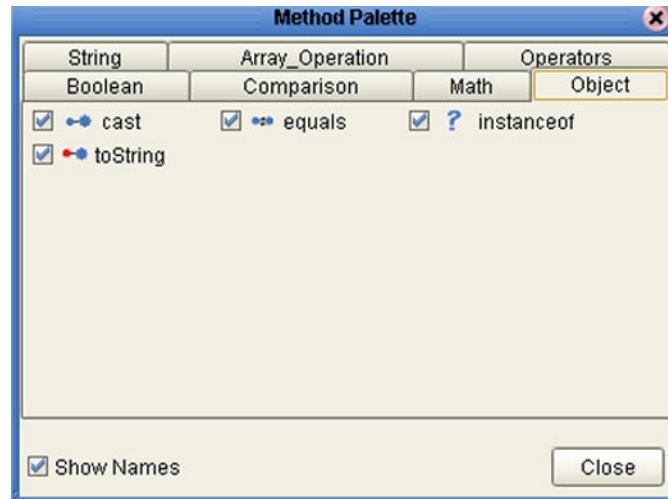


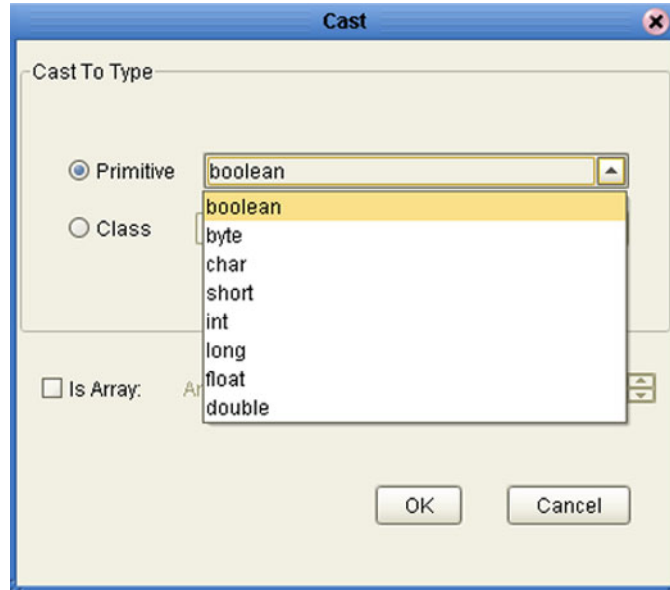
Table 37 Java Object Methods

Method Box	Description/Usage
	<p>Dragging the icon first displays the <i>CAST</i> dialog box (see Figure 148 on page 187), in which you specify the desired type (can be either primitive or class). The cast method then converts the given type (<i>any</i>) to the type specified in the <i>CAST</i> dialog box.</p>
	<p>The equals method returns Boolean true if <i>Object</i> is logically equal to <i>obj</i>, otherwise, returns Boolean false.</p>
	<p>Dragging the icon first displays the <i>InstanceOf</i> dialog box (see series beginning with Figure 149 on page 187), in which you specify the desired type. The instanceof method returns Boolean true if <i>Object</i> is of the specified type; otherwise, returns Boolean false .</p>
	<p>The toString method converts the object in the input field into a string.</p>

Object Method Dialog Boxes

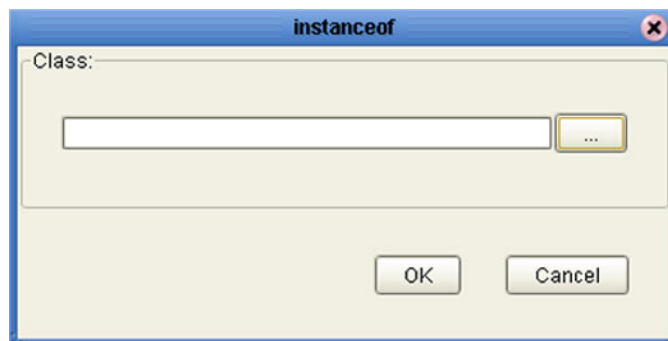
CAST Method

Figure 148 CAST Dialog Box



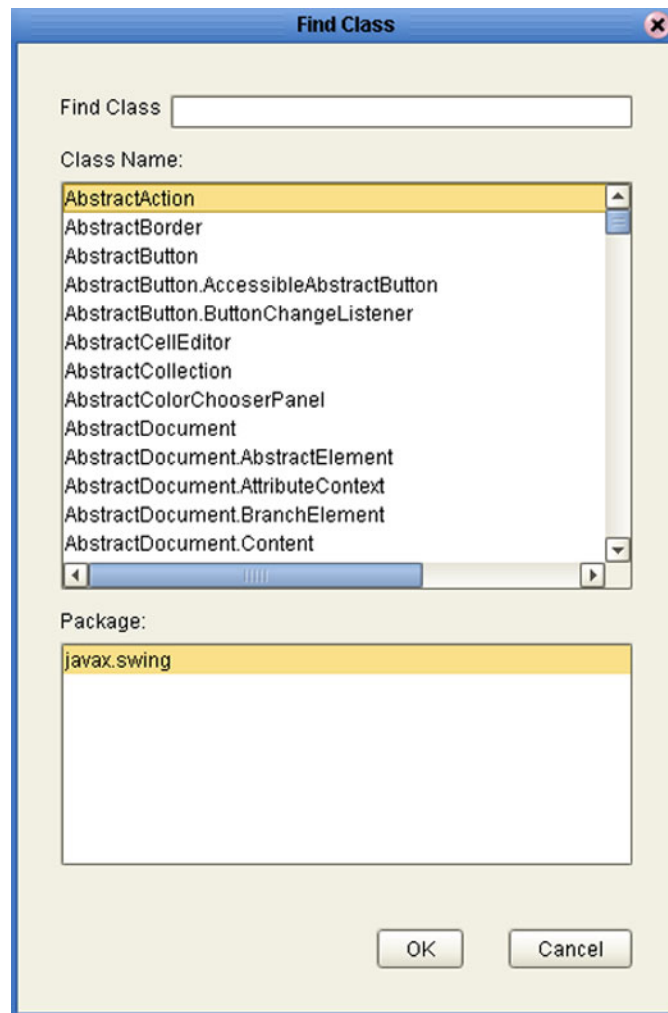
InstanceOf Method

Figure 149 InstanceOf Dialog Box (1)



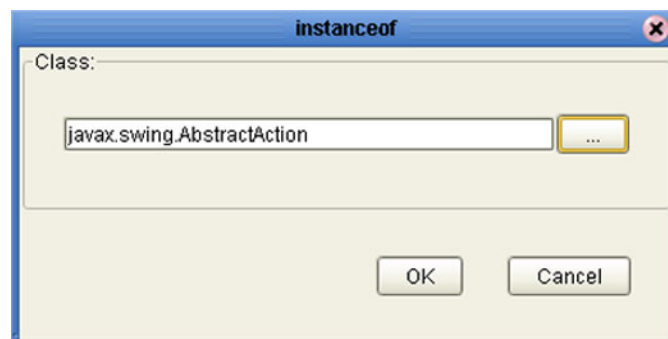
Clicking the ellipsis (...) button displays the *Find Class* dialog box (see [Figure 150 on page 188](#)), in which you select the desired class name.

Figure 150 Find Class Dialog Box



Clicking **OK** returns you to the *InstanceOf* dialog box (see Figure 151), with the field displaying *Package Name Class Name*.

Figure 151 InstanceOf Dialog Box (2)



Clicking **OK** displays the *InstanceOf* method box.

8.4.5 String Methods

Figure 152 Method Palette: String Methods

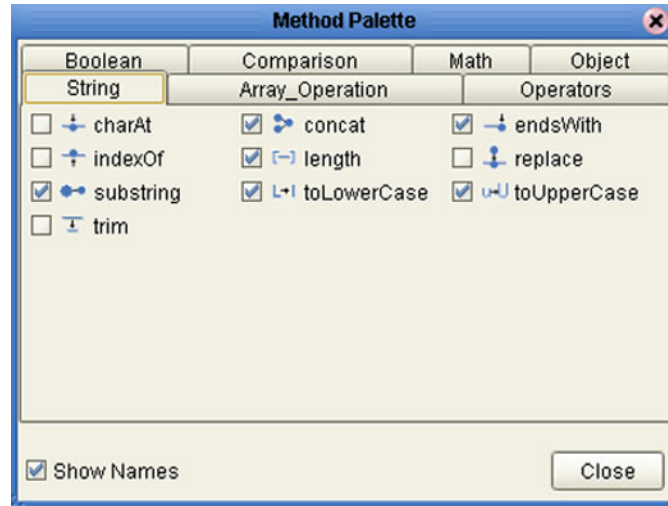


Table 38 Java String Methods

Method Box	Description/Usage
	<p>The charAt method returns the character at the specified index, where <i>index</i> is the number of characters from the beginning of <i>String</i>.</p>
	<p>The concat method returns the string created by concatenating <i>str</i> to the end of <i>String</i>.</p>
	<p>The endsWith method returns Boolean true if <i>String</i> ends with the string <i>suffix</i>; otherwise, returns Boolean false.</p>

Table 38 Java String Methods

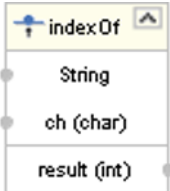



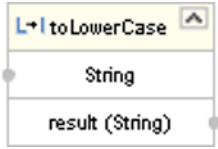
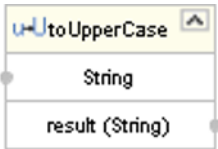
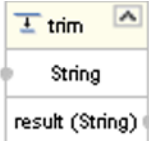
Method Box	Description/Usage
	<p>The indexOf method returns the index within <i>String</i> corresponding to the location of the specified character (<i>ch</i>), where the index represents the number of characters from the beginning of <i>String</i>.</p>
	<p>The length method returns the length (number of characters) of <i>String</i>.</p>
	<p>The replace method returns a new string in which all occurrences of <i>oldChar</i> are replaced with <i>newChar</i>.</p>
	<p>The substring method returns a string that is a substring of <i>String</i>, beginning from <i>beginIndex</i> (inclusive) and ending at <i>endIndex</i> (exclusive). The indices represent the number of characters from the beginning of <i>String</i>.</p>
	<p>The toLowerCase method converts all characters in <i>String</i> to lower case, using the rules of the default locale.</p>
	<p>The toUpperCase method converts all characters in <i>String</i> to upper case, using the rules of the default locale.</p>

Table 38 Java String Methods

Method Box	Description/Usage
 A screenshot of a software tool's method box for the <code>trim</code> method. The box is titled "trim" and shows a parameter of type <code>String</code> and a return value of type <code>String</code> . The return value is labeled "result (String)".	The trim method trims leading and trailing whitespace from <i>String</i> .

8.4.6 Array Operation Methods

Figure 153 Method Palette: Array Operation Methods

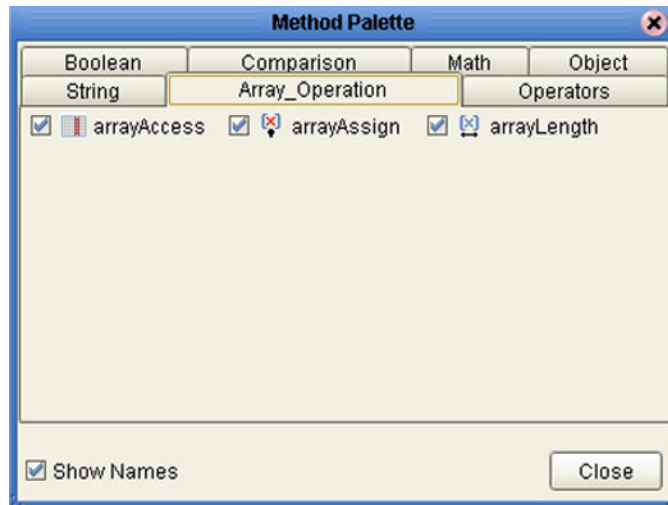


Table 39 Java Array Operation Methods

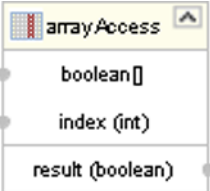
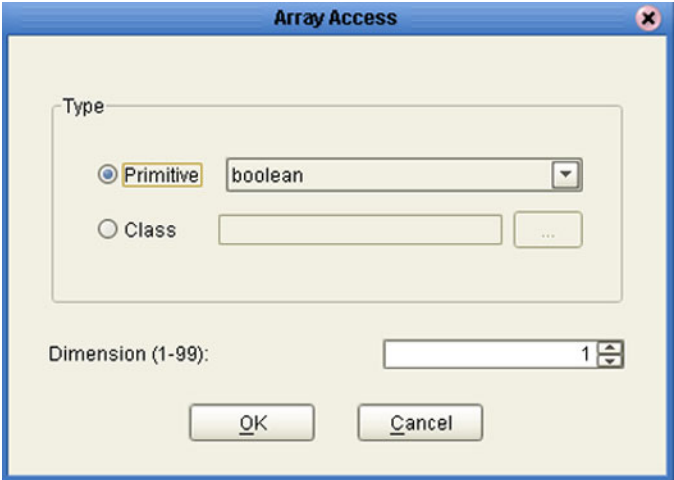
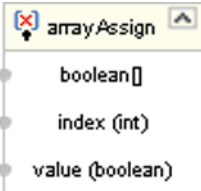
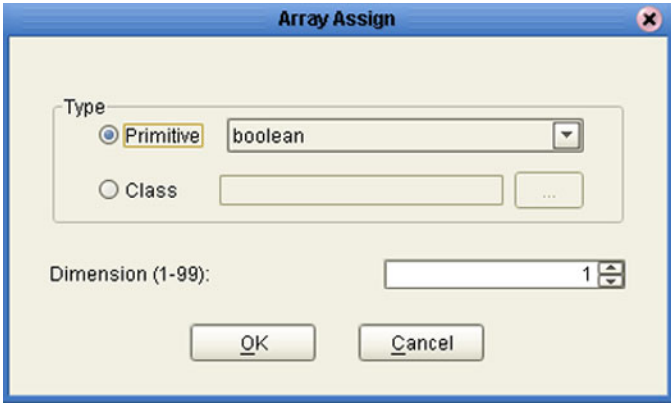
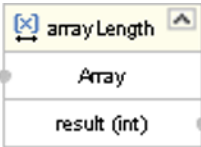
Method Box	Description/Usage
	<p>Dragging the icon first displays the <i>Array Access</i> dialog box, in which you specify the desired array type (can be either primitive or class) and dimension. The arrayAccess method then returns the element located at the specified index within the array, where the index represents the number of elements from the beginning of the array.</p> 

Table 39 Java Array Operation Methods

Method Box	Description/Usage
 <p>The method box for <code>arrayAssign</code> shows the following parameters: <code>boolean[]</code>, <code>index (int)</code>, and <code>value (boolean)</code>.</p>	<p>Dragging the icon first displays the <i>Array Assign</i> dialog box, in which you specify the desired array type (can be either primitive or class) and dimension. The arrayAssign method then sets the element located at the specified index within the array, where the index represents the number of elements from the beginning of the array.</p>  <p>The <i>Array Assign</i> dialog box has a title bar with a close button. It contains a 'Type' section with two radio buttons: 'Primitive' (selected) and 'Class'. The 'Primitive' option has a dropdown menu showing 'boolean'. The 'Class' option has an empty text field and an ellipsis button. Below this is a 'Dimension (1-99):' label with a text field containing '1' and a spinner control. At the bottom are 'OK' and 'Cancel' buttons.</p>
 <p>The method box for <code>arrayLength</code> shows the following parameters: <code>Array</code> and <code>result (int)</code>.</p>	<p>The arrayLength method returns the length (number of elements) of the array.</p>

8.4.7 Operator Methods

Figure 154 Method Palette: Operator Methods

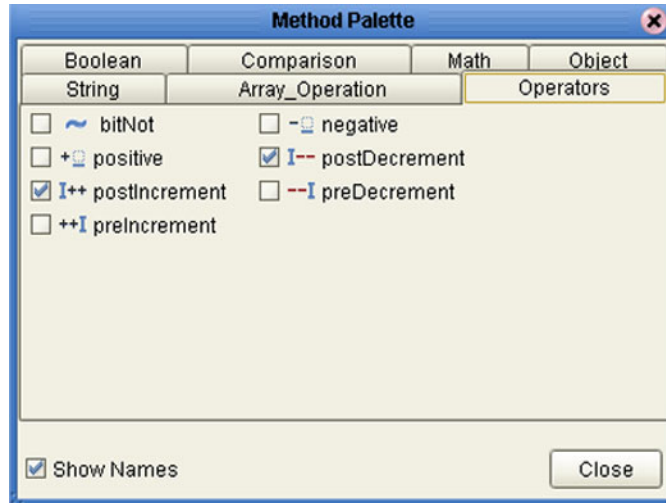
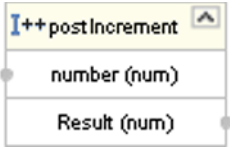
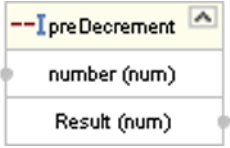
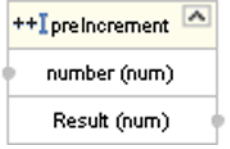


Table 40 Java Operator Methods

Method Box	Description/Usage
	The bitNot method returns the inverse of the bit value of the input number.
	The negative method returns the arithmetic negation of the input number.
	The positive method returns the value of the input number.
	The postDecrement method decrements the input number by one, after other operations.

Table 40 Java Operator Methods

Method Box	Description/Usage
 <p>The image shows a method box for the <code>postIncrement</code> method. It has a title bar with the text <code>I++postIncrement</code> and a small upward-pointing arrow icon. Below the title bar, there is an input field labeled <code>number (num)</code> with a small dot on its left side. Below the input field, there is an output field labeled <code>Result (num)</code> with a small dot on its right side.</p>	<p>The postIncrement method increments the input number by one, after other operations.</p>
 <p>The image shows a method box for the <code>preDecrement</code> method. It has a title bar with the text <code>--IpreDecrement</code> and a small upward-pointing arrow icon. Below the title bar, there is an input field labeled <code>number (num)</code> with a small dot on its left side. Below the input field, there is an output field labeled <code>Result (num)</code> with a small dot on its right side.</p>	<p>The preDecrement method decrements the input number by one, before other operations.</p>
 <p>The image shows a method box for the <code>preIncrement</code> method. It has a title bar with the text <code>++IpreIncrement</code> and a small upward-pointing arrow icon. Below the title bar, there is an input field labeled <code>number (num)</code> with a small dot on its left side. Below the input field, there is an output field labeled <code>Result (num)</code> with a small dot on its right side.</p>	<p>The preIncrement method increments the input number by one, before other operations.</p>

8.5 Java Encapsulation

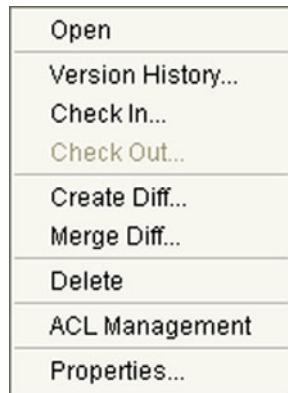
Java Collaboration Definitions can be saved to a Diff (.sdf) file. This feature allows two sites working with the same Java Collaboration Definition to seamlessly merge changes.

8.5.1 Creating a Modified Java Collaboration Definition

To create a *different* version of a Java Collaboration Definition file

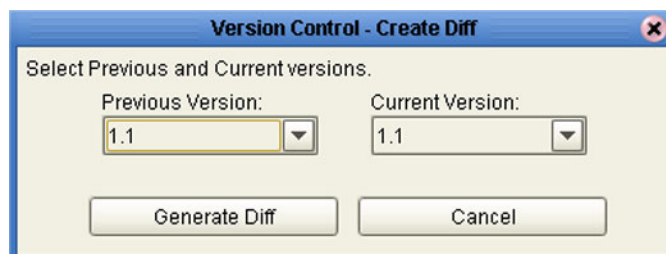
- 1 In Project Explorer, select a **Java Collaboration Definition** icon.
- 2 Right-click to display the context menu shown in Figure 155.

Figure 155 Java Collaboration Definition Context Menu



- 3 Select **Create Diff** to display the dialog box shown in Figure 156.

Figure 156 Version Control - Create Diff Dialog Box



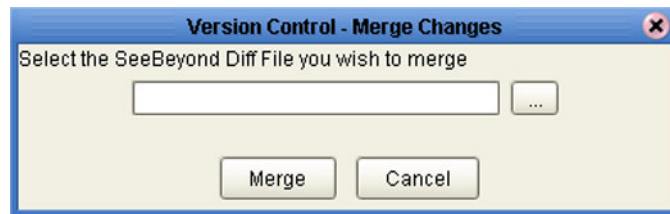
- 4 Click **Generate Diff** to display the *Specify Name ...* dialog box.
- 5 Enter a name for the Diff file in the **File Name** box (or use the default file name).
- 6 Click **Save** to save the Diff file.

8.5.2 Merging Two Versions of a Java Collaboration Definition

To merge two versions of a Java Collaboration Definition

- 1 From the Project Explorer tab, select a **Java Collaboration Definition** icon.
- 2 Right-click to display the Java Collaboration Definition context menu.
- 3 Select **Check Out**.
- 4 Right-click to again display the context menu.
- 5 Select **Merge Diff...** to display the dialog box shown in Figure 157.

Figure 157 Version Control - Merge Changes Dialog Box



- 6 Click the **Ellipsis (...)** button to display the *Specify Name ...* dialog box.
- 7 Locate and select the Diff file.
- 8 Click **Open** add the Diff file to the dialog box shown in Figure 157.
- 9 Click **Merge** to merge the Diff file with the corresponding Java Collaboration Definition in your Project.
- 10 Resolve any conflicts from the merge.
- 11 **Commit** the merged Collaboration.
- 12 **Save** the merged Collaboration.

8.6 Setting Up a Java Collaboration Definition Variable

The Java Collaboration Editor includes a variable creation feature. The following sections describe how to set up and assign a constructor to a variable.

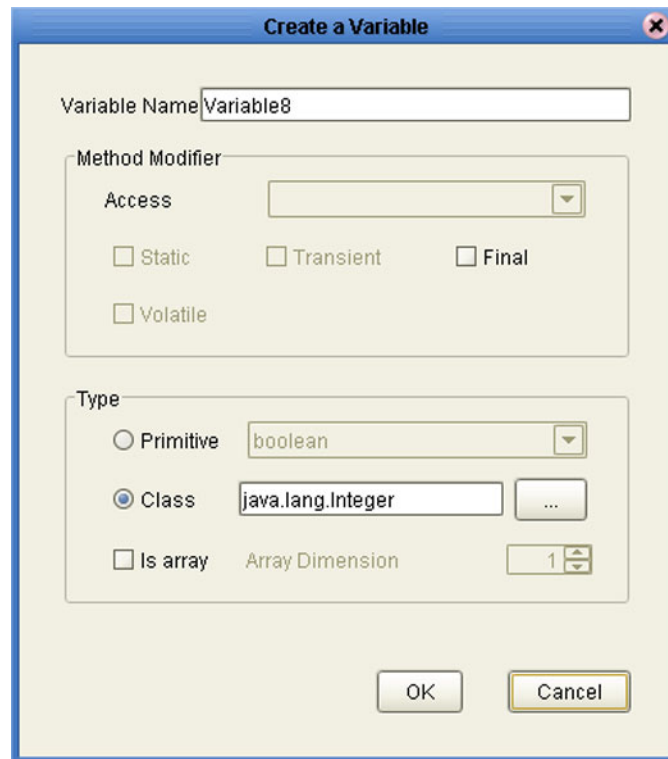
8.6.1 Creating a Variable

This section describes how to add a variable to a Java Collaboration Definition.

To create the variable

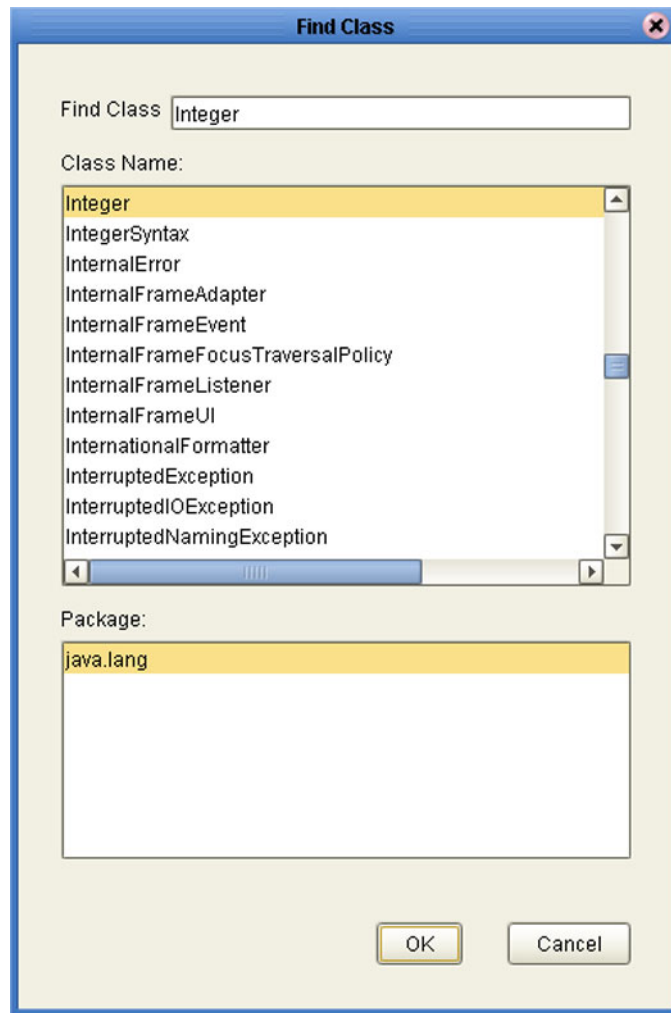
- 1 In the Business Rules area, click the **Local Variable** button to display the dialog box shown in Figure 158.

Figure 158 Create a Variable Dialog Box



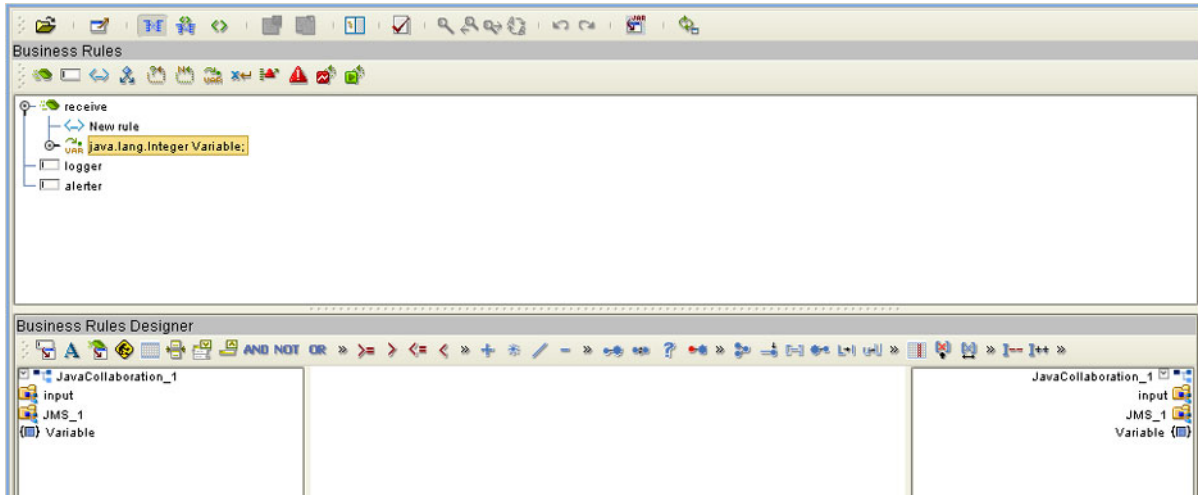
- 2 Enter a name for the variable in the **Variable Name** box.
- 3 Select the **Class** option button.
- 4 Click the **Ellipsis (...)** button to display the **Find Class** dialog box shown in Figure 159.

Figure 159 Find Class Dialog Box



- 5 Select a class from the **Class Name** list.
- 6 Click the **OK** button to add the variable, with selected class, to the Java Collaboration Editor window as shown in Figure 160.

Figure 160 Java Collaboration Definition with Variable



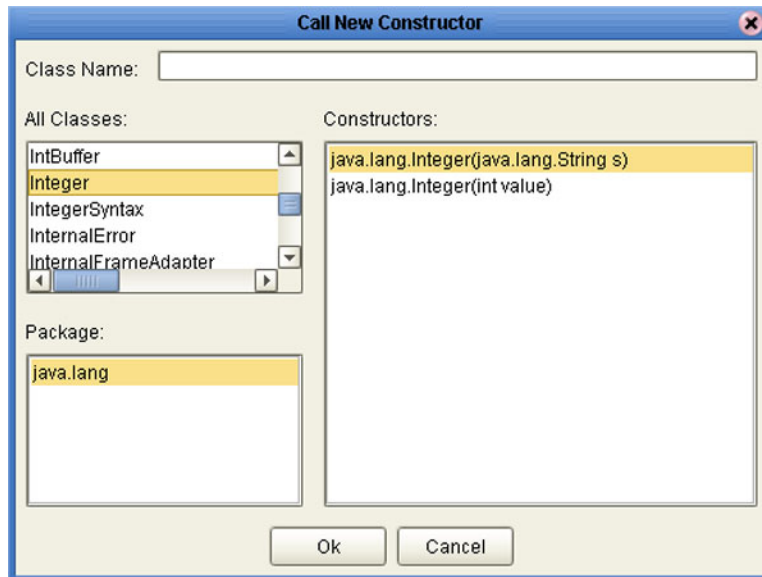
8.6.2 Invoking Variable Constructor

This section describes how to invoke a constructor for the variable.

To invoke a constructor

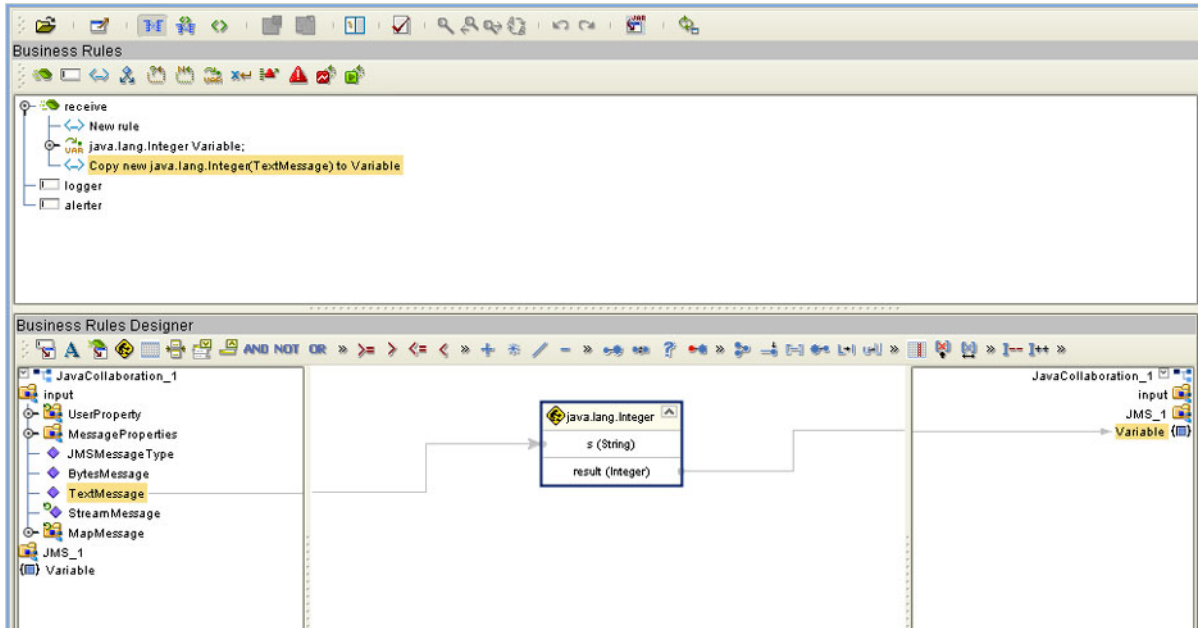
- 1 In the Business Rules Designer area, click the **Call New Constructor** button to display the dialog box shown in Figure 161.

Figure 161 Call New Constructor Dialog Box



- 2 Select the class used for the variable in the **All Classes** list.
- 3 Click **OK** to add an integer box to the Business Rules Designer area as shown in Figure 162.
- 4 Link an item in the left column through the integer to an item in the right column.

Figure 162 Java Collaboration Definition with Constructor



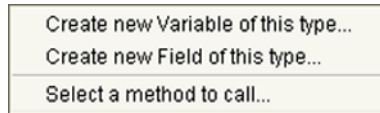
8.6.3 Displaying Method Classes

This section describes how to display a list of method classes.

To view the available method classes

- 1 Select a Variable in the Business Rules Designer area.
- 2 Right-click to display its context menu, as shown in Figure 163

Figure 163 Variable Context Menu



- 3 Click **Select a Method to Call** to display a list of method classes as shown in Figure 164.

Figure 164 Method Selection List Box



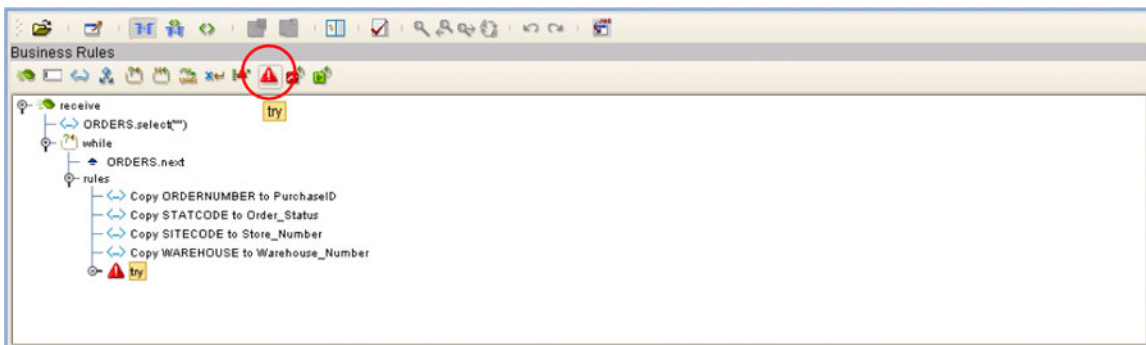
8.7 Using Try-Catch

Clicking the **try** icon in the toolbar adds a **try** statement to the Business Rules tree, initiating a number of programming statements that are monitored to see whether they succeed or fail. A **finally** statement is added automatically; if you want to perform a **try-catch**, you must use the following procedure.

To create a try-catch operation

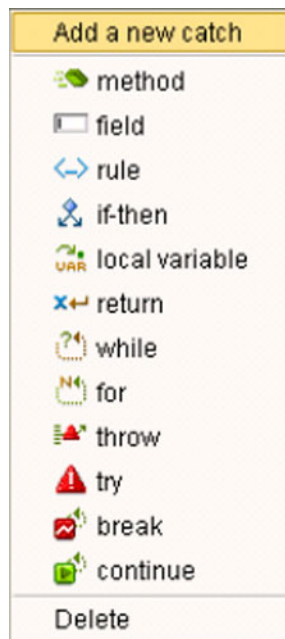
- 1 Click the **try** icon in the toolbar (see Figure 165), which adds the **try** rule to the business rules tree.

Figure 165 Try Icon



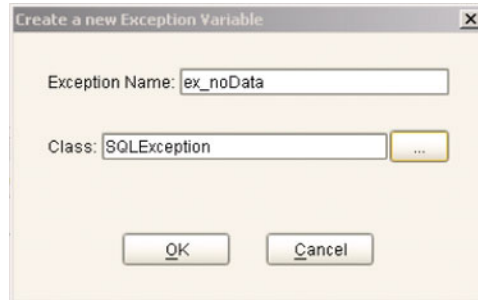
- 2 Right-click on the **try** rule to display the catch context menu (see Figure 166).

Figure 166 Catch Context Menu



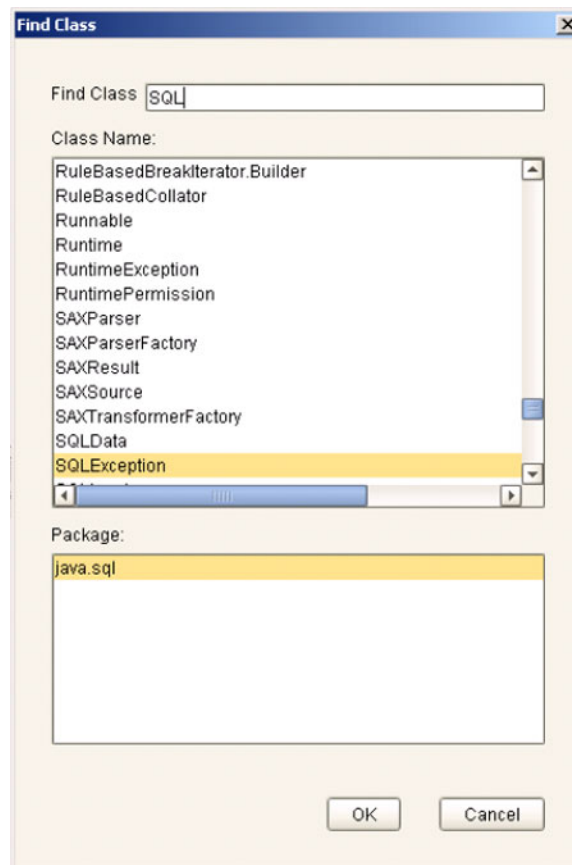
- 3 Select the **Add a new catch** option to add an exception variable, which displays a dialog box (see Figure 167).

Figure 167 Create a New Exception Variable Dialog Box



- 4 Enter a name for the exception, for example: **ex_noData**.
- 5 Click the button to the right of the Class text box to display a list of available classes (see Figure 168).

Figure 168 Find Class Dialog Box



- 6 To specify that the exception represents a database error, for example, select **SQLException** and click **OK** on both dialog boxes.

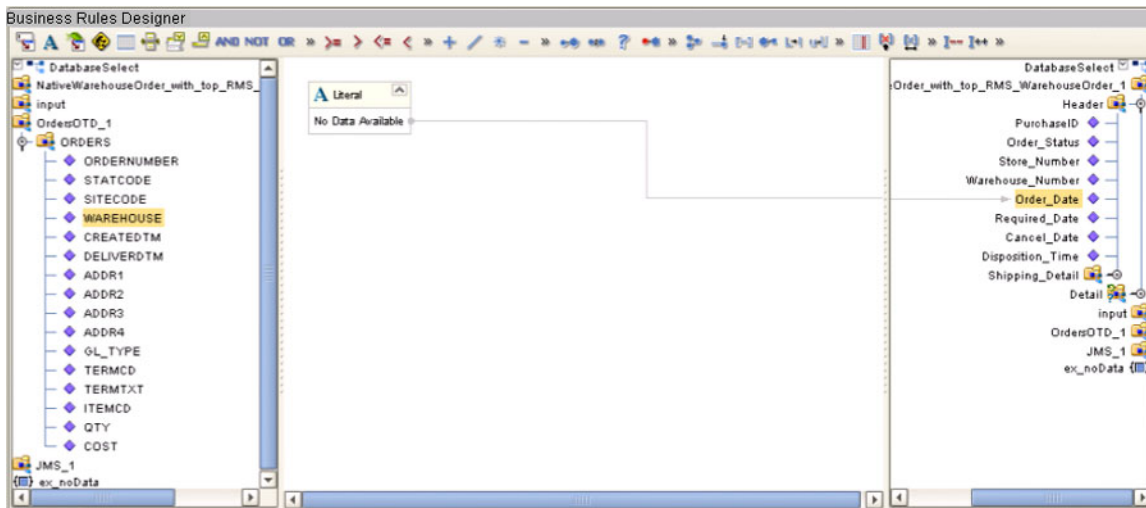
- 7 The catch rule is now added to the business rules tree (see Figure 169).

Figure 169 Catch SQLException Rule



- 8 The rule shown in Figure 169 contains a literal that represents the message that will be assigned to an outbound element Order_Date when the rule is executed (see Figure 170).

Figure 170 Exception Message



8.8 Adding and Using Third-Party Java Classes

Enterprise Designer allows you to add any file to the repository. Certain products in the ICAN Suite can then instruct the repository to distribute these files to Logical Hosts as needed. Selecting **New > File ...** in a Project context menu (see Figure 171) displays a file selection dialog box (see Figure 172).

Figure 171 Project Context Menu: New File ...

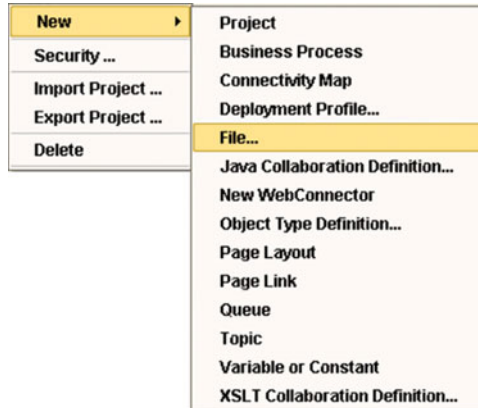
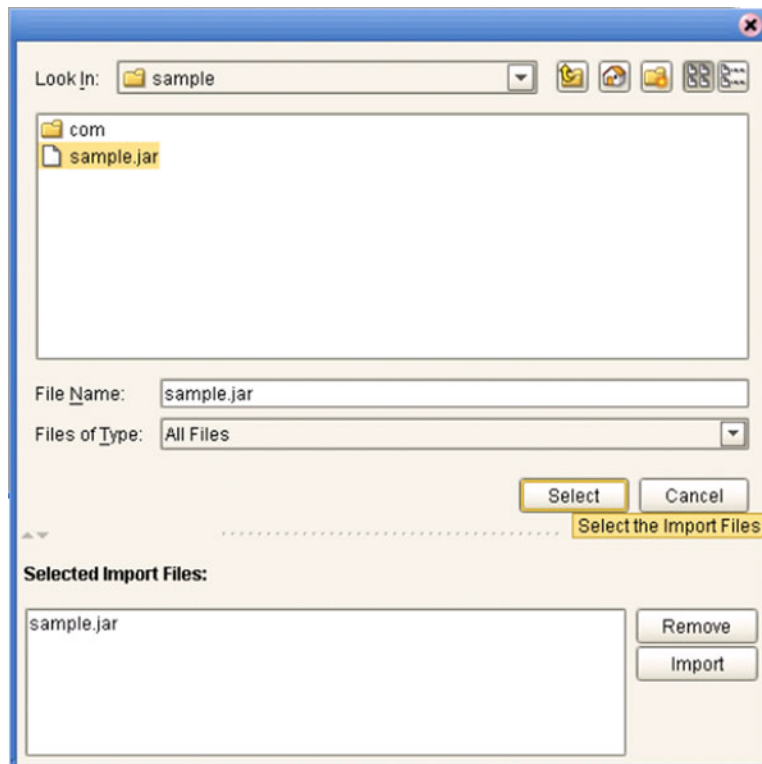


Figure 172 File Selection Dialog Box

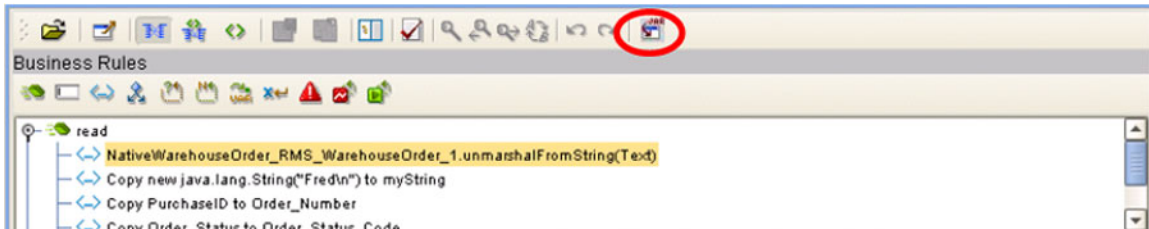


Locate the desired .jar files and click **Select** for each; then click **Import**.

To add the file(s) to a Collaboration Definition:

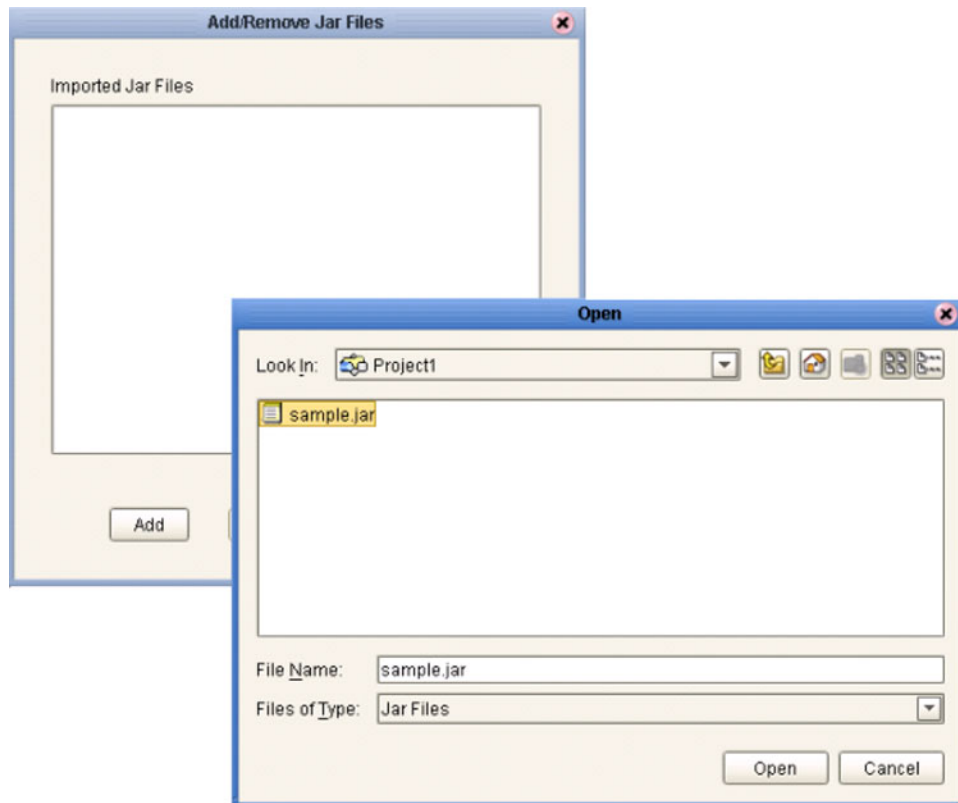
- 1 Open or create a Collaboration.
- 2 Click the **Import JAR File** icon in the toolbar (see Figure 173), which displays an *Add/Remove Jar Files* dialog box.

Figure 173 Import JAR File Icon



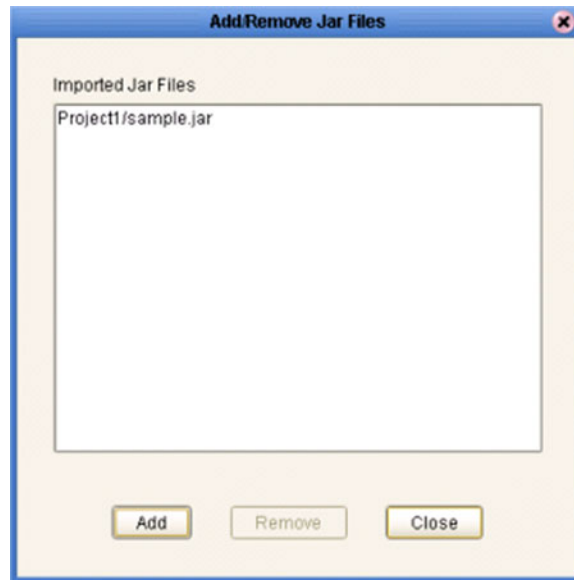
- 3 Click **Add** in the dialog box to display a file browser box (see Figure 175).
- 4 Locate the desired .jar files and click **Open** for each.

