

iWay

iWay Application Systems Adapter for Amdocs
ClarifyCRM for BEA WebLogic User's Guide
Version 5 Release 5

EDA, EDA/SQL, FIDEL, FOCCALC, FOCUS, FOCUS Fusion, FOCUS Vision, Hospital-Trac, Information Builders, the Information Builders logo, Parlay, PC/FOCUS, SmartMart, SmartMode, SNAPpack, TableTalk, WALDO, Web390, WebFOCUS and WorldMART are registered trademarks, and iWay and iWay Software are trademarks of Information Builders, Inc.

Due to the nature of this material, this document refers to numerous hardware and software products by their trademarks. In most, if not all cases, these designations are claimed as trademarks or registered trademarks by their respective companies. It is not this publisher's intent to use any of these names generically. The reader is therefore cautioned to investigate all claimed trademark rights before using any of these names other than to refer to the product described.

Copyright © 2005, by Information Builders, Inc. and iWay Software. All rights reserved. Patent Pending. This manual, or parts thereof, may not be reproduced in any form without the written permission of Information Builders, Inc.

Preface

This document is written for system integrators with programming backgrounds and an understanding of ClarifyCRM. Extensive knowledge of ClarifyCRM is not required, but may be helpful in learning about the adapter.

This document provides details on working with the iWay Applications System Adapter for Amdocs ClarifyCRM to develop online interconnections to ClarifyCRM.

How This Manual Is Organized

The following table provides the title and a brief description of each chapter and appendix found in this manual.

Chapter/Appendix		Contents
1	Introducing the iWay Application Systems Adapter for Amdocs ClarifyCRM	This section introduces the iWay Applications System Adapter for Amdocs ClarifyCRM and provides information to help you accomplish your integration projects.
2	Creating XML Schemas and Web Services With ClarifyCRM Business Objects	This section describes how to use the iWay Application Explorer to create XML schemas and Web services for ClarifyCRM using business objects.
3	Creating XML Schemas and Web Services With ClarifyCRM ClearBasic	This section describes how to use the iWay Application Explorer to create XML schemas and Web services for ClarifyCRM using ClarifyCRM ClearBasic.
4	Listening for Events Using ClarifyCRM Business Objects	This section describes how to use iWay Application Explorer to configure the adapter to listen for events in ClarifyCRM tables using ClarifyCRM Business Objects.
5	Listening for Events Using ClarifyCRM ClearBasic	This section describes how to use iWay Application Explorer to configure the adapter to listen for events in ClarifyCRM tables using ClearBasic.
6	Using Web Services Policy-Based Security	Describes how Web services policy-based security works and how to configure it.

Chapter/Appendix		Contents
7	Management and Monitoring	Describes the management and monitoring tools provided by iBSE and JCA.
A	Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services With CBO	Describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas and business services (Web services) for ClarifyCRM using ClarifyCRM Business Objects (CBOs).
B	Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services with ClarifyCRM ClearBasic	Describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas and business services (Web services) for ClarifyCRM using ClarifyCRM ClearBasic.
C	Using Application Explorer in BEA WebLogic Workshop for Event Handling with CBOs	Describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop for event handling for ClarifyCRM using ClarifyCRM Business Objects (CBOs).
D	Using Application Explorer in BEA WebLogic Workshop for Event Handling with ClarifyCRM ClearBasic	Describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop for event handling using ClarifyCRM ClearBasic.

Documentation Conventions

The following table lists and describes the conventions that apply in this manual.

Convention	Description
THIS TYPEFACE or <i>this typeface</i>	Denotes syntax that you must enter exactly as shown.
<i>this typeface</i>	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
<u>underscore</u>	Indicates a default setting.
<i>this typeface</i>	Represents a placeholder (or variable), a cross-reference, or an important term.
this typeface	Highlights a file name or command.
Key + Key	Indicates keys that you must press simultaneously.
{ }	Indicates two or three choices; type one of them, not the braces.
	Separates mutually exclusive choices in syntax. Type one of them, not the symbol.
...	Indicates that you can enter a parameter multiple times. Type only the parameter, not the ellipsis points (...).
.	Indicates that there are (or could be) intervening or additional commands.

Related Publications

To view a current listing of our publications and to place an order, visit our World Wide Web site, <http://www.iwaysoftware.com>. You can also contact the Publications Order Department at (800) 969-4636.

Customer Support

Do you have questions about the iWay Applications System Adapter for Amdocs ClarifyCRM?

If you bought the product from a vendor other than iWay Software, contact your distributor.

If you bought the product directly from iWay Software, call Information Builders Customer Support Services (CSS) at (800) 736-6130 or (212) 736-6130. Customer Support Consultants are available Monday through Friday between 8:00 A.M. and 8:00 P.M. EST to address all your iWay Applications System Adapter for Amdocs ClarifyCRM questions. Information Builders consultants can also give you general guidance regarding product capabilities and documentation. Please be ready to provide your six-digit site code (xxxx.xx) when you call.

You can also access support services electronically, 24 hours a day, with InfoResponse Online. InfoResponse Online is accessible through our World Wide Web site, <http://www.informationbuilders.com>. It connects you to the tracking system and known-problem database at the Information Builders support center. Registered users can open, update, and view the status of cases in the tracking system and read descriptions of reported software issues. New users can register immediately for this service. The technical support section of www.informationbuilders.com also provides usage techniques, diagnostic tips, and answers to frequently asked questions.

To learn about the full range of available support services, ask your Information Builders representative about InfoResponse Online, or call (800) 969-INFO.

Help Us to Serve You Better

To help our consultants answer your questions effectively when you call, please provide the following information:

- Your six-digit site code number (xxxx.xx).
- Your software configuration.

The following table lists the information to provide about your software configuration.

	Version-Build Date	HF/Service Pack	Patches	OS	Java Version
iWay Product					
Third-party Application Server					
EIS (adapter target)					

Note: For the EIS, ensure you record the application or database name and release level, including minor versions, for example, 4.6.1.

- The exact nature of the error or problem, specified as follows:
 - Steps to reproduce the problem.
 - Problem description (be as specific as possible).
 - Error message(s).
- To best define the problem, provide the following:
 - Screen captures of the error
 - Error output files
 - Trace files and log files
 - Log transaction
 - XML schemas and/or document instances
 - Other input documents (for example, transformations)
 - Configuration files (all are applicable):
 - .xch files
 - config.xml file
 - base.xml file
 - repository.xml file
 - ibserrepo.xml file
 - .dic files
 - .rules files
 - Environment variable settings:
 - IWAY55
 - IWAY55OEM
 - CLASSPATH
 - JAVA_HOME
 - ACBDIR
 - CBDIR (UNIX)
- Has the process, procedure, or query ever worked in its current form? Has it changed recently? If so, how (provide specific details)? How often does the problem occur?
- Can this problem be reproduced? If so, how? Can it be consistently reproduced?
- Have you tried to reproduce your problem in the simplest form possible?

- Do you have a trace file?
- How is the problem affecting your business? Is it halting development or production?
- Do you just have questions about functionality or documentation?