Figure 174 Add/Remove Jar File Dialog Box (1)



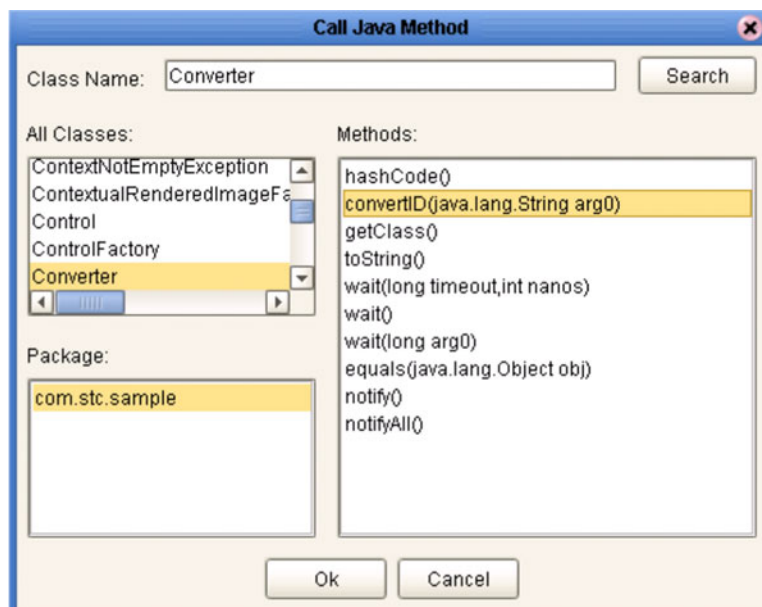
- 5 When all the .jar files are listed in the *Add/Remove Jar Files* dialog box (see Figure 175), click **Close**.

Figure 175 Add/Remove Jar File Dialog Box (2)



- 6 Click the **Call Java Method** icon in the Business Rules Designer, and a *Call Java Method* dialog box opens (see Figure 176).

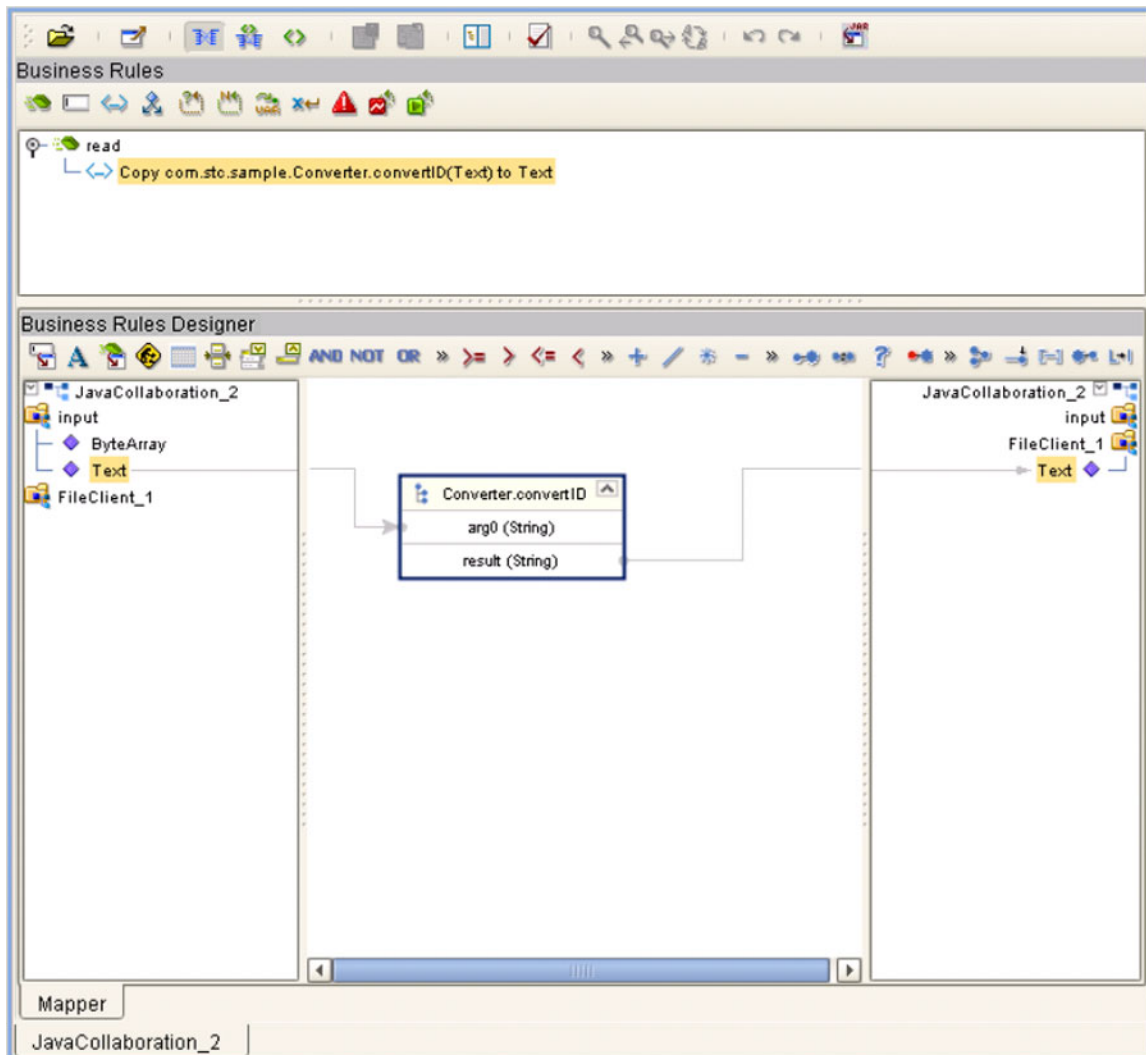
Figure 176 Call Java Method Dialog Box



- 7 Select the desired class from the **All Classes** list.
- 8 Select the desired method from the **Methods** list.
- 9 Click **Ok**.

To use the third-party method, add the business rule containing the method. In the example shown, the method is a **convertID** operation that converts text from upper case to lower case. Connect an inbound text file to the input node of the method, and the outbound node to an outbound text file (see Figure 177).

Figure 177 Using the Third-Party Java Method



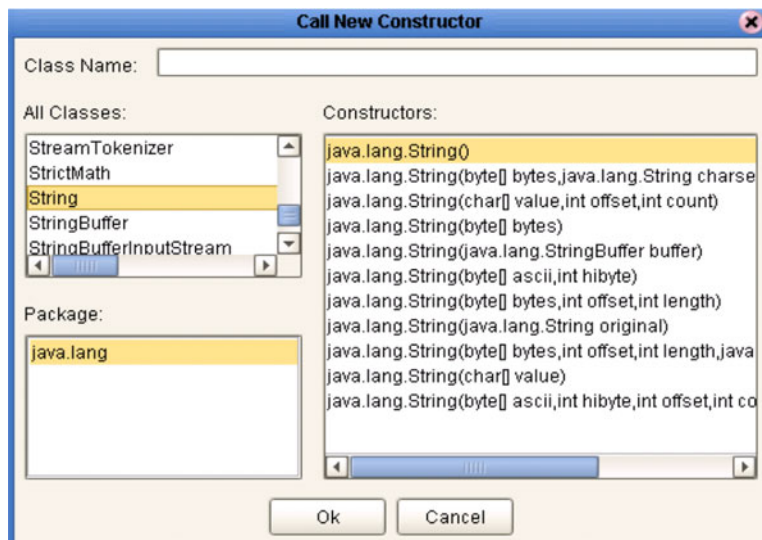
8.9 Adding Class Instances to a Collaboration

Instances of Java classes are added by means of the Java constructors, as described in the following procedure.

To add a class instance to a Collaboration

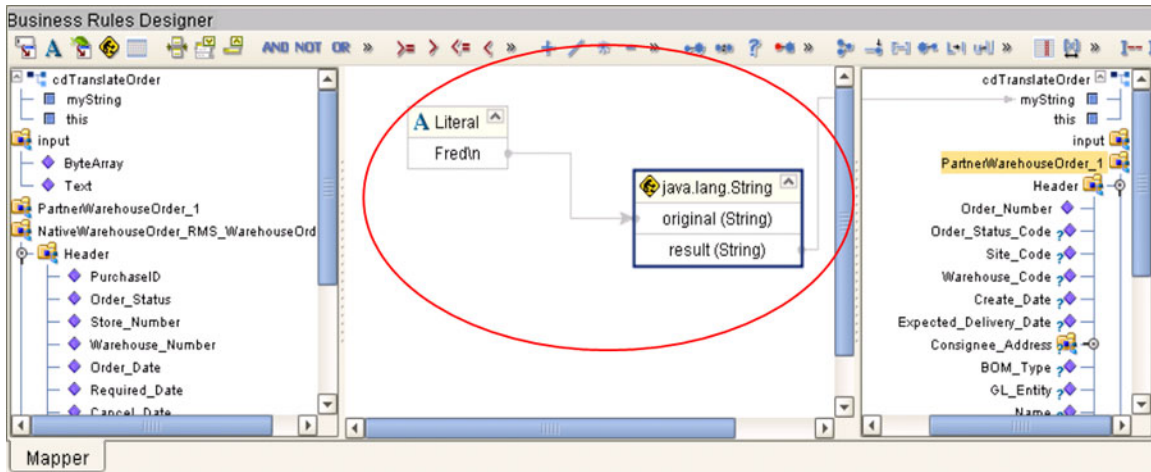
- 1 Create or open a Collaboration and create a new rule.
- 2 Click the **Constructor** icon on the toolbar (see [Business Rules Designer Toolbar Icons](#) on page 175) to display the *Call New Constructor* dialog box (see Figure 178). This dialog box presents all available classes, including any third-party classes that have been uploaded.

Figure 178 Call New Constructor Dialog Box



- 3 Select a constructor method from the Constructors list and click **OK**. This method is placed on the Business Rules Designer's canvas, and can now be used as shown in Figure 179.

Figure 179 Constructor Example 1

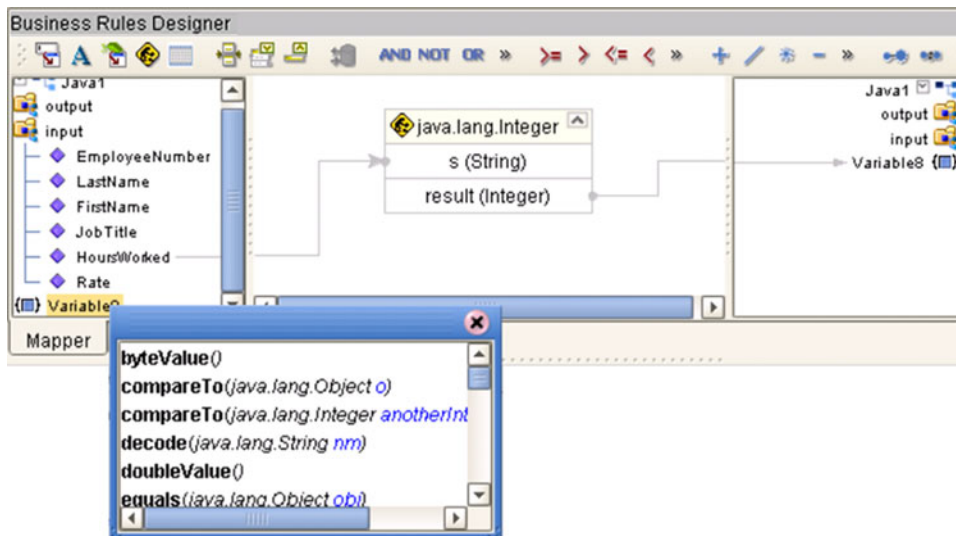


Alternative procedure to add a class instance to a Collaboration

An alternative way to invoke a constructor is to right-click on an element in the left panel of the Business Rules Designer and select a constructor method from the list box that appears, as illustrated in Figure 180. The procedure is:

- 1 Create a local string variable.
- 2 Right-click on the variable element.
- 3 Select a constructor from the list box.

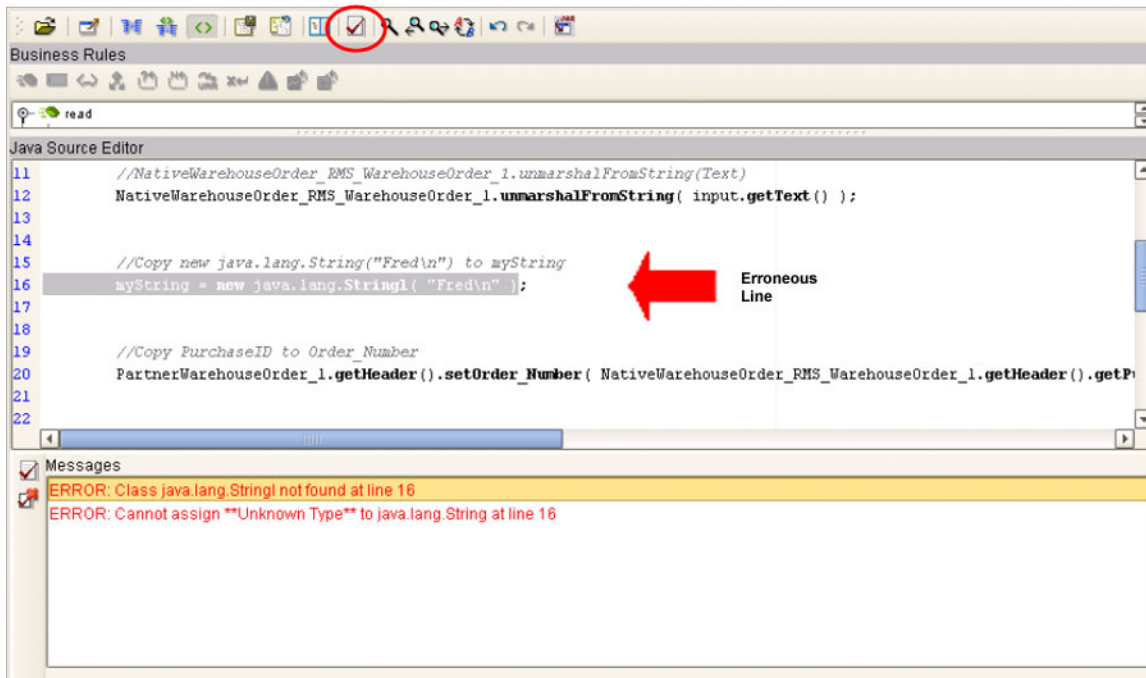
Figure 180 Constructor Example 2



8.10 Validating Java Collaborations

Clicking the **Validate** icon allows you to “precompile” the Java Collaboration Definition and display the errors in a validation panel, as shown in Figure 181. To locate the error, double-click on the error message and the Java Source Editor will be displayed, showing the erroneous line of code.

Figure 181 Validating a Collaboration Definition



The JCE tester is enabled only when the OTDs involved in a particular Collaboration is are of the following types:

- XSD
- DTD
- User-defined

If the Collaboration contains any other type (for example, JMS receive) the tester is disabled.

8.11 Debugging Java Collaborations

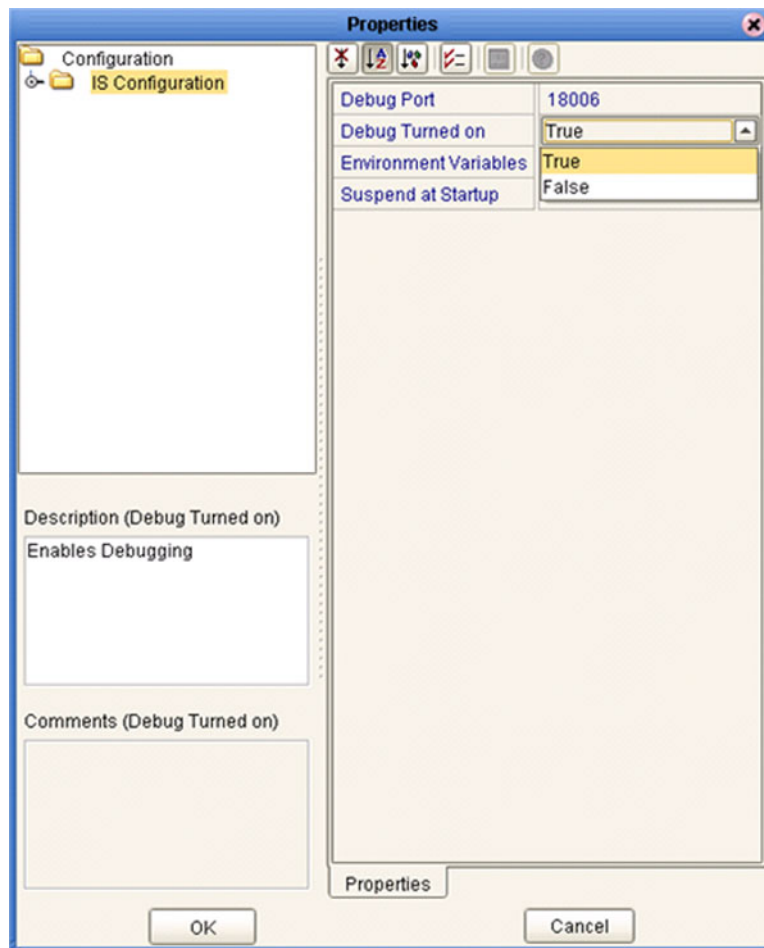
The Java Debugger enables you to debug Java Collaborations as deployed within an integration server on a Logical Host, and offers an alternative to creating logs and warnings in an Java Collaboration and subsequently inspecting them via the Enterprise Manager.

8.11.1 Enabling the Debugger

To enable the Java Debugger

- 1 In Enterprise Explorer, select the appropriate integration server.
- 2 Right-click to display the context menu.
- 3 Click **Properties** to display the Properties Dialog Box (see Figure 182).

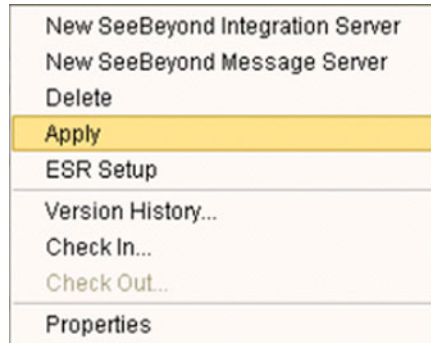
Figure 182 Integration Server Properties Dialog Box



- 4 Set the value for *Debug Turned On* property to **True** and click **OK**.

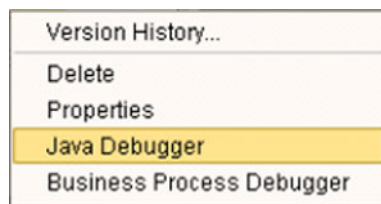
- 5 In Enterprise Explorer, select the Logical Host containing the integration server you have just configured.
- 6 Right-click to display the context menu and click **Apply** (see Figure 183).

Figure 183 Logical Host Context Menu



- 7 The **Java Debugger** option now is enabled in the Integration Server context menu (see Figure 184).

Figure 184 Integration Server Context Menu

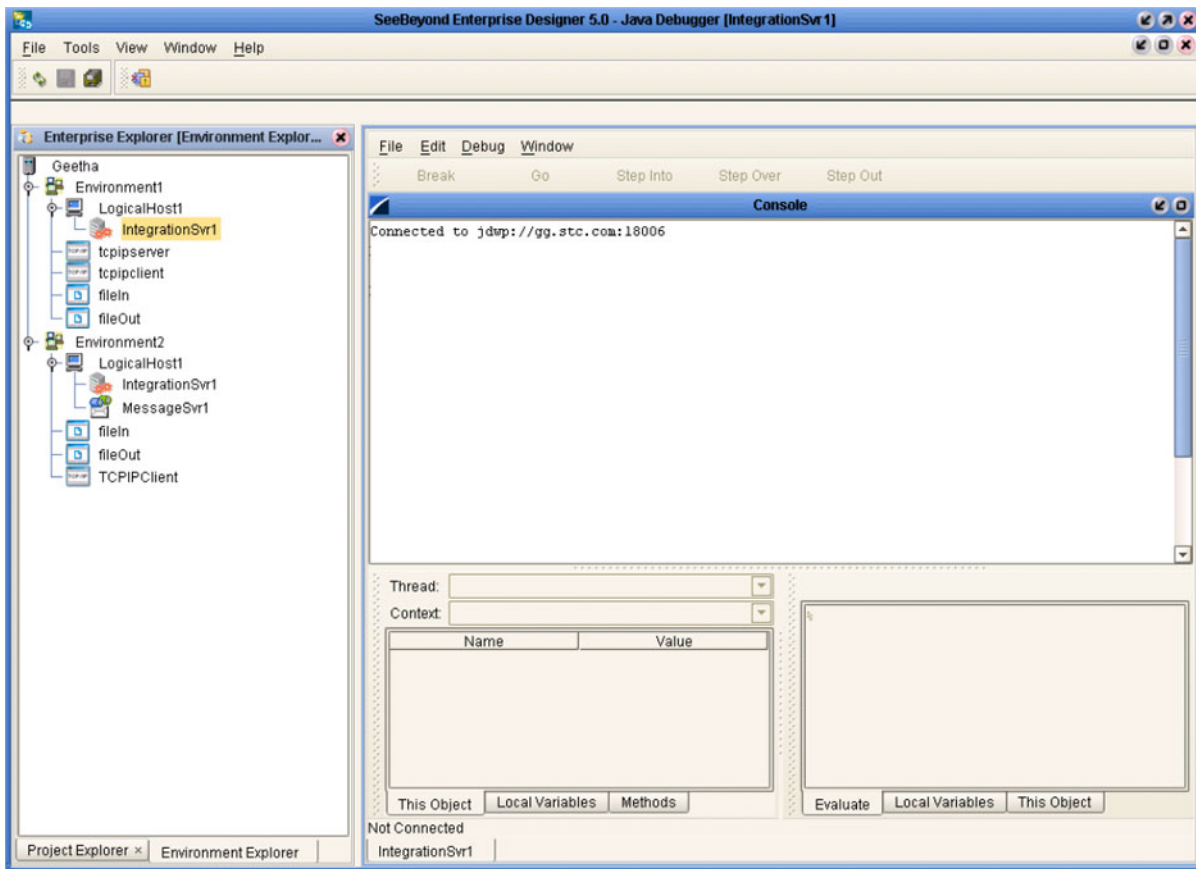


8.11.2 Invoking the Java Debugger

To invoke the Java Debugger

- 1 In Enterprise Explorer, select the appropriate integration server.
- 2 Right-click to display the context menu.
- 3 Click **Java Debugger** (see Figure 184), and the Java Debugger appears in the Enterprise Designer Editor panel (see Figure 185).

Figure 185 Java Debugger



Note: The Java Debugger appears whether or not the connection was successful. If there is no **Connected to ...** message, try the following procedure:

- A Select **Attach to JVM ...** from the File menu (see Figure 186), which displays the *Attach to JVM* dialog box (see Figure 187).

Figure 186 File Menu

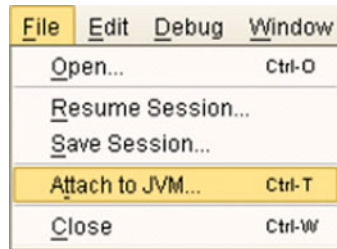
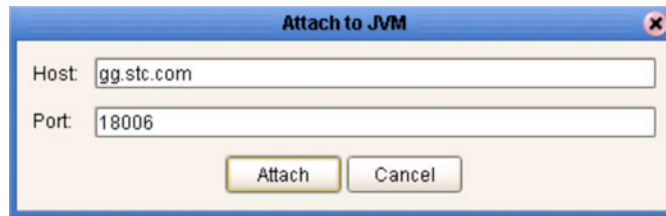
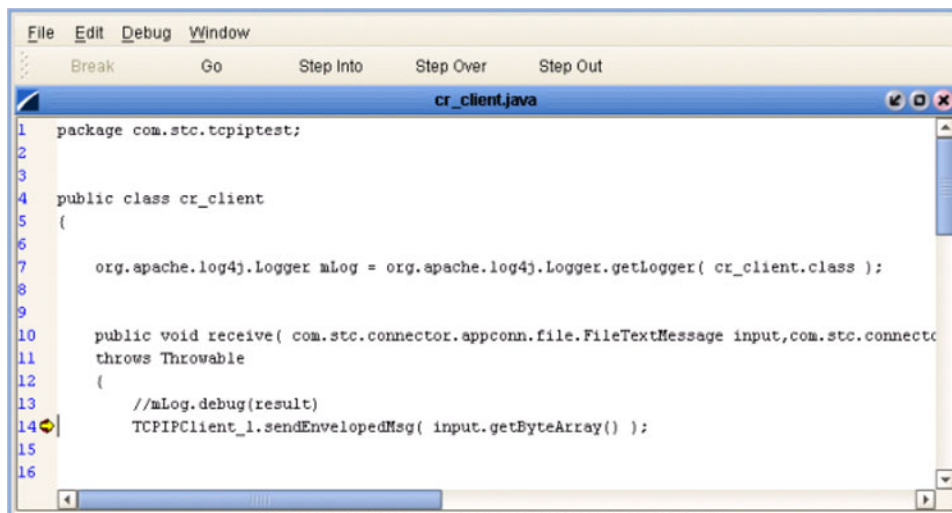


Figure 187 Attach to JVM Dialog Box



- B Enter the integration server’s host name and port number into the text boxes and click **Attach**. The debugger then re-attempts to connect to the integration server.
- 4 Once the Java Debugger is running, Java source code is displayed as soon as a Java Collaboration executes (see Figure 188).

Figure 188 Collaboration Source Code Display



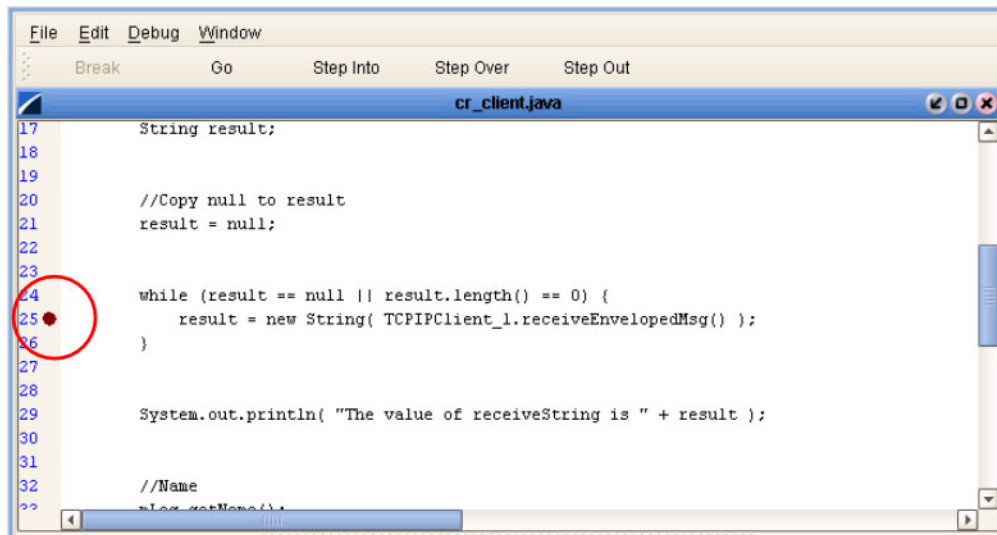
- 5 You can now set breakpoints to assist in examining and debugging the code.

8.11.3 Setting Breakpoints

To set a breakpoint

- 1 Click next to a line number in the executed source code. A red dot is displayed as a marker (see Figure 189).

Figure 189 Breakpoint Example



- 2 Alternatively, you can set stops in a specific class or method, or have the debugger break on an exception. These options are available from the **Debug** menu.
 - A To set a stop in a method, for example, select **Stop in Method ...** from the Debug menu (see Figure 190). A *Stop in Method* dialog box is displayed, in which you can select the desired method (see Figure 191).

Figure 190 Debug Menu

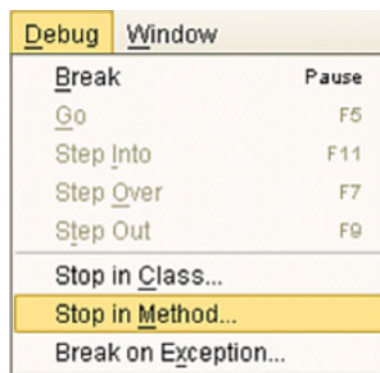
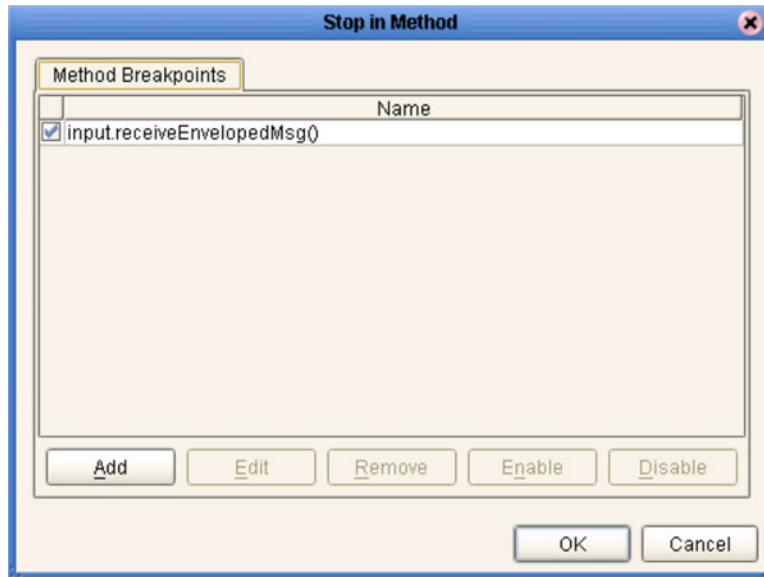


Figure 191 Stop in Method Dialog Box



- B To break on an exception, select **Break on Exception ...** from the Debug menu, which displays a *Choose Exception* dialog box (see Figure 192). All occurrences of the specified exception are then trapped and reported (see Figure 193).

Figure 192 Choose Exception Dialog Box

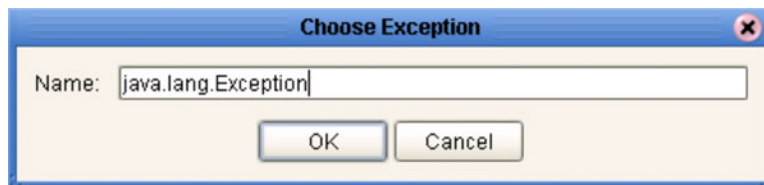


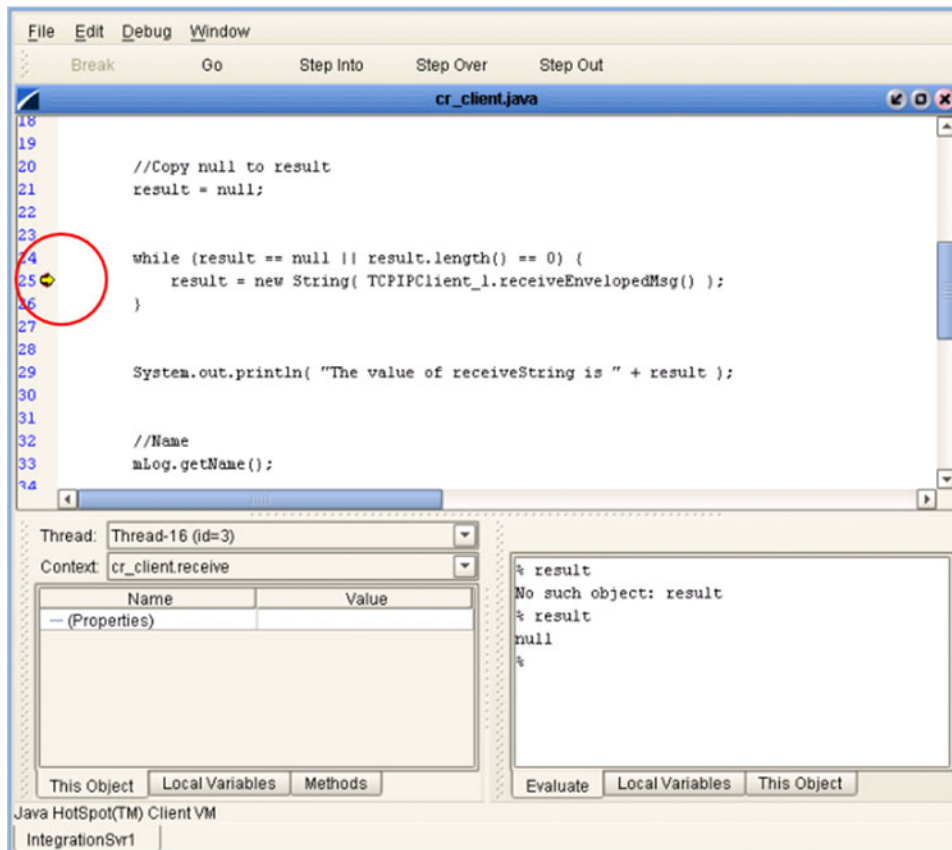
Figure 193 Break on Exception Dialog Box



8.11.4 Inspecting and Editing the Source Code

As soon as the execution of the Java Collaboration arrives at a set breakpoint, it stops executing and displays an right arrow indicator next to the line number in the source code (see Figure 194).

Figure 194 Breakpoint Indicator

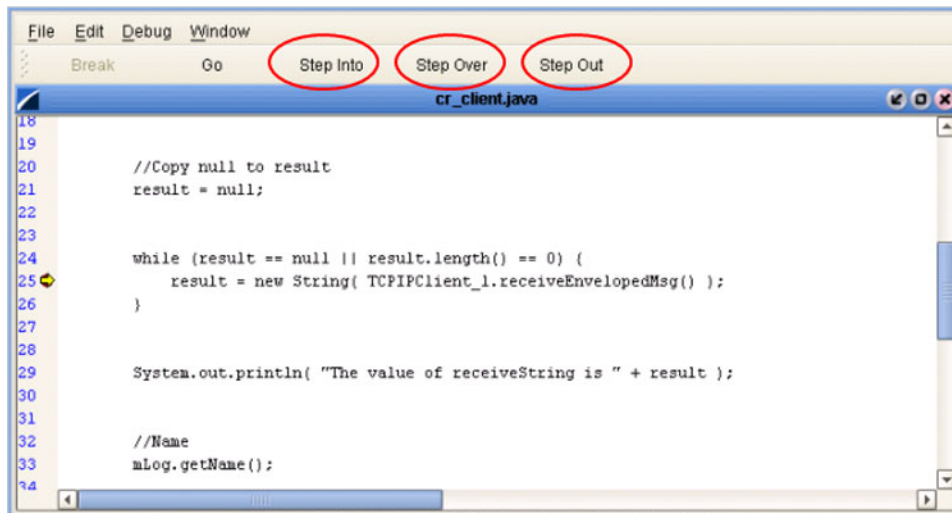


At this point, you can continue by (for example):

- ◆ Stepping Into
- ◆ Stepping Over
- ◆ Stepping Out
- ◆ Inspecting a local variable
- ◆ Setting a local variable

Stepping Into, Over, or Out

Figure 195 Stepping Into, Over, and Out Commands

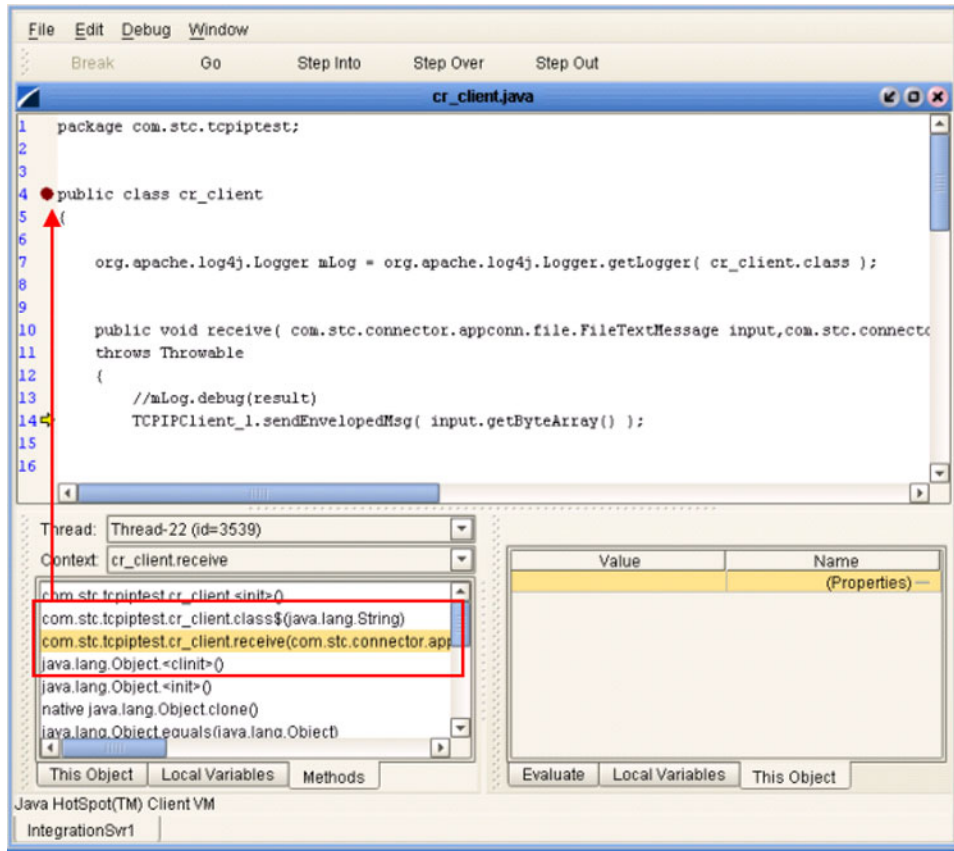


- By selecting the **Step Into** option (see Figure 195), the breakpoint is lifted and execution of the Collaboration will continue, *including* the line of code at the breakpoint.
- By selecting the **Step Over** option, the breakpoint is lifted and execution of the Collaboration will continue, *ignoring* the line of code at the breakpoint.
- By selecting the **Step Out** option, execution of the Collaboration is terminated.

Inspecting Java Threads and Methods

Selecting the **Methods** tab in the left bottom panel of the debugger displays the currently executed Java thread and method (see Figure 196).

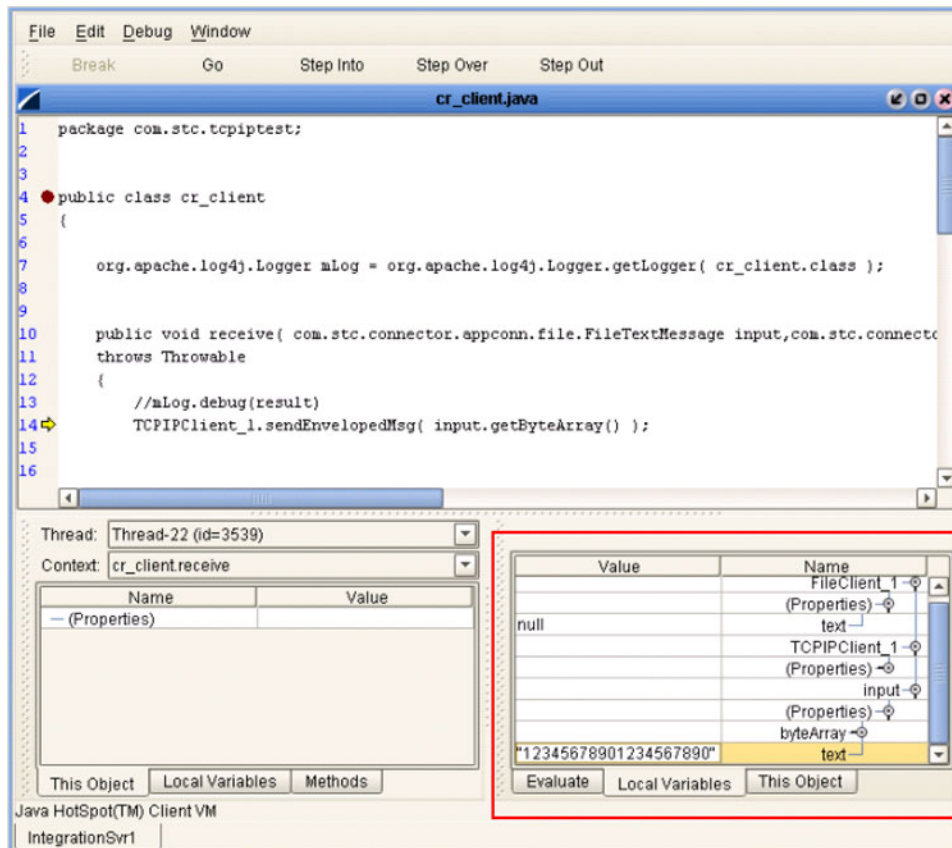
Figure 196 Java Thread and Method Display



Inspecting a Local Variable or Method

You can inspect a local variable by selecting the **Local Variables** tab in the right bottom panel of the debugger (see Figure 197). All nodes of the currently executed Java class are displayed here, and you can expand or collapse certain nodes to search for the value of the desired variable.

Figure 197 Local Variables Tab

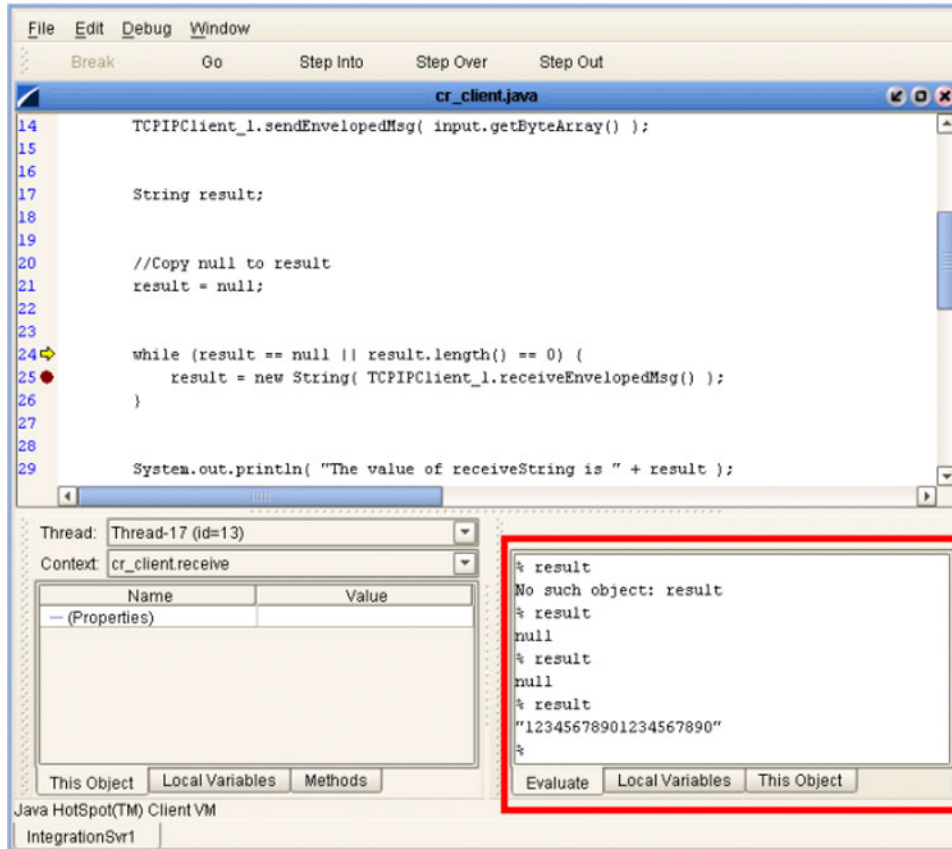


You also can inspect a local variable by selecting the **Evaluate** tab and entering the variable name in the panel using the following syntax:

```
% <variable_name>
```

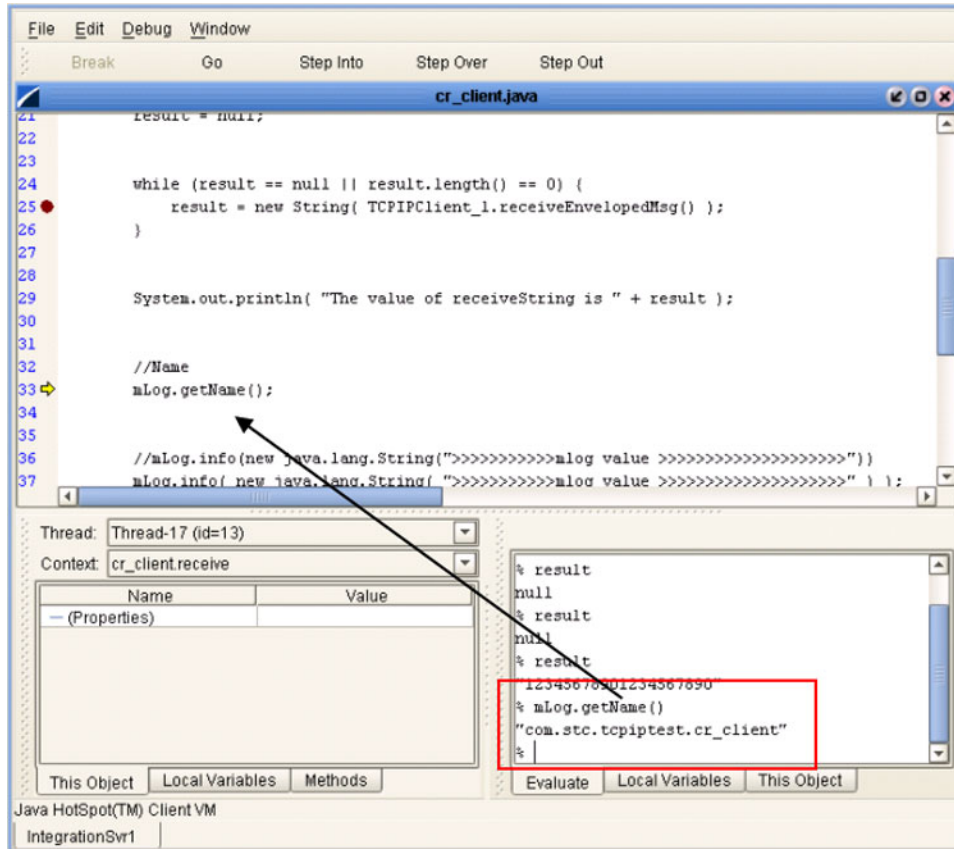
If the code has not initialized a variable, *No such object* is displayed; otherwise, the current value of the variable is displayed (see Figure 198).

Figure 198 Evaluate Local Variable



You can also inspect the result of a method by entering the method name, following the same syntax format (see Figure 199).

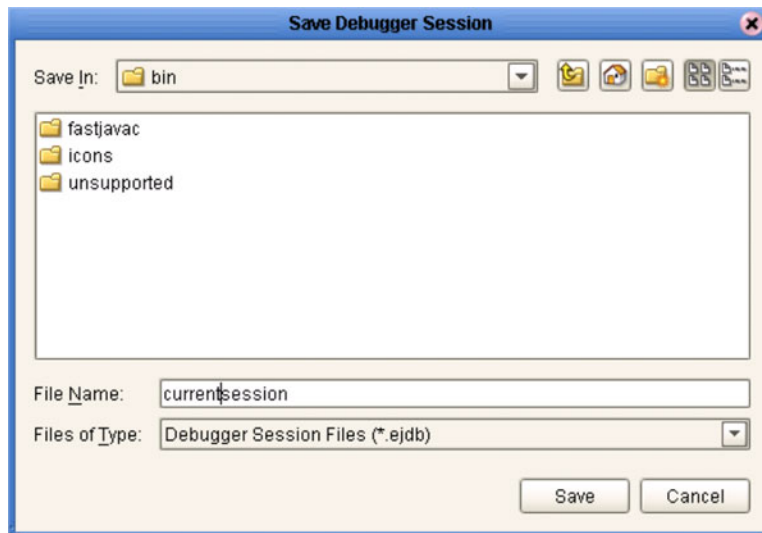
Figure 199 Evaluate Method



Saving and Resuming Debug Sessions

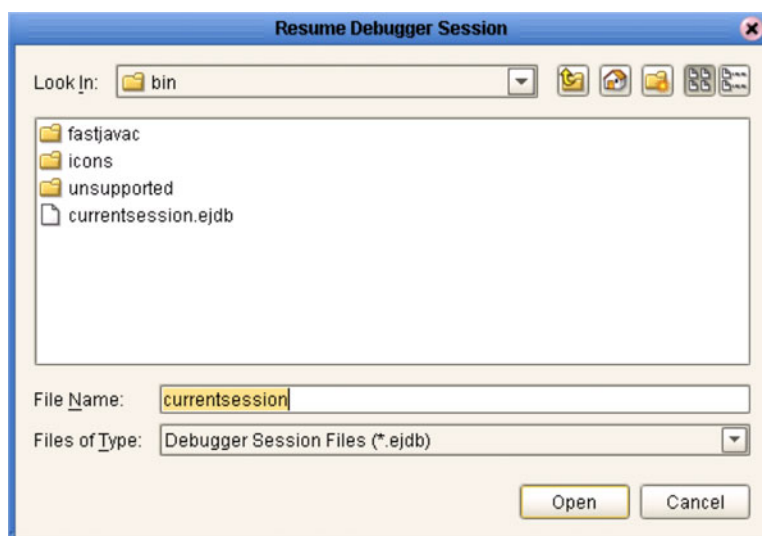
You can pause the debugging process by saving the session to a file. Selecting **Save Session** from the debugger File menu displays the *Save Debugger Session* dialog box (see Figure 200), in which you specify the file name and location.

Figure 200 Save Debugger Session Dialog Box



You can continue a debugging process that was paused by saving to a file. Selecting **Resume Session** from the debugger File menu displays the *Resume Debugger Session* dialog box (see Figure 201), in which you specify the file name and location.

Figure 201 Resume Debugger Session Dialog Box



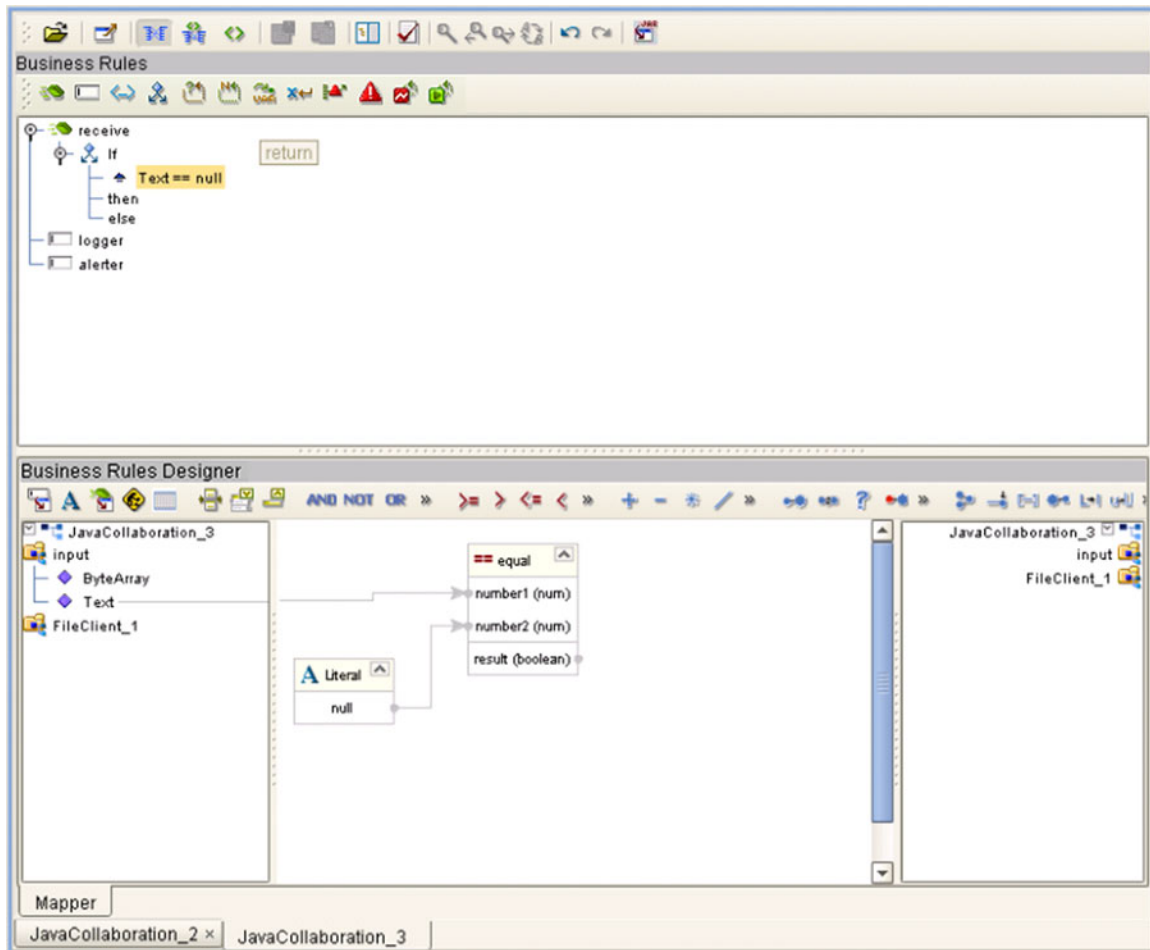
8.12 Creating Alerts

You can create alerts in the Java Collaboration Editor by means of the following procedure.

To create alerts using the JCE

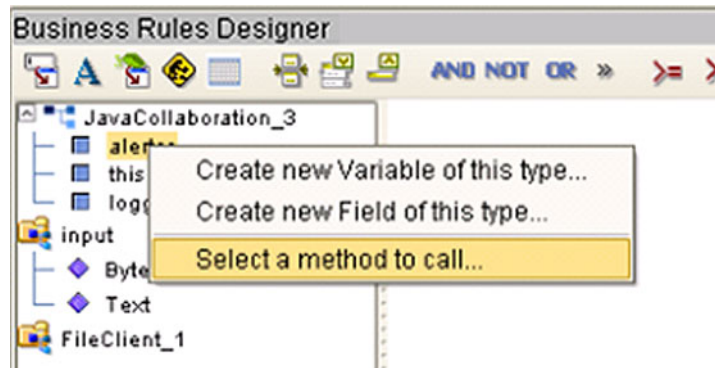
- 1 Create a new Collaboration.
- 2 Add your business rules. As an example, you might perform a test to determine whether or not a file is empty—if it is empty, then raise an alert (see Figure 202).

Figure 202 Empty File Test



- 3 Initiate an alert object.
- 4 Right-click the alerter node of the Collaboration in the left pane of the Business Rules Designer, which displays a menu (see Figure 203).

Figure 203 Alert Menu



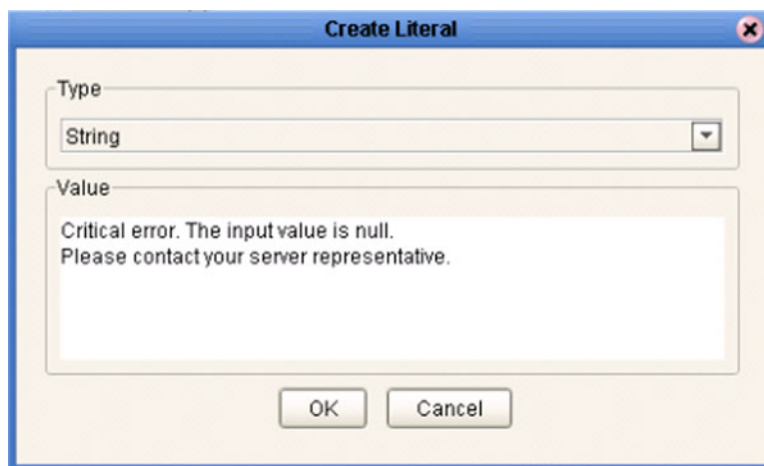
- 5 Click *Select a method to call ...*
- 6 Select the severity of the alert from the selection window (see Figure 204). As an example, we will select **Critical**.

Figure 204 Alert Severity Selection Window



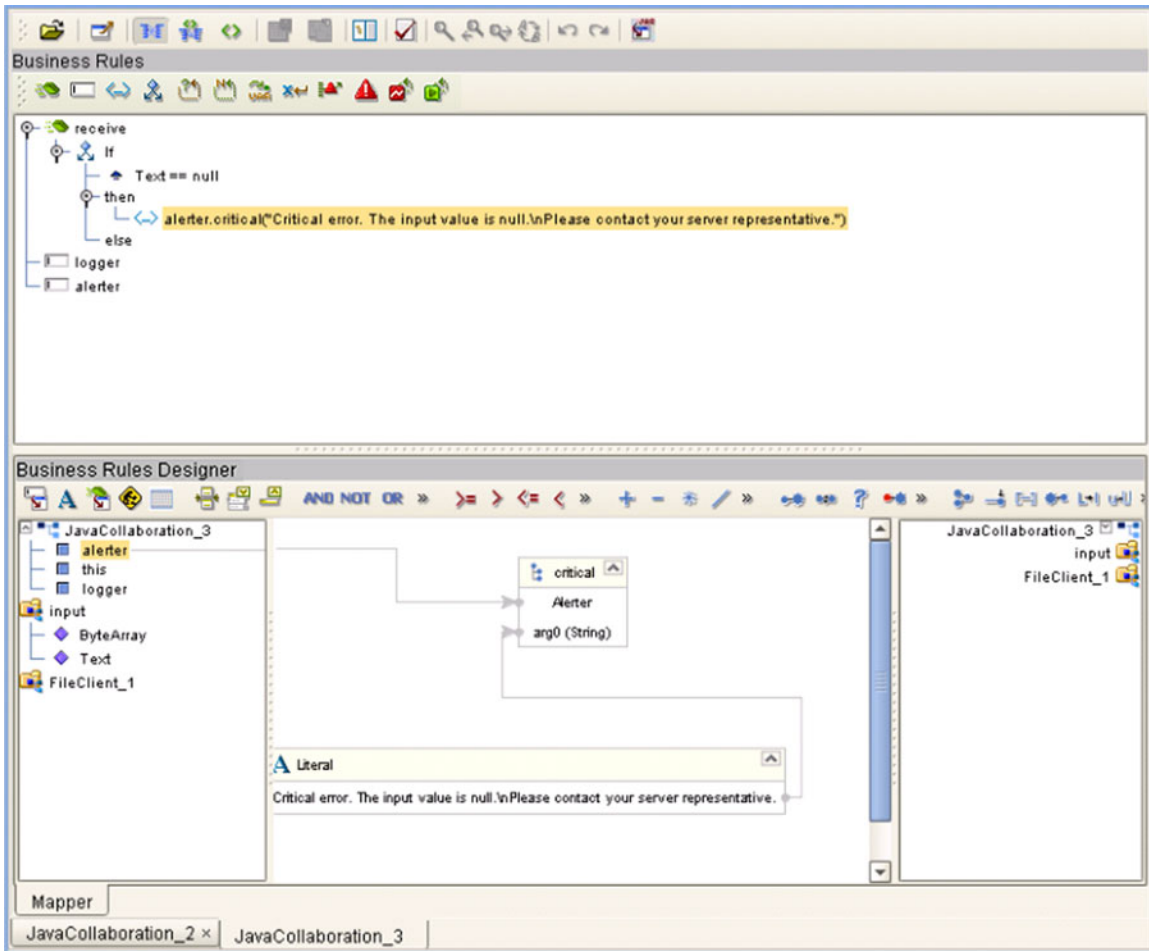
- 7 Create your alert message, which can be a literal, a constant, or an OTD field name. As an example, we will create a **literal** (see Figure 205).

Figure 205 Create Literal Dialog Box



- 8 Pass the alert message to the alert event by dragging the message to the argument of the alerter object (see Figure 206).

Figure 206 Pass Alert Message to Object Argument



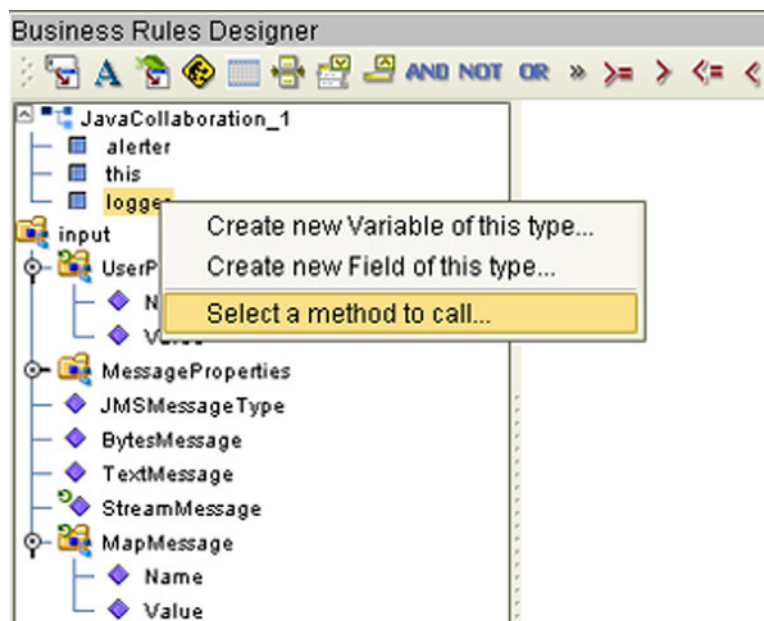
8.13 Creating Log Entries

You can create log file entries in the Java Collaboration Editor by means of the following procedure.

To create log entries using the JCE

- 1 Create a new Collaboration.
- 2 Add your Collaboration rules and initiate a logging event.
- 3 Right-click the **logger** node for the Collaboration in the left pane of the Business Rules Designer, which displays a context menu (see Figure 207).

Figure 207 Logging Menu



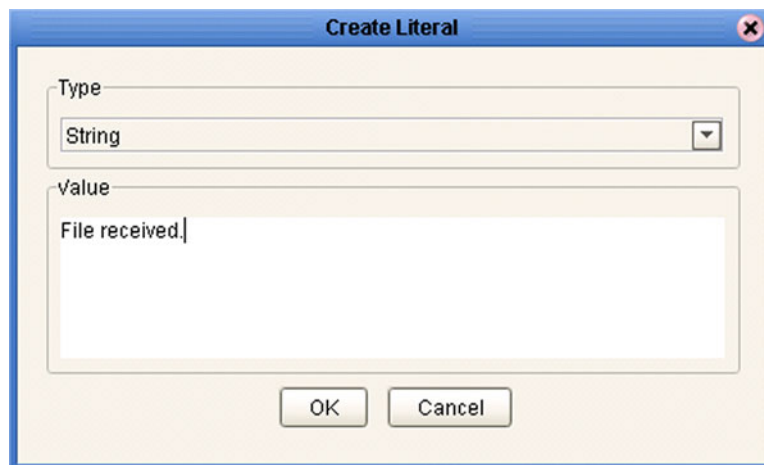
- 4 Click *Select a method to call ...*
- 5 Select the logging level with desired method from the selection window (see Figure 208). As an example, we will select **Debug**.

Figure 208 Logging Level/Method Selection Window



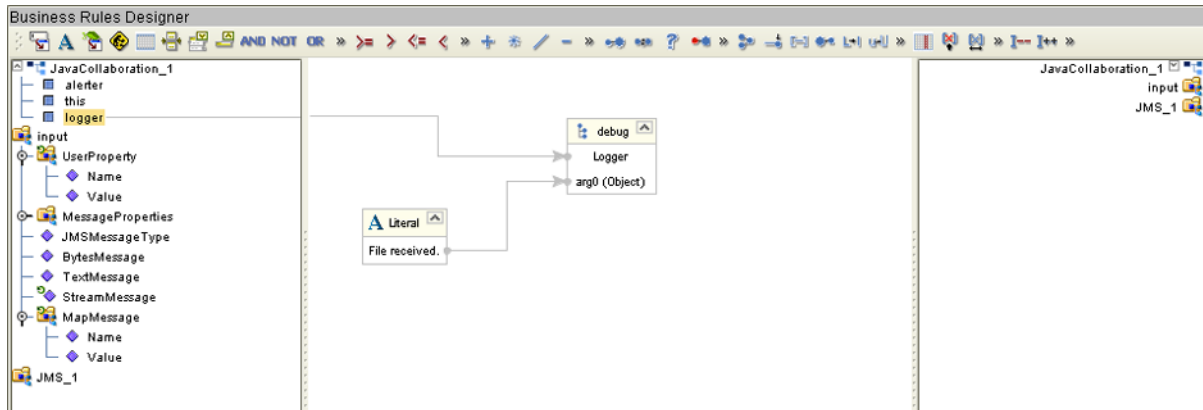
- 6 Create your log message, which can be a literal, a constant, or an OTD field name. As an example, we will create a **literal** (see Figure 209).

Figure 209 Create Literal Dialog Box



- 7 Pass the log message to the logging event by dragging the message to the argument of the logging object (see Figure 210).

Figure 210 Pass Log Message to Object Argument



XSLT Collaboration Definitions

This chapter describes the process for building XSLT Collaboration Definitions.

9.1 Overview

Collaborations use Collaboration Definitions to define how data should be routed between Project components. Collaborations also define how databases should be queried in response to requests and how APIs to one or more applications should be invoked. Collaborations are used when data translation is required.

The Enterprise Designer includes two tools, the XSLT Collaboration Wizard and XSLT Collaboration Editor, that are used to create and customize your XSLT Collaboration Definitions. You must have OTDs available to use as the foundation for creating an XSLT Collaboration Definition. See [Object Type Definitions](#) on page 120 for more details.

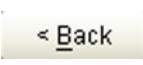




Important: *If you delete an OTD in the Project Explorer, any XSLT Collaboration Definitions that have been built using that OTD will be affected. It is recommended that you run the Impact Analyzer before attempting to delete any OTDs (see [Impact Analyzer](#) on page 69).*

Note: *Any changes made to the names of the Collaborations should be done when the Collaboration is created. If the name is changed later, the Collaboration should be opened, regenerated if applicable, and saved again. For safe measure, this should also be done before creating the Connectivity Map and Deployment Profile.*

9.2 Using the XSLT Collaboration Wizard

The XSLT Collaboration Wizard guides you through the initial phases of creating an XSLT Collaboration Definition, and then invokes the XSLT Collaboration Editor. The user interface is highly self-explanatory, but details of the navigation buttons are listed in Table 41 for your reference.

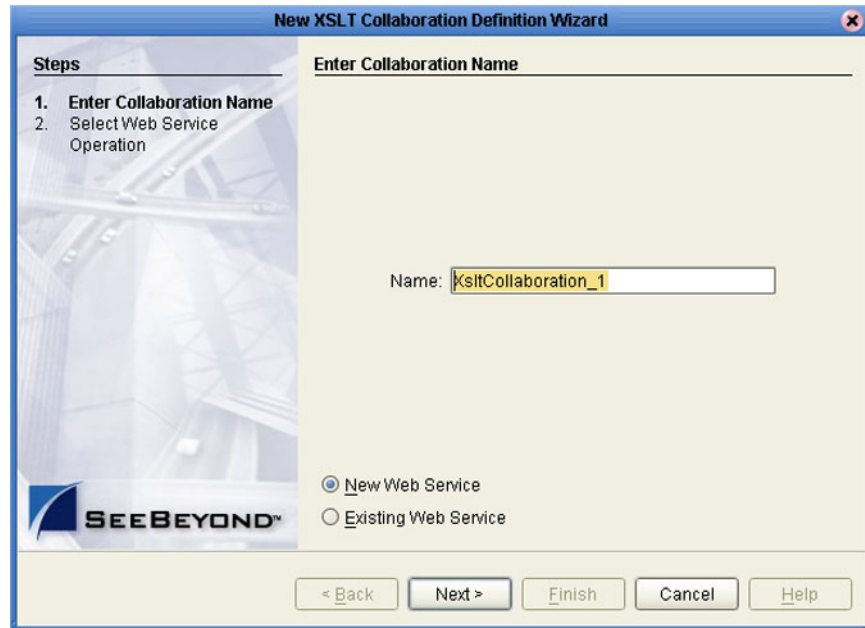
Table 41 Wizard Navigation Buttons

Button	Function
	Returns to the previous step in the wizard. This button is disabled on the first step.
	Goes to the next step in the wizard. This button is disabled on the last step.
	Saves all Collaboration Definition settings and closes the wizard. This button is only enabled on the last step.
	Closes the wizard without saving the Collaboration Definition.
	Displays the online help documentation for the Collaboration Definition Wizard dialog box.

9.2.1 Creating an XSLT Collaboration Definition

- 1 Right-click on a Project in the Enterprise Explorer to display the Project context menu.
- 2 Select **New XSLT Collaboration Definition** to invoke the XSLT Collaboration Wizard.
- 3 Enter a **Name** for your Collaboration, as shown in Figure 211.

Figure 211 XSLT Collaboration Wizard Dialog Box



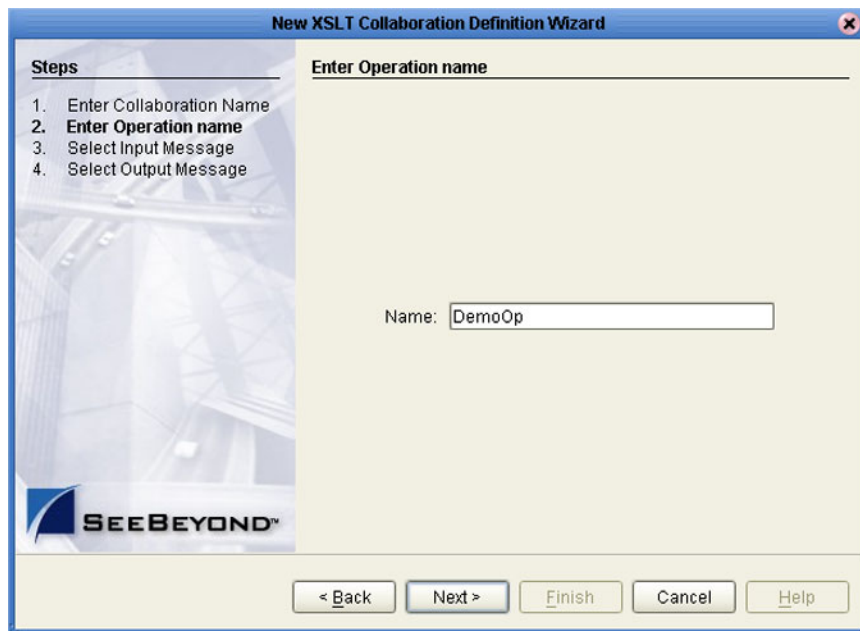
- 4 Select a Web service, which can be either:
 - A New Web Service.
 - An Existing Web Service (for example, an eInsight process or a Java Web Service Operation).
- 5 Click **Next** to proceed to the next Wizard dialog, which is dependent upon your Web Service selection.

New Web Service

If you selected a New Web Service, you will be presented with the following set of Wizard dialogs.

- 1 Enter an operation name, as shown in Figure 212. This will become the *method* that can be used to invoke the XSLT Collaboration as a Web service.

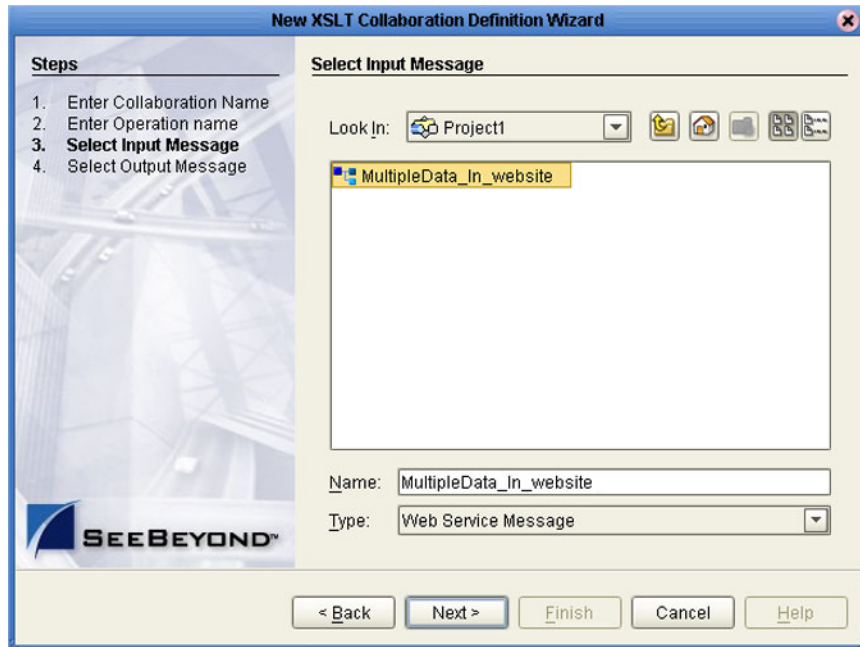
Figure 212 New Web Service: Operation Name



- 2 Click **Next** to proceed to the next Wizard dialog.

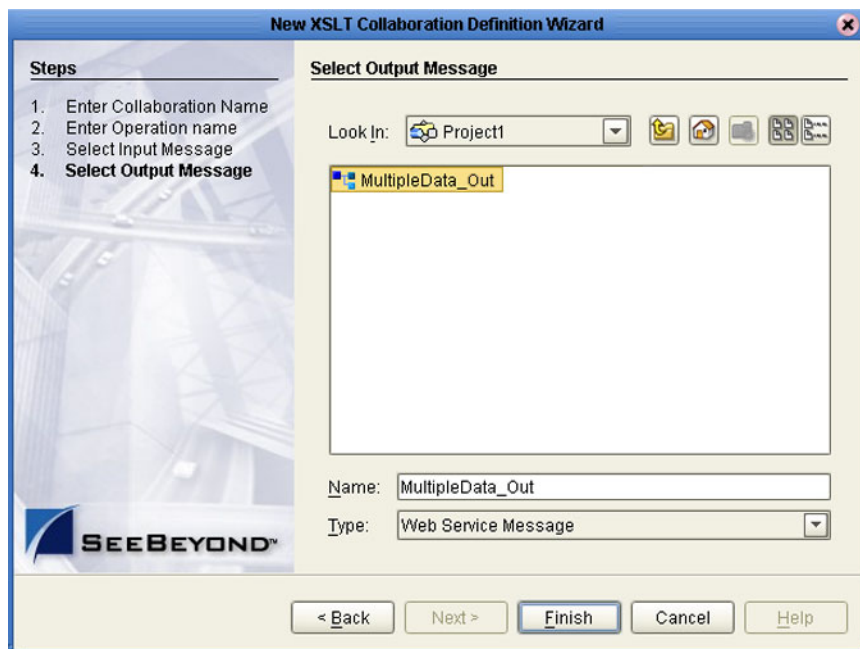
- 3 Select the input Web service message, as shown in Figure 213.

Figure 213 New Web Service: Input Message



- 4 Click **Next** to proceed to the next Wizard dialog.
- 5 Select the output Web service message, as shown in Figure 214.

Figure 214 New Web Service: Output Message



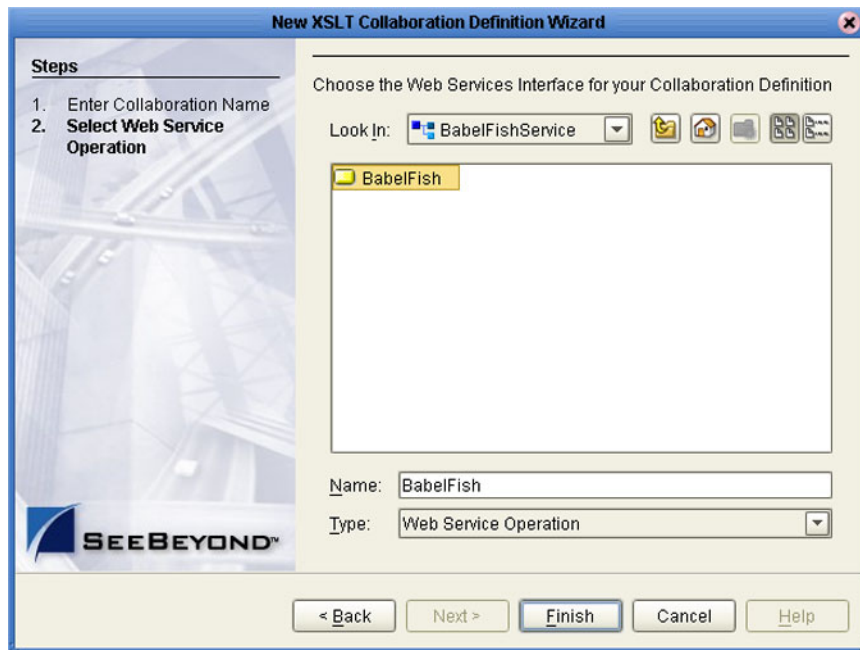
- 6 Click **Finish** to proceed to the XSLT Collaboration Editor.

Existing Web Service

If you selected an Existing Web Service, you will be presented with the Wizard dialog shown in Figure 215.

- 1 Select a Web service operation, which can be either:
 - An installed ICAN Web Service.
 - A custom Web Service (for example, something that has been created in an eGate Project).

Figure 215 Existing Web Service: Select Operation



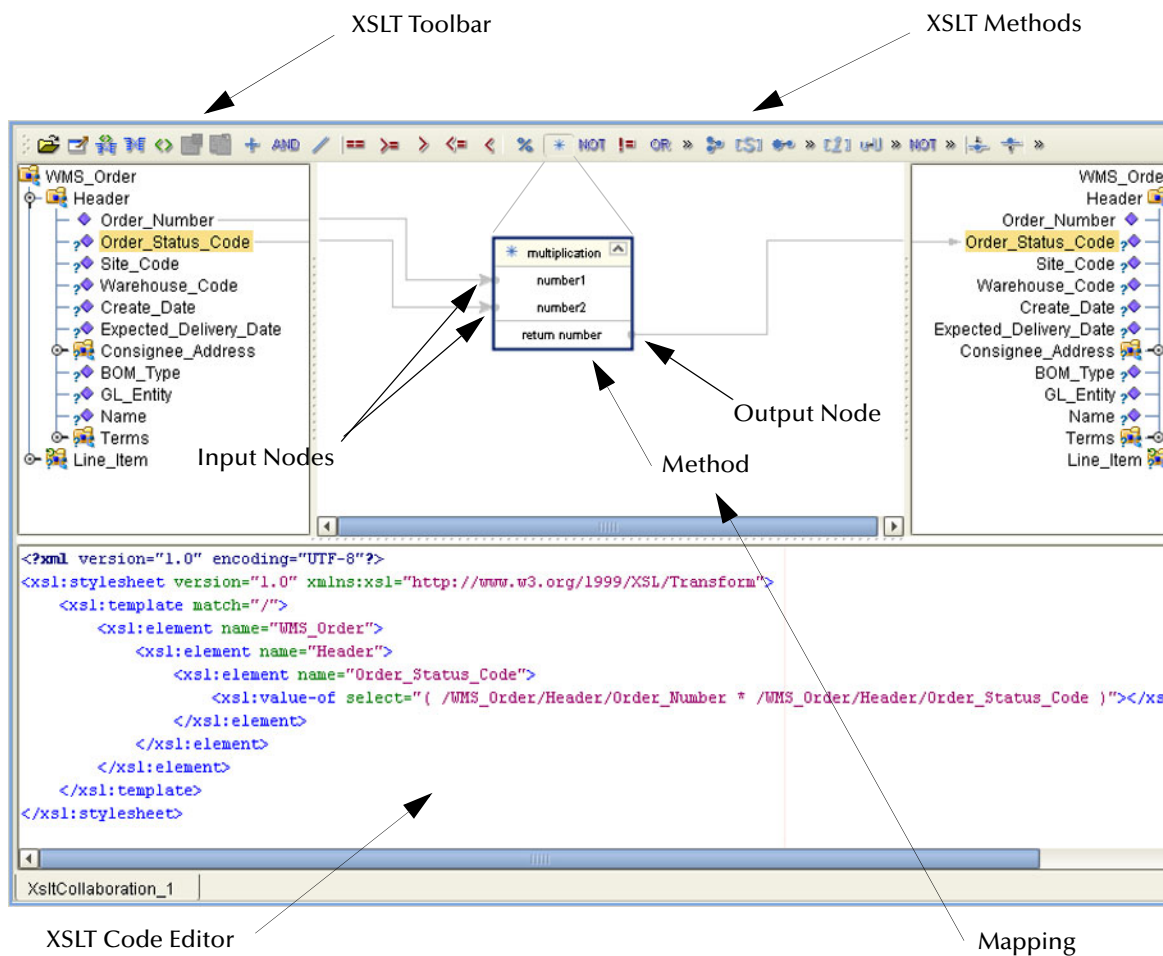
- 2 Click **Finish** to proceed to the XSLT Collaboration Editor.

9.3 Using the XSLT Collaboration Editor

After you have created an XSLT Collaboration Definition using the XSLT Collaboration Wizard, the XSLT Collaboration Editor appears in the Editor panel of the Enterprise Designer. You can also invoke the XSLT Collaboration Editor by selecting **Open** in the context menu for an existing XSLT Collaboration Definition in the Enterprise Explorer.

You can use the XSLT Collaboration Editor to customize Collaboration Definitions. The components of this window are identified in Figure 216. This figure also shows how a Multiplication method displays in the Mapping area after it is dragged down from the Method Palette area and mapped to fields in the left and right panels.








Figure 216 XSLT Collaboration Editor



The XSLT Mapping panel is used to map fields and add methods to the Collaboration Definition. At the top left of the Mapping panel is the toolbar, containing icons as described in Table 42. At the top right of the Mapping area is the XSLT Method Palette, which contains a collection of XSLT methods. The XSLT Code Editor panel allows you to view, enter and edit the XSLT code for the Collaboration Definition.

9.3.1 XSLT Toolbar Icons

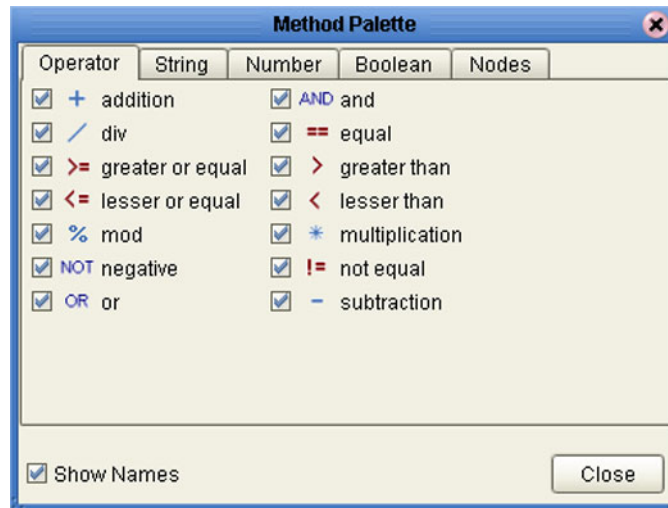
Table 42 XSLT Toolbar Icons

Icon	Command	Function
	Import XSL from a Local File	Displays the Open dialog box, which you can use to locate and select an XSL file to import.
	Save XSL to a Local File	Displays the Save dialog box, which you can use to save the selected Java Collaboration Definition to a file.
	Show Maps and Code	Displays both the Mapping and XSLT Code areas. This is the default view setting.
	Show Mapping Only	Displays the Mapping area and hides the XSLT Code area.
	Show XSLT Code Only	Displays the XSLT Code area and hides the Mapping area.
	Commit Code Changes	Commits changes made to the XSLT code since the last time it was committed. Changes will now be shown in the Mapping panel.
	Roll Back Code Changes	Cancels changes made to the XSLT code since the last time it was committed.

XSLT Method Palette

The XSLT Method Palette includes a series of method icons that you can drag onto the Mapping area. Click the Chevron (>>) to the right of the method groups to display the dialog box shown in Figure 217. Select a check box to add the method to the toolbar; clear a check box to remove the method from the toolbar. The methods are described in detail in [XSLT Methods](#) on page 242.

Figure 217 XSLT Method Palette Dialog Box



XSLT Method Boxes

The method boxes are placed in the mapping area by dragging the corresponding icon from the method palette toolbar. As shown in [Figure 216 on page 239](#), the method boxes typically have input and output nodes that you link to fields in the left and right panels, respectively. The method boxes are expanded by default (see Figure 218); you can collapse them (see Figure 219) by clicking the caret icon in the upper right corner of the box, which can be useful when the mapping area becomes crowded. Some boxes expand further as needed to provide additional arguments.

Figure 218 Expanded Method Box

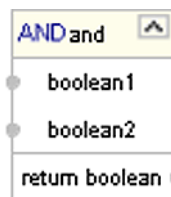
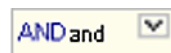


Figure 219 Collapsed Method Box



9.4 XSLT Methods

9.4.1 Operator Methods

Figure 220 Method Palette: Operator Methods

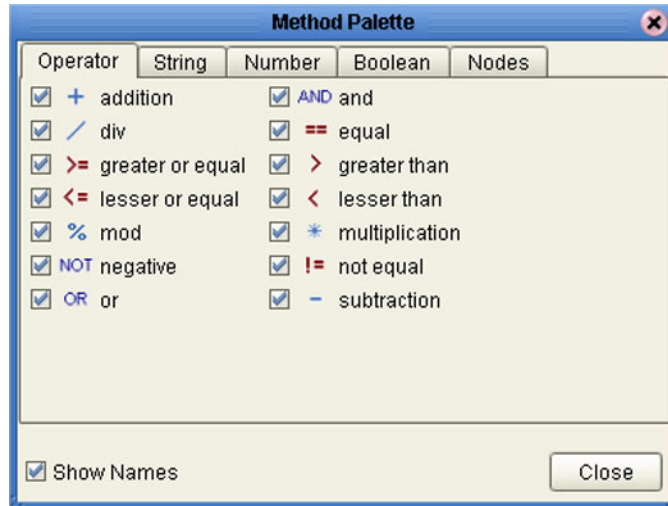


Table 43 XSLT Operator Methods

Method Box	Description/Usage
	The addition method adds the value of <i>number1</i> to the value of <i>number2</i> , returns the sum.
	The AND method returns Boolean true if both <i>boolean1</i> and <i>boolean2</i> are true; otherwise, returns Boolean false.
	The division method divides the value of <i>number1</i> by the value of <i>number2</i> , returns the quotient.

Table 43 XSLT Operator Methods

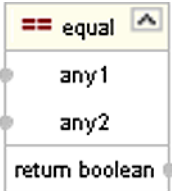
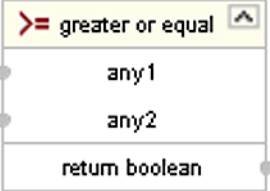
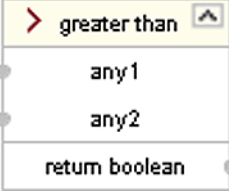
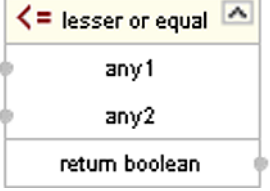
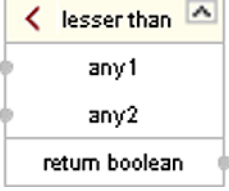
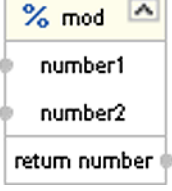
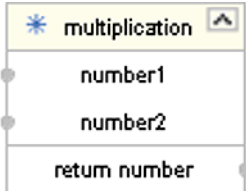
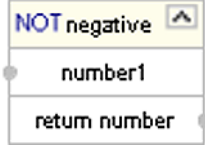
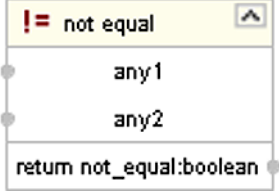
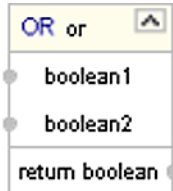
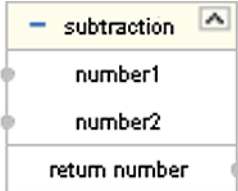
Method Box	Description/Usage
 <p>The method box for 'equal' shows the operator symbol '==', the name 'equal', two input ports labeled 'any1' and 'any2', and a return port labeled 'return boolean'.</p>	<p>The equal method returns Boolean true if <i>any1</i> is equal to <i>any2</i>; otherwise, returns Boolean false.</p>
 <p>The method box for 'greater or equal' shows the operator symbol '>=', the name 'greater or equal', two input ports labeled 'any1' and 'any2', and a return port labeled 'return boolean'.</p>	<p>The greater_or_equal method returns Boolean true if <i>any1</i> is greater than or equal to <i>any2</i>; otherwise, returns Boolean false.</p>
 <p>The method box for 'greater than' shows the operator symbol '>', the name 'greater than', two input ports labeled 'any1' and 'any2', and a return port labeled 'return boolean'.</p>	<p>The greater_than method returns Boolean true if <i>any1</i> is greater than <i>any2</i>; otherwise, returns Boolean false.</p>
 <p>The method box for 'lesser or equal' shows the operator symbol '<=', the name 'lesser or equal', two input ports labeled 'any1' and 'any2', and a return port labeled 'return boolean'.</p>	<p>The lesser_or_equal method returns Boolean true if <i>any1</i> is less than or equal to <i>any2</i>; otherwise, returns Boolean false.</p>
 <p>The method box for 'lesser than' shows the operator symbol '<', the name 'lesser than', two input ports labeled 'any1' and 'any2', and a return port labeled 'return boolean'.</p>	<p>The lesser_than method returns Boolean true if <i>any1</i> is less than <i>any2</i>; otherwise, returns Boolean false.</p>
 <p>The method box for 'modulus' shows the operator symbol '%', the name 'mod', two input ports labeled 'number1' and 'number2', and a return port labeled 'return number'.</p>	<p>The modulus method divides the numerical value of <i>number1</i> by the numerical value of <i>number2</i>, and returns the remainder.</p>

Table 43 XSLT Operator Methods

Method Box	Description/Usage
 <p>The method box for 'multiplication' features a yellow header with a blue asterisk icon and the text 'multiplication'. Below the header are two input fields labeled 'number1' and 'number2', each with a small grey dot on its left side. At the bottom is an output field labeled 'return number' with a small grey dot on its right side.</p>	<p>The multiplication method multiplies the value of <i>number1</i> by the value of <i>number2</i>, returns the product.</p>
 <p>The method box for 'NOT negative' features a yellow header with the text 'NOT negative'. Below the header is one input field labeled 'number1' with a small grey dot on its left side. At the bottom is an output field labeled 'return number' with a small grey dot on its right side.</p>	<p>The negative method returns the arithmetic negation of <i>number1</i>.</p>
 <p>The method box for 'not equal' features a yellow header with a red '!=' icon and the text 'not equal'. Below the header are two input fields labeled 'any1' and 'any2', each with a small grey dot on its left side. At the bottom is an output field labeled 'return not_equal:boolean' with a small grey dot on its right side.</p>	<p>The not_equal method returns Boolean true if <i>any1</i> is not equal to <i>any2</i>; otherwise, returns Boolean false.</p>
 <p>The method box for 'OR or' features a yellow header with a blue 'OR' icon and the text 'or'. Below the header are two input fields labeled 'boolean1' and 'boolean2', each with a small grey dot on its left side. At the bottom is an output field labeled 'return boolean' with a small grey dot on its right side.</p>	<p>The OR method returns Boolean false if both <i>boolean1</i> and <i>boolean2</i> are false; otherwise, returns Boolean true.</p>
 <p>The method box for 'subtraction' features a yellow header with a blue '-' icon and the text 'subtraction'. Below the header are two input fields labeled 'number1' and 'number2', each with a small grey dot on its left side. At the bottom is an output field labeled 'return number' with a small grey dot on its right side.</p>	<p>The subtraction method subtracts the numerical value of <i>number2</i> from the numerical value of <i>number1</i>, returns the difference.</p>

9.4.2 String Methods

Figure 221 Method Palette: String Methods

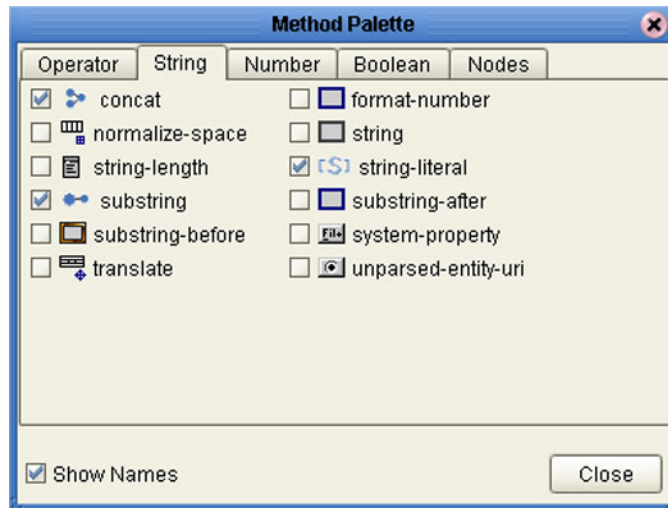


Table 44 XSLT String Methods

Method Box	Description/Usage
	<p>The concat method returns the string created by concatenating <i>string2</i> to the end of <i>string1</i>, and <i>string3*</i> (if present) to the end of <i>string2</i>.</p>
	<p>The format-number method converts its first argument (<i>number1</i>) to a string using the format pattern string specified by the second argument (<i>string2</i>) and the decimal-format named by the third argument (<i>string3</i>) or the default decimal-format, if there is no third argument.</p>
	<p>The normalize-space method returns the argument <i>string1?</i> with whitespace normalized by stripping leading and trailing whitespace and replacing sequences of whitespace characters by a single space. If the argument is omitted, it defaults to the string-value of the context node.</p>

Table 44 XSLT String Methods

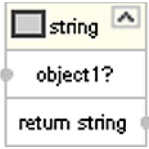
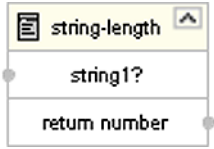

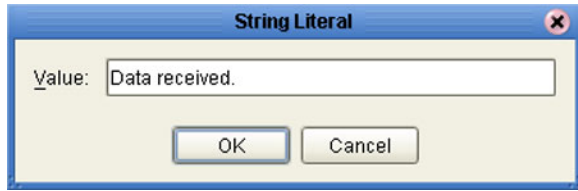
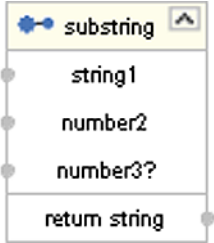
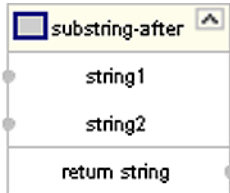
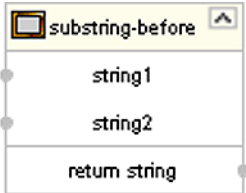
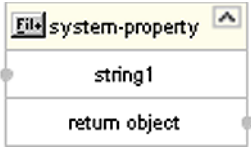

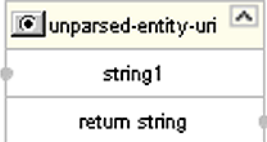
Method Box	Description/Usage
	<p>The string method returns a string representation of the input object.</p>
	<p>The string-length method returns the number of characters in the string.</p>
	<p>Dragging the icon first displays the <i>String Literal</i> dialog box, where you enter the literal value, such as “Data received”:</p> <div data-bbox="711 800 1284 989" style="text-align: center;">  </div> <p>The string-literal method then returns a string having the specified value.</p>
	<p>The substring method returns the substring of the first argument (<i>string1</i>) starting at the position specified in the second argument (<i>number2</i>) with length specified in the third argument (<i>number3?</i>). If the third argument is not specified, it returns the substring starting at the position specified in the second argument and continuing to the end of the string.</p>
	<p>The substring-after method returns the substring of the first argument (<i>string1</i>) that follows the first occurrence of the second argument (<i>string2</i>) in the first argument string. Returns an empty string if the first argument string does not contain the second argument string.</p>
	<p>The substring-before method returns the substring of the first argument (<i>string1</i>) that precedes the first occurrence of the second argument (<i>string2</i>) in the first argument string. Returns an empty string if the first argument string does not contain the second argument string.</p>