User Feedback

In an effort to produce effective documentation, the Documentation Services staff welcomes your opinions regarding this manual. Please use the Reader Comments form at the end of this manual to communicate suggestions for improving this publication or to alert us to corrections. You also can go to our Web site, <http://www.iwaysoftware.com> and use the Documentation Feedback form.

Thank you, in advance, for your comments.

iWay Software Training and Professional Services

Interested in training? Our Education Department offers a wide variety of training courses for iWay Software and other Information Builders products.

For information on course descriptions, locations, and dates, or to register for classes, visit our World Wide Web site, <http://www.iwaysoftware.com> or call (800) 969-INFO to speak to an Education Representative.

Interested in technical assistance for your implementation? Our Professional Services department provides expert design, systems architecture, implementation, and project management services for all your business integration projects. For information, visit our World Wide Web site, <http://www.iwaysoftware.com>.

Contents

User Feedback	1-viii
1. Introducing the iWay Application Systems Adapter for Amdocs ClarifyCRM ...	1-1
ClarifyCRM Application Suite	1-2
ClarifyCRM Architecture	1-2
The Adapter Interface	1-3
Accessing a ClarifyCRM Business Object Through the CBO Interface	1-3
Accessing the Database Through a ClarifyCRM Business Object	1-3
Service and Event Processing Using the CBO Interface	1-3
Processing Services	1-5
Processing Events	1-6
Deployment Options for the iWay Applications System Adapter for Amdocs ClarifyCRM	1-7
Deployment Information Roadmap	1-7
The iWay Business Services Engine	1-8
The iWay Enterprise Connector for J2EE Connector Architecture	1-9
2. Creating XML Schemas and Web Services With ClarifyCRM Business Objects ..	2-1
Generating Schemas and Business Services Overview	2-2
Starting iWay Application Explorer	2-2
Opening a Connection to ClarifyCRM	2-5
Creating a New Target	2-5
Connecting to an Existing Target	2-8
Closing or Deleting a Target	2-9
Modifying a Target	2-10
Generating a Schema	2-11
Schema Location	2-15
Request and Response Document Formats	2-16
Request Document Examples	2-16
Response Document Examples	2-25
Generating a Business Service	2-32
Testing a Business Service	2-35
Identity Propagation	2-36
Deleting a Business Service	2-39
3. Creating XML Schemas and Web Services With ClarifyCRM ClearBasic	3-1
Generating Schemas and Business Services Overview	3-2
Starting iWay Application Explorer	3-2
Opening a Connection to ClarifyCRM	3-4
Creating a New ClarifyCRM CB Target	3-4
Connecting to an Existing Target	3-7
Closing or Deleting a Target	3-9

Modifying a Target	3-10
Generating a Schema	3-11
Schema Location	3-15
Request and Response Documents	3-16
Generating a Business Service	3-16
Testing a Business Service	3-20
Identity Propagation	3-21
Deleting a Business Service	3-23
4. Listening for Events Using ClarifyCRM Business Objects	4-1
Understanding iWay Event Functionality	4-2
Creating an Event Port	4-3
FILE Disposition URL Format	4-7
iBSE Disposition URL Format	4-8
MSMQ Disposition URL Format	4-9
JMS Disposition URL Format	4-10
SOAP Disposition URL Format	4-12
HTTP Disposition URL Format	4-13
MQSeries Disposition URL Format	4-14
Editing or Deleting an Event Port	4-15
Creating a Channel	4-17
Post Query Parameter Operators	4-23
Editing or Deleting a Channel	4-24
5. Listening for Events Using ClarifyCRM ClearBasic	5-1
Understanding iWay Event Functionality	5-2
Creating an Event Port	5-2
FILE Disposition URL Format	5-7
iBSE Disposition URL Format	5-8
MSMQ Disposition URL Format	5-8
JMS Disposition URL Format	5-9
SOAP Disposition URL Format	5-11
HTTP Disposition URL Format	5-12
MQSeries Disposition URL Format	5-13
Editing or Deleting an Event Port	5-14
Creating a Channel	5-16
Editing or Deleting a Channel	5-20
6. Using Web Services Policy-Based Security	6-1
iWay Business Services Policy-Based Security	6-2
Configuring iWay Business Services Policy-Based Security	6-3
7. Management and Monitoring	7-1
Managing and Monitoring Services and Events Using iBSE	7-2

Managing and Monitoring Services and Events Using the JCA Test Tool	7-16
Setting Engine Log Levels	7-21
Configuring Connection Pool Sizes	7-22
Migrating Repositories	7-23
File Repositories	7-23
iBSE Repositories	7-23
JCA Repositories	7-28
Migrating Event Handling Configurations	7-28
Exporting or Importing Targets	7-32
Retrieving or Updating Web Service Method Connection Information	7-36
Starting or Stopping a Channel Programmatically	7-40
A. Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services With CBO	A-1
Starting Application Explorer in BEA WebLogic Workshop	A-2
Creating a New Configuration	A-3
Connecting to and Managing a Target	A-5
Creating and Connecting to a Target	A-5
Managing a Target	A-8
Creating an XML Schema	A-10
Schema Location	A-11
Creating an Integration Business Service	A-12
Exporting iWay WSDL for Use in BEA WebLogic Workshop Workflows	A-16
Identity Propagation	A-17
Adding a Control for an iWay Resource in BEA WebLogic Workshop	A-18
Adding a Web Service Control to a BEA WebLogic Workshop Application	A-18
Adding an Extensible CCI Control	A-19
Overview	A-19
Using the Extensible CCI Control	A-27
B. Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services with ClarifyCRM ClearBasic	B-1
Starting Application Explorer in BEA WebLogic Workshop	B-2
Creating a New Configuration	B-3
Connecting to and Managing a Target	B-5
Creating and Connecting to a Target	B-5
Connecting to an Existing Target	B-8
Managing a Target	B-9
Creating an XML Schema	B-10
Creating an Integration Business Service	B-12
Exporting iWay WSDL for Use in BEA WebLogic Workshop Workflows	B-16
Identity Propagation	B-17

Adding a Control for an iWay Resource in BEA WebLogic Workshop	B-18
Adding a Web Service Control to a BEA WebLogic Workshop Application	B-18
Adding an Extensible CCI Control	B-19
Overview	B-19
Using the Extensible CCI Control	B-27
C. Using Application Explorer in BEA WebLogic Workshop for	
Event Handling with CBOs	C-1
Starting Application Explorer in BEA WebLogic Workshop	C-2
Understanding iWay Event Functionality	C-3
Creating an Event Port	C-3
FILE Disposition URL Format	C-7
iBSE Disposition URL Format	C-8
MSMQ Disposition URL Format	C-8
JMS Disposition URL Format	C-9
SOAP Disposition URL Format	C-10
HTTP Disposition URL Format	C-12
MQSeries Disposition URL Format	C-12
Post Query Parameter Operators	C-13
Modifying an Event Port	C-14
Creating a Channel	C-16
Modifying a Channel	C-23
Deploying iWay Components in a Clustered BEA WebLogic Environment	C-26
D. Using Application Explorer in BEA WebLogic Workshop for	
Event Handling with ClarifyCRM ClearBasic	D-1
Starting Application Explorer in BEA WebLogic Workshop	D-2
Understanding iWay Event Functionality	D-3
Creating an Event Port	D-3
Modifying an Event Port	D-23
Creating a Channel	D-24
Modifying a Channel	D-28
Deploying iWay Components in a Clustered BEA WebLogic Environment	D-31

CHAPTER 1

Introducing the iWay Application Systems Adapter for Amdocs ClarifyCRM

Topics:

- ClarifyCRM Application Suite
- Accessing a ClarifyCRM Business Object Through the CBO Interface
- Deployment Options for the iWay Applications System Adapter for Amdocs ClarifyCRM

The iWay Applications System Adapter for Amdocs ClarifyCRM provides a means to exchange real-time business data between ClarifyCRM systems and other application, database, or external business partner systems.