Table 44 XSLT String Methods

Method Box	Description/Usage
 <p>The diagram shows a method box titled 'system-property'. It has a single input parameter 'string1' and a return value 'return object'.</p>	<p>The system-property method returns an object representing the value of the system property identified by <i>string1</i>. If there is no such system property, the empty string should be returned.</p>
 <p>The diagram shows a method box titled 'translate'. It has three input parameters: 'string1', 'string2', and 'string3'. The return value is 'return string'.</p>	<p>The translate method returns the first argument (<i>string1</i>) with occurrences of characters in the second argument (<i>string2</i>) replaced by the character at the corresponding position in the third argument (<i>string3</i>). If a character occurs more than once in the second argument (<i>string2</i>), then the first occurrence determines the replacement character. If the third argument (<i>string3</i>) is longer than the second argument (<i>string2</i>), then excess characters are ignored. Refer to the W3C <i>XML Path Language</i> documentation for additional conditions.</p>
 <p>The diagram shows a method box titled 'unparsed-entity-uri'. It has a single input parameter 'string1' and a return value 'return string'.</p>	<p>The unparsed-entity-uri method returns the URI of the unparsed entity with the specified name (<i>string1</i>) in the same document as the context node. It returns an empty string if there is no such entity.</p>

9.4.3 Number Methods

Figure 222 Method Palette: Number Methods

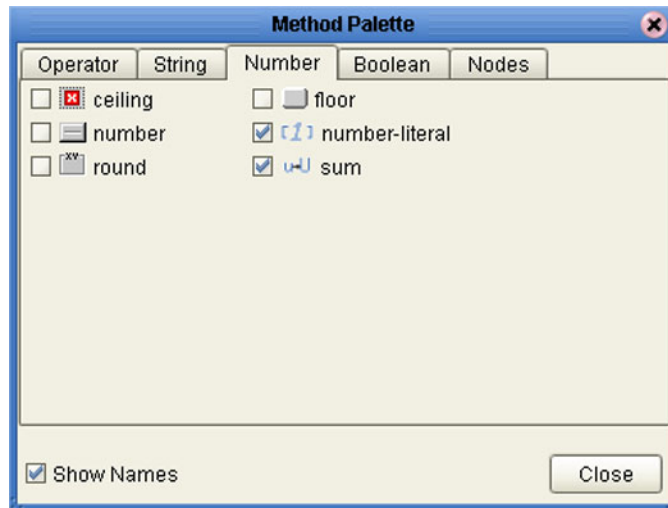
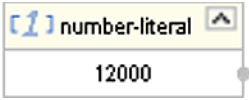
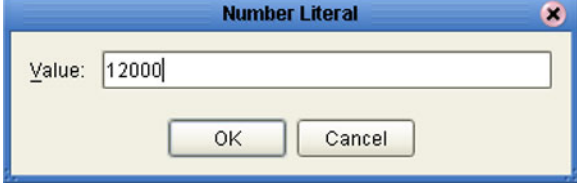

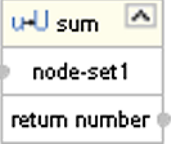


Table 45 XSLT Number Methods

Method Box	Description/Usage
	<p>The ceiling method returns the smallest (closest to negative infinity) number that is not less than the argument and that is an integer.</p>
	<p>The floor method returns the largest (closest to positive infinity) number that is not greater than the argument and that is an integer.</p>
	<p>The number method converts its argument (<i>object1</i>) to a number. An object of a type other than the four basic types is converted to a number in a way that is dependent on that type.</p>

Table 45 XSLT Number Methods

Method Box	Description/Usage
 <p>The method box for <code>number-literal</code> shows a blue icon with the number 1, the text "number-literal", and a value of "12000" in a text field.</p>	<p>Dragging the icon first displays the <i>Number Literal</i> dialog box, where you enter the literal value, such as "12000":</p>  <p>The number-literal method then returns a number having the specified value.</p>
 <p>The method box for <code>round</code> shows an icon with "xv", the text "round", an input field for "number1", and a return value of "return number".</p>	<p>The round method returns the number that is closest to the argument <i>number1</i> and that is an integer.</p> <ul style="list-style-type: none"> ▪ If there are two such numbers, then the one that is closest to positive infinity is returned. ▪ If the argument is not a number (NaN), then NaN is returned. ▪ If the argument is positive infinity, then positive infinity is returned. ▪ If the argument is negative infinity, then negative infinity is returned. ▪ If the argument is positive zero, then positive zero is returned. ▪ If the argument is negative zero, then negative zero is returned. ▪ If the argument is less than zero, but greater than or equal to -0.5, then negative zero is returned.
 <p>The method box for <code>sum</code> shows a blue icon with a summation symbol, the text "sum", an input field for "node-set1", and a return value of "return number".</p>	<p>The sum method returns the sum, for each node in the argument <i>node-set1</i>, of the result of converting the string-values of the node to a number.</p>

9.4.4 Boolean Methods

Figure 223 Method Palette: Boolean Methods

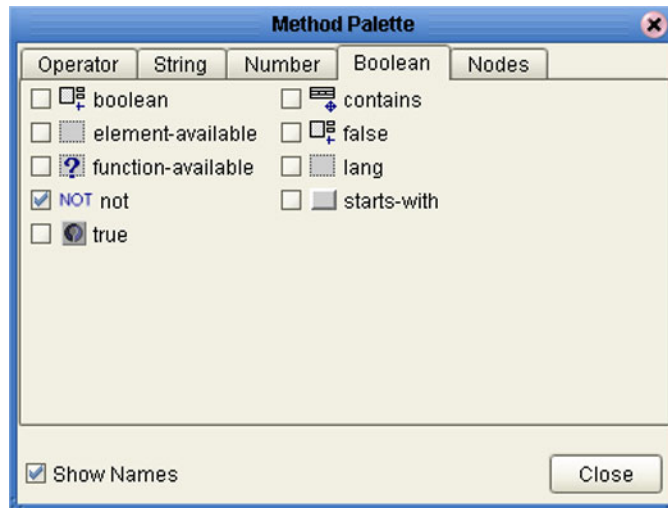
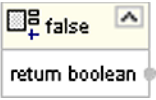
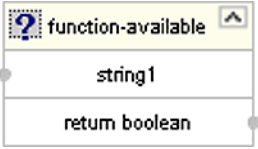

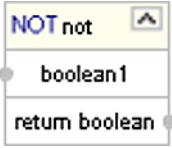
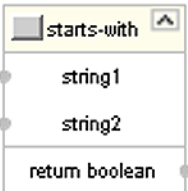
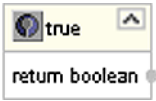


Table 46 XSLT Boolean Methods

Method Box	Description/Usage
	<p>The boolean method converts the argument <i>object1</i> to a Boolean true or false as follows:</p> <ul style="list-style-type: none"> ▪ A number is true if and only if it is neither \pmzero nor NaN (not a number). ▪ A node-set is true if and only if it is non-empty. ▪ A string is true if and only if it is non-zero. ▪ An object of a type other than the four basic types is converted to a Boolean in a way that is dependent on that type.
	<p>The contains method returns Boolean true if the first argument (<i>string1</i>) contains the second argument (<i>string2</i>); if not, returns Boolean false.</p>
	<p>The argument <i>string1</i> must be a QName, which is expanded into an expanded-name using the namespace declarations in scope for the expression. The element-available method returns true if and only if the expanded-name is the name of an instruction. If the expanded-name has a namespace URI equal to the XSLT namespace URI, then it refers to an element defined by XSLT. Otherwise, it refers to an extension element. If the expanded-name has a null namespace URI, the element-available function will return false.</p>

Table 46 XSLT Boolean Methods

Method Box	Description/Usage
 <p>The method box for 'false' shows a yellow header with a plus icon and the text 'false'. Below the header is a white box containing the text 'return boolean'.</p>	<p>The false method returns Boolean false.</p>
 <p>The method box for 'function-available' shows a yellow header with a magnifying glass icon and the text 'function-available'. Below the header is a white box containing the text 'string1' and 'return boolean'.</p>	<p>The argument <i>string1</i> must be a QName, which is expanded into an expanded-name using the namespace declarations in scope for the expression. The function-available method returns true if and only if the expanded-name is the name of a function in the function library. If the expanded-name has a non-null namespace URI, then it refers to an extension function; otherwise, it refers to a function defined by XPath or XSLT.</p>
 <p>The method box for 'lang' shows a yellow header with the text 'lang'. Below the header is a white box containing the text 'string1' and 'return boolean'.</p>	<p>The language method returns Boolean true or false depending upon whether the language of the context node as specified by <i>xml:lang</i> attributes is the same as, or is a sub-language of, the language specified by the argument string (<i>string1</i>). Returns Boolean false if the attribute <i>xml:lang</i> does not exist.</p>
 <p>The method box for 'NOT not' shows a yellow header with a blue 'NOT' icon and the text 'not'. Below the header is a white box containing the text 'boolean1' and 'return boolean'.</p>	<p>The NOT method returns the inverse of <i>boolean1</i>.</p>
 <p>The method box for 'starts-with' shows a yellow header with the text 'starts-with'. Below the header is a white box containing the text 'string1', 'string2', and 'return boolean'.</p>	<p>The starts-with method returns Boolean true if the first argument (<i>string1</i>) starts with the second argument (<i>string2</i>); if not, returns Boolean false.</p>
 <p>The method box for 'true' shows a yellow header with a magnifying glass icon and the text 'true'. Below the header is a white box containing the text 'return boolean'.</p>	<p>The true method returns Boolean true.</p>

9.4.5 Nodes Methods

Figure 224 Method Palette: Nodes Methods

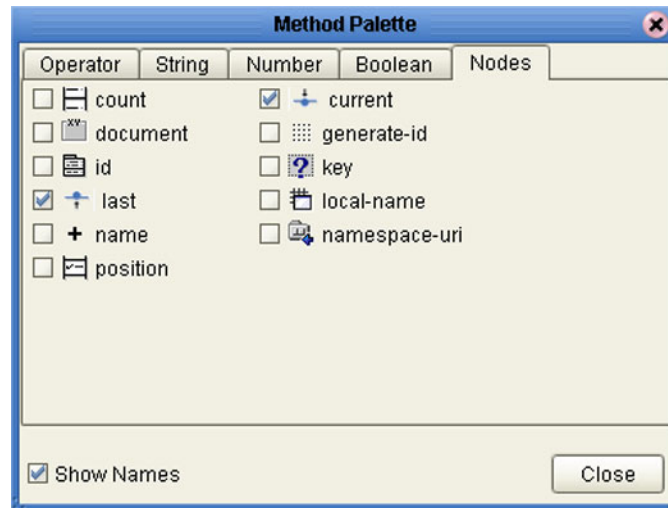


Table 47 XSLT Nodes Methods

Method Box	Description/Usage
	<p>The count method returns the number of nodes in the argument <i>node-set1</i>.</p>
	<p>The current method returns a node-set that has the current node as its only member.</p>
	<p>The document method allows access to XML documents other than the main source document.</p> <ul style="list-style-type: none"> ▪ When the method has exactly one argument and the argument is a node-set, then for each node in the argument node-set, the result is the union of the result of calling the document function with the first argument being the string-value of the node, and the second argument being a node-set with the node as its only member. ▪ When the method has two arguments and the first argument is a node-set, then for each node in the argument node-set, the result is the union of the result of calling the document function with the first argument being the string-value of the node, and with the second argument being the second argument passed to the document function. When the first argument is not a node-set, the first argument is converted to a string as if by a call to the string function. This string is treated as a URI reference; the resource identified by the URI is retrieved.

Table 47 XSLT Nodes Methods

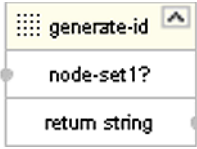
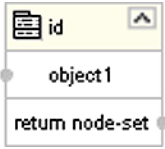
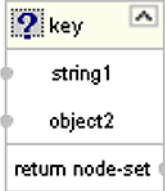
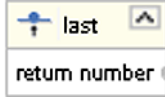
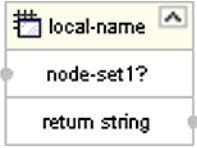
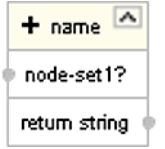
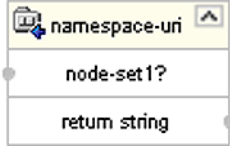
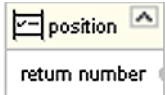
Method Box	Description/Usage
	<p>The generate-id method returns a string that uniquely identifies the node in the argument <i>node-set1?</i> that is first in document order. The unique identifier must consist of ASCII alphanumeric characters and must start with an alphabetic character. Thus, the string is syntactically an XML name.</p>
	<p>When the argument <i>object1</i> is not of type node-set, the id method converts the argument to a string as if by a call to the string function; the string is split into a whitespace-separated list of tokens; the result is a node-set containing the elements in the same document as the context node that have a unique ID equal to any of the tokens in the list.</p>
	<p>The key method does for keys what the id function does for IDs. The value of the first argument (<i>string1</i>), which specifies the name of the key, must be a QName—which is expanded into an expanded-name using the namespace declarations in scope for the expression.</p> <ul style="list-style-type: none"> ▪ When the second argument is of type node-set, then the result is the union of the result of applying the key function to the string value of each of the nodes in the argument node-set. ▪ When the second argument is of any other type, the argument is converted to a string as if by a call to the string function; it returns a node-set containing the nodes in the same document as the context node that have a value for the named key equal to this string.
	<p>The last method returns a number equal to the context size from the expression evaluation context.</p>
	<p>The local-name method returns the local part of the expanded-name of the node in the argument <i>node-set1?</i> that is first in document order. If the argument is empty or the first node has no expanded-name, an empty string is returned. If the argument is omitted, it defaults to a node-set with the context node as its only member.</p>
	<p>The name method returns a string containing a QName representing the expanded-name of the node in the argument <i>node-set1?</i> that is first in document order. If the argument is empty or the first node has no expanded-name, an empty string is returned. If the argument it omitted, it defaults to a node-set with the context node as its only member.</p>
	<p>The namespace-uri method returns the namespace URI of the expanded-name of the node in the argument <i>node-set1?</i> that is first in document order. If the argument is empty, the first node has no expanded-name, or the namespace URI of the expanded-name is null, an empty string is returned. If the argument is omitted, it defaults to a node-set with the context node as its only member.</p>

Table 47 XSLT Nodes Methods

Method Box	Description/Usage
 The image shows a method box for the 'position' method. It consists of two parts: a top section with a yellow background containing a list icon, the text 'position', and an upward-pointing arrow; and a bottom section with a white background containing the text 'return number' and a small grey circle to its right.	The position method returns a number equal to the context position from the expression evaluation context.

9.5 XSLT Encapsulation

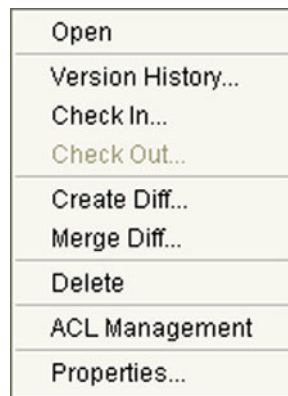
XSLT Collaboration Definitions can be saved to a Diff (.sdf) file. This feature allows two sites working with the same XSLT Collaboration Definition to seamlessly merge changes.

9.5.1 Creating a Modified XSLT Collaboration Definition

This section explains how to create a *different* version of a XSLT Collaboration Definition file.

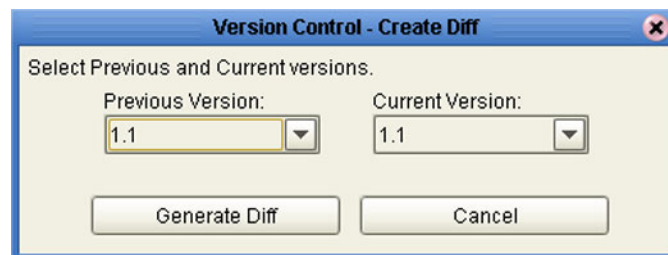
- 1 From Project Explorer, select a **XSLT Collaboration Definition** icon.
- 2 Right-click to display the context menu shown in Figure 225.

Figure 225 XSLT Collaboration Definition Context Menu



- 3 Select **Create Diff** to display the dialog box shown in Figure 226.

Figure 226 Version Control - Create Diff Dialog Box



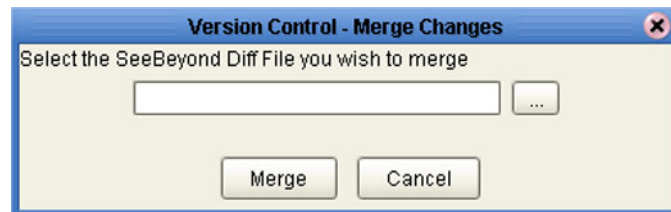
- 4 Click **Generate Diff** to display the *Specify Name ...* dialog box.
- 5 Enter a name for the Diff file in the **File Name** box (or use the default file name).
- 6 Click **Save** to save the Diff file.

9.5.2 Merging Two Versions of a XSLT Collaboration Definition

This section explains how to merge two versions of a XSLT Collaboration Definition.

- 1 From the Project Explorer tab, select a **XSLT Collaboration Definition** icon.
- 2 Right-click to display the XSLT Collaboration Definition context menu.
- 3 Select **Check Out**.
- 4 Right-click to again display the context menu.
- 5 Select **Merge Diff...** to display the dialog box shown in Figure 227.

Figure 227 Version Control - Merge Changes Dialog Box



- 6 Click the **Ellipsis (...)** button to display the *Specify Name ...* dialog box.
- 7 Locate and select the Diff file.
- 8 Click **Open** add the Diff file to the dialog box shown in Figure 227.
- 9 Click **Merge** to merge the Diff file with the corresponding XSLT Collaboration Definition in your Project.
- 10 Resolve any conflicts from the merge.
- 11 **Commit** the merged Collaboration.
- 12 **Save** the merged Collaboration.

Environments

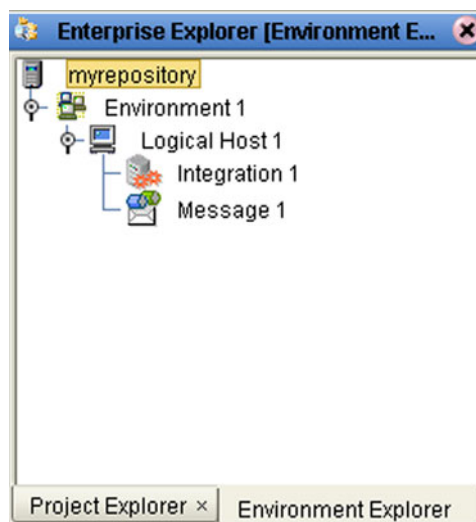
Projects are run within Logical Hosts, which are individual, runtime instances of eGate Integrator. Logical Hosts are defined within *Environments*, which represent the physical resources required to implement the Project. This chapter describes the process of defining eGate Environments, and the components of an Environment.

10.1 Overview

An Environment consists of Logical Hosts capable of hosting eGate components and information about external systems which may be involved with an eGate configuration.

10.2 Environment Explorer

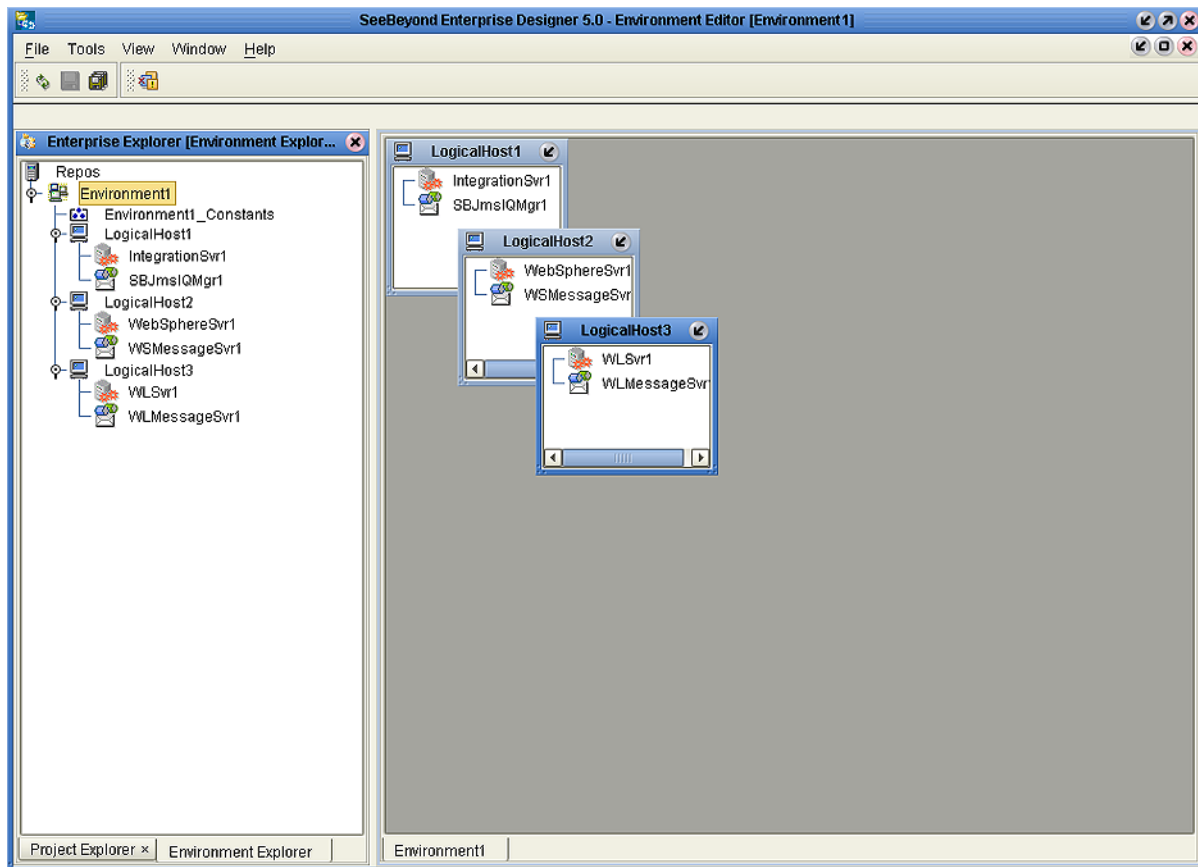
Figure 228 Enterprise Explorer: Environment Explorer View



10.3 Environment Editor

Clicking on an Environment icon in the Environment Explorer invokes the Environment Editor, which provides a canvas in which you can create and customize an Environment (see Figure 229).

Figure 229 Environment Editor











Here you can see the various components (Logical Hosts, servers, and external systems) included in the selected Environment. New Environments are added through the use of the Repository context menu (see [Repository Menu](#) on page 260). Components are added to the Environment by selecting options in the Environment and Logical Host context menus (see [Environment Menu](#) on page 261 and [Logical Host Menu](#) on page 262, respectively).

10.3.1 Environment Explorer Icons

The icons described in Table 48 appear in the Environment Explorer.

Table 48 Environment Icons

Icon	Function
	Represents the Repository , which is the database where all Projects and contents are saved.
	Represents the Environment , which contains Logical Hosts and information about external systems.
	Represents a Logical Host , which contains the software and other installed components that are required at runtime.
	Represents an external file .
	Represents an Environmental constant, which you can use to automate eWay and message destination configuration changes.
	Represents a Scheduler component of an Environment, which you can use to set data transfer to occur at set intervals.
	Represents an Integration Server .
	Represents a JMS IQ Manager or third-party message server , which is used to store and forward eGate system messages.

10.3.2 Context Menus

Right-clicking on a component in the Environment Explorer displays a context menu for that component. Included here are descriptions of options for the following component context menus:

- [Repository Menu](#) on page 260
- [Environment Menu](#) on page 261
- [Logical Host Menu](#) on page 262

Repository Menu

Figure 230 Repository Menu



Table 49 Repository Menu Options

Option	Function
New Environment	Displays a dialog box with which you can create a new Environment.
Configure SNMP Agent	Displays a dialog box in which you can modify the SNMP agent properties.
Save Changes to Repository	Saves all changes made in the Environment Explorer.
Refresh All from Repository	Updates Enterprise Designer with Project/Environment configurations stored in the Repository. (Open editors are not refreshed.)

Environment Menu

Figure 231 Environment Menu

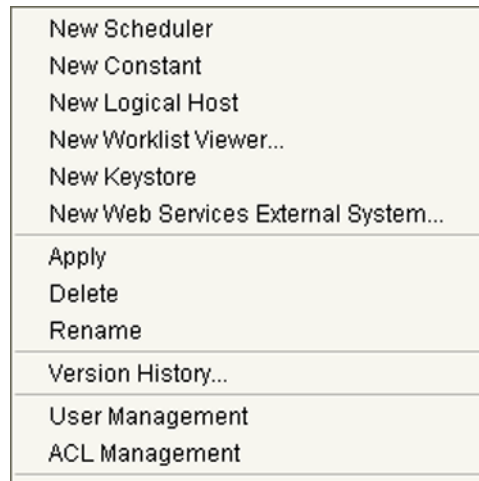


Table 50 Environment Menu Options

Option	Function
New Scheduler	Displays a dialog box with which you can add a new scheduling component to the selected Environment.
New Constant	Displays a dialog box with which you can add a constant to the Environment. See Defining Environmental Constants on page 264.
New Logical Host	Adds a new Logical Host to the selected Environment.
New Worklist Viewer	This option is present only when eInsight Business Process Manager is installed. See the <i>eInsight Business Process Manager User's Guide</i> for information.
New Keystore	Adds a new keystore to the selected Environment.
New Web Service ...	Adds a third-party Web service application to the Project Explorer. See SeeBeyond Web Services on page 102.
Apply	Saves changes to the selected Environment.
Delete	Displays a dialog box in which you confirm that you want to delete the selected Environment. Clicking Yes then deletes the Environment.
Rename	Allows you to rename the selected Environment.
Version History	Displays a dialog box with which you can track the version history for Environments. Version control allows users to maintain multiple versions of the same Environment. See Viewing a Component's Version History on page 73 for more information.
User Management	Displays a dialog box with which you can manage Repository access. See Configuration User Management on page 294.

Table 50 Environment Menu Options

Option	Function
ACL Management	Displays the ACL Properties dialog box, with which you can assign read and/or write privileges to users for the selected Project. See ACL Management on page 321.

Logical Host Menu

Figure 232 Logical Host Menu

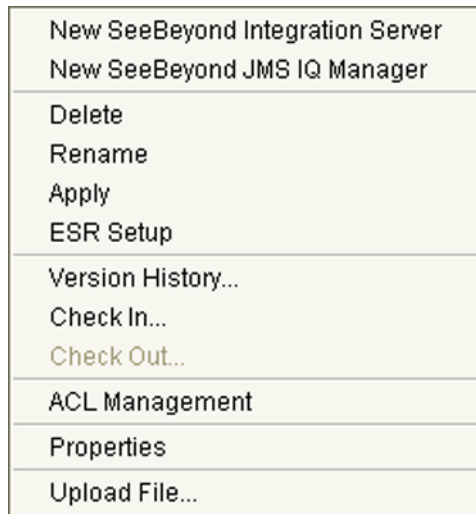


Table 51 Logical Host Menu Options

Option	Function
New SeeBeyond Integration Server	Adds a new SeeBeyond integration server to the selected Logical Host.
New SeeBeyond JMS IQ Manager	Adds a new SeeBeyond JMS IQ Manager to the selected Logical Host.
Delete	Displays a dialog box in which you confirm that you want to delete the selected Logical Host. Clicking Yes then deletes the Logical Host.
Rename	Allows you to rename the selected Logical Host.
Apply	Saves changes to the selected Logical Host.
ESR Setup	Displays a dialog box with which you can select emergency software releases (ESRs) to add to the Environment.
Version History	Displays a dialog box with which you can track the version history for Logical Hosts. Version control allows users to maintain multiple versions of the same Logical Host. See Viewing a Component's Version History on page 73 for more information.

Table 51 Logical Host Menu Options

Option	Function
Check In	Displays a dialog box, with which you can check in the current version of an Environment. Refer to Checking a Component In on page 71 for more details.
Check Out	Displays a dialog box with which you can check out a version of an Environment. See Checking a Component Out on page 72 for more information.
ACL Management	Displays the ACL Properties dialog box, with which you can assign read and/or write privileges to users for the selected Environment See ACL Management on page 321.
Properties	Displays a dialog box with which you can modify the default settings for the selected Logical Host.

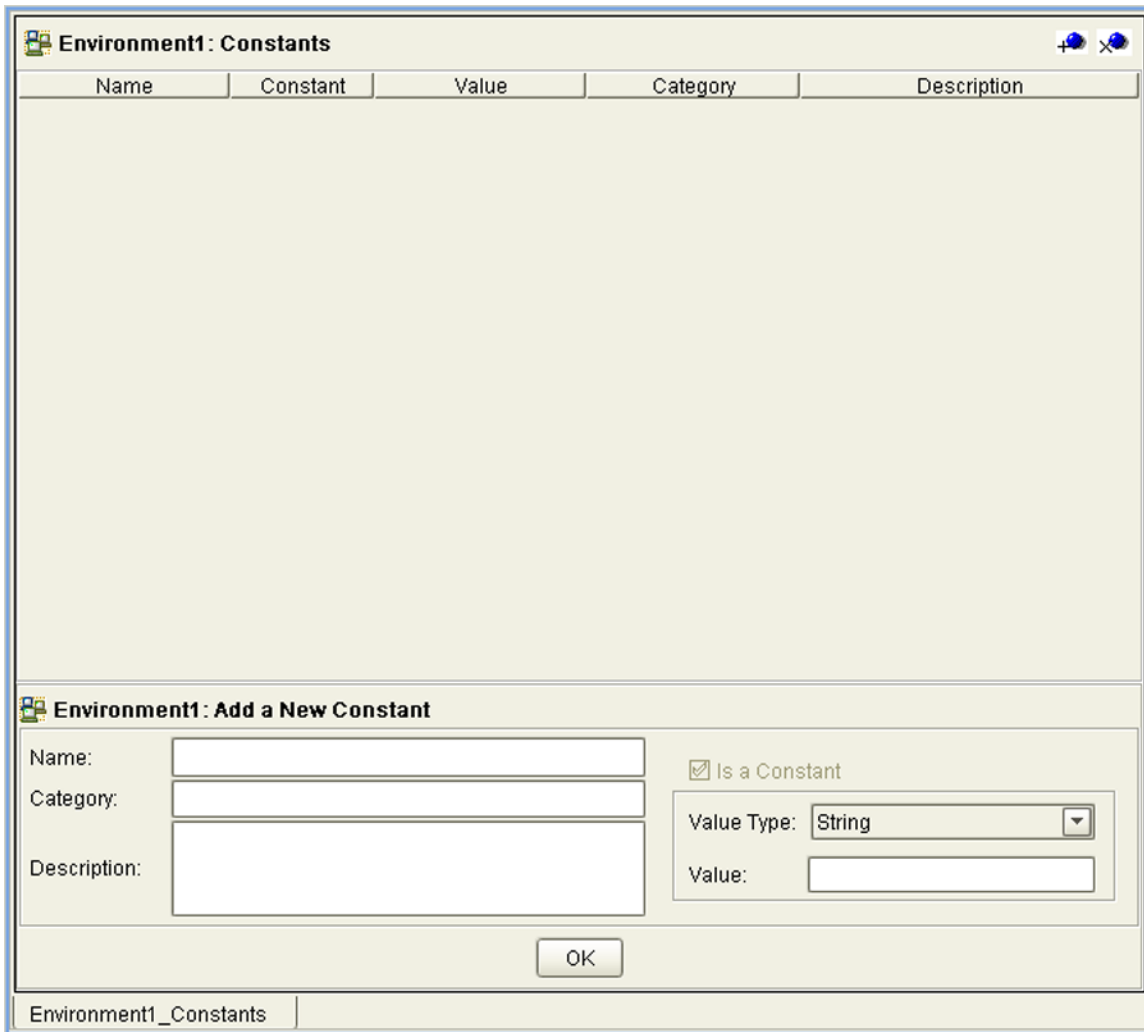
***Note:** If you are using BEA WebLogic and/or IBM WebSphere, the Integration Servers and JMS Message Servers for these products will also appear in the context menu.*

10.4 Defining Environmental Constants

You can define constants for a specific Environment. Environmental constants are name/value pairs that are visible across the Environment. When you create a constant you assign a permanent value to it, which cannot be overridden.



Selecting the **New Constant** option from the Environment context menu displays the Constants panel in the Environment Editor (see Figure 233).

Figure 233 Environmental Constants Editor



New constants are added using the bottom panel in the Constants editor. All constants defined for the specific Environment are listed in the top panel, along with their various properties.

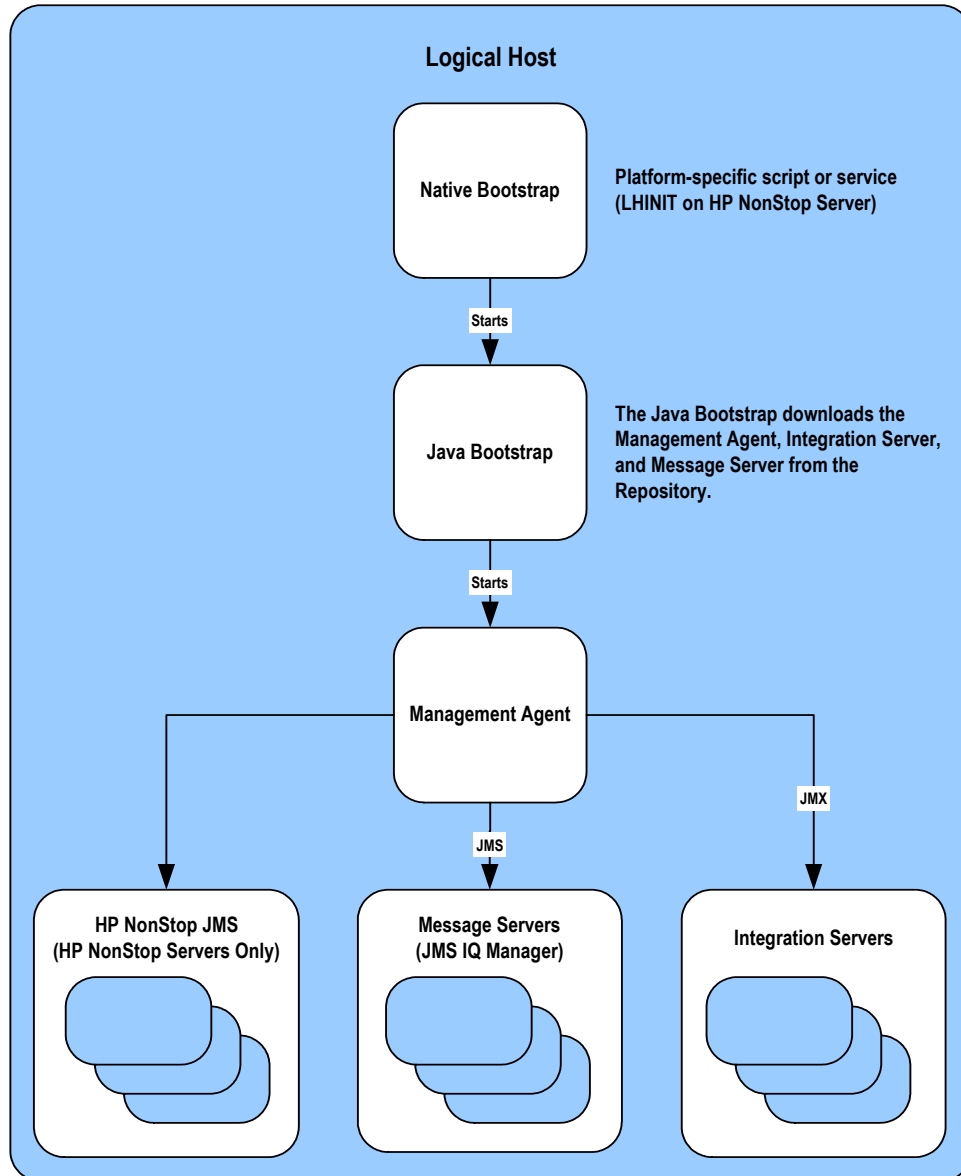
Table 52 Environmental Constants Editor Icons

Icon	Name	Function
	Add a New Constant	Adds a new constant to the list.
	Delete a Highlighted Constant	Deletes the selected constant from the list.

10.5 Logical Hosts

A Logical Host is an instance of the eGate runtime environment that is installed on a host hardware platform. The architecture of a Logical Host is illustrated in [Figure 234 on page 266](#). A Logical Host can be a member of only one Environment, but each Environment can contain multiple Logical Hosts.

Figure 234 Logical Hosts



At run time, the bootstrap script starts the bootstrap Java program that downloads the Management Agent, the Message Server, and the Integration Server from the Repository. The Management Agent is then started, which in turn starts the Message Server(s) and Integration Server(s).

Each Logical Host has a separate bootstrap process. The process is started from a batch file (bootstrap/bin/bootstrap.bat). It finds the Repository via command-line parameters or from the configuration file (bootstrap/config/logical-host.properties). See [Starting the Logical Host](#) on page 275.

Note: *When multiple Logical Hosts reside on a single hardware platform, you must configure the base port numbers (see [Configuring the Base Port Number](#) on page 274).*

10.5.1 Management Agent

The Management Agent is the master service of the Logical host. The service starts the other services on the Logical Host as part of the bootstrap process. The Management Agent also communicates with the Enterprise Manager via JMX (Java Management Extensions) to report the status of the JMS IQ Managers and Integration Servers.

10.5.2 Message Servers

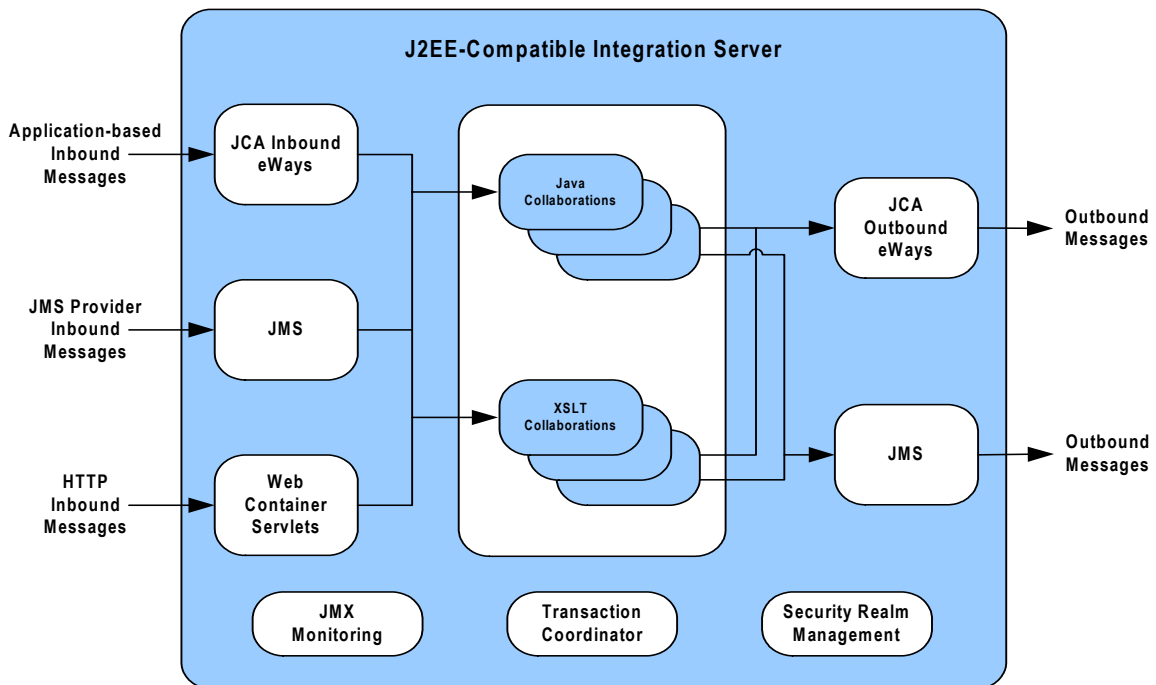
The Logical Host contains one or more Message Servers, which manage JMS topics (publish-and-subscribe messaging) and queues (point-to-point messaging). eGate Integrator includes the SeeBeyond JMS IQ Manager as its Java Messaging Service (JMS) implementation. The JMS IQ Manager conforms to the Java Message specification 1.0.2b, and supports both topic (publish-and-subscribe) and queue (point-to-point) messaging styles.

eGate also includes support for HP NonStop JMS for eGate implementations on HP NonStop Server platforms. Third-party integration servers such as BEA WebLogic and IBM WebSphere incorporate their own Message Servers. For more information on the JMS IQ Manager, and deploying Project components to third-part message servers, see the *eGate Integrator JMS Reference Guide*.

10.5.3 Integration Servers

The Logical Host contains one or more Integration Servers, which are the engines that run eGate Collaborations for processing business logic, and eWays that communicate with external applications. It provides services for security, transactions, business rules execution, and connectivity management. The SeeBeyond Integration Server is based on Java 2 Enterprise Edition (J2EE). eGate Integrator also runs on third-party integration servers such as BEA WebLogic and IBM WebSphere (see [Deploying Projects to Third-Party Servers](#) on page 286).

Figure 235 Integration Server (J2EE Compatible)



The Integration Server runs the JCA eWays and Collaborations. For HTTP/HTTPS, rather than using a JCA adapter, the Integration Server uses servlets running in the Web container to provide J2EE HTTP facilities.

10.6 Configuring the Logical Host

10.6.1 Modifying the Logical Host Properties File

If you want to install a Logical Host as a Windows service, you must set the Logical Host default startup configuration in the **logical-host.properties** file. Setting the default configuration also avoids having to specify the arguments if you choose to start the Logical Host manually at the command prompt. If you do specify the arguments when you manually start the Logical Host, what you enter will override the default values; see [Starting the Logical Host](#) on page 275.

To modify the Logical Host Properties file

- 1 Ensure that the Logical Host is not running.
- 2 Navigate to `ICAN_root\logicalhost\bootstrap\config\logical-host.properties`.
- 3 Use a text editor (such as Windows WordPad) to open the file.

Figure 236 Example logical-host.properties File

```
#####
#
#                               Logical Host Properties
#
#####

# These properties are automatically persisted by the bootstrap sequence.
# They are used by default if none are provided at the command line.

#####
# repository.url: (USER MODIFIABLE)
#   Specifies the remote URL for connecting to the repository.
#   Takes the form:
#       http://<repository-server-hostname>:<port>/
#       <repository-name>
#   For example:
#       http://localhost:10000/myRep
#####
repository.url=

#####
# repository.username: (USER MODIFIABLE)
#   Username for connecting to the repository.
#####
repository.username=

#####
# repository.password: (USER MODIFIABLE)
#   Plain text form of password used for connecting to the
#   repository. Any value provided here will be cleared out
#   by the system and written in encrypted form to the
#   repository.password.encrypted field.
#####
repository.password=

#####
# repository.password.encrypted:
#   Encrypted form of the repository password. NOTE: This value
#   is generated by the system, so it is improper to edit this
#   field manually.
#####
repository.password.encrypted=

#####
# physical.host.name: (USER CONFIGURABLE)
#   Specifies the physical host on which this logical host is
#   running. The host name should include the domain name.
#   Example: host.company.com
#####
physical.host.name=

#####
# logical.host.environment.name: (USER MODIFIABLE)
#   Specifies the name of the environment containing the
#   current logical host.
#
```

```

#####
logical.host.environment.name=

#####
# logical.host.name: (USER MODIFIABLE)
# Specifies the name of the current logical host.
#####
logical.host.name=

#####
# logical.host.root.dir:
# Specifies the root directory of a logical host
# installation.
#####
logical.host.root.dir=

#####
# os.type:
# Specifies the OS type of the machine on which logical host
# is going to run
#####
os.type=

# *** THE FOLLOWING PROPERTIES ARE PRIVATE; MODIFY AT YOUR OWN RISK ***

# bootstrap
managementagent.jar.path=lib/managementagent.jar

# deployment manager
deployment.manager.jar.relative.path=deploymentmanager.jar

# management agent
managementagent.config.file=./config/ManagementAgent-config.xml
managementagent.command.line.windows=cmd.exe /c ..\jre\bin\java.exe -Xrs -
Dlogical.host.properties.file=..\bootstrap\config\logical-host.properties -classpath
.\config;.\lib\managementagent.jar;.\lib\com.stc.hnsems.jar;.\lib\jta.jar;.\lib
\jms.jar;.\lib\jta.jar;.\lib\concurrent.jar;.\lib\log4j.jar;.\lib\jargs.jar;.\
lib\mx4j.jar;.\lib\mx4j-
tools.jar;.\lib\stcjms.jar;.\lib\stcrepository.jar;.\lib\stccore.jar;.\lib\stcr
epositoryclient.jar;.\lib\stcrepositoryserver.jar;.\lib\velocity.jar;.\lib\ejb.ja
r;.\lib\commons-
collections.jar;.\stcis\lib\com.stc.isbootstrap.jar;.\stcis\lib\gnu-regexp-
1.1.4.jar;.\lib\com.stc.isapi.jar;.\stcis\lib\com.stc.isimpl.jar;.\lib\com.stc.
compatimpl.jar;.\lib\com.stc.eventmanagementapi.jar;.\lib\xmlrpc.jar;.\lib\smscli
ent.jar;.\lib\com.stc.stcutil4Impl.jar;.\lib\xercesimpl.jar;.\lib\xalan.jar;.\l
ib\xml-apis.jar;.\lib\stcqueueviewer.jar;.\lib\stcjournaller.jar;.\lib\jakarta-
regexp-1.2.jar;;.\lib\commons-codec-1.1.1-dev.jar;
com.stc.processmanager.ma.ManagementAgent

managementagent.command.line.unix=sh ../bootstrap/bin/magent.sh

managementagent.shutdown.windows=cmd.exe /c ..\jre\bin\java.exe -
Djava.ext.dirs=.\lib com.stc.sms.mbeans.ManagementAgentControl -f
.\config\ManagementAgent-config.xml -c shutdown

managementagent.shutdown.unix=sh ./jre/bin/java -Djava.ext.dirs=./lib
com.stc.sms.mbeans.ManagementAgentControl -f ./config/ManagementAgent-config.xml -c
shutdown

# misc
repository.relative.url=data/repository/SBYN00.properties
log4j.config.path=bootstrap/config/log4j.xml
jre.zip.name=jre.zip

```

- 4 In the beginning of the **logical-host.properties** file, locate the properties listed in [Table 53 on page 271](#) (most are marked USER MODIFIABLE, and may appear in a different order than that shown):
- 5 Enter the appropriate values for your system.

Note: Do not enter spaces before or after the equal (=) sign and the property values. Spaces are allowed only in the value itself.

Table 53 User-Modifiable Logical Host Properties

Property	Description
logical.host.environment.name	Specifies the name of the Logical Host environment deployment; for example, EnvironmentDeployment1 .
logical.host.name	Specifies the name for the Logical Host; for example, logicalhost1 .
logical.host.root.dir	Specifies the full path of the Logical Host directory; for example, c:\ican50\logicalhost .
os.type	Specifies the operating system type under which logical host is going to run.
physical.host.name	Specifies the physical host on which this logical host is running. The host name should include the domain name, for example: host.company.com .
repository.url	Specifies the path to your Repository; for example, http://hostname:port/repositoryname where: <ul style="list-style-type: none"> ▪ hostname is the physical name of the computer on which the Repository resides; for example, localhost. ▪ port is the port number that the Repository uses to receive requests; for example, 9999. ▪ repositoryname is the name you specified for the Repository; for example, Repository1.
repository.username	Specifies the user name you are using to access the Repository; for example, Administrator .
repository.password	Specifies the password you are using to access the Repository; for example, myPassword . When you launch the bootstrap process, this password is encrypted and written to the repository.password.encrypted property. After the encrypted password has been written, this repository.password value is removed from the logical-host.properties file.
repository.password.encrypted	This property is automatically updated based on changes made to the repository.password property. <i>Do not enter anything for this property or modify its contents.</i>

- 6 Save the **logical-host.properties** file.
- 7 The Logical Host may now be started, either manually or as a Windows service.

10.6.2 Installing the Logical Host as a Windows Service

Installing the Logical Host as a Windows service configures the Logical Host to start up automatically at system startup, and restart automatically after an abnormal system shutdown.

Note: You must have Administrator privileges on the local Windows computer in order to configure the Logical Host to start as a service (the installation script writes to the Windows registry, which requires Administrator privileges).

To install the Logical Host as a Windows Service

- 1 Configure the **logical-host.properties** file as described in **Modifying the Logical Host Properties File** on page 269.
- 2 Change to the directory containing **bootstrap.bat**:

```
cd ICAN_root\logicalhost\bootstrap\bin\
```
- 3 Double-click **install-as-nt-service.bat**.

This runs the script that configures the Windows service to automatically run the Logical Host at system startup. The only parameter is the name of your Logical Host Service; the default is “ICAN 5.0 Logical Host” (see Figure 237).

Figure 237 Install as Service Script

```
C:\ican50\logicalhost\bootstrap\bin>install-as-nt-service.bat

-----
--- eGate 5.0 Management Agent Install Bootstrap NT Service ---
--- Copyright (c) 2003, SeeBeyond Technology Corporation, ---
--- All Rights Reserved ---
-----

To start the service use: net start "ICAN 5.0 LogicalHost"
To stop the service use: net stop "ICAN 5.0 LogicalHost"
C:\ican50\logicalhost\bootstrap\bin\..\..
C:\ican50\logicalhost\bootstrap\bin\..\..
C:\ican50\logicalhost\bootstrap\bin\..\..\bootstrap\lib
C:\ican50\logicalhost\bootstrap\bin\..\..\bootstrap\config
The service was successfully installed.
```

- 4 Verify the installation by opening the Windows Services facility and searching for the Logical Host name (see Figure 238). Note that by default, the service is listed as *Automatic*—it will not be running, however, until you click **Start** or reboot the computer.

Figure 238 Windows Services List

Hummingbird Ineco	The Hummi...		Manual	LocalSystem
Hummingbird Jconfi...	Jconfig Da...		Manual	LocalSystem
ICAN 5.0 LogicalHost			Automatic	LocalSystem
IIS Admin Service	Allows adm...	Started	Automatic	LocalSystem
Indexing Service		Started	Automatic	LocalSystem
Informatica			Manual	SYSTEM

To remove the Logical Host Windows Service

- 1 Change to the directory containing **bootstrap.bat**:

```
cd ICAN_root\logicalhost\bootstrap\bin\
```

- 2 Double-click **uninstall-nt-service.bat**.

This runs the script that removes the Windows service that automatically starts the Logical Host at system startup (see Figure 239).

Figure 239 Uninstall as Service Script

```
C:\ican50\logicalhost\bootstrap\bin>install-as-nt-service.bat

-----
eGate 5.0 Management Agent Install Bootstrap NT Service
Copyright (c) 2003, SeeBeyond Technology Corporation,
All Rights Reserved
-----

To start the service use: net start "ICAN 5.0 LogicalHost"
To stop the service use: net stop "ICAN 5.0 LogicalHost"
C:\ican50\logicalhost\bootstrap\bin\..\..
C:\ican50\logicalhost\bootstrap\bin\..\..
C:\ican50\logicalhost\bootstrap\bin\..\..\bootstrap\lib
C:\ican50\logicalhost\bootstrap\bin\..\..\bootstrap\config
The service was successfully installed.
C:\ican50\logicalhost\bootstrap\bin>uninstall-nt-service.bat

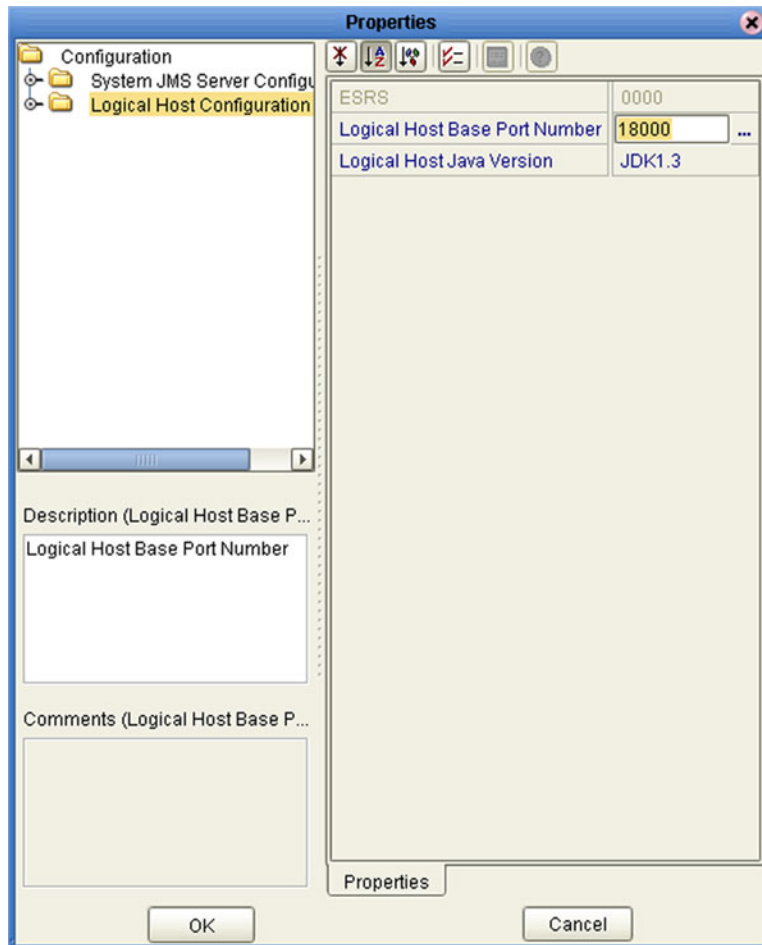
-----
eGate 5.0 Management Agent Uninstall Bootstrap NT Service
Copyright (c) 2003, SeeBeyond Technology Corporation,
All Rights Reserved
-----

The service was successfully uninstalled.
```

10.6.3 Configuring the Base Port Number

Port numbers assigned to the Logical Host and its components (Integration Server, JMS IQ Manager, etc.) can be viewed and modified in the Enterprise Explorer by selecting **Properties** from the component's context menu (see Figure 240).

Figure 240 Logical Host Properties Dialog Box



If multiple Logical Hosts reside in the same Environment, you must ensure that each Logical Host has a different base port number to avoid conflicts. This base port number is propagated throughout the Logical Host, so that the various components are automatically given successive port numbers following that assigned to the Logical Host itself.

The number of port numbers used in a Logical Host varies according to the specific implementation, so when assigning new base port numbers you need to skip successive numbers by an adequate amount. The default base port number is 18000, so base port numbers of 19000, 20000, and so on are recommended.

If you need to assign a specific port number to a particular Logical Host component, the automatic numbering process will skip the component port number you have assigned manually (*be sure this port number is not used elsewhere*).

10.7 Starting the Logical Host

The Logical Host is started by means of a bootstrap script that is specific to the platform on which it is running. You can start the Logical Host manually by opening a command prompt and typing a startup command having a string of arguments appended to it. These include both optional and required initial (first-time only) arguments, as listed in Table 54. If you have configured the logical-host.properties file (see [Modifying the Logical Host Properties File](#) on page 269), you can use the default values by omitting the arguments.

The startup command string uses the following syntax:

```
bootstrap argument1 ... argumentN
```

For example, the command

```
bootstrap -h
```

displays the online *Help* entry that explains the command line arguments.

Table 54 Startup Command Arguments

Parameter	Description	Req/Opt
-d <i>debug</i>	Overrides bootstrap sequence. Displays all cached (default) argument values.	Optional
-e <i>environment name</i>	The name of the Environment to which this Logical Host belongs.	Required (first time only)
-h <i>help</i>	Overrides bootstrap sequence. Displays the usage report.	Optional
-i <i>id</i>	The user ID used for accessing the Repository. Note that the user ID is the same as the username, and that the Administrator can set up more than one user ID.	Required (first time only)
-l <i>logicalhost name</i>	The name of this Logical Host.	Required (first time only)
-n <i>physical host name</i>	The name of this Physical Host.	Required (first time only)
-p <i>password</i>	The password used for accessing the Repository.	Required (first time only)
-r <i>repository URL</i>	The root URL for the Repository containing the Logical Host data.	Required (first time only)

Note: *Required (first time only)* indicates that the argument is required the first time you start the Repository. You do not need to use it again unless you change the name of the Logical Host.

10.7.1 Starting the Logical Host on a Windows System

To start the Logical Host for the first time, or to override existing default values

- 1 Open a command prompt on your desktop.
- 2 Change to the directory containing **bootstrap.bat**:

```
cd ICAN_root\logicalhost\bootstrap\bin\
```
- 3 Type the startup command including all of the required parameters shown in Table 54:

```
bootstrap argument1 ... argumentN
```
- 4 Click **Enter**.

To start the Logical Host after the first time, or to use existing default values

- 1 Open a command prompt on your desktop.
- 2 Change to the directory containing **bootstrap.bat**:

```
cd ICAN_root\logicalhost\bootstrap\bin
```
- 3 Double-click **bootstrap.bat**.

10.7.2 Starting the Logical Host on a UNIX System

To start the Logical Host

- 1 Open a command prompt on your desktop.
- 2 Change to the directory containing **bootstrap.bat**:

```
cd ICAN_root/logicalhost/bootstrap/bin/
```
- 3 Run the bootstrap script using the following command:

```
./bootstrap.sh arguments
```

10.7.3 Starting the Logical Host on a Red Hat Linux System

To start the Logical Host

- 1 Open a command prompt on your desktop.
- 2 Change to the directory containing **bootstrap.bat**:

```
cd ICAN_root/logicalhost/bootstrap/bin/
```
- 3 Run the bootstrap script using the following command:

```
./bootstrap.sh arguments
```
- 4 After the bootstrap command is executed, the script prompts you for the RedHat server release number.

10.8 Stopping the Logical Host

The Logical Host can be shut down either from the Enterprise Monitor or by means of the following command-line procedure.

To stop the Logical Host

- 1 Open a command prompt on your desktop.
- 2 Change to the directory containing **bootstrap.bat**:

```
cd ICAN_root/logicalhost/bootstrap/bin/
```
- 3 Run the file **shutdown.bat** (or **shutdown.sh**).

Project Deployment

This chapter describes the process of creating deployment profiles and activating the deployed projects.

11.1 Deployment Profiles

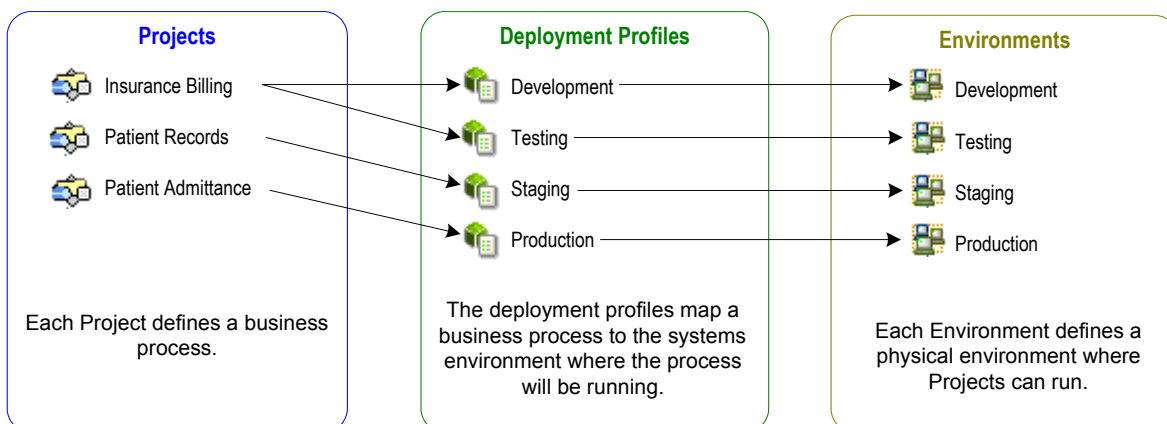
Deployment Profiles define specific instances of a Project in a particular Environment. A deployment profile contains information about the assignment of Services and Message Destinations to integration and message servers (JMS IQ Managers). It also contains version information for all relevant objects in the Project. The Enterprise Designer includes a Deployment Editor, which you can use to create and customize deployment profiles.

Note that:

- Each Project can have zero or more Deployment Profiles, but each of a Project's active Deployment Profiles must be in a separate Environment.
- Each Environment can have zero or more Deployment Profiles assigned to it, but any given Environment can have only one Deployment Profile from a given Project.

Repeating Figure 3 from the [System Overview](#) on page 25:

Figure 241 eGate Integrator Implementation Model



11.2 Using the Deployment Editor

The Deployment Editor (see Figure 242) appears when you create a new Deployment Profile or click on an existing Deployment Profile icon in the Project Explorer tab.

Figure 242 Deployment Editor Window

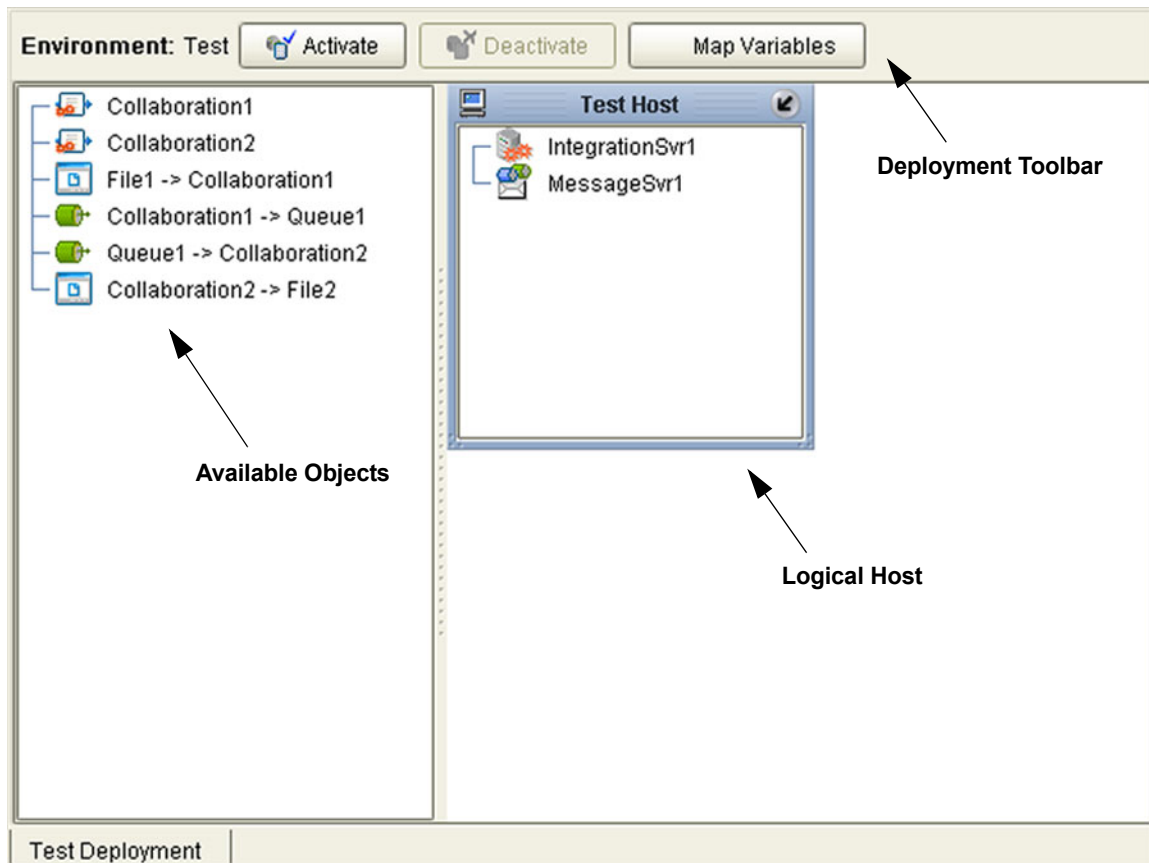




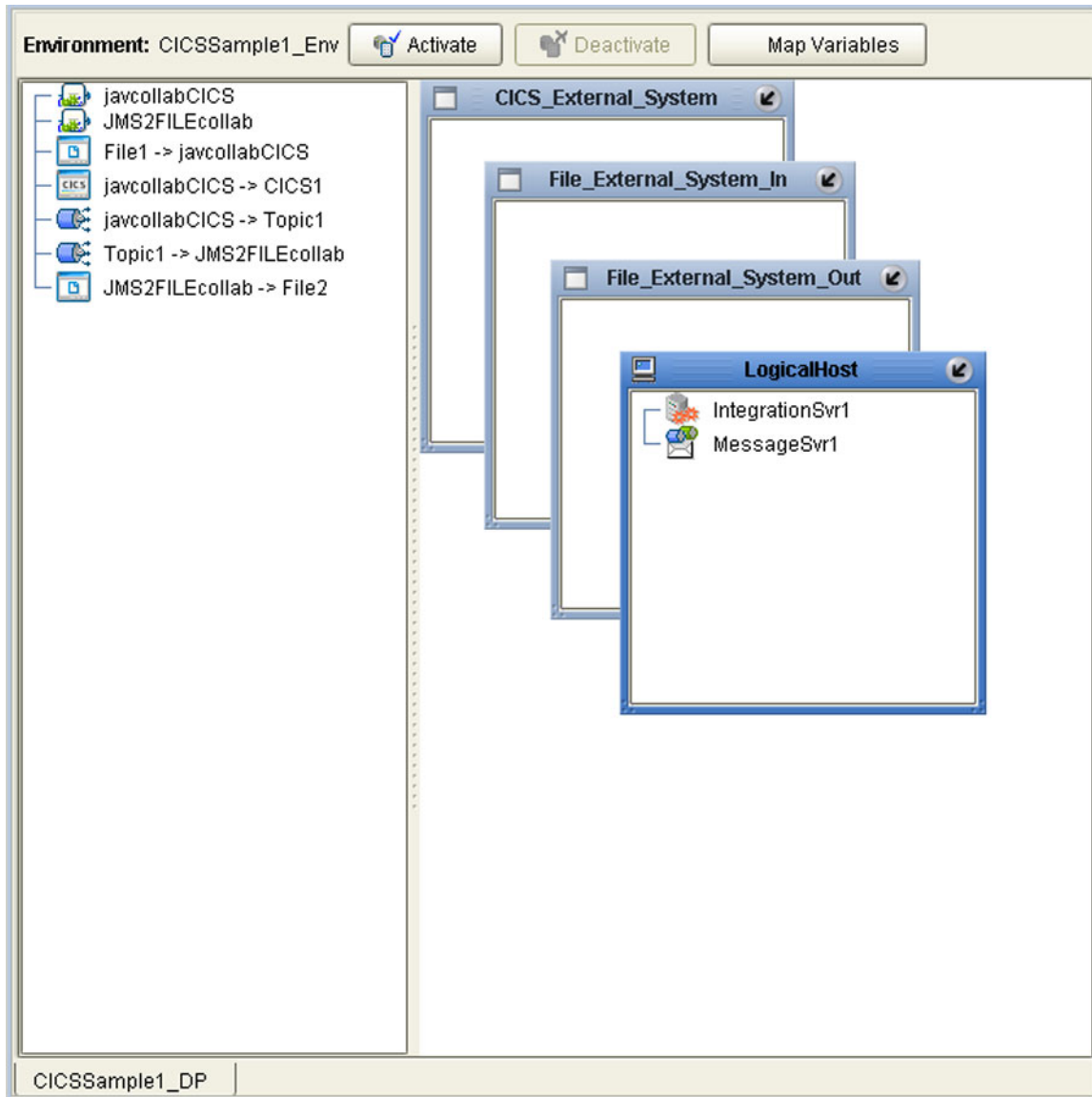
Table 55 Deployment Toolbar Buttons

Button	Function
 Activate	Starts the Project by creating an enterprise archive (EAR) file based on the Connectivity Map and linking this file with the SeeBeyond Integration Server. See Activating and Deactivating Deployment Profiles on page 283.
 Deactivate	Stops the Project by terminating the link between the EAR file and the SeeBeyond Integration Server, sets the Deployment Profile to <i>inactive</i> , and saves to the Repository.
Map Variables	Allows you to assign names and values to Project variables for the specific Deployment Profile. See Mapping Variables on page 285.

11.3 Creating a Deployment Profile

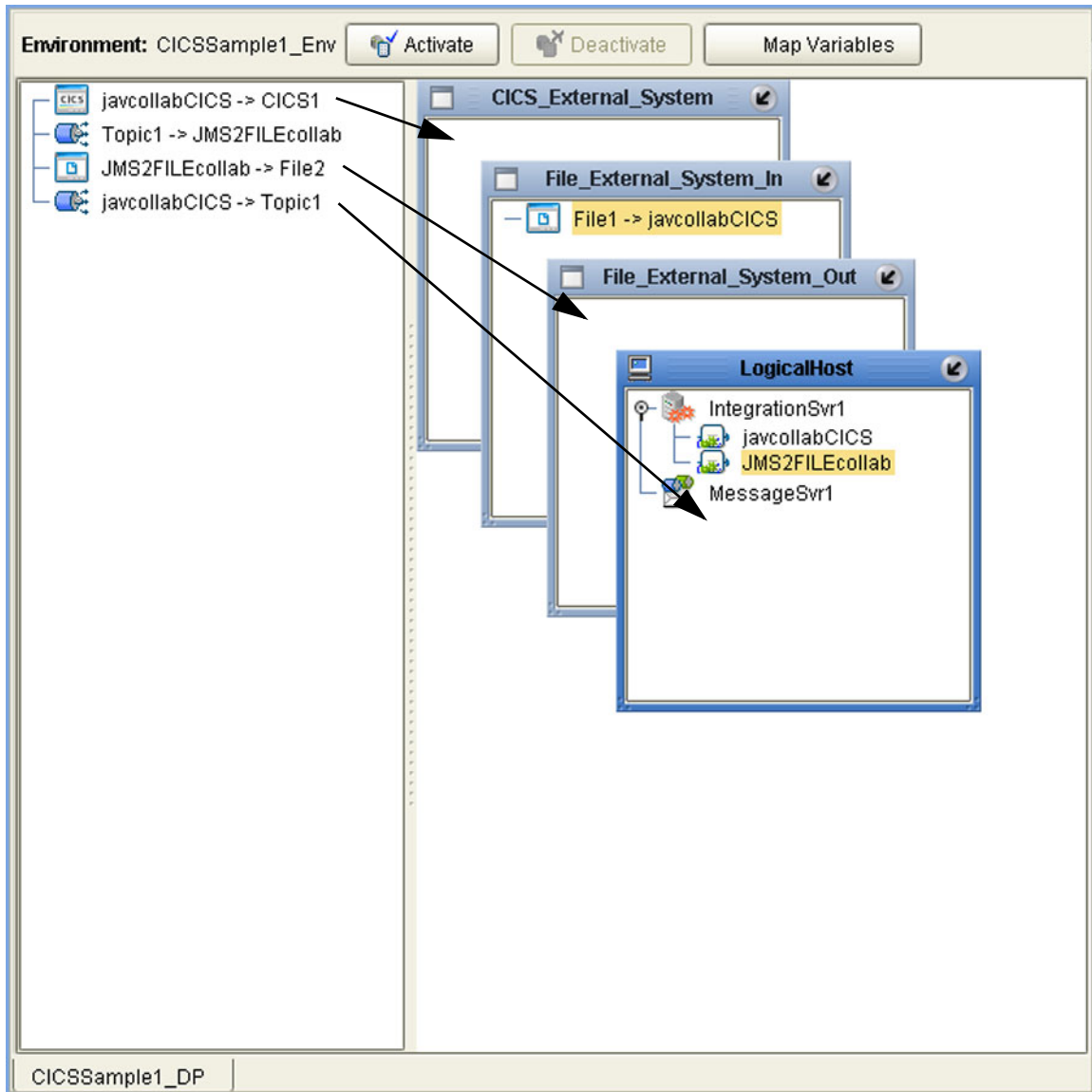
In the Environment Explorer, create an Environment and right-click on the Environment to display its context menu. From the menu, select the Environment components you need and name them appropriately. They will appear on the Deployment Editor canvas, as shown in Figure 243.

Figure 243 Example Deployment Profile (1)



Drag the Project components from the left panel and drop them into the appropriate Environment components in the right panel, as illustrated in Figure 244. As you do so, they will disappear from the left panel.

Figure 244 Example Deployment Profile (2)

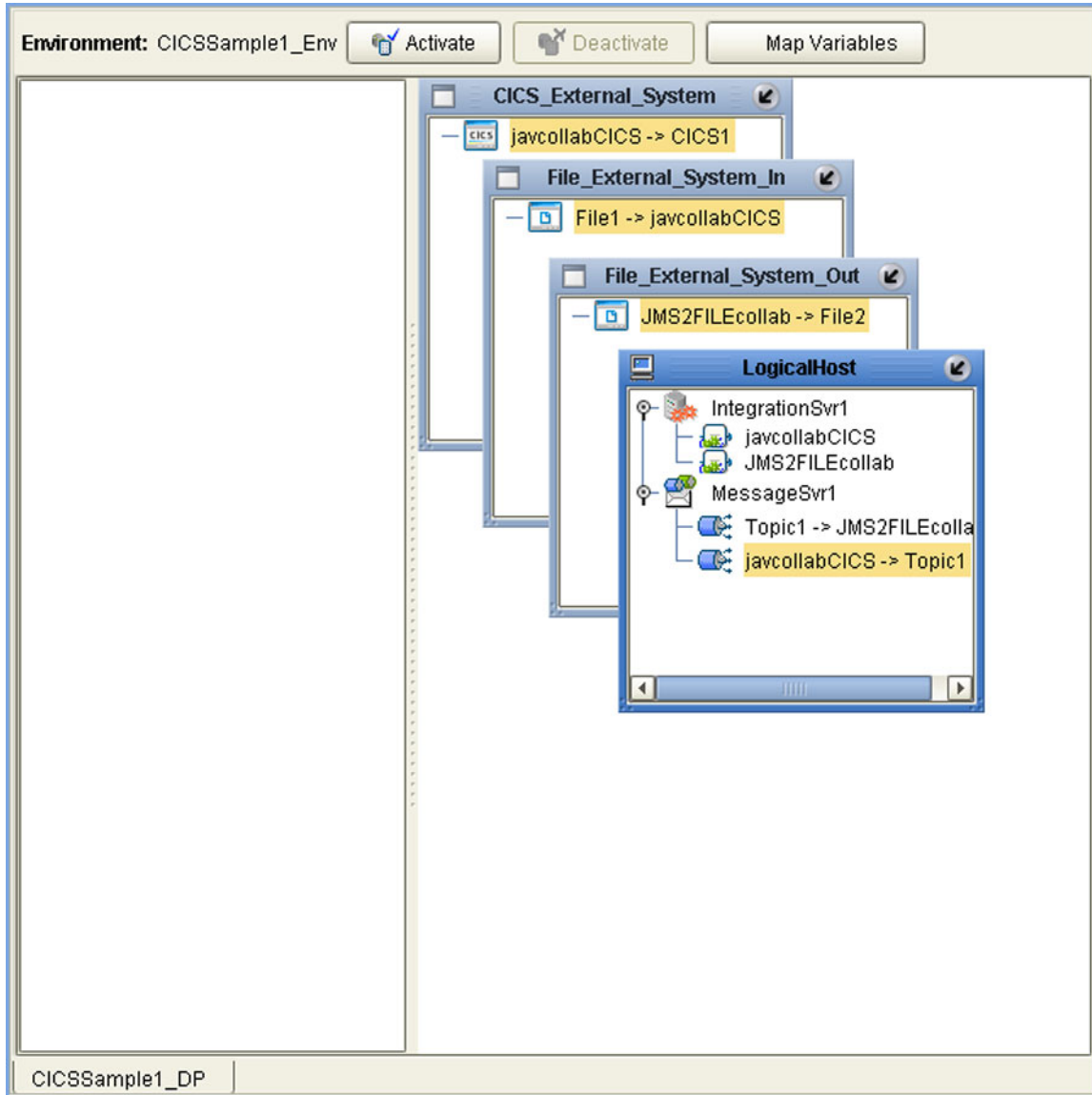


Note that:

- The eWay objects are placed into their appropriate External Systems.
- Collaboration objects are placed into the appropriate Integration Server on the appropriate Logical Host.
- Topic and queue objects are placed into the appropriate Message Server (JMS IQ Manager) on the appropriate Logical Host.

When the Environment components are fully populated, the left panel will be blank, as shown in Figure 245. The Deployment Profile is now ready to be saved and Activated.

Figure 245 Example Deployment Profile (3)



11.4 Activating and Deactivating Deployment Profiles

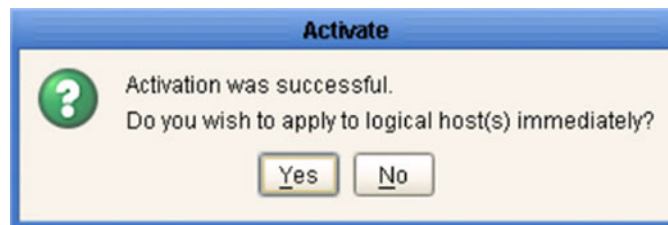
Using the Activate and Deactivate toolbar buttons, you have the option of immediately applying the changes to the Logical Host or deferring the changes to a later time. Activating the Deployment Profile without applying the changes checks the validity of the entire Deployment Profile.

Another advantage to activating the Deployment Profile without applying the changes comes into play when you have multiple Deployment Profiles to deploy at once. To save time, you can activate each of the Deployment Profiles without applying the changes. Then when you do apply all of the changes to the Logical Host in one batch.

To activate a Deployment Profile

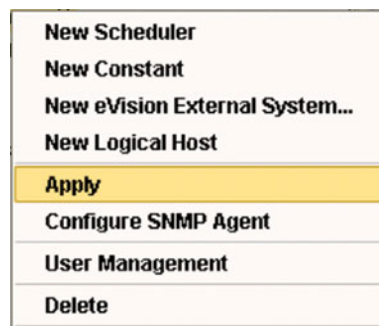
- 1 In the Deployment Profile, select the Deployment you wish to activate.
- 2 Click the **Activate** button. The following message appears:

Figure 246 Activate Dialog Box



- 3 Answer the question following these criteria:
 - If the Logical Host is running, and you wish to apply the changes immediately, click **Yes**.
 - If the Logical Host has not yet been bootstrapped, or you wish to apply the changes at a later time, click **No**. To apply the changes later, right-click the Logical Host and select **Apply** from the menu (see Figure 247). This will apply all of the changes for that Logical Host.

Figure 247 Logical Host Context Menu - Apply

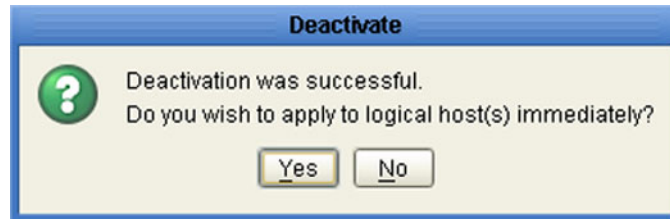


Note: The **Apply** action assumes that the Logic Host is running, since it invokes a trigger to the Logical Host causing it to download the latest settings from the Repository and deploy those settings to all components on the Logical Host.

To deactivate a Deployment Profile

- 1 In the Deployment Profile, select the Deployment you wish to deactivate.
- 2 Click the **Deactivate** button. The following message appears:

Figure 248 Activate Dialog Box



- 3 Answer the question following these criteria:
 - If the Logical Host is running, and you wish to apply the changes immediately, click **Yes**.
- 4 If the Logical Host has not yet been bootstrapped, or you wish to apply the changes at a later time, click **No**. To apply the changes later, right-click the Logical Host and select **Apply** from the menu (see Figure 247). This will apply all of the changes for that Logical Host. See the **Note** following the activation procedure.

Note: In Windows and NFS, application working directories cannot be deleted during deactivation. This is because .jar files in these directories have been added to a ClassLoader and the JVM maintains locks/handles on any such files. At the subsequent startup of the Integration Server, leftover work directories in the repository/application directory are deleted.

11.5 Mapping Variables

Project variables function as placeholders, having values that are determined when you create a specific Deployment Profile. These values can be literals or Environmental constants. Clicking the **Map Variable** button displays the Deployment Profile Mappings panel, where you can assign names (see Figure 249) and values (see Figure 250).

Figure 249 Deployment Profile Mappings

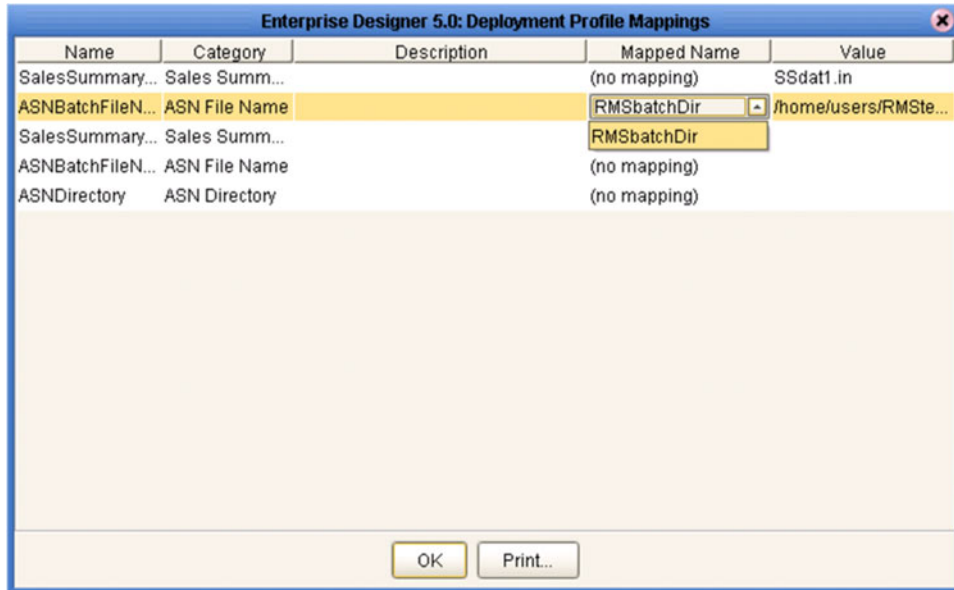
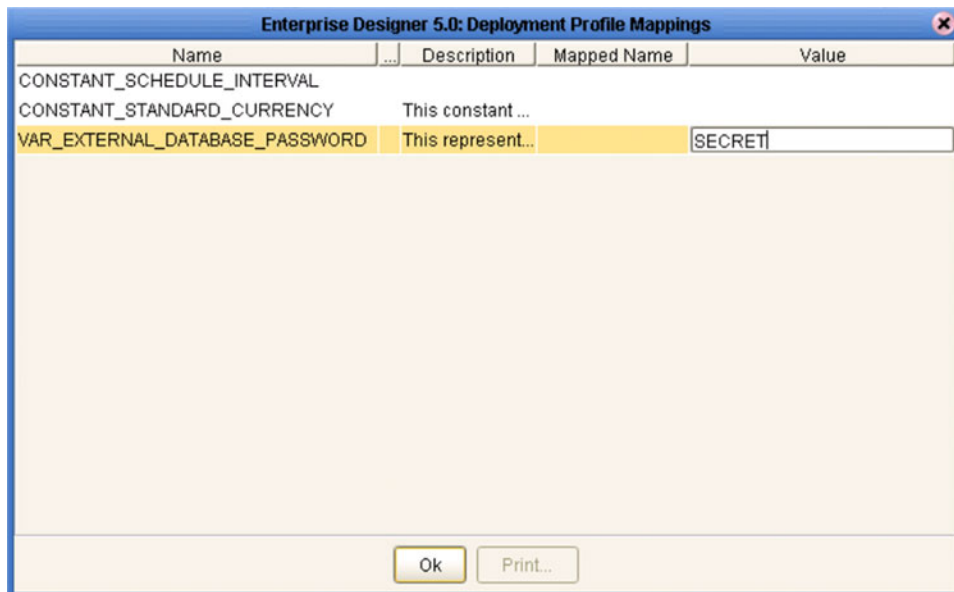


Figure 250 Project Variable Value Entry



11.6 Deploying Projects to Third-Party Servers

SeeBeyond's eGate Integrator allows you to develop Projects using Enterprise Designer and deploy them to a BEA WebLogic or IBM WebSphere environment. The SAR files for these third-party products must be installed prior to deployment, as described in the *eGate Integrator JMS Reference Guide*.

Important: The file `log4j.jar` must be added to the classpath for both WebLogic and WebSphere. This file can be obtained from the following URL:

<http://jakarta.apache.org/log4j/docs/>

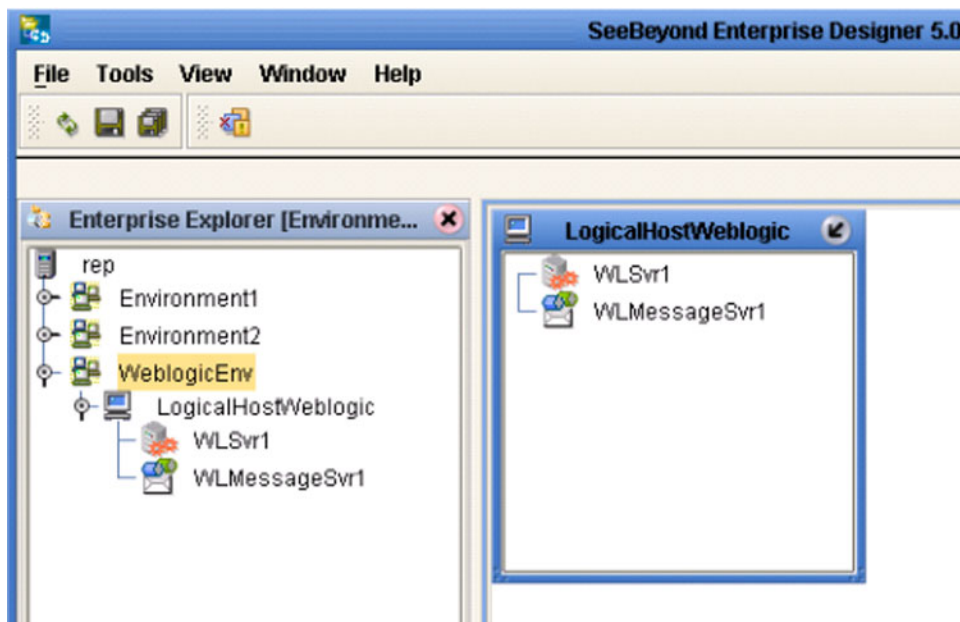
Note: The initial capability provided by eGate Integrator 5.0 is restricted to topic-to-topic Collaborations and JMS connections.

11.6.1 BEA WebLogic

To deploy an eGate Project to a BEA WebLogic environment

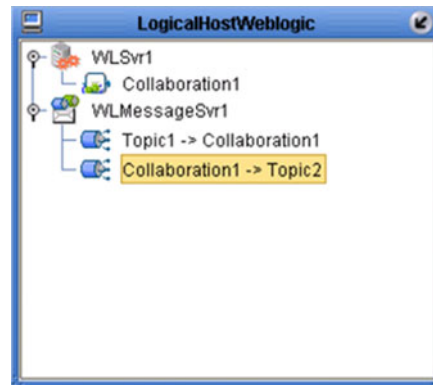
- 1 Create the following components in Enterprise Designer (see Figure 251):
 - A A new environment
 - B A Logical Host
 - C A WebLogic Integration Server
 - D A WebLogic JMS Message Server

Figure 251 WebLogic Deployment (1)



- 2 Create a new Deployment Profile to bind the Connectivity Map to the new WebLogic environment (see Figure 252).
 - A Drag the two topics and drop onto the WebLogic message server.
 - B Drag the Collaboration and drop onto the WebLogic integration server.

Figure 252 WebLogic Deployment (2)



- 3 Activate the Deployment Profile.

Activating the Deployment Profile creates an Environment Archive (EAR) file, which contains all files necessary to create and run an application in WebLogic. This file can be found in the following location:

```
ICAN-root\repository\data\files\WLEnvironmentName\  
ProjectName_DeploymentProfileName.ear
```

Note: The remainder of this procedure is performed in the WebLogic user interface, and is only outlined here. Please refer to your BEA WebLogic documentation for current information regarding interface layout and deployment details.

- 4 Start the BEA WebLogic server.
- 5 Navigate to **Server Administration Console > Deployments > Applications**.
- 6 Perform the following steps:

- A Add a new JMS Connection Factory.
- B Enter a JNDI name for the JMS Connection Factory:

```
jms/connectionfactory/xa-topic/  
LogicalHostName_MessageServerName
```

For example, the default name would be:

```
jms/connectionfactory/xa-topic/LogicalHost1_WLMessageSvr1
```

- C Verify that the WebLogic JMS Server Destination names for topics match those in eGate.
- D Select **Deploy a new Application**.
- E Upload and install the EAR file described in step 3.
- F Select the EAR file you just installed as the archive for the new application.

- G Enter a name for the new application.
- H Click **Deploy**.
- I Verify the success of the deployment (see Figure 253 , which shows a WebLogic 8.1 example).

Figure 253 WebLogic Deployment Verification

mydomain> Applications> weblogic-demo

Connected to : localhost :7001 | You are logged in as : rm | Logout

Configuration | Targets | **Deploy** | Notes

This page allows you to view the deployment status of each module in the application, and to stop or redeploy individual modules. You may also choose to stop and redeploy all modules within the application using the buttons at the bottom of the page. (To configure additional deployment targets for this application, click the Targets tab.)

Deployment status for EJB Modules

Module	Module Status	Target	Target Type	Status of Last Action
Collaboration1.jar	Active	myserver	Server	Success
Topic1_C290971529.jar	Active	myserver	Server	Success
Collabor_u002D_1977709066.jar	Active	myserver	Server	Success

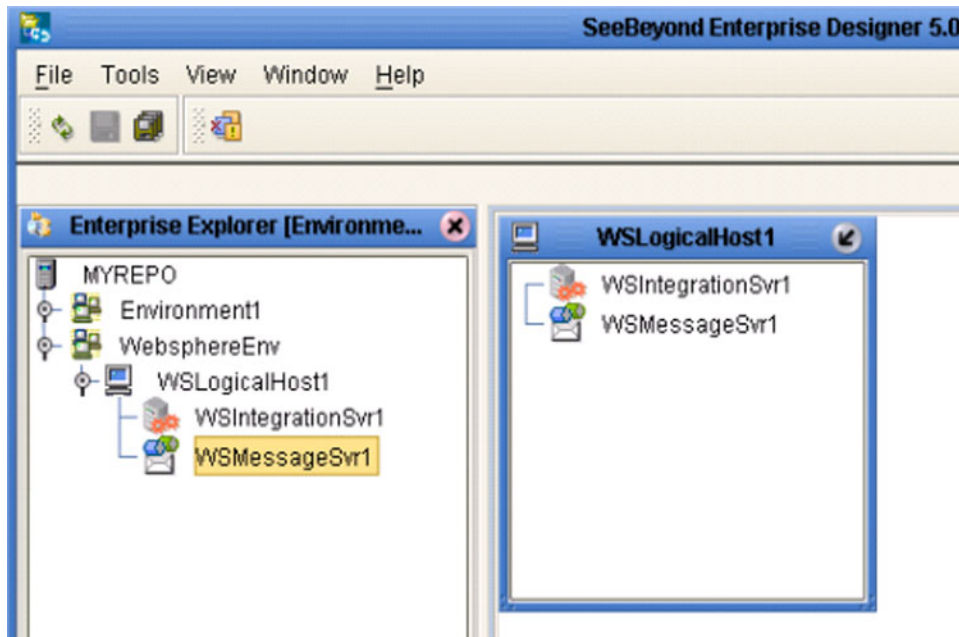
Stop Application Redeploy Application

11.6.2 IBM WebSphere

To deploy an eGate Project to an IBM WebSphere environment

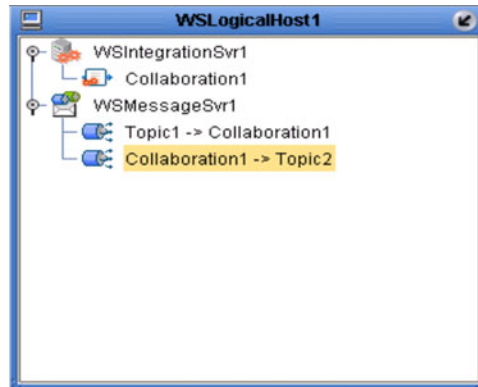
- 1 Create the following components in Enterprise Designer (see Figure 254):
 - A A new environment
 - B A Logical Host
 - C A WebSphere Integration Server
 - D A WebSphere JMS Message Server

Figure 254 WebSphere Deployment (1)



- 2 Create a new Deployment Profile to bind the Connectivity Map to the new WebSphere environment (see Figure 255).
 - A Drag the two topics and drop onto the WebSphere message server.
 - B Drag the Collaboration and drop onto the WebSphere integration server.

Figure 255 WebSphere Deployment (2)



- 3 Activate the Deployment Profile.

The activated Deployment Profile creates an Environment Archive (EAR) file, which contains all files necessary to create and run an application in WebSphere. This file can be found in the following location:

```
ICAN-root\repository\data\files\WSEnvironmentName\  
ProjectName_DeploymentProfileName.ear
```

Note: *The remainder of this procedure is performed in the WebSphere user interface, and is only outlined here. Please refer to your IBM WebSphere documentation for current information regarding interface layout and deployment details.*

- 4 Start the IBM WebSphere server.
- 5 From the Administrative Console, navigate to **Servers > Application Servers > server_name > Message Listener Service > Listener Ports**.
- 6 Add a new Listener port.
- 7 Enter a Connection Factory JNDI name for the new port:

```
jms/connectionfactory/xa-topic/  
LogicalHostName_MessageServerName
```

For example, the default name would be:

```
jms/connectionfactory/xa-topic/LogicalHost1_WSMMessageSvr1
```

This binds the JNDI name with the WebSphere Message Server Listener port.