The adapter enables inbound and outbound processing with ClarifyCRM systems and provides scalable, reliable, and secure access, enabling you to use ClarifyCRM data and functions within your business processes.

ClarifyCRM Application Suite

ClarifyCRM eFrontOffice (CeFO) provides an integrated solution that enables you to manage interactions with your customers. With CeFO, you can track sales, contracts, customers, product development, inventory, and repair operations. You can capture customer interactions made by telephone, e-mail, or over the Web. You can follow interactions with a customer from the first request for information, through the sales cycle, and through subsequent requests for additional products or services.

Examples of CeFO client applications include the following:

- ClearSupport, which enables you to manage incoming calls to customer support.
- ClearQuality, which enables you to record and monitor product defects, enhancement requests, and changes.
- ClearLogistics, which enables you to manage service-related inventory and on-site service calls.
- ClearSales, which enables you to collect and manage information for marketing campaigns and sales opportunities.
- ClearContracts, which enables you to manage service contracts.

ClarifyCRM Architecture

CeFO is a client-server solution, which means that you can run applications on your computer (the client) to work with information stored in a central database on another computer (the server).

When you run a CeFO application on the client, you can view forms for your business processes, such as bills of material, shipping labels, inventory counts, and contract information. You use these forms to view and change information in a central database on the server.

For example, in ClearSupport you can enter information about an incoming customer call in a form. The information that you enter is stored in the main database. Other employees can use ClearSupport to view and update the information that you entered.

CeFO applications are available in the following two types of clients:

- ClarifyCRM Desktop LAN/WAN Client
- ClarifyCRM Desktop Web Client

The ClarifyCRM Desktop LAN/WAN and Web Clients are virtually identical.

The Adapter Interface

The iWay Applications System Adapter for Amdocs ClarifyCRM supports synchronous and asynchronous, bidirectional message interactions for ClarifyCRM. The adapter provides integration with ClarifyCRM Business Objects (CBOs) through the CBO interface.

Accessing a ClarifyCRM Business Object Through the CBO Interface

ClarifyCRM Business Objects (CBOs) are part of the ClarifyCRM eBusiness Framework for developing applications for the ClarifyCRM eFrontOffice (CeFO) database.

These business objects are C++ objects with a Java layer that provide access to data in the CeFO database. Each type of business object implements part of the ClarifyCRM data model and encapsulates the application logic for working with that part of the model.

Accessing the Database Through a ClarifyCRM Business Object

The iWay Applications System Adapter for Amdocs ClarifyCRM provides controlled access to the data contained in the ClarifyCRM database through the CBO interface. You can use the iWay Applications System Adapter for Amdocs ClarifyCRM to perform inserts, updates, gets, and deletes against ClarifyCRM objects stored in the ClarifyCRM database. You also can use the adapter to integrate ClarifyCRM with non-ClarifyCRM systems.

A business object has methods and properties that are used to query, manipulate, and update data in a CeFO database table. The GET method is issued through an XML document the same way as Insert, Update, and Delete. The results of the GET are contained in an XML response document.

Each business object contains a row set, which is an in-memory copy of a set of rows from a CeFO database table. When a database table is queried, the business object holds the results of the query in its row set. You can select a row in the row set and obtain the values of the fields in that row.

You can use the row set to make changes to data in the database table. You can modify the values of fields in rows and commit the changed rows back to the database. If a new row must be added to the database table, you add the rows to the row set in the business object and commit the new rows to the database.

Service and Event Processing Using the CBO Interface

The iWay Applications System Adapter for Amdocs ClarifyCRM connects one application to another when those applications are not designed to communicate with one another. The adapter sends requests to ClarifyCRM and manipulates data. In some cases, rows are returned. In other cases, data is modified or inserted.

The iWay Applications System Adapter for Amdocs ClarifyCRM supports both services and events, that is, it supports full bidirectional access to the ClarifyCRM applications.

Services Applications use this capability to initiate a service request to access or manipulate data through CBOs. For example, the adapter can perform inserts, updates, gets, and deletes against ClarifyCRM objects stored in the ClarifyCRM database. The adapter receives an XML request document from a client and issues the appropriate CBO method call. The ClarifyCRM application processes the calls and returns a service response. There are two kinds of services:

- **Asynchronous** The client application issues a service request and then proceeds with its processing. It does not wait for the response.
- **Synchronous** The client application waits for the response before proceeding with further processing.

Service processing includes:

- Receiving an XML-based service request from an external client.
- Interpreting the request document and making appropriate CBO method calls.
- Transforming a response from a ClarifyCRM-specific data format to an XML document. The XML document conforms to the response XML schema for the service.

The schema is based on ClarifyCRM metadata.

Events Applications use this capability if they require access to ClarifyCRM data only when a ClarifyCRM business event occurs. When the adapter detects an event in ClarifyCRM, it is processing an event.

The adapter monitors a ClarifyCRM database that represents current data in a ClarifyCRM system. When that information is retrieved by the adapter, it is formatted as an XML document and sent to the iWay Integration Server. Adapter-supplied database triggers move only the relevant data about a ClarifyCRM business process into an events database. After the event is processed successfully, the event data is deleted from the event database.

Event processing includes:

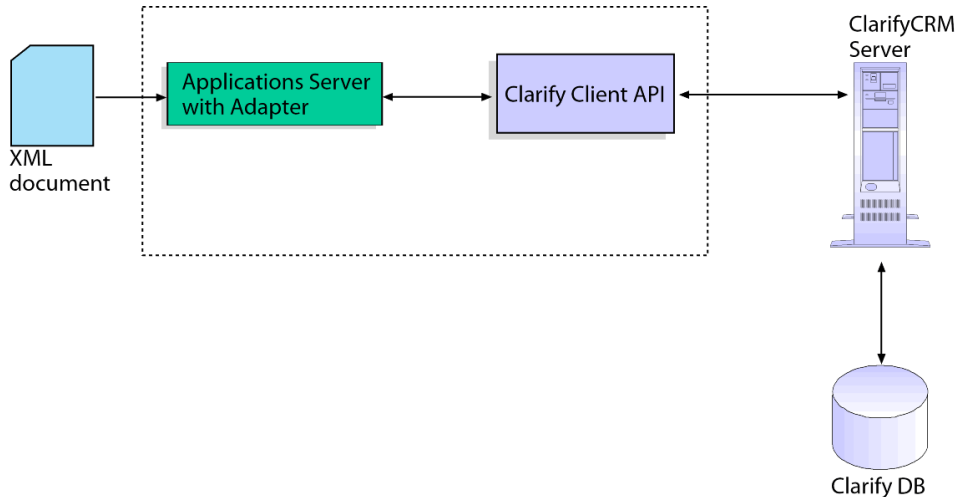
- Applying triggers to specified tables within the ClarifyCRM application.
- Polling the ClarifyCRM events database at user-configured time intervals.
- Translating the event information to XML.

The adapter provides a Java-based utility that enables the browsing of CBO-based tables and the creation of adapter-based triggers within the desired CBO tables. When a row within one of the tables is affected by an update, insert, or delete, the trigger moves key related information to a custom table supplied by the adapter installation process.

The adapter polls this custom event table and returns the data in an XML response document. The adapter maintains the logic to only process the current (“unprocessed”) row from the event table.

Processing Services

The following diagram illustrates the framework for processing a service with the iWay Applications System Adapter and the iWay Integration Server. It shows the flow of an XML document passing to the application server with the adapter, to the Clarify client API, then to the ClarifyCRM server and the Clarify database.

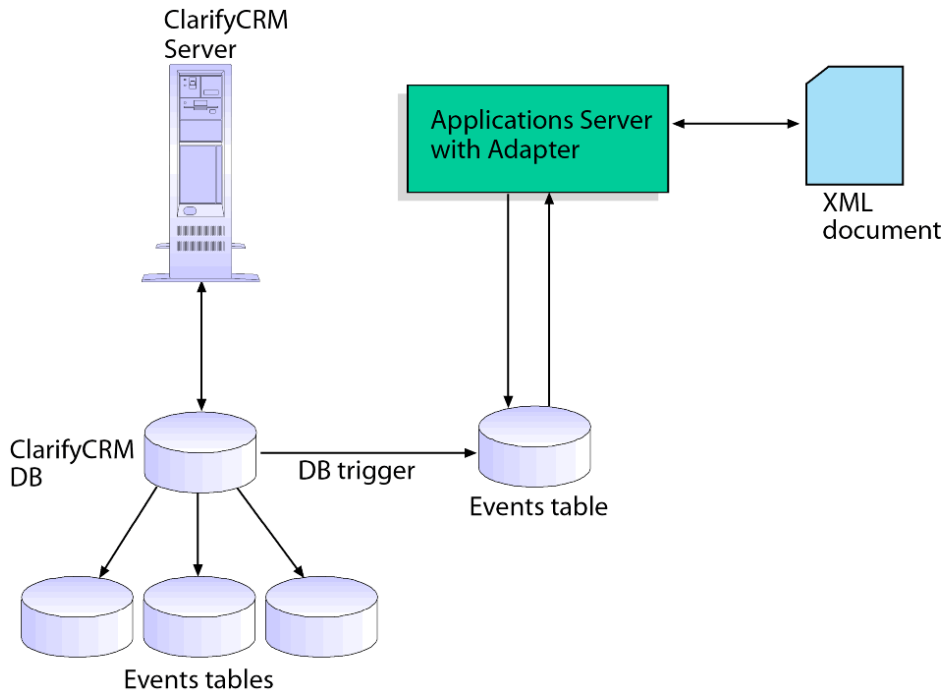


When the iWay Applications System Adapter for Amdocs ClarifyCRM processes a service:

1. The adapter receives an XML service request document.
2. The adapter validates the request to ensure that it conforms to the service request schema.
3. The adapter interprets the document and issues the corresponding CBO method calls to the ClarifyCRM Server.
4. The ClarifyCRM Server processes the calls and returns the service response, if any, to the adapter.
5. From the response, the adapter generates an XML response document. The document conforms to the service response schema.
6. The adapter returns the XML response document to the iWay Integration Server.

Processing Events

The following diagram illustrates the framework for processing an event with the iWay Applications System Adapter and the iWay Integration Server. It shows the ClarifyCRM server connected to the ClarifyCRM database event tables, which are being monitored for a change by the adapter. A database trigger (an event) is acknowledged by the adapter, which then produces an XML document.



When the iWay Applications System Adapter for Amdocs ClarifyCRM processes an event:

1. A business process (for example, a booked service request or a completed sales call) is completed against a ClarifyCRM application.
2. Data changes within the ClarifyCRM system, which triggers the relevant event data to be copied to the event table.
3. The adapter periodically polls the events table with a specified SQL query.

When the events table contains rows that satisfy the query, the iWay Applications System Adapter for Amdocs ClarifyCRM obtains the data and packages it in XML format.

Deployment Options for the iWay Applications System Adapter for Amdocs ClarifyCRM

The iWay Applications System Adapter for Amdocs ClarifyCRM works with iWay Application Explorer in conjunction with one of the following components:

- iWay Business Services Engine (iBSE)
- iWay Enterprise Connector for J2EE™ Connector Architecture (JCA)

Application Explorer, used to access ClarifyCRM metadata and create Web services and events, can be configured to work in a Web services environment in conjunction with iBSE or the iWay Enterprise Connector for JCA. When working in a JCA environment, the connector uses the Common Client Interface (CCI) to provide fast integration services using an iWay adapter instead of Web services.

Both iBSE and the iWay Connector for JCA are deployed to your application environment with Application Explorer and the adapters.

Deployment Information Roadmap

The following table lists the location of deployment information for the iWay Applications System Adapter for Amdocs ClarifyCRM. A description of the iWay Business Services Engine (iBSE) and the iWay Enterprise Connector for J2EE Connector Architecture (JCA) follow the table.

Deployed Component	For more information, see
iWay Application Explorer	Chapters 2 and 3, 4 and 5 of this guide Appendices A, B, C, and D of this guide when using Application Explorer inside BEA WebLogic Workshop <i>iWay Installation and Configuration for BEA WebLogic</i> <i>iWay Servlet Application Explorer for BEA WebLogic User's Guide</i>
iWay Business Services Engine (iBSE)	<i>iWay Installation and Configuration for BEA WebLogic</i>
iWay Enterprise Connector for J2EE Connector Architecture (JCA)	<i>iWay Connector for JCA for BEA WebLogic User's Guide</i> <i>iWay Installation and Configuration for BEA WebLogic</i>

The following versions of Application Explorer are available when deploying the adapter with BEA WebLogic Server:

- **Servlet.** Deployed as a Web application on BEA WebLogic Server, this version is accessible through a Web browser. In addition, the servlet Application Explorer can be used with iWay Business Services Engine (iBSE) and iWay Enterprise Connector for J2EE Connector Architecture (JCA). For more information, see the following chapters:
 - Chapter 2, *Creating XML Schemas and Web Services With ClarifyCRM Business Objects*
 - Chapter 4, *Listening for Events Using ClarifyCRM Business Objects*
- **Integrated Java Swing.** Tightly integrated within the BEA WebLogic toolset, the integrated Java Swing Application Explorer can be accessed directly from BEA WebLogic WorkShop, where WSDL (Web Services Description Language) files generated from Integration Business Services and XML schemas can be shared as resources within a BEA WebLogic WorkShop application. For more information, see the following appendices:
 - Appendix A, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services With CBO*
 - Appendix C, *Using Application Explorer in BEA WebLogic Workshop for Event Handling with CBOs*

Note: To use Application Explorer within BEA WebLogic WorkShop, you must deploy the iWay Business Services Engine (iBSE). For more information, see the *iWay Installation and Configuration for BEA WebLogic* manual.