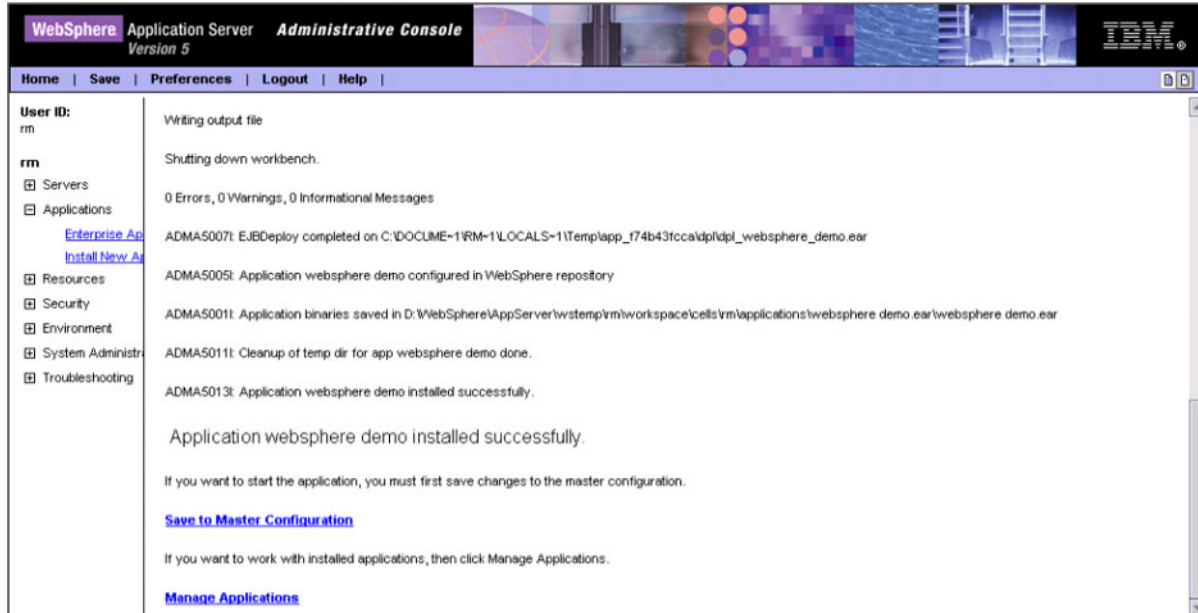
- 8 From the Administrative Console, navigate to **Applications > Enterprise Applications > Install New Application**.
- 9 In *Preparing for the application installation*:
 - A Enter the path for the EAR file described in step 3 and click **Next**.
 - B Select **Generate Default Bindings** and click **Next**.

- 10 In *Step 1, Provide options ...*:
 - A Check **Deploy EJBs**.
 - B Enter the application name.
 - C Click **Next**.
- 11 In *Step 2, Provide options ...*, click **Next**.
- 12 In *Step 3, Provide Listener Ports ...*, accept the default value and click **Next**.

Note: The Listener port number should match the port number entered in step 6.

- 13 In *Step 4, Provide JNDI Names ...*, accept the default value and click **Next**.
- 14 In *Step 5, Provide EJB references ...*, accept the default value and click **Next**.
- 15 In *Step 6, Map resource references ...*, enter the JNDI name from step 7, and click **Next**.
- 16 In *Step 7, Map modules ...*, check all modules and click **Next**.
- 17 In *Step 8, (protection levels)*, check all modules and click **Next**.
- 18 In *Step 9, Summary*, click **Finish**.
- 19 Verify the success of the deployment (see Figure 256 , which shows a WebSphere 5 example).

Figure 256 WebSphere Deployment Verification



ICAN Security Features

This chapter contains information on the various security features provided in the ICAN Suite.

12.1 Overview

ICAN users can be classified into two categories:

- 1 Users of the ICAN Suite toolset.

This category includes those who perform the development, administration, and management activities. The deployment and bootstrap tasks also fall into this category. These users are also logically mapped to the *all*, *administration*, and *management* roles, respectively.

- 2 Users of J2EE applications running in the Environment.

This category includes those who access the deployed J2EE applications in the Logical Host in an Environment. Potentially, these users are the Customers of the Enterprise accessing the J2EE applications.

Security features described in this chapter are as follows:

- **Configuration User Management** on page 294 describes the management of users in the ICAN Suite.
- **Environment User Management** on page 319 describes the management of users who would access the applications deployed in an enterprise, using the ICAN Suite.
- **ACL Management** on page 321 describes the management of access control to various components and features in the ICAN Suite.
- **ICAN Component Security** on page 324 briefly describes the security settings for message servers and JMS Client connections. Additional information is contained in the *eGate Integrator JMS Reference Guide*.
- **Using SSL/HTTPS in ICAN** on page 326 describes the use of Secure Sockets Layer protocols in Web communications.

12.1.1 Multiple Environments

Deploying Projects to multiple Environments requires special considerations regarding security.

To prepare for deployment to multiple Environments

- 1 Create the users who will develop, administer, or manage the multiple Environments in the Repository.
- 2 Set the ACL on the Environments to isolate them and grant access to only the specific Environment users (such as Administrators).
- 3 Create the J2EE application-specific users and roles in the respective Environments.
- 4 Set the environment-specific settings for the application using the users and roles you have created for the Environment.

12.2 User Management

12.2.1 Configuration User Management

In order to access the ICAN Suite, an individual must be registered as a *user* in the ICAN security system by a system administrator. Once entered into the system, the user can then be assigned privileges allowing access to different parts and features of the ICAN Suite. User management takes effect immediately, so it is not necessary to reboot the Repository to reflect any changes.

Roles

The Enterprise Designer contains facilities that allow a system administrator to manage user access, based on *roles* and user IDs. There are three predefined roles in the ICAN Suite, as described in the following table. Note that a user can have more than one role, in which case his privileges will be the combined privileges from both roles.

Role	Description
all	Users assigned this role can log in to ICAN 5.0. Once logged in, they can inherently connect to the Repository, perform downloads, and access documentation in Enterprise Manager. This is the most basic role, and offers the minimum permission level.
administration	Users assigned this role can log in and connect to the Repository, perform downloads and uploads, and access documentation in Enterprise Manager.
management	Users assigned this role can log in and connect to the Repository, perform downloads but not uploads, and access documentation in Enterprise Manager. They also can start and stop components using the eGate Monitor.

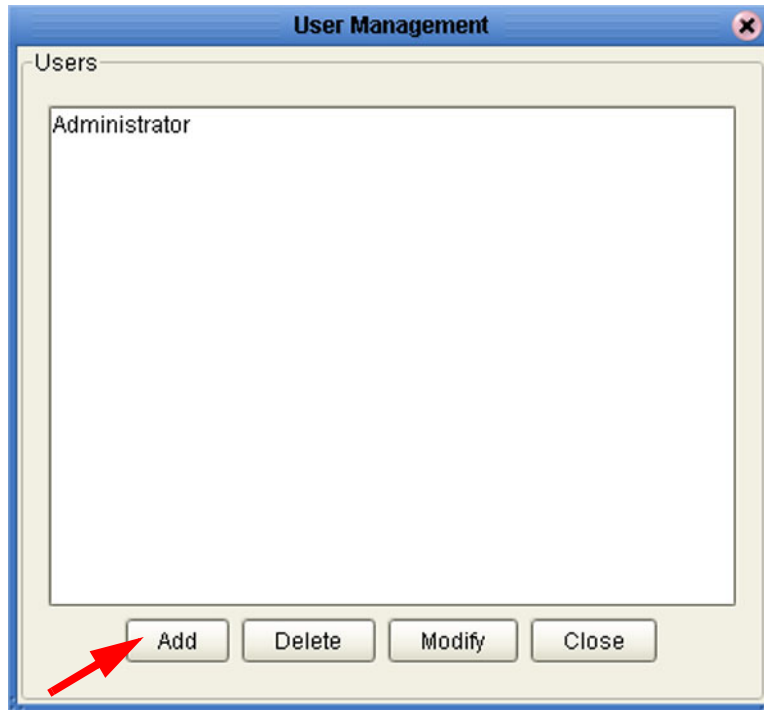
A mechanism is included in the Enterprise Designer for ICAN 5.0 to create roles in addition to the three predefined roles described above. The role you create using this mechanism provides a means for organizing users into groups that can be referenced in your own Web-based application (or perhaps in a third-party application that you are using). Future versions of the ICAN Suite will make additional use of this capability within the Suite itself. See [Creating User Roles](#) on page 303.

Adding and Deleting Repository Users

To add a new user

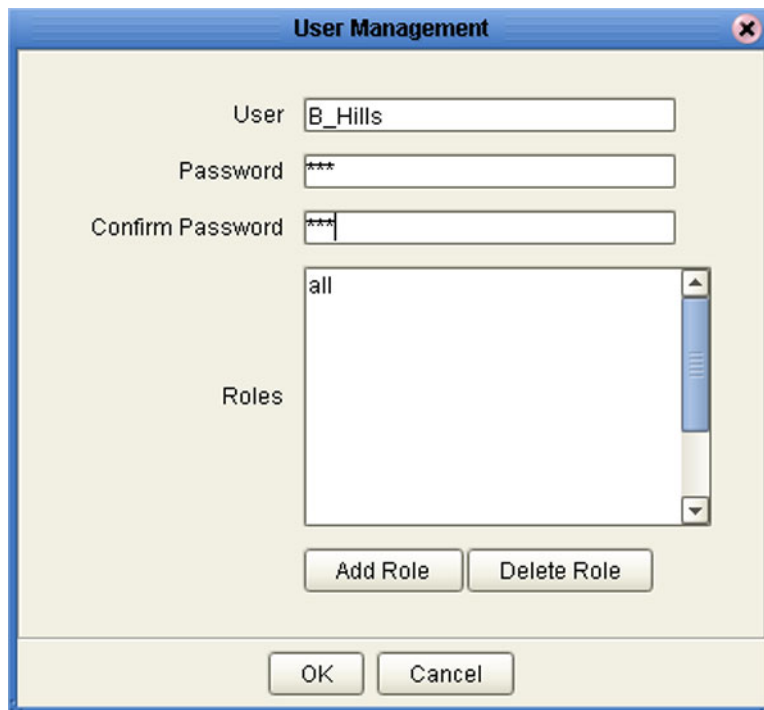
- 1 In the Project Explorer, select the **Repository** icon.
- 2 Right-click to display the Repository context menu.
- 3 Select **User Management** to display the User Management dialog box shown in Figure 257.

Figure 257 User Management Dialog Box (1)



- 1 Click **Add** to display a second User Management dialog box shown in Figure 258.

Figure 258 User Management - Add User

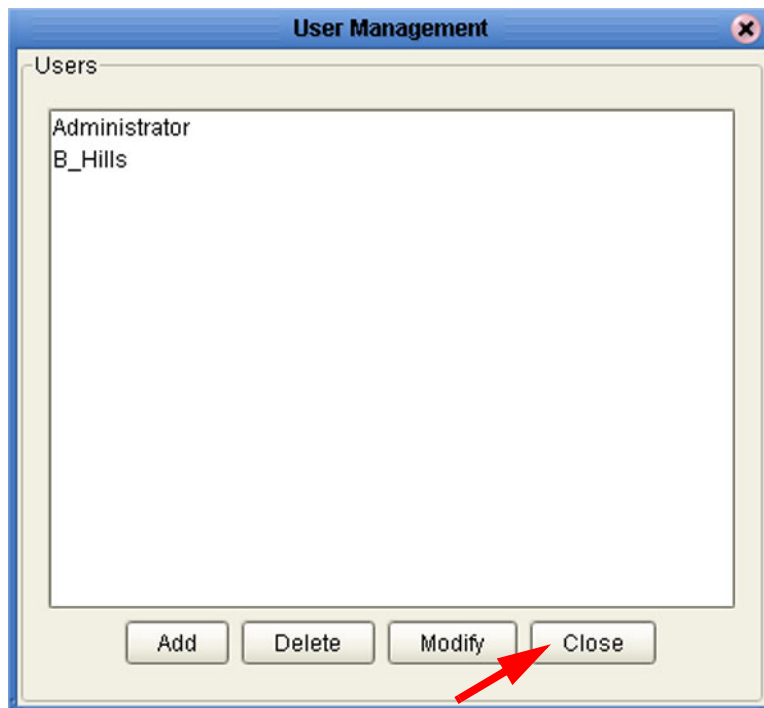


- 2 Enter a name for the user in the **User** box. This is the name the user will enter as their login ID during system login.
- 3 Enter a code for the user in the **Password** box. This is the code the user will enter as their password during system login.

Note: Every user entered into the system is automatically assigned to the **all** role, which is required to connect to the Repository.

- 4 Click **OK** to create the new user. This user can now access the Enterprise Designer and Repository with the assigned login ID and password, and this user name will be added to the list in the initial User Management dialog box (see Figure 259).
- 5 To add an another role for this user, see [Adding and Deleting User Roles](#) on page 298.

Figure 259 User Management Dialog Box (2)

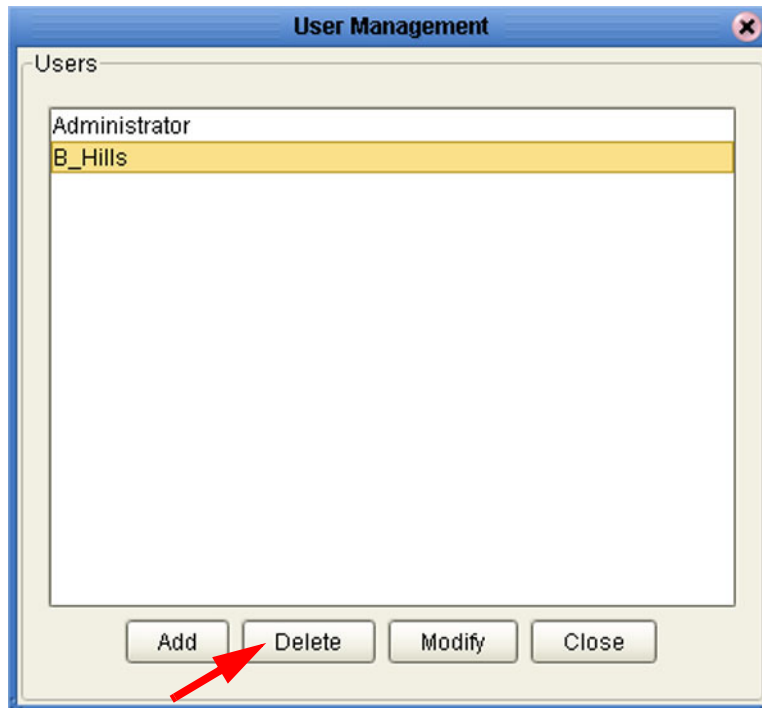


- 6 Click **Close** to return to the Enterprise Designer.

To delete a current user

- 1 In the Project Explorer, select the **Repository** icon.
- 2 Right-click to display the Repository context menu.
- 3 Select **User Management** to display the User Management dialog box shown in Figure 260.

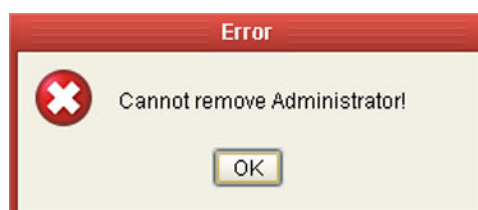
Figure 260 User Management Dialog Box



- 4 To delete a current user, select the user from the displayed list and click **Delete**. The user entry will be removed from the list.
- 5 Click **Close** to return to the Enterprise Designer.

Note: *The Administrator cannot be deleted from the system (see Figure 261).*

Figure 261 Error Warning Box

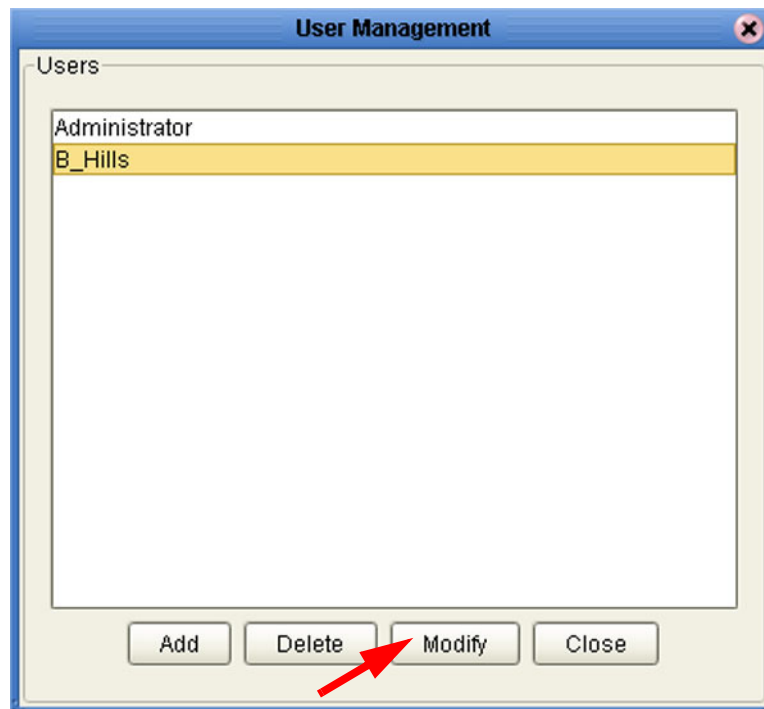


Adding and Deleting User Roles

To add a new role for a current user

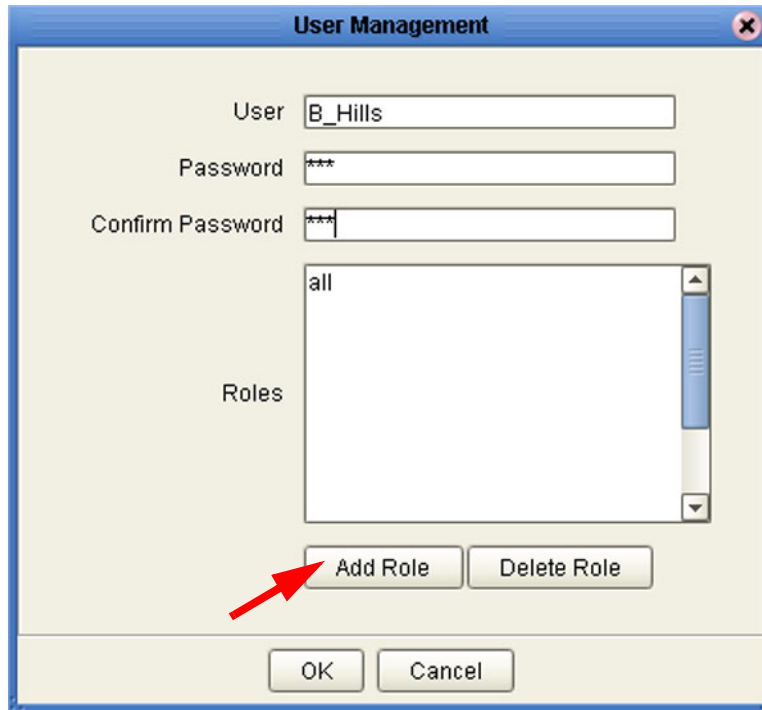
- 1 In the Project Explorer, select the **Repository** icon.
- 2 Right-click to display the Repository context menu.
- 3 Select **User Management** to display the User Management dialog box shown in Figure 262.

Figure 262 User Management Dialog Box



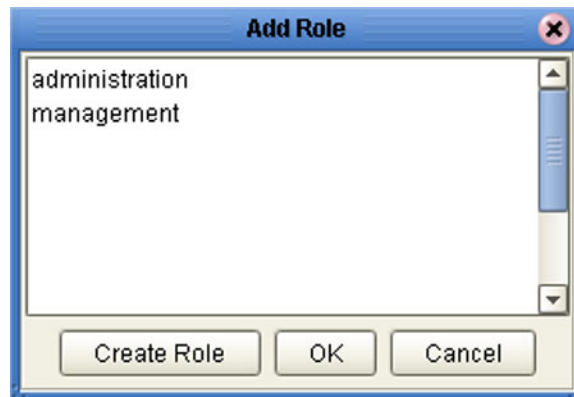
- 4 To add a role for a current user, select the user from the displayed list and click **Modify** to display the second User Management dialog box shown in Figure 263.

Figure 263 User Management - Add Role (1)



- 5 Click the **Add Role** button, which displays the dialog box shown in Figure 264.

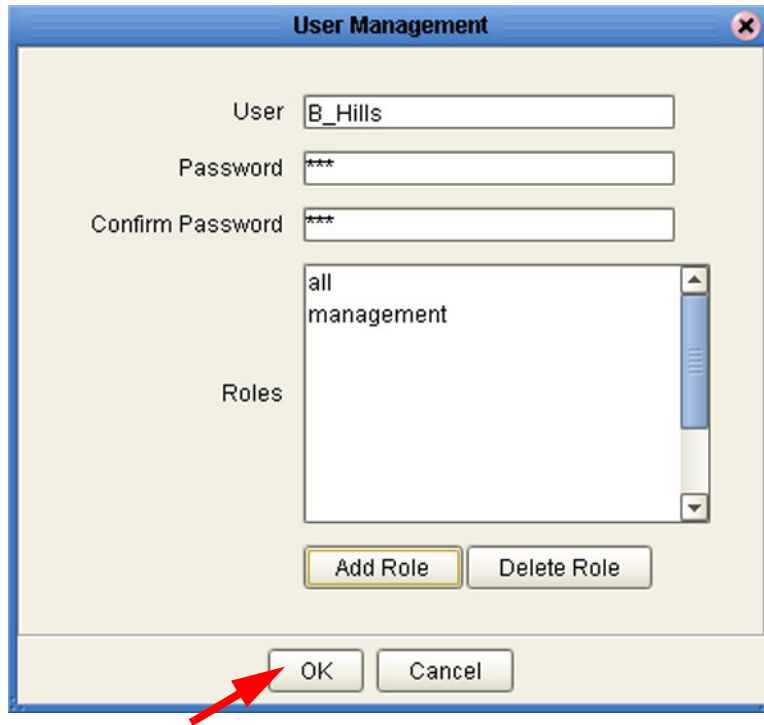
Figure 264 Add Role Dialog Box



- 6 Select the desired role, for example **management**, and click **OK**. The new role appears in the list for the selected user (see Figure 265).

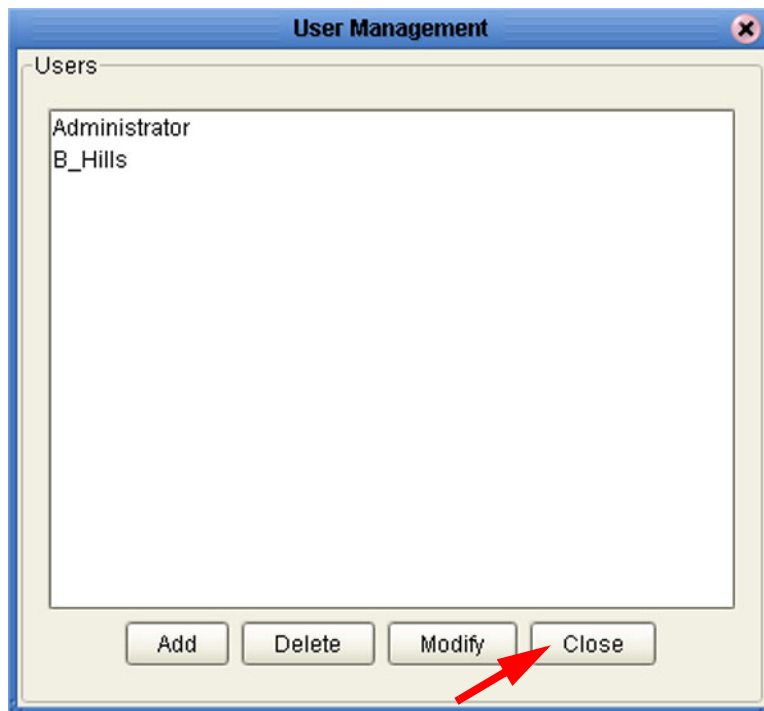
Note: If the desired role is not listed in the Add Role dialog box, you can create a new role; see [Creating User Roles](#) on page 303.

Figure 265 User Management - Add Role (2)



- 7 Click **OK** to return to the initial User Management dialog box.

Figure 266 User Management Dialog Box

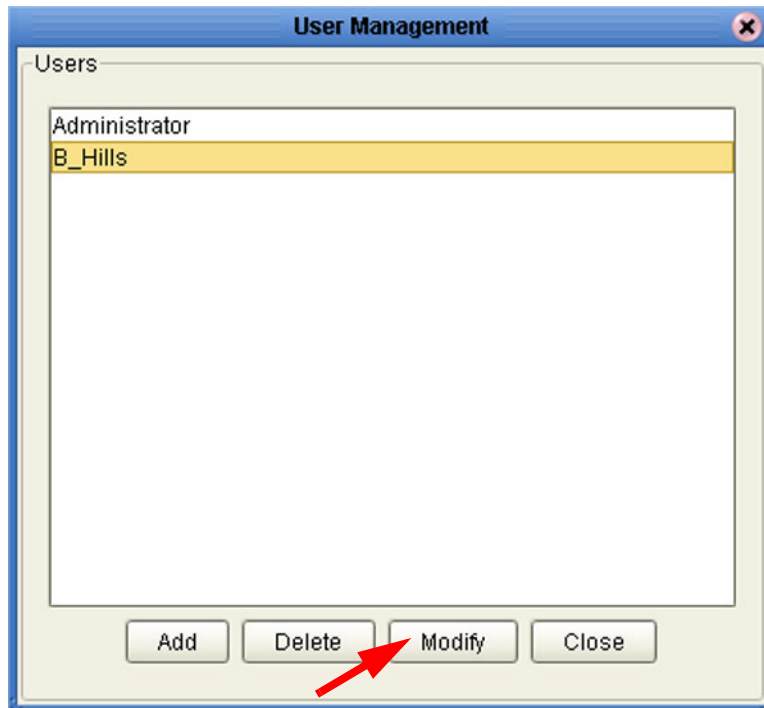


- 8 Click **Close** to return to the Enterprise Designer.

To delete an existing role for a current user

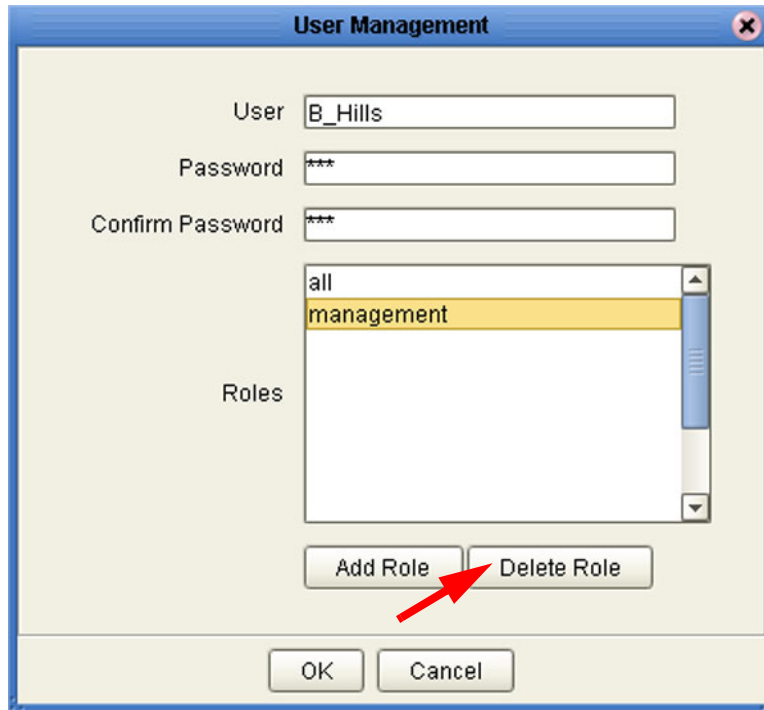
- 1 In the Project Explorer, select the **Repository** icon.
- 2 Right-click to display the Repository context menu.
- 3 Select **User Management** to display the User Management dialog box shown in Figure 267.

Figure 267 User Management Dialog Box



- 4 To delete a role for a current user, select the user from the displayed list and click **Modify** to display the second User Management dialog box shown in Figure 268.

Figure 268 User Management - Delete Role



- 5 Select the role to be deleted, and click the **Delete Role** button to delete the role for the selected user. The role will disappear from the list.
- 6 Click **OK** to return to the initial User Management dialog box.
- 7 Click **Close** to return to the Enterprise Designer.

Note: Every user entered into the system is automatically assigned to the **all** role, which is required to connect to the Repository; therefore, the **all** role cannot be deleted from the user (see Figure 269).

Figure 269 Error Warning Box



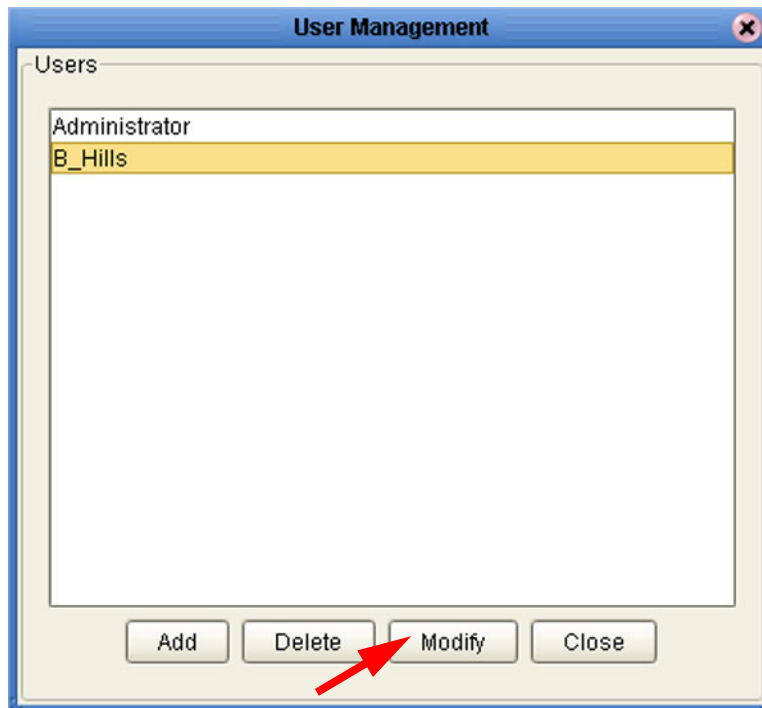
Creating User Roles

New roles having limited functionality can be created for your own use. See the discussion in [Roles](#) on page 294.

To create a new role for a current user

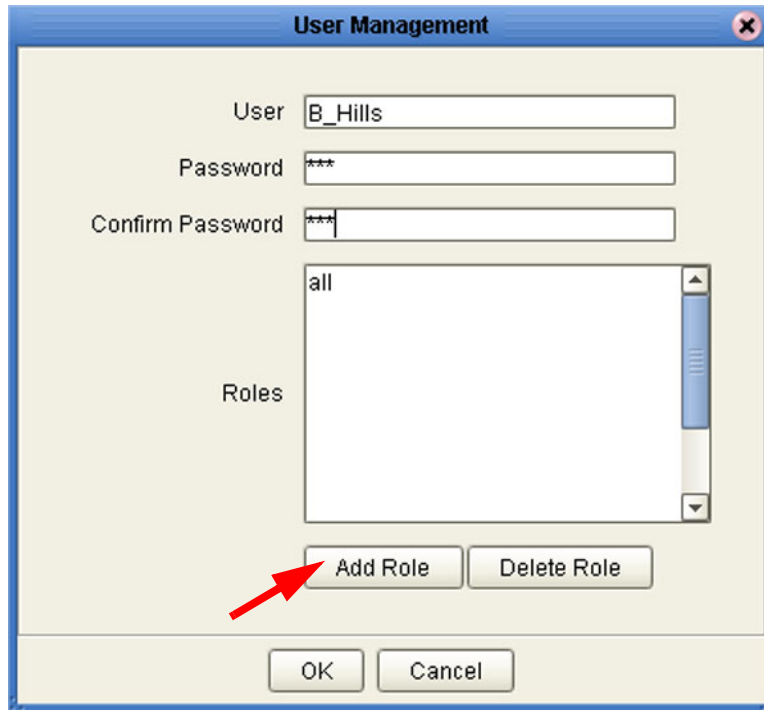
- 1 In the Project Explorer, select the **Repository** icon.
- 2 Right-click to display the Repository context menu.
- 3 Select **User Management** to display the User Management dialog box shown in Figure 270.

Figure 270 User Management Users List



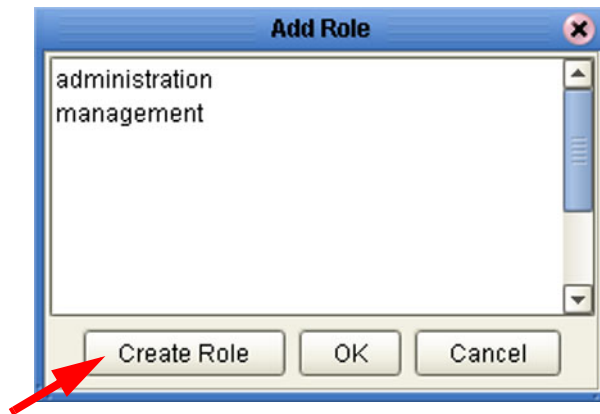
- 4 Select the desired user from the displayed list and click **Modify** to display the second User Management dialog box shown in Figure 271.

Figure 271 User Management - Create Role (1)



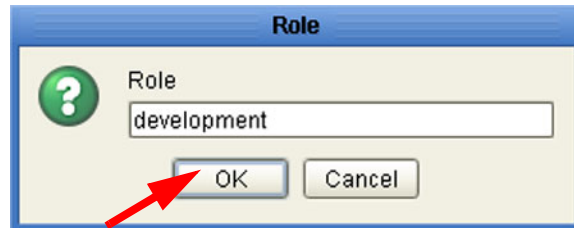
- 5 Click the **Add Role** button, which displays the dialog box shown in Figure 272.

Figure 272 Add Role Dialog Box (1)



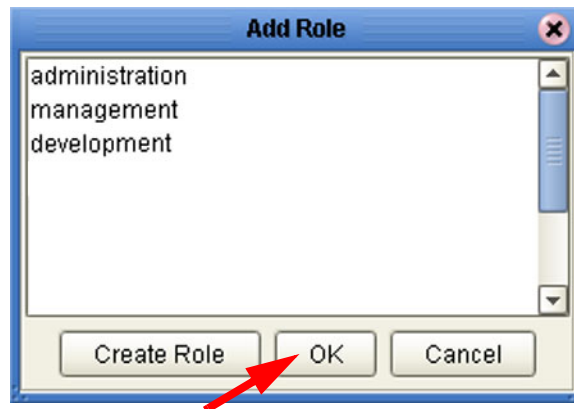
- 6 Click the **Create Role** button, which displays the dialog box shown in Figure 273.

Figure 273 Role Dialog Box



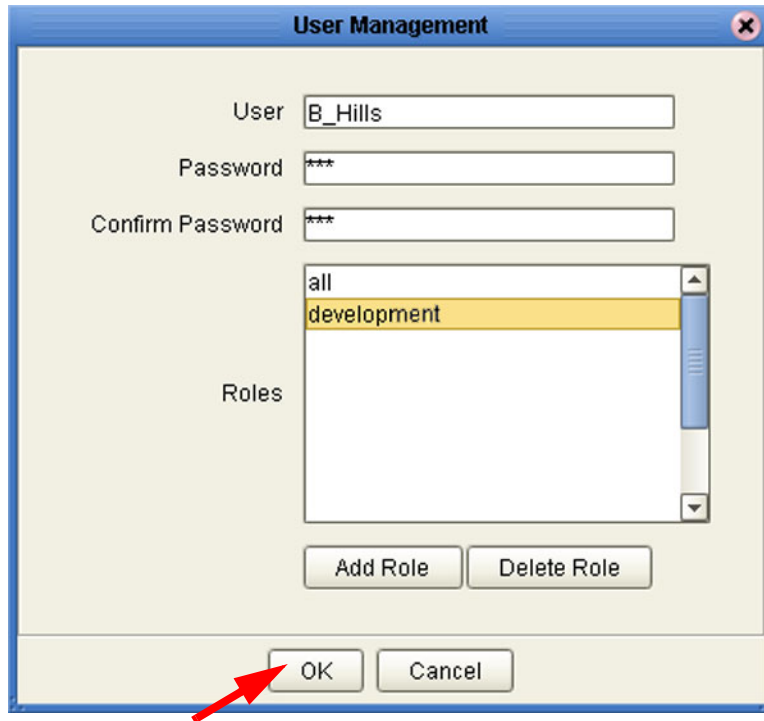
- 7 Type in the name of the new role you are creating, for example **development**.
- 8 Click **OK** to return to the *Add Role* dialog box, where the new role has been added to the list (see Figure 274).

Figure 274 Add Role Dialog Box (2)



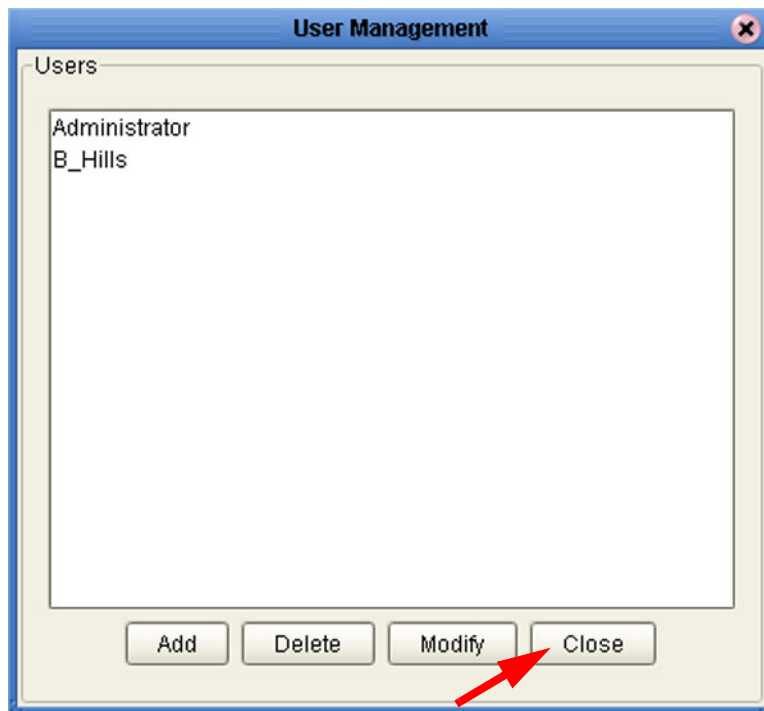
- 9 Select the new role and click **OK**. The role is now added for the selected user (see Figure 275).

Figure 275 User Management - Create Role (2)



- 10 Click **OK** to return to the initial User Management dialog box.

Figure 276 User Management Users List



- 11 Click **Close** to return to the Enterprise Designer.

Using LDAP

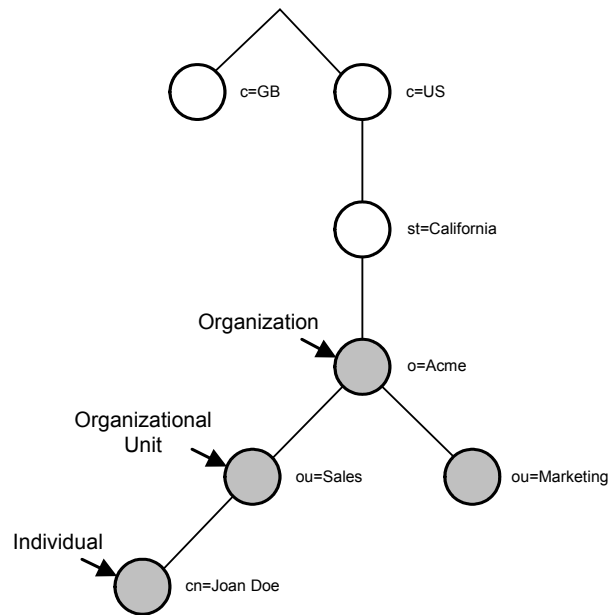
The Lightweight Directory Access Protocol (LDAP) is a protocol for accessing X.500-based directory services. LDAP runs over TCP/IP or other connection oriented transfer services. LDAP provides a mechanism for a client to authenticate, or prove, its identity to a directory server, paving the way for rich access control to protect the information the server contains. LDAP also supports privacy and integrity security services.

The LDAP directory service is based on a client-server model. One or more LDAP servers contain the data making up the directory information tree (DIT). The client connects to servers and asks it a question. The server responds with an answer and/or with a pointer to where the client can get additional information (typically, another LDAP server). No matter which LDAP server a client connects to, it sees the same view of the directory; a name presented to one LDAP server references the same entry it would at another LDAP server.

The LDAP information model is based on *entries*. An entry is a collection of attributes that has a globally-unique Distinguished Name (DN). The DN is used to refer to the entry unambiguously. Each of the entry's attributes has a *type* and one or more *values*. The types are typically mnemonic strings, like **cn** for common name, or **mail** for E-mail address. The syntax of values depend on the attribute type. For example, a **cn** attribute might contain the value Joan Doe. A mail attribute might contain the value jdoe@example.com. A **jpegPhoto** attribute would contain a photograph in the JPEG (binary) format.

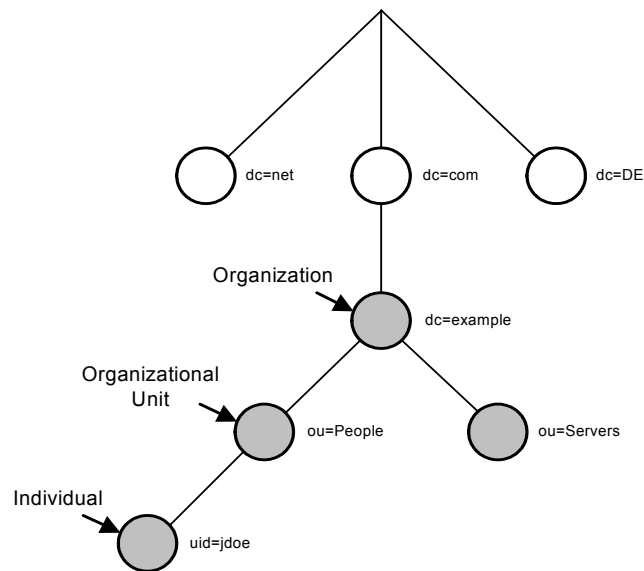
In LDAP, directory entries are arranged in a hierarchical tree-like structure. Traditionally, this structure reflected the geographic and/or organizational boundaries. Entries representing countries appear at the top of the tree. Below them are entries representing states and national organizations. Below them might be entries representing organizational units, people, printers, documents, or just about anything else you can think of. Figure 277 shows an example LDAP directory tree using traditional naming.

Figure 277 LDAP Directory Tree (Traditional Naming)



The tree may also be arranged based upon Internet domain names. This naming approach is becoming increasingly popular, since it allows for directory services to be located using the DNS. Figure 278 shows an example LDAP directory tree using domain-based naming.

Figure 278 LDAP Directory Tree (Internet Naming)



In addition, LDAP allows you to control which attributes are required and allowed in an entry through the use of a special attribute called **objectClass**. The values of the **objectClass** attribute determine the schema rules the entry must obey.

Referencing and Accessing Information

An entry is referenced by its distinguished name (DN), which is constructed by taking the name of the entry itself (called the Relative Distinguished Name or RDN) and concatenating the names of its ancestor entries. For example, the entry for Joan Doe in the previous Internet naming example has an RDN of `uid=jdoe` and a DN of `uid=jdoe,ou=People,dc=example,dc=com`.

LDAP defines operations for interrogating and updating the directory. Operations are provided for adding and deleting an entry from the directory, changing an existing entry, and changing the name of an entry. Most of the time, though, LDAP is used to search for information in the directory. The LDAP search operation allows some portion of the directory to be searched for entries that match some criteria specified by a search filter. Information can be requested from each entry that matches the criteria.

For example, you might want to search the entire directory sub-tree at and below `dc=example, dc=com` for people with the name Joan Doe, retrieving the E-mail address of each entry found. On the other hand, you might want to search the entries directly below the `st=California, c=US` entry for organizations with the string Acme in their name, and that have a fax number. LDAP allows you to do either.

Configuring LDAP Servers for Configuration User Management

ICAN 5.0 supports Microsoft's Active Directory Server (ADS) and Sun Microsystem's Sun ONE Directory Server. When a user attempts to log into the Repository, his user name and password are checked against the user name and password that is stored in either Active Directory or the Sun ONE Directory Server. The list of roles for that user is also retrieved from the respective server to authorize his access to various objects in the repository.

By default, the ICAN Suite comes with the three pre-configured roles *all*, *administration*, and *management*. Additionally, ICAN users are also allowed to create their own, custom roles. To support these roles as defined in the ICAN Suite, you must create the roles in the LDAP servers (we will demonstrate only the predefined roles).

Configuring the Active Directory Server

ADS does not support the concept of Roles, so you must simulate Roles in ADS using the ADS concept of Groups. To avoid the confusion of ADS's own Groups and ICAN's Roles, the ICAN Roles need to be located under a directory other than the ADS Groups directory.

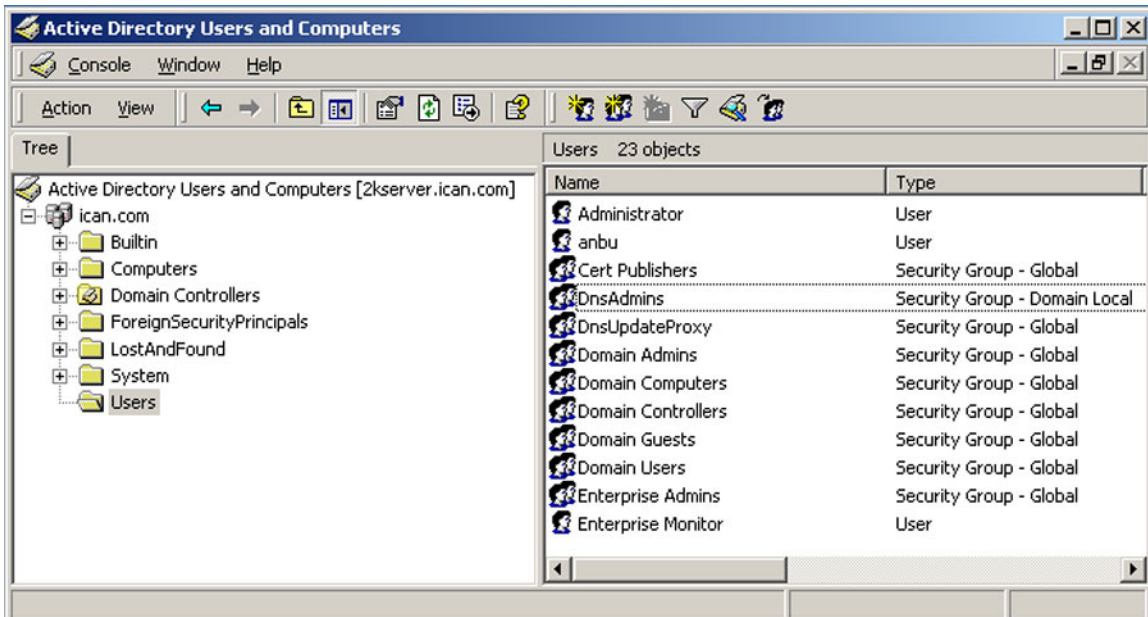
To create the ICAN Roles under their own node in ADS

- 1 Start the Active Directory User Management console by following the path indicated below on the computer where the Active Directory Server is running:

Start > Programs > Administrative Tools > Active Directory Users and Computers

This displays the window shown in Figure 279.