The iWay Business Services Engine

The iWay Business Services Engine (iBSE) exposes—as Web services—enterprise assets that are accessible from adapters regardless of the programming language or the particular operating system.

iBSE simplifies the creation and execution of Web services when running:

- Custom and legacy applications
- Database queries and stored procedures
- Packaged applications
- Terminal emulation and screen-based systems
- Transactional systems

Web services is a distributed programming architecture that solves Enterprise Application Integration (EAI) hurdles that other programming models cannot. It enables programs to communicate with one another using a text-based but platform and language independent message format called XML.

Coupled with a platform and language independent messaging protocol called SOAP (Simple Object Access Protocol), XML enables application development and integration by assembling previously built components from multiple Web services.

The iWay Enterprise Connector for J2EE Connector Architecture

The iWay Enterprise Connector for J2EE Connector Architecture (JCA) enables developers of JCA-compliant applications to deploy iWay adapters as JCA resources.

The iWay Connector for JCA is distributed as a standard Resource Adapter Archive (RAR) for deployment to the application server. Thus, the connector can be used in systems that are non-compliant, although services such as pooled connections are not available.

CHAPTER 2

Creating XML Schemas and Web Services With ClarifyCRM Business Objects

Topics:

- Generating Schemas and Business Services Overview
- Starting iWay Application Explorer
- Opening a Connection to ClarifyCRM
- Closing or Deleting a Target
- Modifying a Target
- Generating a Schema
- Request and Response Document Formats
- Generating a Business Service

This section describes how to use iWay Servlet Application Explorer to create XML schemas and Web services for ClarifyCRM using ClarifyCRM Business Objects (CBOs).

Although this section describes the Java™ servlet implementation of Application Explorer, other implementations provide the same functionality by means of similar graphical user interfaces. For information on running Application Explorer in BEA WebLogic Workshop, see Appendix A, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services With CBO*.

Generating Schemas and Business Services Overview

The iWay Applications System Adapter for Amdocs ClarifyCRM enables you to create and open connections to the ClarifyCRM database using the iWay Servlet Application Explorer. Through a connection to the database you can generate XML schemas to define request and response documents, and generate a business service.

1. **Start iWay Servlet Application Explorer and open a new or existing connection** to the ClarifyCRM database, as described in *Opening a Connection to ClarifyCRM* on page 2-5.
2. **Generate XML schemas** that define request and response documents for your SQL statements and stored procedures, as described in *Generating a Schema* on page 2-11.

You can use the schemas when you create request documents and when you develop logic to process responses.
3. **Create request documents** for each operation against each table and for each stored procedure.

You can use a third-party XML tool to generate a request document from the XML schema. For examples of request and response documents, see *Request and Response Document Formats* on page 2-16.
4. **Generate a business service** (also known as a Web service) for an SQL statement or stored procedure. For more information see *Generating a Business Service* on page 2-32.

Starting iWay Application Explorer

You can use Servlet Application Explorer to:

- Generate XML schemas that define request and response documents for your database tables. You can use these schemas when you create request documents and when you develop logic for processing response documents.
- View metadata that describes your database tables. You can use this metadata when you create request documents and when you develop logic for processing response documents.
- Create business services (also known as Web services).

Before you can use iWay Servlet Application Explorer, you must start BEA WebLogic Server.

Procedure: How to Start Application Explorer

1. Start BEA WebLogic Server, as follows:

On Windows:

Click the Start menu and select *Programs, BEA WebLogic Platform 8.1, User Projects*, then *your domain for iWay*, and then click *Start Server*.

On UNIX or from a command line:

Type the following at the prompt:

```
BEA_HOME\user_projects\domains\DOMAIN_NAME\startWebLogic.cmd
```

where:

BEA_HOME

Is the directory where BEA WebLogic is installed.

DOMAIN_NAME

Is the domain you are using for iWay.

2. After BEA WebLogic Server is running, enter the following URL in your browser window

```
http://hostname:port/iwae/index.html
```

where:

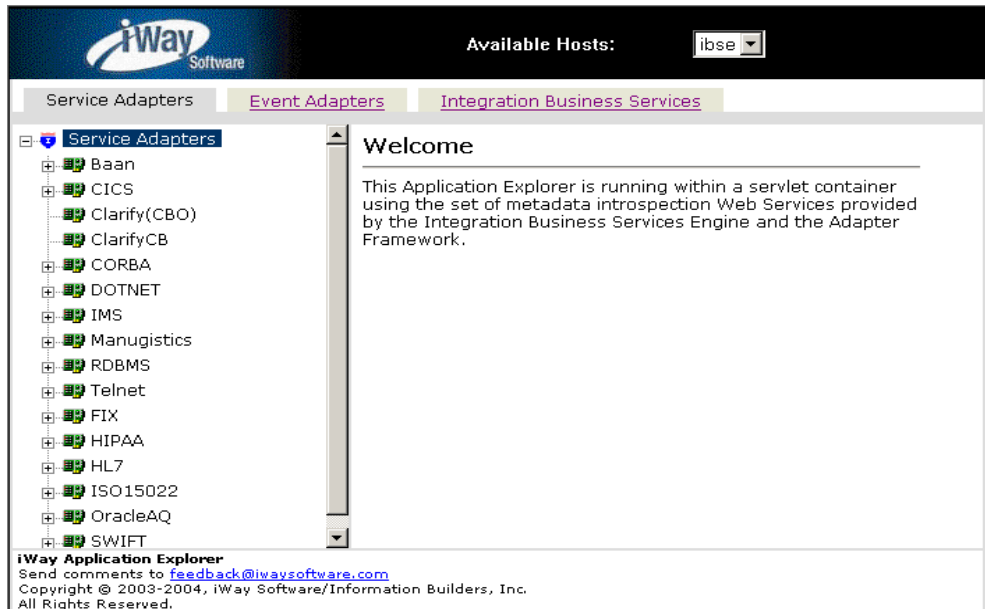
hostname

Is the machine where your application server is installed.

port

Is the port number on which the application server is listening.

iWay Application Explorer opens, as shown in the following image. The main features of this window are an Available Hosts drop-down list in the upper-right corner, three tabs across the top window, an expandable Service Adapters node in the left pane, and a welcome message in the right pane.



The three tabs are:

Service Adapters, where you create and manage connections to the ClarifyCRM database.

Event Adapters, where you configure ClarifyCRM database event listening.

iWay Business Services, where you generate XML schemas and create and view business services.

Your selection in the Available Hosts drop-down list reflects the instances you can access. The default choices are iWay JCA Connector (JCA) and Servlet iBSE (ibse). For information about adding instances, see the *iWay Installation and Configuration Guide*.