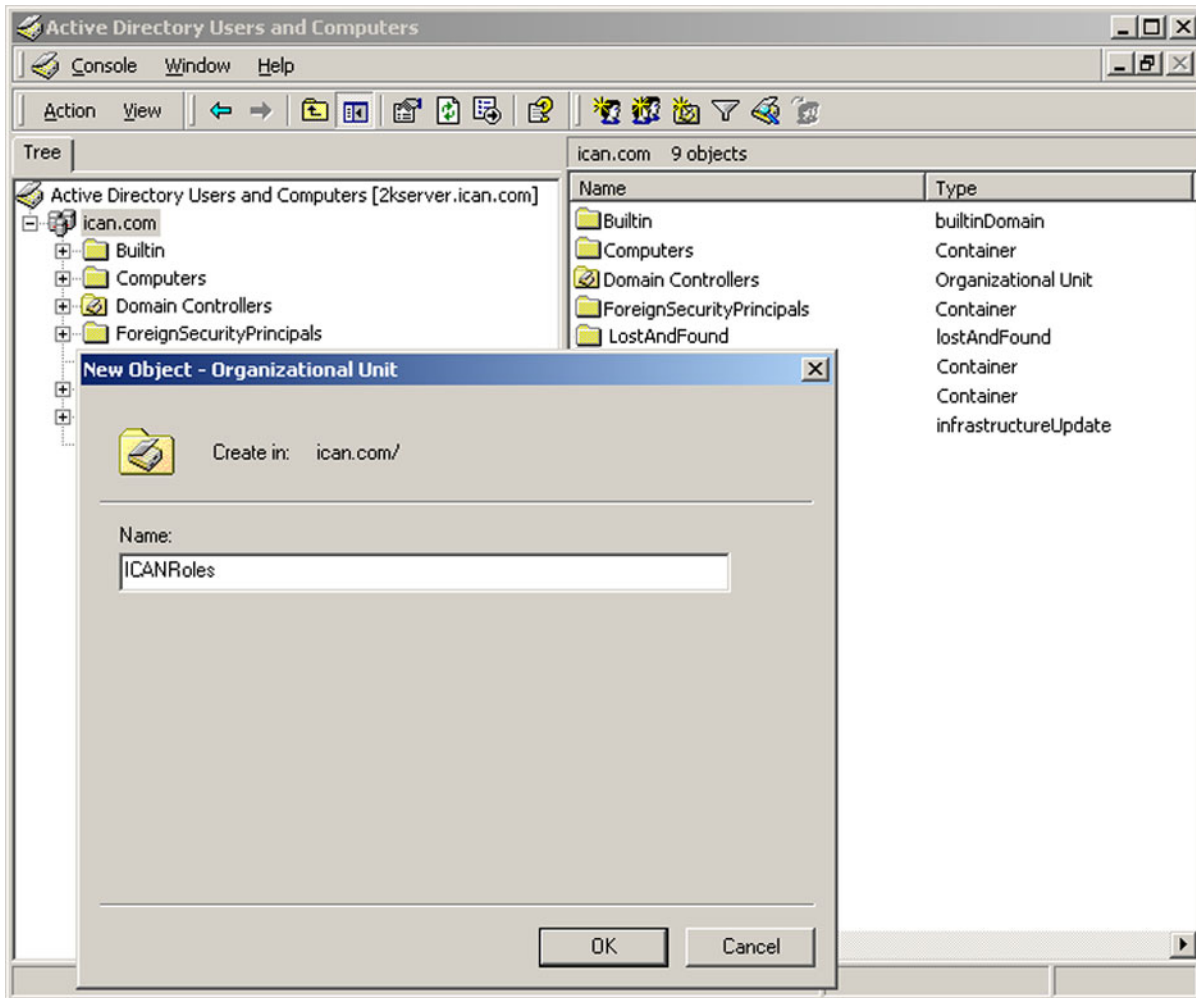
Figure 279 Active Directory Users and Computers Window



Note: "ican.com" is a fictitious URL.

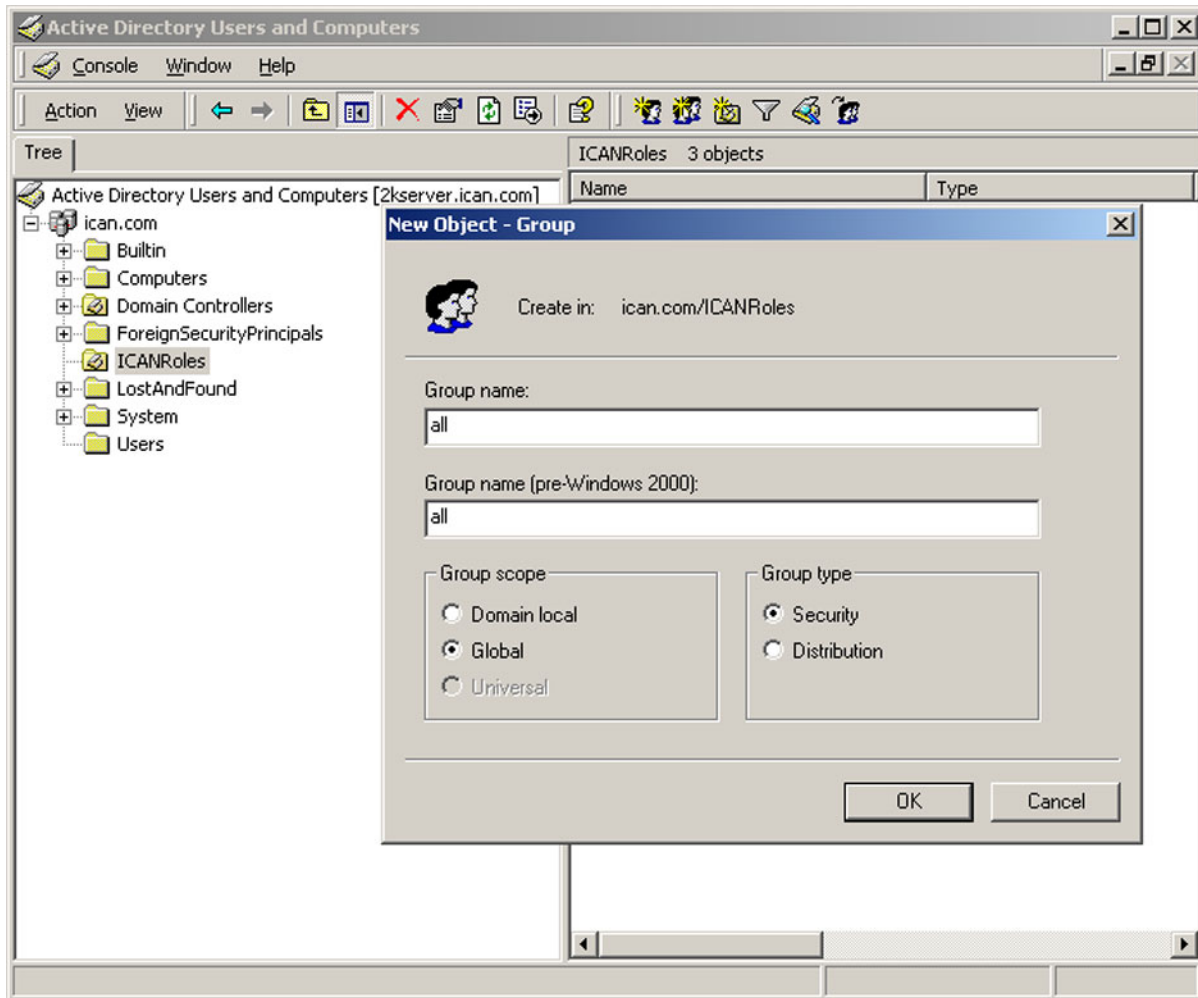
- 2 On the root node, right-click and select **New > Organizational Unit** to display the *New Object - Organization Unit* dialog box (see Figure 280).
- 3 Enter **ICANRoles** for the Name and click **OK**.

Figure 280 Active Directory - Create Organizational Unit



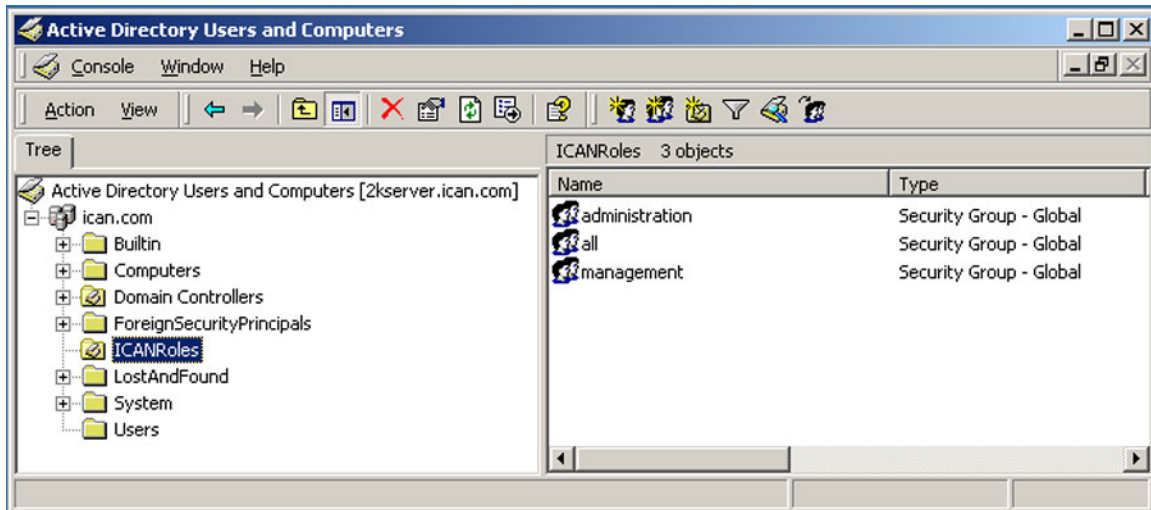
- 4 Under the *ICANRoles* directory, create new groups **all**, **administration**, and **management** (see Figure 281).

Figure 281 Active Directory - Create Roles Groups



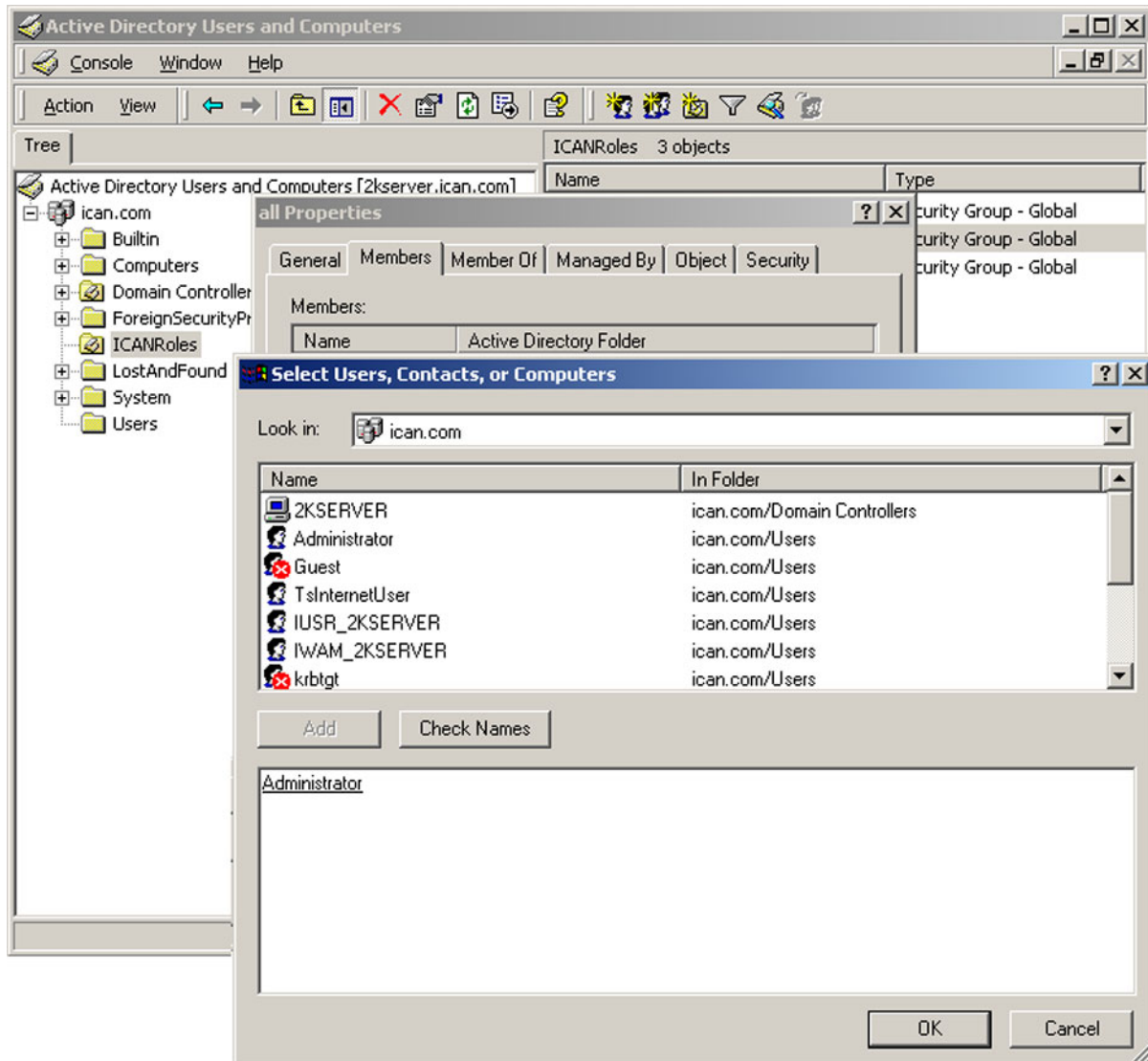
After adding all the three groups, you will see them under the *ICANRoles* as shown in Figure 282.

Figure 282 Active Directory - ICANRoles Directory



- 5 Add the *Administrator* user as a member of these groups by double-clicking on each of the groups and selecting **Administrator** from the dialog box (see Figure 283).

Figure 283 Active Directory - Select Administrator



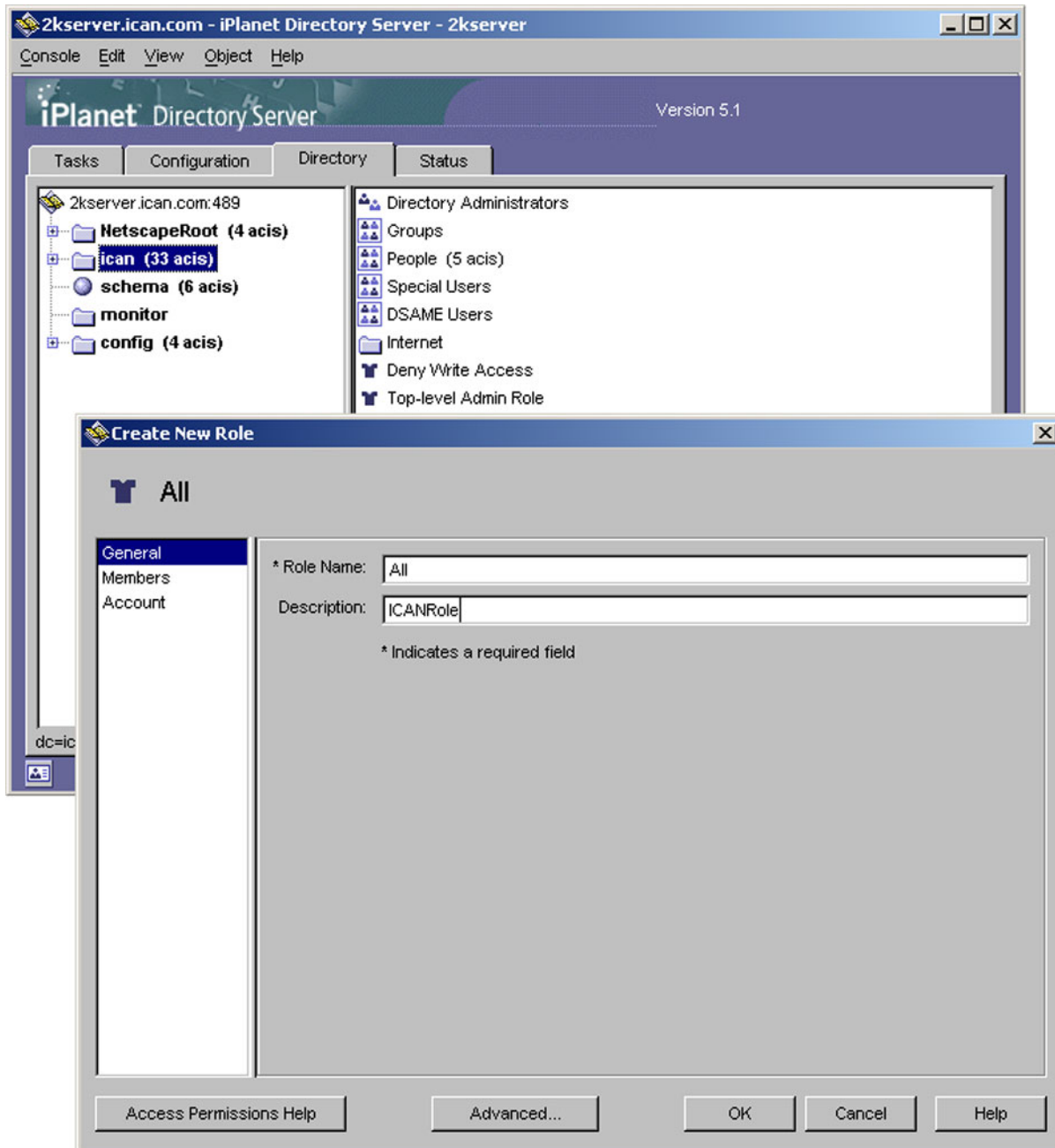
6 Configure the Active Directory Server for *anonymous read*.

Configuring the SunONE Server

To create the ICAN Roles in the SunONE Directory Server

- 1 Create the user **Administrator** under the *People* directory.
- 2 Create the roles **all**, **administration**, and **management** under the top node as shown in Figure 284.

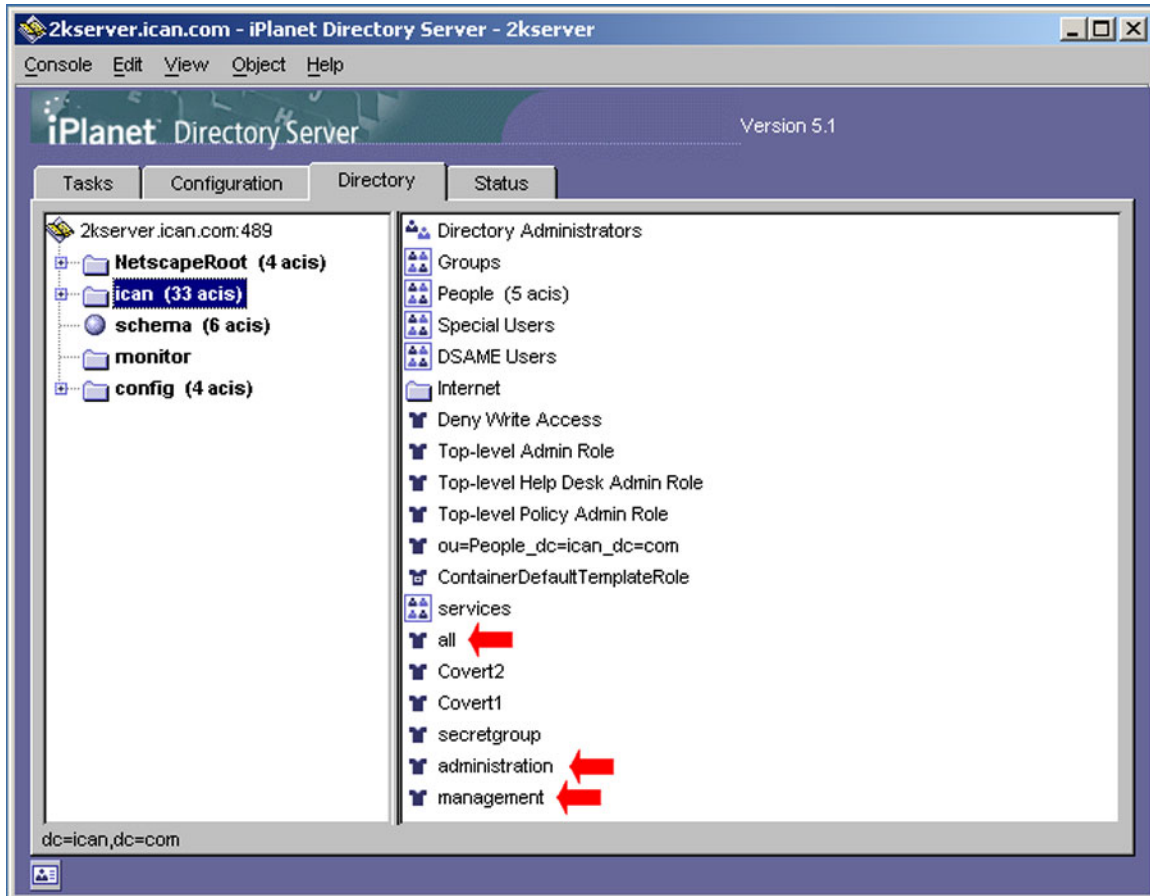
Figure 284 Sun ONE - Create New Role



Note: "ican.com" is a fictitious URL.

- 3 After creating the roles, you will see them listed as shown in Figure 285.

Figure 285 Sun ONE - Roles



Configuring the ICAN Repository to Use LDAP

To use an LDAP server with the ICAN system, a *Realm* element needs to be added to the ICAN Repository server's `server.xml` file. The Realm element attributes used by the default configuration are shown in Table 56. For a detailed description of all the possible attributes of this Realm class, please refer to the Tomcat documentation for JNDI Realm.

Table 56 Realm Element Attributes

Attribute	Parameter	Description/Notes
className		Always use the default className: "org.apache.catalina.realm.JNDIRealm"
connectionURL		Identifies the location of the LDAP server.
	LDAP_host	The LDAP server name; for example, 'localhost'.
	LDAP_port	The port your LDAP server listens for requests; for example, 389.
roleBase		The base entry for the role search. If not specified, the search base is the top-level directory context.
roleName		The attribute in a role entry containing the name of that role.
roleSearch		The LDAP search filter for selecting role entries. It optionally includes pattern replacements "{0}" for the distinguished name and/or "{1}" for the username of the authenticated user.
userBase		The entry that is the base of the subtree containing users. If not specified, the search base is the top-level context.
userPattern		A pattern for the distinguished name (DN) of the user's directory entry, following the syntax supported by the <code>java.text.MessageFormat</code> class with {0} marking where the actual username should be inserted.
userRoleName		The name of an attribute in the user's directory entry containing zero or more values for the names of roles assigned to this user. In addition you can use the roleName property to specify the name of an attribute to be retrieved from individual role entries found by searching the directory. If userRoleName is not specified, then all roles for a user derive from the role search.
userRoleNamePattern		A pattern for the distinguished name (DN) of the role's directory entry, following the syntax supported by the <code>java.text.MessageFormat</code> class with {0} marking the actual role name. This pattern is used to parse the DN to get the actual role name for authorization purposes in ICAN, where the actual username should be inserted.

Attribute	Parameter	Description/Notes
userSearch		The LDAP search filter to use for selecting the user entry after substituting the username in {0}.

Configuring the ICAN Suite for ADS

To add the Realm element for Microsoft Active Directory Server

- 1 Open *ICAN-root\repository\server\conf\server.xml*.
- 2 Add the following code snippet inside the **ENGINE** tag:

```
<Realm className="org.apache.catalina.realm.JNDIRealm"
        connectionURL="ldap://localhost:389"
        userBase="CN=Users,dc=ican,dc=com"
        userSearch="(cn={0})"
        roleBase="ou=ICANRoles,dc=ican,dc=com"
        roleName="cn"
        roleSearch="(member={0})"
/>
```

- 3 Save your changes and close the file.
- 4 Start your Active Directory server.
- 5 Shut down and restart your ICAN repository server.

Configuring the ICAN Suite for SunONE

To add the Realm element for the Sun ONE Directory Server

- 1 Open *ICAN-root/repository/server/conf/server.xml*.
- 2 Add the following code snippet inside the **ENGINE** tag:

```
< Realm className="org.apache.catalina.realm.JNDIRealm"
        connectionURL="ldap://localhost:489"
        userBase="CN=People,dc=ican,dc=com"
        userPattern="uid={0},ou=People,dc=ican,dc=com"
        userSearch="(uid={0})"
        roleName="nsroledn"
        roleNamePattern="cn={0},dc=ican,dc=com"
/>
```

- 3 Save your changes and close the file.
- 4 Start your Sun ONE Directory Server.
- 5 Shut down and restart your ICAN repository server.

12.2.2 Environment User Management

This section describes the management of users who would access the applications deployed in an enterprise, using the ICAN suite. Unlike the Configuration User Management, changes are not dynamic and the Logical Host must be rebooted to reflect the changes.

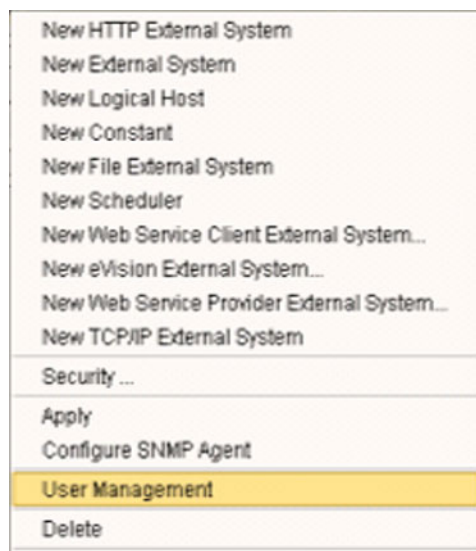
Creating and Configuring Users

When you create an Environment, it comes with one default user: *Administrator*. If you specify any user other than the Administrator in any of your application settings (such as in the Connectivity Map links), you need to create that user in that Environment by right-clicking on the Environment and selecting the **User Management** option.

To Open the User Management Dialog Box

- 1 In the Environment Explorer, right-click on an **Environment** to display its context menu (see Figure 286).

Figure 286 Environment Context Menu



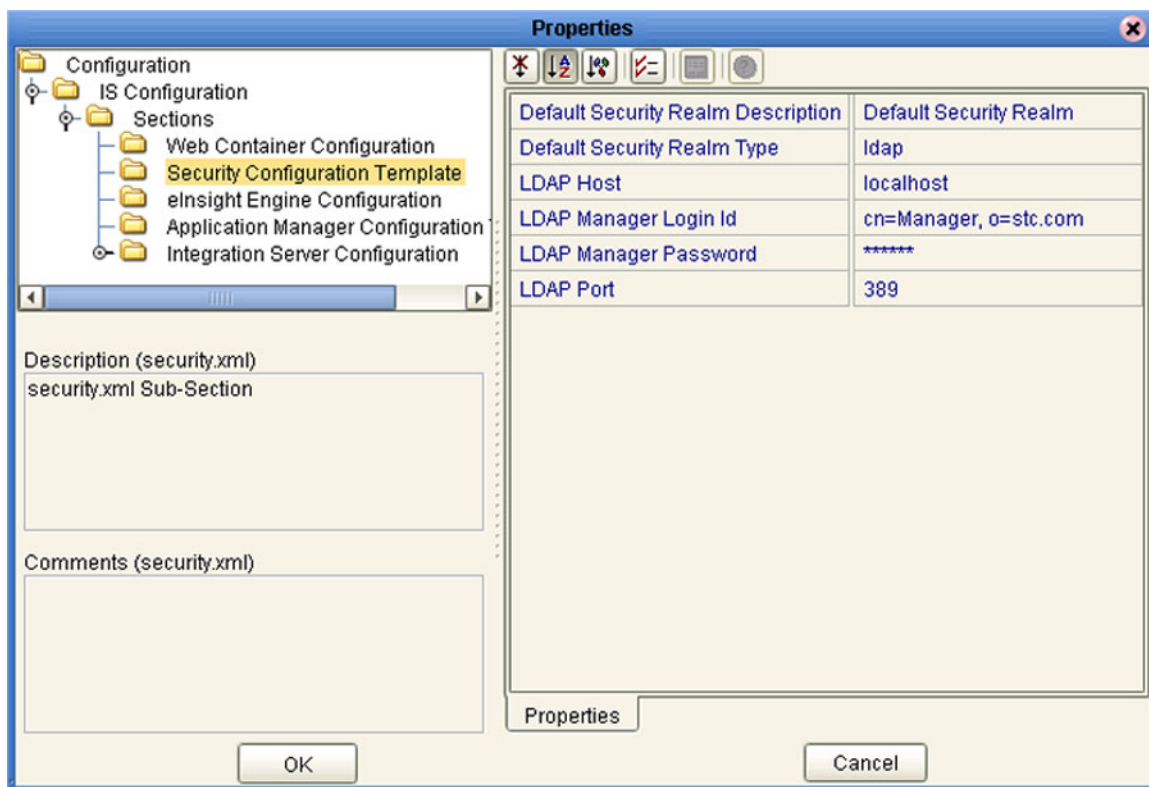
- 2 Select **User Management** to display the series of User Management dialog boxes for the Environment, which are identical to those shown in [Configuration User Management](#) on page 294. Follow the same procedure described in that section.
- 3 Reboot the Logical Host so that the run-time components can pick up the changes.

Configuring for LDAP Servers in Environment User Management

Integration servers running on the Logical Hosts can be configured to use the LDAP server for their authentication. Run-time authentication for an Integration Server is configured in the Properties Dialog Box for the Security Configuration Template (see Figure 287), where:

- **LDAP Host** is the computer on which the LDAP server is running, for example: *localhost*.
- **LDAP Manager Login Id** is the root distinguished name, for example: *cn=Manager, o=stc.com*.
- **LDAP Manager Password** is the root password, for example: *secret*.
- **LDAP Port** is the port on which the LDAP server is running, for example: *389*.

Figure 287 Security Configuration Template Properties Dialog Box



12.3 ACL Management

When you create any object such as Project, Connectivity Map, or Environment in the Enterprise Designer and store it in the repository, by default no Access Control List (ACL) is set on these objects. Therefore, no permission checks are triggered on these objects when actions are performed involving these objects such as activation or bootstrap. Every repository user will have access to every one of these objects if no ACLs have been specified.

ACL Properties allow you to assign **Read** and/or **Write** access to registered users for a selected object within a Repository. If an object does not have an ACL, all users are authorized to access that object. If an ACL is created for an object, an Administrator is added automatically with **Read** and **Write** permissions.

The actions on a node in Enterprise Designer are enabled or disabled based on the ACL of the repository object associated with the node.

- A user without the **Read** or **Write** permissions will not be able to expand a node to see the children; all the actions on that node will also be disabled.
- A user having only the **Read** permission can expand the node to see the child nodes. The enabling or disabling of the actions on that node will vary, however, based on the type of action. This is based on the ACL of the Repository Object and the Version Control status.

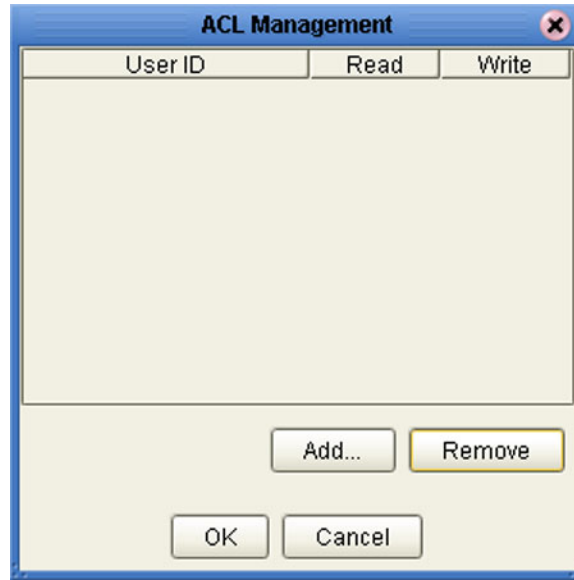
This logic for this is dependent upon the type of action and the module to which it belongs. For example, the *Delete* action on the Project Elements is disabled if the user does not have the **Write** permission on *both* the Project Element and the parent Project.

- If the user has both the **Read** and **Write** permissions, or if the object does not have any ACL, all the actions on that node are enabled for that user.

To add and assign access rights to a user

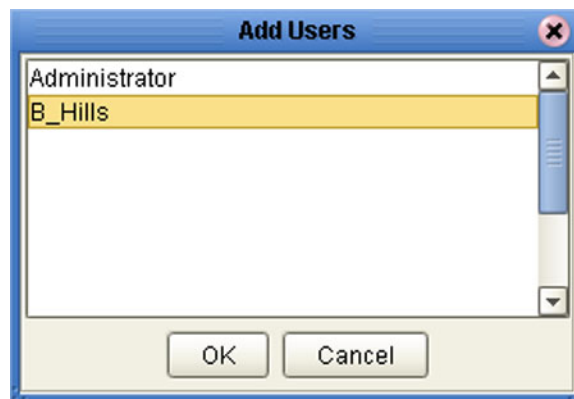
- 1 From the Project Explorer, select an object icon.
- 2 Right-click to display the object's context menu.
- 3 Select **ACL Management** to display the dialog box shown in Figure 288.

Figure 288 ACL Management Dialog Box (1)



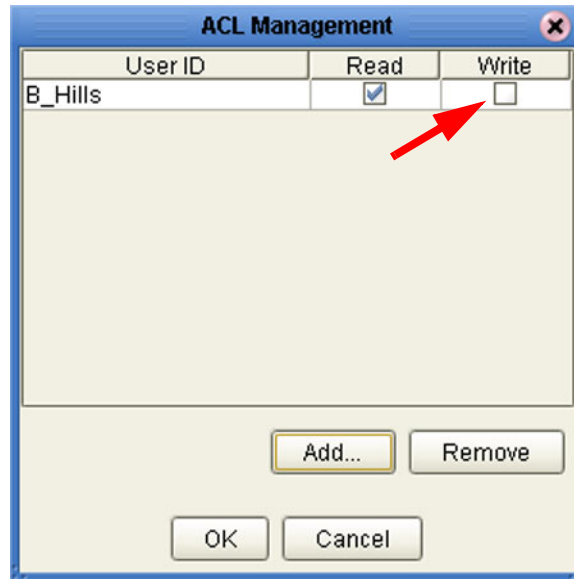
- 4 Click **Add** to display the dialog box shown in Figure 289.

Figure 289 ACL Add Users Dialog Box



- 5 Select the *existing Repository user* to whom you want to grant access to the object.
- 6 Click **OK** to add the user to the ACL Management list. The user is automatically assigned **Read** access to the object.

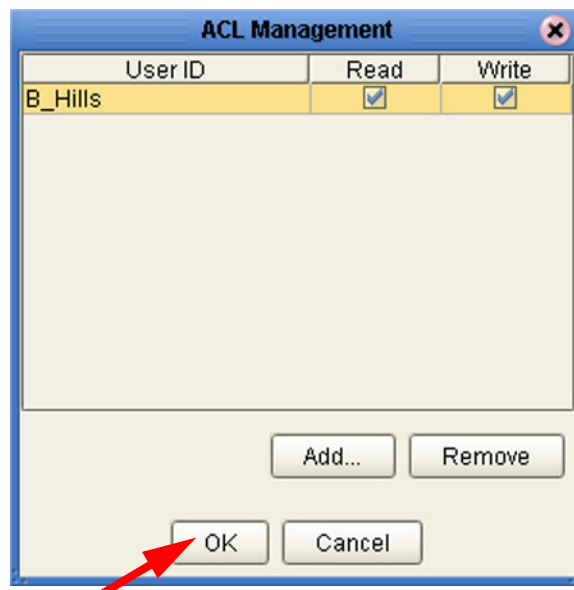
Figure 290 ACL Management Dialog Box (2)



- 7 Select the **Write** check box for the user if you want them to be able to edit the Project (see Figure 291). This check box can be cleared later if you need to remove **Write** access for this user.

Note: The Administrator's permissions are preset and cannot be modified.

Figure 291 ACL Management Dialog Box (3)



- 8 Click **OK** to save your changes.

Since all ACL users have **Read** access, you cannot clear the **Read** check box; you simply select the user and click **Remove** to delete the user from the ACL for the object.

12.4 ICAN Component Security

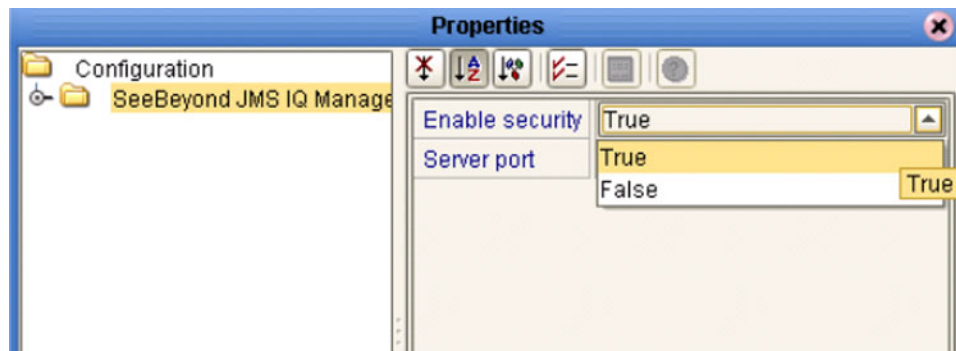
12.4.1 Message Server Security

Message server security can be enabled and disabled by modifying the configuration properties for the message server.

To enable message server security

- 1 In the Environment Explorer, right-click on a message server to display its context menu.
- 2 Select **Properties** to display the message server Properties Dialog Box.
- 3 Select **JMS IQ Manager** in the left panel of the Properties Dialog Box to display its properties (see Figure 292).

Figure 292 JMS IQ Manager Properties



- 4 Select **True** from the drop-down menu to enable security.

Note: See the *eGate Integrator JMS Reference Guide* for more information on this subject.

12.4.2 JMS Client Security

Security settings for the generated J2EE applications can be specified by modifying the configuration properties for the JMS Client connections in the Connectivity Map. JMS Client security properties must be specified if security is enabled for the message server.

To enable JMS Client security

To enable authentication of Collaborations in JMS Client connections, set the User ID and Password properties as demonstrated in Figure 293.

Figure 293 JMS Client Security Properties



Note: See the *eGate Integrator JMS Reference Guide* for more information on this subject.

12.5 Using SSL/HTTPS in ICAN

12.5.1 Overview

Secure Socket Layer (SSL) is a technology that allows Web browsers and Web servers to communicate over a secured connection. In this secure connection, the data that is being sent is encrypted before being sent, then decrypted upon receipt and prior to processing. Both the browser and the server encrypt all traffic before sending any data.

Another important aspect of the SSL protocol is *authentication*. During your initial attempt to communicate with a Web server over a secure connection, that server will present your Web browser with a set of credentials in the form of a server certificate. The purpose of the certificate is to verify that the site is who and what it claims to be. In some cases, the server may request a certificate that the client is also who and what it claims to be (which is known as client authentication).

Certificates and Keys

In order to implement SSL, a Web server must have an associated certificate for each external interface, or IP address, that accepts secure connections. The theory behind this design is that a server should provide some kind of reasonable assurance that its owner is who you think it is, particularly before receiving any sensitive information. It may be useful to think of a certificate as a “digital driver's license” for an Internet address. It states with which company the site is associated, along with some basic contact information about the site owner or administrator.

A certificate is a digitally-signed statement from one entity (person, company, etc.), saying that the public key (and some other information) of some other entity has a particular value. When data is digitally signed, the signature can be verified to check the data integrity and authenticity. *Integrity* means that the data has not been modified or tampered with, and *authenticity* means the data indeed comes from whoever claims to have created and signed it.

The certificate is cryptographically signed by its owner and is difficult for anyone else to forge. For sites involved in e-commerce, or any other business transaction in which authentication of identity is important, a certificate can be purchased from a well-known Certificate Authority (CA) such as **Verisign** or **Thawte**.

Certificates are used with the HTTPS protocol to authenticate Web clients. The HTTPS service of the ICAN Repository server will not run unless a server certificate has been installed. Use the procedure outlined below to set up a server certificate that can be used by the ICAN repository server to enable SSL.

The Keytool Utility

One tool that can be used to set up a server certificate is **keytool**, a key and certificate management utility that ships with the J2SE SDK. It enables users to administer their own public/private key pairs and associated certificates for use in self-authentication (where the user authenticates himself/herself to other users/services) or data integrity

and authentication services, using digital signatures. It also allows users to cache the public keys (in the form of certificates) of their communicating peers.

Keytool stores the keys and certificates in a so-called *keystore*. The default keystore implementation implements the keystore as a file. It protects private keys with a password.

The keytool utility enables you to create the certificate. The version that ships with the J2SE SDK programmatically adds a Java Cryptographic Extension provider that has implementations of RSA algorithms. This provider enables you to import RSA-signed certificates.

12.5.2 Installation and Configuration

To install and configure SSL support, you need to perform the following steps:

- 1 Generate a key pair and a self-signed signature.
- 2 Obtain a digitally-signed certificate from a Certificate Authority.
- 3 Import/install the certificate.
- 4 Configure the SERVER.XML file.
- 5 Test the new SSL connection.

The following procedures employ the **keytool** utility.

To generate a key pair and a self-signed signature

- 1 From the command prompt, enter:

```
JAVA_HOME\bin\keytool -genkey -keyalg RSA -alias ICAN -keystore  
keystore_filename
```

Where, for example:

```
keystore_filename =  
ICAN_HOME\repository\server\conf\ssl\mykeystore
```

- 2 Enter your keystore password: (for example, **seebeyond**)
- 3 The keytool program will ask some questions, such as the following, which you need to answer appropriately:
 - A What is your first and last name?
 - B What is the name of your organizational unit?
 - C What is the name of your organization?
 - D What is the name of your City or Locality?
 - E What is the name of your State or Province?
 - F What is the two-letter country code for this unit?
- G Is CN=*your_first_and_last_name*, OU=*your_organizational_unit*, O=*your_organization_name*, L=*your_city*, ST=*your_state_or_province*, C=*your_two_letter_country_code* correct?
- 4 Enter key password for ICAN: (RETURN, if same as keystore password)

Note: The example used the following name for the keystore file to be generated: **ICAN-root\repository\server\conf\ssl\mykeystore**. You can use this name or another, as long as you use the same name throughout the configuration process.

To obtain a digitally-signed certificate

- 1 From the command prompt, enter the following to generate a Certificate Signing Request (CSR):

```
JAVA_HOME\bin\keytool -certreq -alias ICAN -keyalg RSA -file
csr_filename -keystore keystore_filename
```

- 2 Send the CSR for signing.
 - ♦ For example, if you are using Verisign CA, go to **http://digitalid.verisign.com/**. Verisign will send the signed certificate in E-mail.
- 3 Store the signed certificate in a file.

Note: You may skip the following step if you are using only the self-signed certificate. If you are using a self-signed certificate or a certificate signed by a CA that your browser does not recognize, a dialog will be triggered the first time you try to access the server. You can then choose to trust the certificate for this session only, or permanently.

To import the certificate

From the command prompt, enter the following to install the CA certificate:

```
JAVA_HOME\bin\keytool -import -trustcacerts -alias ICAN -file ca-
cert_filename -keystore keystore_filename
```

Note: You must have the required permissions to modify the **JAVA_HOME\jre\lib\security\cacerts** file.

An example Connector element for an SSL connector is included in the default server.xml file. This Connector element is commented out by default. To enable the SSL Connector for the ICAN repository, remove the comment tags around the SSL Connector element. Changes to the server.xml file are read by the ICAN repository server when it is started.

To configure SERVER.XML

- 1 Shut down the ICAN repository server if it is running.
- 2 Open the file **ICAN_HOME/repository/conf/server.xml** using a text editor.
- 3 Find the following section of code in the file (try searching for *Connector*). The following snippet can be put before the first instance of *Connector* in the server.xml file.

```
<Connector
  className="org.apache.catalina.connector.http.HttpConnector"
  port="8443"
  minProcessors="5"
  maxProcessors="75"
  enableLookups="true"
  acceptCount="10"
  debug="0"
```



```
        scheme="https"  
        secure="true"  
    >  
        <Factory  
            className="org.apache.catalina.net.SSLServerSocketFactory"  
            clientAuth="false"  
            protocol="TLS"  
            keystoreFile="ICAN_HOME\repository\server\conf\ssl"  
            keystorePass="seebeyond"  
        />  
    </Connector>
```

- 4 Save and close the file.
- 5 Start the ICAN repository server.

Step 5: Test the new SSL connection

- 1 For testing purposes, and to verify that SSL support has been correctly installed on the ICAN repository server, load the default ICAN repository server introduction page with the following URL:

```
https://localhost:8443/
```

The *https* in this URL indicates that the browser should be using the SSL protocol. The port of 8443 is where the SSL Connector was created in the previous step.

- 2 The first time you load this application, the New Site Certificate dialog displays. Select **Next** to move through the series of New Site Certificate dialogs, and select **Finish** when you reach the last dialog.

Important: *You should still have the option to use HTTP to connect to the Enterprise Designer—system administrators should **not** block the HTTP port.*

Logging

This chapter provides information about eGate Integrator's logging features.

13.1 Overview

While a deployment profile is active and running, eGate Integrator automatically generates log messages for the run-time components (Logical Host, Integration Servers, and Message Servers). Enterprise Designer, Enterprise Monitor, and the Repository also have log files, as do several other eGate components. The logging level determines what type of information is recorded, as the logs only contain messages of the specified severity level and higher.

The log message files that are generated for the various eGate components, and their locations, are identified in [Basic Log Files and Locations](#) on page 333 and [Run-Time Log Files and Locations](#) on page 337. The corresponding log configuration files, and their configurable properties, are also described.

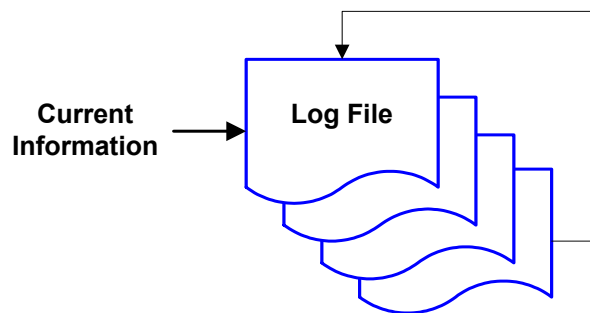
Logs can be viewed and their levels set using the Enterprise Manager, as described in [Viewing Logs](#) on page 48 and [Setting Log Levels](#) on page 50.

13.2 eGate Logs

13.2.1 Log File System

While a deployment profile is active and running, eGate Integrator automatically generates log messages for the various runtime components. Other eGate components, such as the Repository, maintain log files whenever they are being used. The log files constitute a recirculating stack (see Figure 294). You can specify both the file size and number of files in the stack for each Logical Host and Integration Server instance (the respective property names are `MaxFileSize` and `MaxBackupIndex`).

Figure 294 Recirculating Log File Stack



As soon as the maximum file size is reached in the currently active log file, a new log file is created. When the number of files in the stack reaches the specified limit, the oldest one is deleted when the new one is created. The effect is that the oldest one is emptied and moved to the top of the stack. A separate stack is maintained for each log file type.

Run-time log files are initialized during the installation of a new logical host, so if you reinstall a logical host, all existing log files are deleted. If you want to preserve log files (for example, on a weekly basis), you can copy the log files to a backup storage location.

13.2.2 Log File Management

Log files can be managed from the Enterprise Manager, as described in:

- [Viewing Logs](#) on page 48
- [Setting Log Levels](#) on page 50

Additional management can be performed from the command line. Locations of all log files, their corresponding configuration files, and configurable properties, are listed in:

- [Basic Log Files and Locations](#) on page 333
- [Run-Time Log Files and Locations](#) on page 337

13.2.3 Logging Model

The ICAN logging system is based on the open-source *log4j* API, which is fast, reliable, and extensible, but also relatively simple to use. Log4j has three main components: *Loggers*, *Appenders* and *Layouts*. These three types of components work together to enable the logging of messages according to message type and level, and to allow control (at run time) of how these messages are formatted and where they are reported.

Loggers

The *Logger* is the core component of the logging process, and is responsible for handling the majority of log operations. There are five basic logging levels defined in the log4j API, as shown in the following table.

Table 57 Logging Levels

Level	Designates
FATAL	Very severe error events that will presumably lead eGate to abort.
ERROR	Error events that might still allow eGate to continue running.
WARN	Potentially harmful situations.
INFO	Informational messages that highlight the progress of eGate at a coarse-grained level.
DEBUG	Informational events that are most useful for debug eGate at a fine-grained level. This is the default setting.

↑
Event Severity
↓
Events Logged

A logger will only output messages having a severity level that is higher than or equal to the set level. If the level of a logger is not set, it will default to the DEBUG level, which results in the maximum amount of logging. This situation should be avoided during routine operation because of the negative impact on performance and increased file storage requirements.

Appenders

Appenders are responsible for controlling the output of log operations. Loggers are configured by specifying their Appender properties, as listed in the configuration properties tables. The recirculating stack behavior of the log file system is controlled by the `log4j.RollingFileAppender` method.

Layouts

Layouts are responsible for formatting the output of the loggers, as displayed in the Monitor (see [Viewing Logs](#) on page 48).

13.3 Basic Log Files and Locations

13.3.1 Repository

- Master Repository log:

ICAN-root/repository/logs/repository.log

- Master Repository log configuration file:

ICAN-root/repository/server/webapps/repositoryconfig.properties

Table 58 Properties for the Master Repository Log

Property	Default Values
log4j.appender.RepositoryAppender	org.apache.log4j.RollingFileAppender
log4j.appender.RepositoryAppender.File	<i>ICAN-root/repository/logs/repository.log</i>
log4j.appender.RepositoryAppender.MaxFileSize	1000KB
log4j.appender.RepositoryAppender.MaxBackupIndex	10
log4j.appender.RepositoryAppender.layout	org.apache.log4j.PatternLayout
log4j.appender.RepositoryAppender.layout.Conversion Pattern	%d{ddMM HH:mm:ss} %5p [%t] - %m%n
log4j.logger.com.stc.repository	DEBUG, RepositoryAppender

- UNIX Repository log:

ICAN-root/repository/server/logs/catalina.out

- Repository installation log:

ICAN-root/repository/logs/install.log

- Administration servlet log:

ICAN-root/repository/server/logs/hostname_admin_log.date.txt

- Default repository and manifest servlet log:

ICAN-root/repository/server/logs/hostname_log.date.txt

- Examples servlet log:

ICAN-root/repository/server/logs/hostname_examples_log.date.txt

- Connection log:

ICAN-root/repository/logs/connection.log

- FTP log:

ICAN-root/repository/logs/repoftp.log

- UDDI Repository log:

ICAN-root/repository/logs/stcuddi.log

- UDDI Repository log configuration file:

ICAN-root/repository/server/webapps/stcuddi/conf/log4j.properties

Table 59 Properties for the UDDI Repository Log

Property	Default Values
log4j.appender.juddilog	org.apache.log4j.RollingFileAppender
log4j.appender.juddilog.File	<i>ICAN-root/repository/logs/stcuddi.log</i>
log4j.appender.juddilog.MaxFileSize	10MB
log4j.appender.juddilog.MaxBackupIndex	3
log4j.appender.juddilog.layout	org.apache.log4j.TTCCLayout
log4j.appender.juddilog.layout.ContextPrinting	true
log4j.appender.juddilog.layout.DateFormat	ISO8601
log4j.rootLogger	DEBUG, juddilog

- Deployment Application log:

ICAN-root/repository/lh-deployment-servlet/deployment-servlet.log

Note: The Repository Deployment Application log is related to all deployment actions spawned by invoking either the **Apply** menu option from Enterprise Designer or invoking the bootstrap script. If any errors occur during these invocations, this log will contain the root cause of the problem—if the problem originated from the deployment application residing on the repository server.

13.3.2 ESR Installer Logs

- Log file:

ICAN-root/esrs.log

- Log configuration file:

ICAN-root/ESRs/log4j.properties

Table 60 Properties for the ESR Installer Log

Property	Default Values
log4j.appender.juddilog	org.apache.log4j.RollingFileAppender
log4j.appender.juddilog.File	<i>ICAN-root/esrs.log</i>
log4j.appender.juddilog.MaxFileSize	10MB
log4j.appender.juddilog.MaxBackupIndex	3
log4j.appender.juddilog.layout	org.apache.log4j.PatternLayout
log4j.appender.juddilog.layout.ConversionPattern	%d{ISO8601} %-5p [%c] %m%n
log4j.rootLogger	DEBUG,File,Console

13.3.3 Enterprise Designer

- Master log file:

ICAN-root/edesigner/usrdir/system/ide.log

- Master log configuration file:

ICAN-root/edesigner/bin/log4j.properties

Table 61 Properties for the Enterprise Designer Master Log

Property	Default Values
log4j.appender.R	org.apache.log4j.RollingFileAppender
log4j.appender.R.File	<i>ICAN-root/usrdir/system/ide.log</i>
log4j.appender.R.MaxFileSize	1000KB
log4j.appender.R.MaxBackupIndex	100
log4j.appender.R.layout	org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern	ICAN5.[%d{DATE}] %p (%F:%L) - %m%n
log4j.rootLogger	DEBUG, R, stdout

13.3.4 Enterprise Monitor

- Enterprise monitor log:

ICAN-root/monitor/logs/monitor.log

- Log configuration file:

ICAN-root/monitor/config/log4j.properties

Table 62 Properties for Enterprise Monitor Log

Property	Default Values
log4j.appender.R	org.apache.log4j.RollingFileAppender
log4j.appender.R.File	<i>ICAN-root/monitor/logs/monitor.log</i>
log4j.appender.R.MaxFileSize	1000KB
log4j.appender.R.MaxBackupIndex	100
log4j.appender.R.layout	org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern	%d %5p [%t] %C (%F:%L) - %m%n
log4j.rootLogger	INFO, R, stdout

13.4 Run-Time Log Files and Locations

Run-time log files, and the directory structure in which they reside, are generated when you bootstrap the Logical Host following deployment.

13.4.1 Logical Host

- Master log file:

`ICAN-root/logicalhost/logs/stc_lh.log`

- Master log configuration file:

`ICAN-root/logicalhost/logconfigs/LH/log4j.properties`

Table 63 Properties for the Logical Host Log

Property	Default Values
log4j.appender.FILE	org.apache.log4j.RollingFileAppender
log4j.appender.FILE.File	<code>ICAN-root/logicalhost/logs/stc_lh.log</code>
log4j.appender.FILE.MaxFileSize	10MB
log4j.appender.FILE.MaxBackupIndex	10
log4j.appender.FILE.layout	org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.ConversionPattern	<code>%d{ISO8601} %-5p [%t] [%c] [%x] %m%n</code>
log4j.rootCategory	INFO, FILE

Note: *If it becomes necessary to increase space for Logical Host log files, you must shut down the Logical Host, change the **MaxFileSize** and/or **MaxBackupIndex** properties, and re-bootstrap the Logical Host.*

- Monitor Interface log file

`ICAN-root/logicalhost/logs/stc_ms_stcsysjms.log`

13.4.2 Integration Servers

- Log file:

ICAN-root/logicalhost/logs/stc_is_int-server-name.log

- Log configuration file:

ICAN-root/logicalhost/logconfigs/int-server-name/log4j.properties

Table 64 Properties for the Integration Server Logs

Property	Default Values
log4j.appender.FILE	org.apache.log4j.RollingFileAppender
log4j.appender.FILE.File	<i>ICAN-root/logicalhost/logs/stc_is_int-server-name.log</i>
log4j.appender.FILE.MaxFileSize	10MB
log4j.appender.FILE.MaxBackupIndex	10
log4j.appender.FILE.layout	org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.ConversionPattern	%d{ISO8601} %-5p [%t] [%c] [%x] %m%n
log4j.rootCategory	INFO, FILE

13.4.3 Message Servers

- Log file:

ICAN-root/logicalhost/logs/stc_ms_msg-server-name.log

- Log configuration file:

ICAN-root/logicalhost/stcms/msg-server-name/stcms.default.properties

Table 65 Properties for Integration Server Logs

Property	Default Values
log4j.appender.FILE	org.apache.log4j.RollingFileAppender
log4j.appender.FILE.File	<i>ICAN-root/logicalhost/logs/stc_ms_msg-server-name.log</i>
log4j.appender.FILE.MaxFileSize	10MB
log4j.appender.FILE.MaxBackupIndex	10
log4j.appender.FILE.layout	org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.ConversionPattern	%d{ISO8601} %-5p [%t] [%c] [%x] %m%n
log4j.rootCategory	INFO, FILE

OTD Interfaces

A.1 The *Bean-like* Interface

At run time, an OTD instance is accessed directly from Java in a Java Collaboration, using the so-called *bean-like* accessors. The nodes comprising the hierarchy of the data structure have an interface similar to Java beans: each node has a set of properties with *get* and *set* methods to manipulate them.

This data structure is represented at run time by a set of generated Java classes, which follow the Java bean rule. They have a set of zero or more properties, each of which has a specific type and a given name, and may be optional and/or repeating. In contrast to regular bean properties, a OTD node property has two distinct names: a display name, that can be a virtually-arbitrary string, and a Java name that is the accessor basename.

For example, if a node has a property with Java name "X", then the implementing class for that node will have a method "getX". The Java name is normally derived from the display name, modified to suit the restrictions on Java identifiers, and supplied automatically by eGate.

Important: Do not modify this Java name property.

A.2 The *Generic* Interface

The **OtdNode** interface describes the generic access provided by any non-leaf, implemented by a generated class for a single OTD node. The OTD metadata is provided separately by an object that implements the **OtdMeta** interface.

- The *meta ()* method gets the run-time metadata for this node in the OTD's grammar. (This is really a static method by nature, because the grammar for the OTD is not tied to any particular instance of it, but Java interfaces do not allow static methods in interface definitions, and this gives more freedom of implementation.)
- The *has (child)* method tests whether given child is present. For repeated children, it tests against the collection as a whole.
- The *has (child, index)* method tests whether given instance of a repeated child is present. This method may not be called for a non-repeating child node, and is only useful if the pertinent OTD builder supports transient “holes” in a repeating child's set of instances.
- The *size (child)* method returns one past the last valid index for a repeating child. For a non-repeating node, this should return 1. If the OTD does not permit holes in repetition, then this value is the same as returned by the *count()* method.
- The *count (child)* method returns the number of repetition instances of the given child. For a non-repeating node, this should return 1.
- For a **choice** node, the *choice* method returns the index of the valid child. A choice node has a number of children, only one of which can occur below any given parent instance. If there is no current valid child, it returns -1.
- The *get (child)* method retrieves the given child instance. This method may only be called for a non-repeating child node.
- The *get (child, index)* method retrieves the given child instance. This method may not be called for a non-repeating child node.
- The *set (child, value)* method stores the given child instance. This method may only be called for a non-repeating child node.
- The *set (child, index, value)* method stores the given child instance. This method may not be called for a non-repeating child node.
- The *remove (child)* method removes the given child instance. The child must be optional. The implementation is allowed (but not required) to throw an exception if the child is not optional.
- The *remove (child, index)* method removes the given occurrence of a child instance. The child must be repeated. The implementation is allowed (but not required) to throw an exception if the child is absent. A subsequent *has (child, index)* call with the same arguments should return false. If the implementation does not support holes, an attempt to remove the non-ultimate occurrence may throw an exception or replace the value with the same default value used by *set (child, index)* to fill unassigned occurrences.


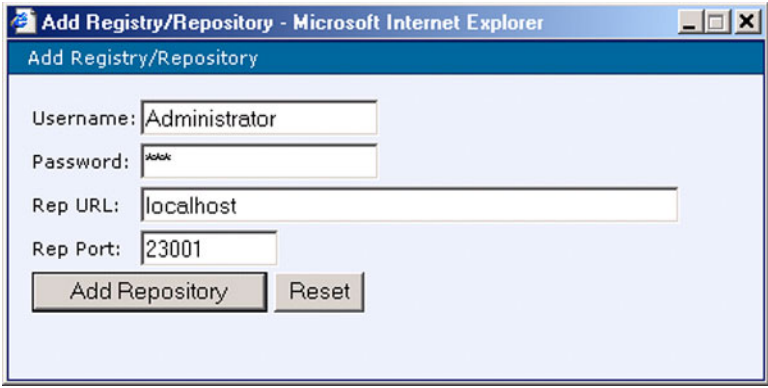

The Schema Runtime Environment

eGate 5.0 provides a completely different operating environment from earlier versions of the product (e*Gate). The Schema Runtime Environment (SRE) allows you to use schemas developed for e*Gate 4.x with eGate 5.0 by providing the necessary environmental components, plus the SRE Monitor.

B.1 The SRE Monitor

The SRE Monitor for the Schema Runtime Environment enables you to manage an e*Gate 4.x system from within eGate 5.0. The SRE Monitor interface generally resembles the Enterprise Monitor, but differs somewhat in detail. Only the Environment Explorer is displayed, which has two additional icons in the upper left corner; these are described in Table 66.

Table 66 SRE Monitor Explorer Icons

Icon	Function
	<p>The Add Registry/Repository icon displays the Add Registry/Repository dialog box, in which you specify the desired Registry's name and port, and your user name and password.</p> 
	<p>The Refresh Repository icon refreshes the SRE Registry and the Explorer tree, following changes to component status.</p>

To run the SRE

To run an e*Gate 4.x schema in eGate 5.0, you must start the following components:

- 1 SRE Repository
- 2 SRE Lightweight Installer
- 3 JIntegra services
- 4 Control Broker

The Control Broker is started by running the following command from the root directory for your e*Gate 4.5.x installation:

```
stccb -rh hostname -rs schemaname -un username -up password -ln  
      logical-host-name
```

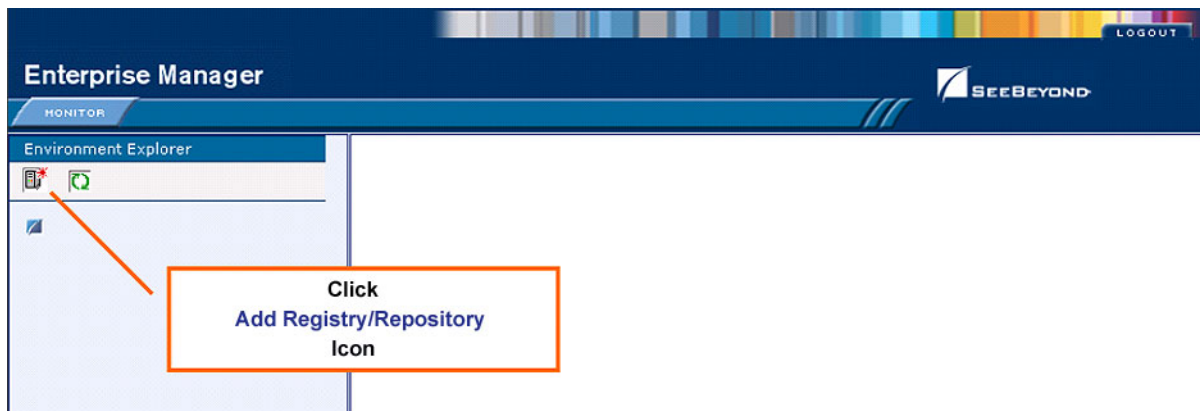
See the *SeeBeyond ICAN Suite Installation Guide* for additional information.

To start the SRE Monitor

- 1 Launch Internet Explorer.
- 2 Enter one of the following commands:
 - A Enter **http://hostname:portnumber/SREMonitor** in the **Address** box to display the SRE Monitor page as shown in Figure 295.
 - B Enter **http://hostname:portnumber/SREMonitor/remote.do** in the **Address** box to display the SRE Monitor page without the SeeBeyond banner.
 - C Enter **http://hostname:portnumber/treemanager** in the **Address** box to display only the Environment Explorer, containing the Environment tree.

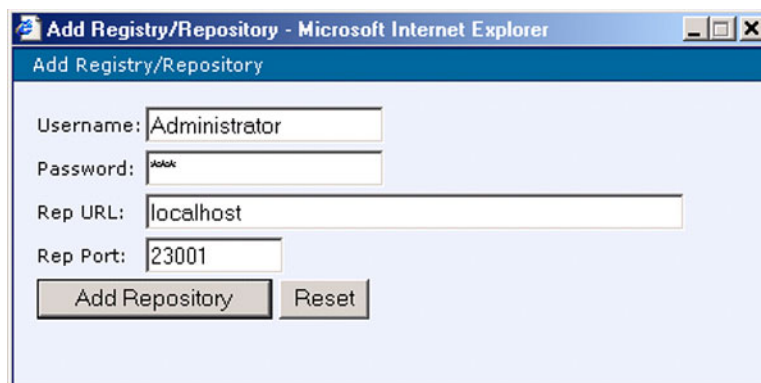
Note: The *hostname* is the name of the host computer on which you installed the SRE Monitor. The *portnumber* is the port number you used when you installed the SRE Monitor.

Figure 295 SRE Monitor Initial Page



- 3 Click the **Add Registry/Repository** icon in the upper left corner of the Explorer. This displays the dialog box shown in Figure 296.

Figure 296 Add Registry/Repository Dialog Box



- 4 Enter your login ID and password, and the Repository URL and Port, and click **Add Repository**.

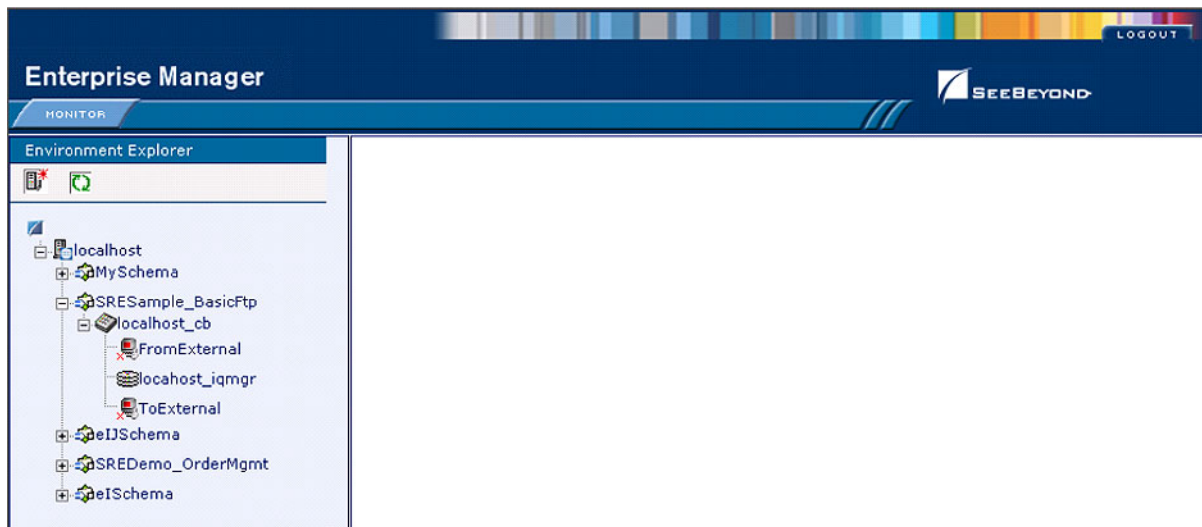
Note: The **Repository URL** is the TCP/IP host name of the server where you installed the Repository, and the **portnumber** is the number of the port you entered during installation of the Repository. See the SeeBeyond ICAN Suite Installation Guide.

Important: The TCP/IP host name must be composed of alphanumeric characters.

To use the SRE Monitor

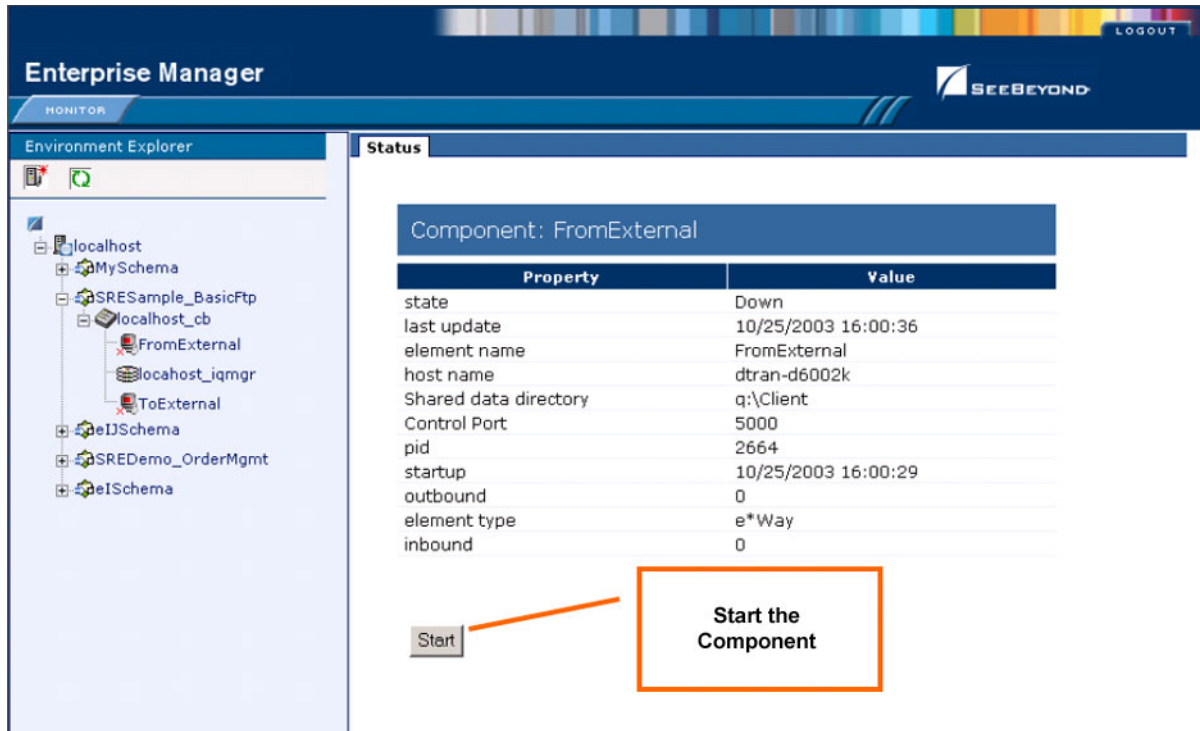
- 1 After adding the Registry, expand the Control Broker in the Explorer tree to view the SRE components, as shown in Figure 297.

Figure 297 Viewing SRE Components



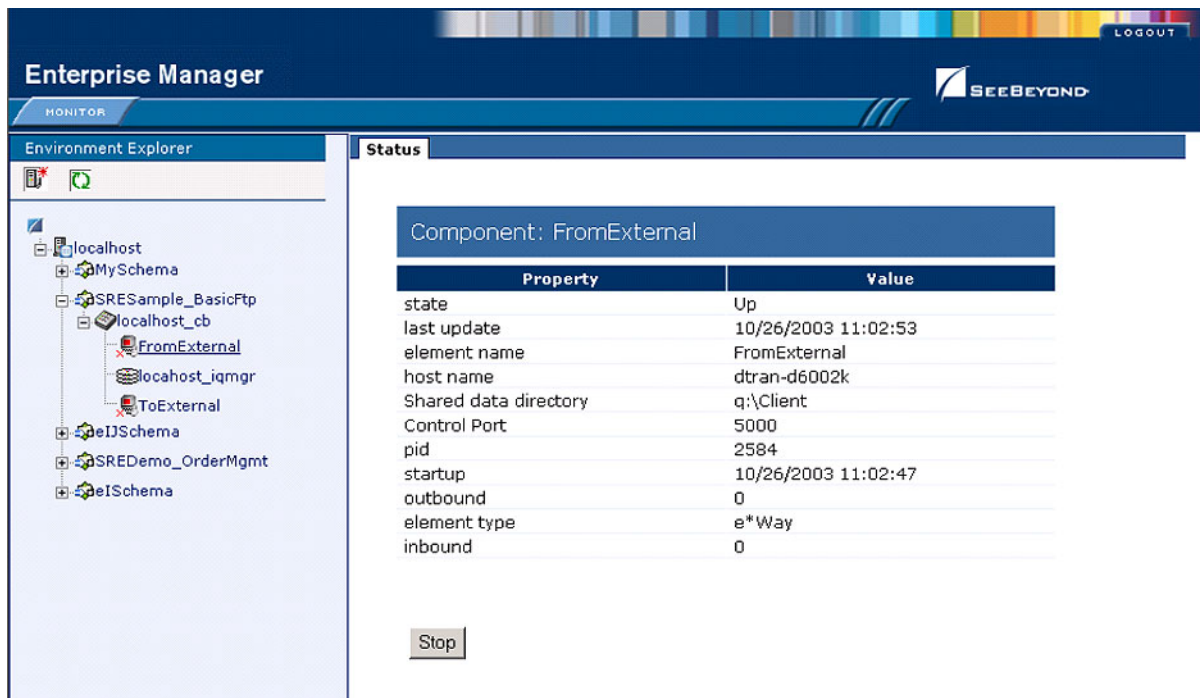
- 2 Click on a component to display its status. As an example, the status of the *FromExternal* e*Way is shown in Figure 298. You can start the component by clicking the **Start** button, which becomes a **Stop** button when the component is running.

Figure 298 FromExternal e*Way Status (Not Running)



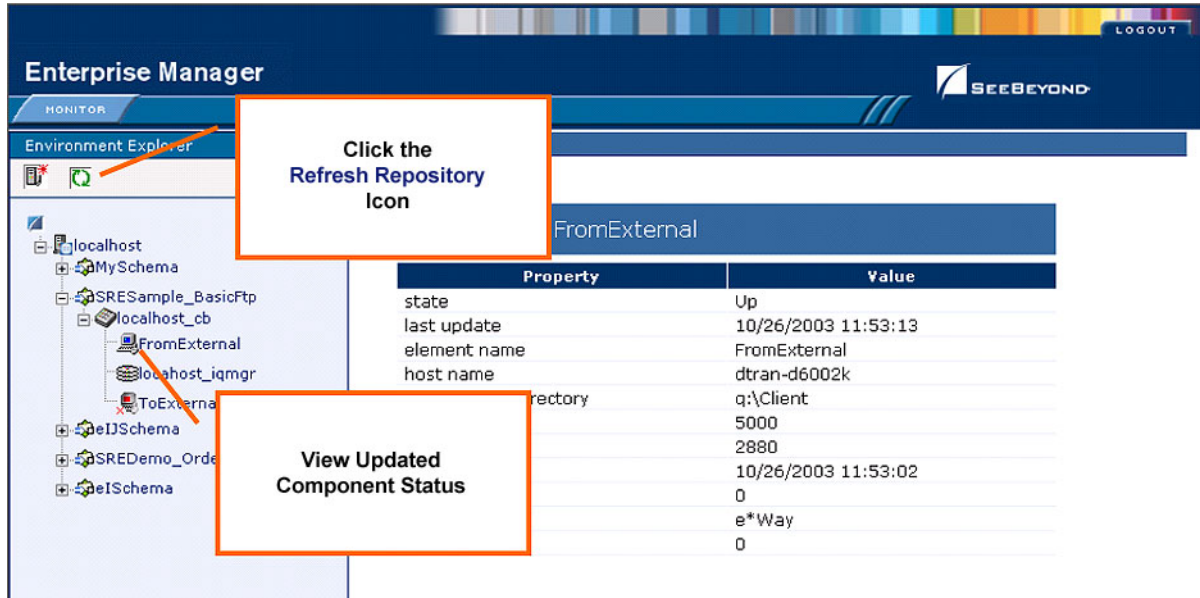
- 3 The status page is updated after you start or stop the component, as shown in Figure 299.

Figure 299 FromExternal e*Way Status (Running)



- 4 To refresh the Explorer tree, click the **Refresh Repository** icon, as shown in Figure 300.

Figure 300 Refreshing the Explorer Tree



B.0.1 Security

Security for the SRE Monitor mimics the monitor in e*Gate 4.5—you must log into each registry instead of authenticating at run-time. See the e*Gate 4.5 documentation.

Glossary

BI

Business integration (also Business Intelligence).

Collaboration

See “**Service**” and “**Collaboration Definition**”.

Collaboration Definition

The encoding of business rules, in Java or XSLT format. Typically, the encoding consists of operations on OTDs (see “**OTD**” on page 350). Several Collaborations can have the same Collaboration Definition.

Connection

Consists of the configuration information that enables an eWay to connect to an external system.

Connectivity Map

Contains business logic and routing information about the data transmission. A Connectivity Map usually includes one or more Collaborations, Passthrough Collaborations, topics, queues, and eWays. A Connectivity Map is created under a Project. A Project may have multiple Connectivity Maps.

Constants

A name or value pair that is visible across a Project.

CRM

Customer Relations Management

Data Cleansing

Data must be cleansed of errors in structure and content before it is useful in data warehousing and integration; this means transforming data for accurate and effective use in a database or data management system by cleansing “dirty” or redundant data.

Data Dictionary

Defines the organization of a database and lists all files in the database, the number of records in each file, and the names and types of each field. The data dictionary is often hidden from end users. Although the dictionary doesn’t contain actual data, it does contain essential information for managing the database.

Data Integrity

Refers to the accuracy and validity of data. Data integrity can be compromised in many ways, including human error through data entry, or through faulty logic in

programming. Computer viruses, software bugs and many other factors can also compromise data integrity.

Data Mapping

In relational databases (RDBMSs) data mapping is the relationship and data flow between source and target objects. Mapping involves structuring the relationship between source and target objects.

Data Mart

A smaller, focused, database designed to help managers make business decisions. (A data warehouse is a larger, enterprise, database(s).)

Data Mining

Used to synthesize or isolate unique data patterns to predict future behaviors or to filter data to select patterns that help discover previously unknown relationships among data. Commonly used by marketers who acquire and distill consumer information.

Data Transformation

Data transformation is necessary after extracting data from legacy data formats, or any format that requires cleansing. Data is transformed for efficient use for Business-to-Business Enterprise Data Integration.

Data Warehouse

A copy or view of enterprise transaction data (sometimes non-transaction data) that is used for reporting. The data is often summarized and always structured for queries and analysis.

Deployment Profile

Contains the information about how the Project components will be deployed in an Environment. A Project can have multiple Deployment Profiles, but only one Deployment Profile can be activated for a Project in any one Environment.

Derived Collaboration

Collaboration that inherits operations from another, according to standard object-oriented practice.

Dimension Table

Dimension tables describe the business entities of an enterprise; also called lookup or reference tables.

Dirty Data

Dirty data contains, but is not limited to, incorrect data including spelling errors, punctuation errors, incorrect data referencing, incomplete, inconsistent, outdated, and redundant data.

Drill Down

To move from summary to more detailed data by “drilling down” to get it. In database terminology this might mean starting with a general category and drilling down to a specific field in a record.

eGate System

See “Project”.

Environment

A collection of physical resources and their configurations that are used to host Project components. An Environment contains logical hosts and external systems.

EPR

Enterprise Resource Management

ETL

Extract, Transform, Load. Extract is the process of reading data from a source database and extracting the desired subset of data. Transform is the process of converting the extracted data from its previous form into the desired form. Load is the process of writing the data into a larger database.

eWay

A link between a Collaboration and an external connection including the message server connection (topic or queue) or external application.

External Application

A logical representation in an eGate Project of an external application.

External System

A representation in an eGate Project of an external application system.

Extraction

Data are extracted from a source using software tools. This first step in ETL initially “gets” the data.

Fact Table

A fact table typically contains two types of columns: those containing facts and those that contain foreign keys to dimension tables. Fact tables contain detail facts and/or summary facts.

ICAN Suite

The SeeBeyond Integrated Composite Application Network Suite.

Integration Server

J2EE software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays). Provides transaction services, persistence, and external connectivity.

JMS IQ Manager

JMS-compliant, guaranteed delivery store, forwarding, and queueing service.

Join

Matches records, which are joined by a common field, in two tables in a relational database. Often part of a Select query.

Link

The JMS Connection between a Collaboration and a topic or queue in a JMS-compliant message server.

Linked Message Destination

A reference to a Message Destination defined in another Connectivity Map.

Logical Host

An instance of the eGate runtime Environment that is installed on a machine. A Logical Host contains the software and other installed components that are required at runtime, such as application and message servers.

Management Agent

Uses J2EE technology to manage and monitor an eGate 5.0 deployment that may contain other application servers in addition to the SeeBeyond Integration Server. Defines management interfaces and services designed for distributed environments, focusing on providing functionality for managing networks, systems, and applications.

Message Destination

A general term for a topic or queue. Two or more Projects can share a message destination that has the same name and is deployed on the same message server. A single Project may also have a single message destination referenced in multiple Connectivity Maps.

Metadata

“Data about data.” Metadata describes “how,” “when,” and “who” about structure and format, of a particular set of data. ETL tools are used to generate and maintain a central metadata repository.

Non-normalized Data

Non-normalized data cannot be cross-referenced accurately, if at all, and causes manageability issues. Non-normalized data may be converted to normalized data.

Normalized Data

Normalization is a common database design process used to remove redundant or incorrect organization and data. The design and normalization of the database will create a maintainable data set that can be cross-referenced.

Normalized data is not only easier to analyze but also easier to expand. Normalization involves removing redundancy and correcting incorrect data structure and organization.

OLAP

Online analytical processing.

OTD

An acronym for Object Type Definition. OTDs contain the data structure and rules that define an object. An OTD is used in Java Collaboration Definitions for creating data transformations and interfacing with external systems.

Project

Contains a collection of logical components, configurations, and files that are used to solve business problems. A Project organizes the files and packages and maintains the settings that comprise an eGate system in SeeBeyond's Enterprise Designer.

Query

A request for information from a database. There are three query methods:

Choose – With this easy-to-use method, the database system presents a list of parameters from which you can choose. This method is not as flexible as other methods.

Query by example (QBE) – With this method, the system lets you specify fields and values to define a query.

Query language – With this method, you have the flexibility and power to make requests for information in the form of a stylized query using a query language. This is the most complex and powerful method.

Queue

A JMS queue is a shareable object that conforms to the *point-to-point* (p2p, or PTP) messaging domain, where one sender delivers a message to exactly one receiver. When the SeeBeyond JMS IQ Manager sends a message to a queue, it ensures it is received once and only once, even though there may be many receivers "listening" to the queue. This is equivalent to the subscriber pooling in other queue implementations. You can reference a queue that exists in another Connectivity Map or Project.

Raw Data

Data that has not been turned into "information," through processing. Although factual and "real," raw data is unorganized.

Relational Database (RDBMS)

Short for Relational Database Management System, most often referred to as RDBMS. Data is stored in related tables. Relational databases can be viewed in many different ways.

In this system a single database can be spread across several tables. (RDBMS differs from flat-file databases where each database is self-contained as a single file or table.)

Repository

Stores and manages the setup, component, and configuration information for eGate Projects. The Repository also provides monitoring services for Projects, which include version control and impact analysis.

Schema Runtime Environment

An add-on in eGate 5.0 that provides the upgrade path for e*Gate 4.x users to upgrade to eGate 5.0. Also known as the SRE.

Service

Contains the information about executing a set of business rules. These business rules can be defined in a Java Collaboration Definition, XSLT Collaboration Definition, Business Process, eTL Definition, or other service. A Service also contains binding information for connecting to JMS Topics, Queues, eWays, and other services.

Staging Data

Data that is to be processed before entering the warehouse.

Subproject

An independent Project that is included as part of another Project and listed on the Enterprise Explorer tree beneath the main Project icon.

Table

Refers to data arranged in rows and columns, like a spreadsheet. In relational database management systems, all information is stored in tables.

Topic

A JMS topic is a shareable object that conforms to the *publish-and-subscribe* (pub/sub) messaging domain, where one publisher broadcasts messages to potentially many subscribers. When the SeeBeyond JMS IQ Manager publishes a message on a topic, it ensures that all subscribers receive the message.

Transformation

Data that are extracted from databases are transformed into a desired form, using various tools that cleanse, merge, purge, aggregate, calculate, audit, remove redundancy, standardize, etc.

XSLT

An acronym for Extensible Stylesheet Language Transformations. A file format used in eGate to generate Collaboration Definitions.

Warehouse

See “Data Warehouse”.

e*Gate 4.x Terms in eGate 5.0

Table 67 provides definitions for the terms that are new with eGate release 5.0, as well as equivalent terms from eGate release 4.x.

Table 67 eGate 5.0 Terms

5.0 Term	4.x Equivalent Term
Collaboration	Collaboration
Collaboration Definition	Collaboration Definition
Connection	e*Way Connection
Connectivity Map	Closest: Network View of an entire Schema
Deploy	Run the Control Broker
Deployment	<none>
Deployment Profile	Closest: Schema
Enterprise Designer	Enterprise Manager
Enterprise Manager	Enterprise Monitor
Environment	Schema (except only includes physical information, not business logic)
eWay	e*Way Connection e*Way
eWay Configuration	e*Way Connection Configuration
External Application	e*Way Connection
External System	e*Way Connection
JMS Connection	e*Way Connection
Integration Server	<none>
Link	JMS e*Way Connection
Linked Message Destination	<none>
Logical Host	Participating Host
Message Destination	Topic or queue
Message Server	JMS IQ Manager
Object Type Definition (OTD)	Event Type Definition (ETD)
Process Manager	Control Broker
Project	Schema (except not including physical layer)
Queue	JMS queue
Repository	Registry
Subproject	Schema

Table 67 eGate 5.0 Terms (Continued)

5.0 Term	4.x Equivalent Term
Topic	JMS topic
XSLT	<none>

Index

A

ACL properties 86, 262–263
 Activate button 279
 activating
 Deployment Profile 283
 addition method 184, 242
 Advanced Mode command 169
 alerts
 viewing 45
 AND method 181, 242
 arrayAccess method 192
 arrayAssign method 193
 arrayLength method 193
 Auto Layout command 178
 Automerger 197, 256

B

BEA WebLogic 286
 BI 347
 bitNot method 194
 boolean method 250
 Break command 172
 business rules
 tree 173
 Business Rules on Left command 169
 Business Rules on Top command 169
 buttons
 Activate 279
 Deactivate 279
 Map Variables 279

C

Call Java Method command 176
 Call New Constructor command 177
 cast method 186
 ceiling method 248
 charAt method 189
 Collaboration 347, 353
 derived 348
 Collaboration definition 347, 353
 Java 159
 XSLT 233

Collaborations
 monitoring 39
 Collapse All Methods command 178
 Commit Changes command 169
 Commit Code Changes command 240
 concat method 189
 concatenate method 245
 connection 347, 353
 Connectivity Map 347, 353
 Editor 88
 constants 347
 Environmental 264
 contains method 250
 Continue command 172
 Control Broker 353
 conventions, writing in document 23
 count method 252
 Create Literal command 176
 CRM 347
 current method 252
 customizer 96

D

data cleansing 347
 data dictionary 347
 data integrity 347
 data mapping 348
 data mart 348
 data mining 348
 data transformation 348
 Data Type Definition (DTD) 124
 data warehouse 348, 352
 Deactivate button 279
 deactivating
 Deployment Profile 283
 deploy 353
 Deployment 353
 Deployment EditorEditors
 Deployment 279
 Deployment Profile 278, 348, 353
 activating 283
 creating 280
 deactivating 283
 map variables 285
 derived Collaboration 348
 Diff file
 create 196, 255
 generate 196, 255
 merge 197, 256
 dimension table 348
 dirty data 348
 division method 184, 242
 document method 252

drill down 348
 DTD Wizard 124

E

e*Way 353
 e*Way Connection 353
 e*Way Connection Configuration 353
 Editor
 Connectivity Map 88
 Java Collaboration 168
 OTD 152
 XSLT Collaboration 239
 eGate system 349
 element-available method 250
 endsWith method 189
 enter file name to export 76
 Enterprise Designer 353
 enterprise explorer 60
 menu bar 56
 starting 55
 Enterprise Explorer
 Environments 61, 257
 Projects 60, 82
 Enterprise Manager 353
 Documentation 36
 Interface 35
 starting 34
 Enterprise Monitor 37, 353
 Environment 29, 349, 353
 constants 264
 Environment Explorer 61, 257
 EPR 349
 equal method 182, 243
 equals method 186
 ETD 353
 ETL 349
 Event Type Definition 353
 eWay 349, 353
 eWay Configuration 353
 Expand All Methods command 178
 Export Java Rule command 169
 external
 application 349, 353
 system 349, 353
 external system 29
 extraction 349

F

fact table 349
 false method 251
 Field command 172
 Find command 170

Find Next command 170
 Find Previous command 170
 floor method 248
 For command 172
 format_number method 245
 function_available method 251

G

generate-id method 253
 greater_or_equal method 182, 243
 greater_than method 182
 greater_then method 243

I

IBM WebSphere 289
 ICAN Suite 349
 Icons
 JCE Business Rules Designer toolbar 175
 JCE Business Rules toolbar 172
 JCE toolbar 169
 XCE toolbar 240
 id method 253
 If-Then command 172
 impact analyzer
 overview 69
 Import a Local File command 169
 Import JAR File command 170
 Import Static Field command 175
 Import XSL from a Local File command 240
 indexOf method 190
 indoubt transaction 51
 Indoubt transaction (XA) 51
 instanceof method 186
 Integration Server 349, 353
 Interfaces
 Enterprise Manager 35

J

Java Collaboration
 definitions 159
 Editor (JCE) 168
 Wizard 160
 Java method palette 179
 Java methods
 addition 184
 AND 181
 arrayAccess 192
 arrayAssign 193
 arrayLength 193
 bitNot 194

- cast 186
 - charAt 189
 - concat 189
 - division 184
 - endsWith 189
 - equal 182
 - equals 186
 - greater_or_equal 182
 - greater_than 182
 - indexOf 190
 - instanceof 186
 - length 190
 - lesser_or_equal 183
 - lesser_than 183
 - multiplication 184
 - negative 194
 - NOT 181
 - not_equal 183
 - OR 181
 - positive 194
 - postDecrement 194
 - postIncrement 195
 - preDecrement 195
 - preIncrement 195
 - remainder 185
 - replace 190
 - substring 190
 - subtraction 185
 - toLowerCase 190
 - toString 186
 - toUpperCase 190
 - trim 191
 - JCE commands
 - Advanced Mode 169
 - Auto Layout 178
 - Break 172
 - Business Rules on Left 169
 - Business Rules on Top 169
 - Call Java Method 176
 - Call New Constructor 177
 - Collapse All Methods 178
 - Commit Changes 169
 - Continue 172
 - Create Literal 176
 - Expand All Methods 178
 - Export Java Rule 169
 - Field 172
 - Find 170
 - Find Next 170
 - Find Previous 170
 - For 172
 - If-Then 172
 - Import a Local File 169
 - Import JAR File 170
 - Import Static Field 175
 - Local Variable 172
 - Method 172
 - New Array 177
 - Redo 170
 - Refresh Collaboration 170
 - Replace 170
 - Return 172
 - Roll Back Changes 169
 - Rule 172
 - Source Code Mode 169
 - Standard Mode 169
 - Throw 172
 - Try 172
 - Undo 170
 - Validate 169
 - While 172
 - JMS
 - connection 353
 - e*Way Connection 353
 - IQ Manager 353
 - queue 353
 - topic 354
 - JMS IQ Manager 349
 - join 349
- K**
- key method 253
- L**
- language method 251
 - last method 253
 - LDAP
 - Configuration User Management 309
 - Environment User Management 320
 - Overview 307
 - length method 190
 - lesser_or_equal method 183, 243
 - lesser_than method 183, 243
 - link 350, 353
 - linked message destination 350, 353
 - Local Variable command 172
 - local-name method 253
 - Logical Host 29, 350, 353
 - starting 275
 - Starting as a Windows Service 272
 - Starting Manually on Windows 276
 - Starting on Linux 276
 - Starting on UNIX 276
 - Startup Configuration File 269
 - stopping 277
 - logs

- setting levels 50
- viewing 48

M

- Management Agent 350
- map variables 285
- Map Variables button 279
- menu bar 56
- message
 - destination 350, 353
 - server 353
- metadata 350
- Method command 172
- method palette
 - Java 179
 - XSLT 241
- mod method 243
- Monitor
 - Enterprise 37
 - SRE 341
- monitoring Collaborations 39
- multiplication method 184, 244

N

- name method 253
- namespace-uri method 253
- negative method 194, 244
- network view 353
- New Array command 177
- non-normalized data 350
- normalize_space method 245
- normalized data 350
- NOT method 181, 251
- not_equal method 183, 244
- number method 248
- number-literal method 249

O

- Object Type Definition 350, 353
 - wizard 125, 130, 134
- Object Type Definition (OTD) 120
- OLAP 350
- Open File command 153
- OR method 181, 244
- OTD 350, 353
 - Editor 152
 - tester 155
- OTD Editor commands
 - Open File 153
 - Refresh OTD 153

- Run Tester 153
- Save as New Name 153
- Save File 153
- Sort by Name 153
- Tester 153
- Toggle Reference Tab Panel 153
- OTD Wizard
 - DTD 124
 - User-Defined 137
 - WSDL 128
 - XSD 133

P

- Participating Host 353
- position method 254
- positive method 194
- postDecrement method 194
- postIncrement method 195
- preDecrement method 195
- preIncrement method 195
- Process Manager 353
- Profile, Deployment 278
- Project 351, 353
- Project Explorer 60, 82

Q

- query 351
- queue 351, 353

R

- raw data 351
- rdbm 351
- Redo command 170
- Refresh Collaboration command 170
- Refresh OTD command 153
- Registry 353
- relational database 351
- remainder method 185
- Replace command 170
- replace method 190
- Repository 29, 351, 353
- Return command 172
- roles 294
- Roll Back Changes command 169
- Roll Back Code Changes command 240
- round method 249
- Rule command 172
- Run Tester command 153

S

- Save as New Name command 153
- Save File command 153
- Save XSL to a Local File command 240
- Scheduler 93
- Schema 353
- Schema Runtime Environment 351
- Schema Runtime Environment (SRE) 341
- security 268
- Security Server 352
- select file to import 78
- setting log levels 50
- Show Mapping Only command 240
- Show Maps and Code command 240
- Show XSLT Code Only command 240
- Sort by Name command 153
- Source Code Mode command 169
- SRE 351
 - Monitor 341
- staging data 352
- Standard Mode command 169
- starting
 - Enterprise Designer 55
 - Enterprise Manager 34
- starts_with method 251
- string method 246
- string_length method 246
- string_literal method 246
- subproject 352–353
- substring method 190, 246
- substring-after method 246
- substring-before method 246
- subtraction method 185, 244
- sum method 249
- supporting documents 24
- system-property method 247

T

- table 352
- tester
 - OTD 155
- Tester command 153
- Throw command 172
- Toggle Reference Tab Panel command 153
- toLowerCase method 190
- topic 352–354
- toString method 186
- toUpperCase method 190
- transformation 352
- translate method 247
- trim method 191
- true method 251

- Try command 172

U

- Undo command 170
- unparsed-entity-uri method 247
- User-Defined OTD Wizard 137

V

- Validate command 169
- variables
 - mapping 285
- Version Control
 - Create Diff 196, 255
 - Generate Diff 196, 255
 - Merge Diff 197, 256
- version control 71
- viewing
 - alerts 45
 - logs 48

W

- warehouse 352
- WebLogic 286
- WebSphere 289
- While command 172
- Wizard
 - Java Collaboration 160
 - OTD (DTD) 124
 - OTD (User-Defined) 137
 - OTD (WSDL) 128
 - OTD (XSD) 133
 - XSLT Collaboration 234
- WSDL Wizard 128

X

- XA transaction
 - Indoubt 51
- XA transactions 51
- XCE commands
 - Commit Code Changes 240
 - Import XSL from a Local File 240
 - Roll Back Code Changes 240
 - Save XSL to a Local File 240
 - Show Mapping Only 240
 - Show Maps and Code 240
 - Show XSLT Code Only 240
- XSD Wizard 133
- XSLT 352, 354
 - method palette 241

- XSLT Collaboration
 - definitions 233
 - Editor 239
 - Wizard 234
- XSLT Collaboration Editor
 - method palette 241
- XSLT methods
 - addition 242
 - AND 242
 - boolean 250
 - ceiling 248
 - concatenate 245
 - contains 250
 - count 252
 - current 252
 - division 242
 - document 252
 - element-available 250
 - equal 243
 - false 251
 - floor 248
 - format_number 245
 - function_available 251
 - generate-id 253
 - greater_or_equal 243
 - greater_than 243
 - id 253
 - key 253
 - language 251
 - last 253
 - lesser_or_equal 243
 - lesser_than 243
 - local-name 253
 - mod 243
 - multiplication 244
 - name 253
 - namespace-uri 253
 - negative 244
 - normalize_space 245
 - NOT 251
 - not_equal 244
 - number 248
 - number-literal 249
 - OR 244
 - position 254
 - round 249
 - starts_with 251
 - string 246
 - string_length 246
 - string_literal 246
 - substring 246
 - substring-after 246
 - substring-before 246
 - subtraction 244
 - sum 249
 - system-property 247
 - translate 247
 - true 251
 - unparsed-entity-uri 247