3. Expand the list of adapters by clicking the *Service Adapters* node.

You are ready to add new targets to ClarifyCRM.

Opening a Connection to ClarifyCRM

To browse ClarifyCRM database tables, you must create targets for ClarifyCRM. These targets serve as your connection points. You must establish a connection to ClarifyCRM every time you start iWay Application Explorer or after you disconnect from the system.

You can connect to ClarifyCRM by:

- Creating a new CBO target.
- Connecting to an existing CBO target.

The left pane of Application Explorer displays the supported application systems, which are based on the iWay adapters you installed and are licensed to use.

Creating a New Target

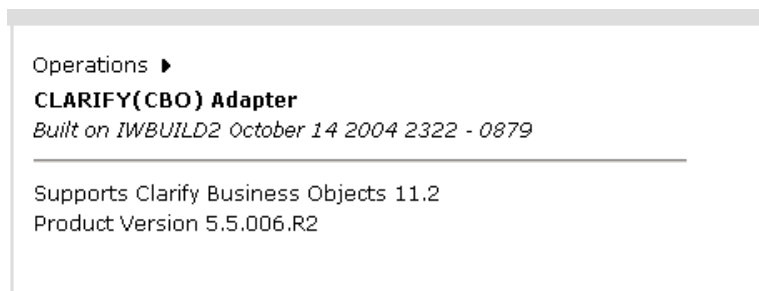
A target serves as the connection point to your Enterprise Information System (EIS) and is automatically saved after you create it. To connect to ClarifyCRM for the first time, you must create a new target from the Service Adapters tab.

Procedure: How to Create a New CBO Target

To create a new CBO target:

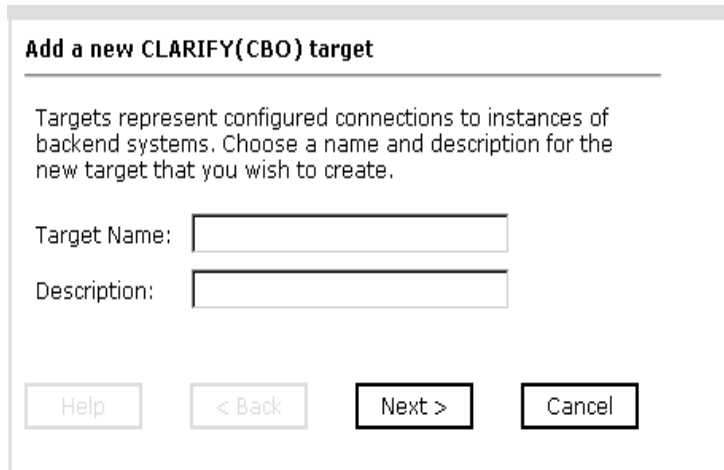
1. In the left pane, expand the *Service Adapters* node and select the *Clarify(CBO)* node.

In the right pane, the Operations option appears along with the title, build date, and product version of the adapter, as shown in the following image.



2. Move the pointer over *Operations* and select *Define a new target*.

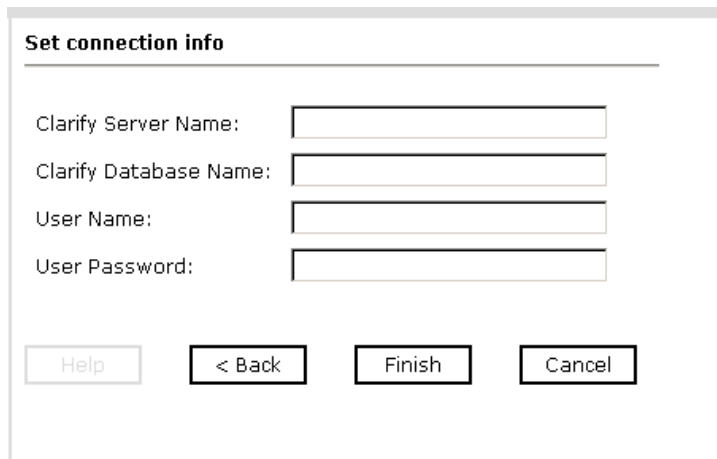
The Add a new CLARIFY(CBO) target pane opens on the right, as shown in the following image. This pane includes a general statement about targets, two target input fields, a help button, and three action buttons.



The screenshot shows a dialog box titled "Add a new CLARIFY(CBO) target". Below the title is a horizontal line, followed by a paragraph of text: "Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create." Below this text are two input fields: "Target Name:" and "Description:". At the bottom of the dialog box are four buttons: "Help", "< Back", "Next >", and "Cancel".

- a. In the Target Name field, type a descriptive name for the target, for example, ClarifyNew.
 - b. In the Description field, type a brief description of the connection.
3. Click *Next*.

The Set connection info pane opens on the right, as shown in the following image. This pane provides fields to define the database connection parameters.

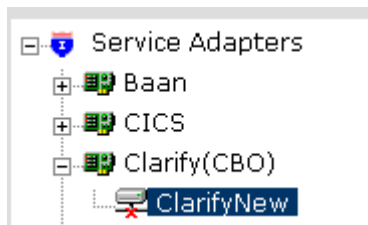


The image shows a dialog box titled "Set connection info". It contains four text input fields: "Clarify Server Name:", "Clarify Database Name:", "User Name:", and "User Password:". Below the fields are four buttons: "Help", "< Back", "Finish", and "Cancel".

Note: The ClarifyCRM connection parameters are consistent with those found in your ClarifyCRM system. For more information on parameter values that are specific to your ClarifyCRM configuration, consult your ClarifyCRM system administrator.

- a. In the Clarify Server Name field, type the name of the server.
 - b. In the Clarify Database Name field, type the name of the database.
 - c. In the User Name field, type the user identification to enter the database.
 - d. In the User Password field, type the password that is associated with the User Name.
4. Click *Finish*.

In the left pane, the new target appears beneath the Clarify(CBO) node. The following images shows an example of a new target, ClarifyNew, under the Clarify(CBO) node. The red X below the ClarifyNew icon indicates that the node is disconnected.



You are ready to connect to the application target you created.

Connecting to an Existing Target

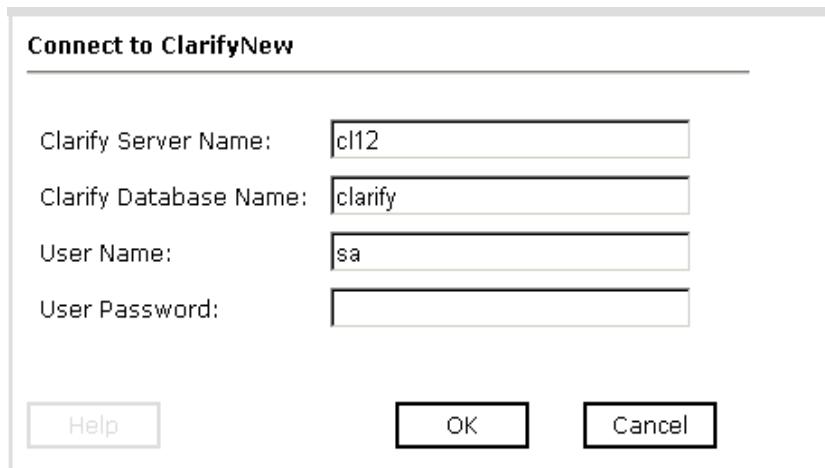
You can use the already defined targets to connect to instances of ClarifyCRM.

Procedure: How to Connect to a Target

To connect to an existing target:

1. In the left pane, expand the *Clarify(CBO)* node and select the target to which you want to connect.
2. In the right pane, move the pointer over *Operations* and select *Connect*.

The Connect to pane with connection parameter fields opens on the right. An example of this pane is shown in the following image.



Connect to ClarifyNew

Clarify Server Name:

Clarify Database Name:

User Name:

User Password:

The first three input fields (Clarify Server Name, Clarify Database Name, and User Name) are pre-populated with the information you provided when you created the target.

- a. In the User Password field, type a valid password.
- b. Click *OK*.

If the parameters are correct and the ClarifyCRM or server component is available, the selected target is highlighted with a plus sign and the Operations menu becomes available on the right, as shown in the following image.



Closing or Deleting a Target

Although you can maintain multiple open connections to different application systems, we recommend that you close connections when they are not in use.

Procedure: How to Disconnect From a Target

To close, or disconnect from, a target:

1. Select the ClarifyCRM target from which you want to disconnect.
2. In the right pane, move the pointer over *Operations* and select *Disconnect*.

Disconnecting from the ClarifyCRM target drops the connection, but the target definition and its node remain visible. The target node displays a red X below the icon to reflect that the connection is closed. The following image shows an example of a disconnected target.



Procedure: How to Delete a Target

You can delete a connection whether or not it is closed; if open, it automatically closes before it is deleted.

To delete a target from Application Explorer:

1. In the left pane, select the ClarifyCRM target.
2. In the right pane, move the pointer over *Operations* and select *Delete*.

A confirmation dialog box opens.

3. To delete the selected target, click *OK*.

The target node disappears from the left pane.

Modifying a Target

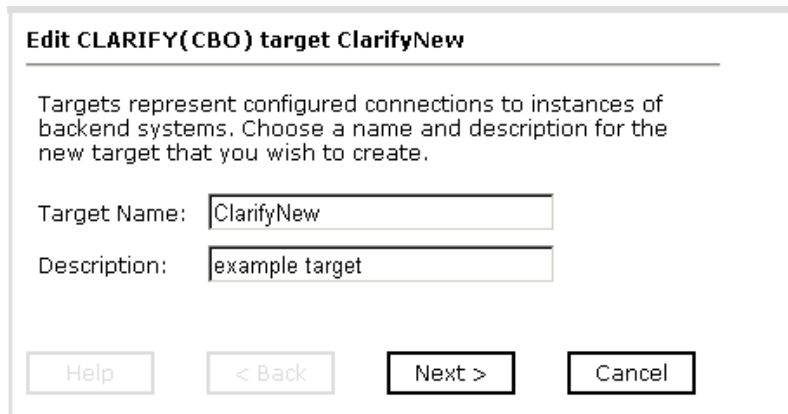
After a target exists in iWay Servlet Application Explorer, you can edit the target name, description, and connection parameters. The Edit function is available only for a disconnected target.

Procedure: How to Edit a Target

To edit a target in Application Explorer:

1. Be sure that the node you want to edit is disconnected. See *Closing or Deleting a Target* on page 2-9.
2. In the left pane, select the target you want to edit.
3. In the right pane, move the pointer over *Operations* and select *Edit*.

The Edit CLARIFY(CBO) target pane appears on the right. This pane provides a general statement about targets followed by two fields, a help button, and three action buttons.



Edit CLARIFY(CBO) target ClarifyNew

Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.

Target Name:

Description:

4. Modify the Target Name and Description fields as required.
5. Click *Next*.

The Set connection info pane opens. This pane is described in *Create a New CBO Target* on page 2-5.

6. Edit the parameters in the Set connection info pane as required.
7. Click *Finish*.

Generating a Schema

XML schemas are used to define service request documents and the corresponding response documents.

When you deploy the adapter in a business services environment, you are not required to generate a schema. For more information, see *Generating a Business Service* on page 2-32.

You can create XML schemas for Clarify(CBO):

- services
- events

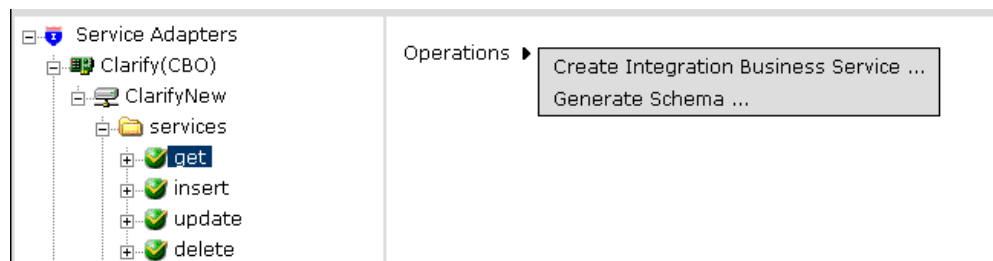
After you create the schema, you can view it or save it and use the schema to create request and response documents.

Procedure: How to Create an XML Schema for a Service

To create a schema for a Clarify(CBO) target:

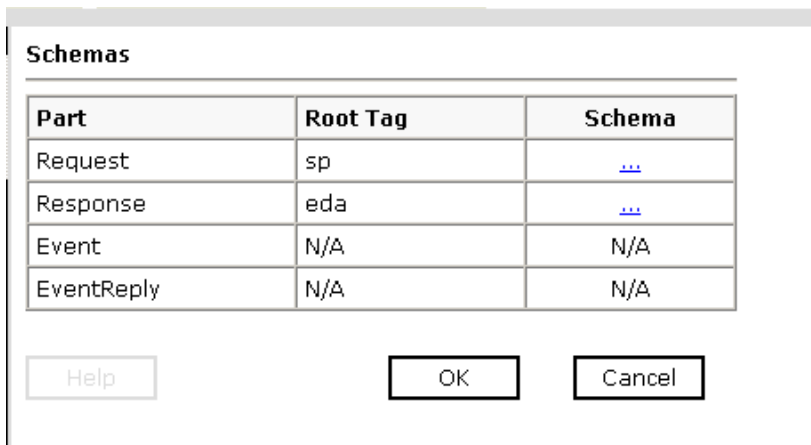
1. Connect to a Clarify(CBO) target, as described in *Opening a Connection to ClarifyCRM* on page 2-5.
2. In the left pane of Application Explorer, expand the target node and then the *services* node.
3. Select the module containing the table for which you want to create schemas.

The following image shows the expanded services node in the left pane, and the available options in the Operations menu in the right pane.



4. In the right pane, move the pointer over *Operations* and select *Generate Schema*.

Application Explorer builds the schema and displays it in a table on the right. The following image shows an example of a Schemas table that has three columns labeled Part, Root Tag, and Schema. The Schema column includes hyperlinks to the request and response schema for the selected table service.



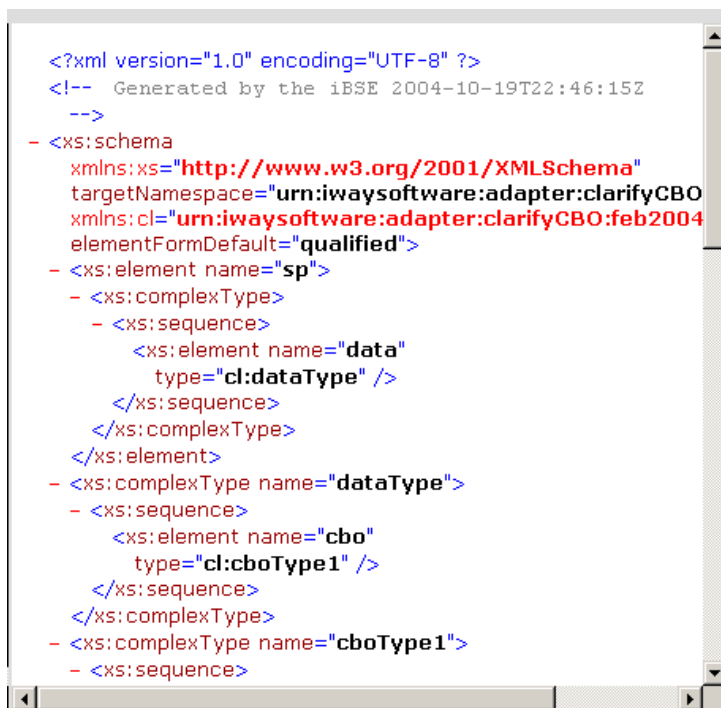
Schemas

Part	Root Tag	Schema
Request	sp	...
Response	eda	...
Event	N/A	N/A
EventReply	N/A	N/A

Help OK Cancel

5. In the Schema column of the Schema table, click the option (...) to view the Request or Response schema.

The selected XML schema code appears in the right pane. The following image is an example of this pane displaying XML schema.



6. To save the schema to a directory other than the default (see *Schema Location* on page 2-15 for the default location):
 - a. Right-click the right pane.
 - b. Select *View Source*.
The source XML schema opens in a text editor.
 - c. Save the schema or copy and paste it to another location.
7. Use the *Back* button on the Web browser to return to the Schemas table.
8. To return to the Operations menu from the Schemas table, click *OK*.

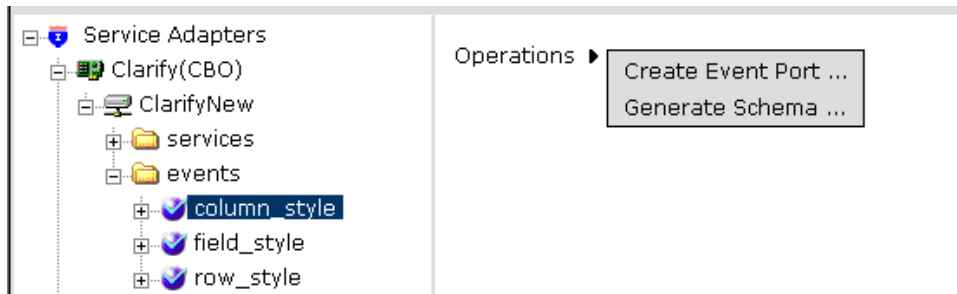
Procedure: How to Create an XML Schema for an Event

To create a schema for a Clarify(CBO) event:

1. Connect to a Clarify(CBO) target, as described in *Opening a Connection to ClarifyCRM* on page 2-5.

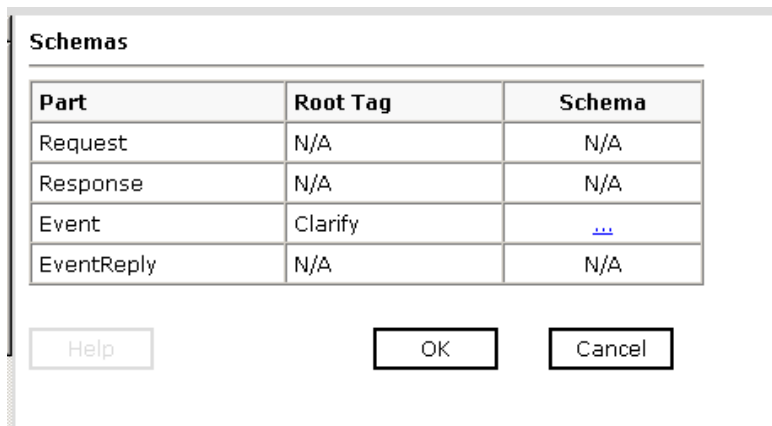
2. In the left pane of Application Explorer, expand the target node, and then the *events* node.
3. Select the events module containing the table for which you want to create schemas.

The following image shows the expanded events node in the left pane and the available options in the Operations menu in the right pane.



4. In the right pane, move the pointer over *Operations* and select *Generate Schema*.

Application Explorer builds the schema and displays a Schemas table on the right. The following image shows an example of a Schemas table that has three columns labeled Part, Root Tag, and Schema. The Schema column includes hyperlinks to the request and response schema for the selected table service.



5. In the Schema column of the Schemas table, click the option (...) in the Event row to view the associated schema.

The XML schema code appears in the right pane.

6. To save the schema to a directory other than the default (see *Schema Location* on page 2-15 for the default location):

- a. Right-click the right pane.

- b. Select *View Source*.

The source XML schema opens in a text editor.

- c. Save the schema or copy and paste it to another location.

Use the *Back* button on the Web browser to return to the Schemas table.

To return to the Operations menu from the Schemas table, click *OK*.

Schema Location

Application Explorer stores the schemas it creates in subdirectories under the iWay home directory of the machine where it is installed. The exact location of the schemas differs depending on whether you deploy Application Explorer with an iBSE or a JCA configuration.

- When using the adapter with an iBSE configuration, the schemas are stored under a \schemas subdirectory of the iWay home directory, for example,

`C:\Program`

`Files\iway55\bea\ibse\wsdl\schemas\service\Clarify(CBO)\ClarifyNew`

where:

`ClarifyNew`

Is the name of the connection to the ClarifyCRM system as defined in Application Explorer. Under this directory, Application Explorer creates subdirectories containing schemas.

- When using the adapter with a JCA configuration, the schemas are stored under a \schemas subdirectory of the iWay home directory, for example,

`C:\Program Files\iWay55\config\base\schemas\ClarifyCRM\NewTarget`

where:

`NewTarget`

Is the name of the connection to the ClarifyCRM system as defined in Application Explorer. Application Explorer stores the schemas in this directory.

Request and Response Document Formats

This topic describes the format of the adapter service request and response documents. You can generate request and response document schemas using Application Explorer and then use a third-party XML tool to generate a request document instance from the schema.

For services, the request and response documents are for:

- tables
- concurrent programs

When processing a request document, the adapter performs all table insertions before it runs any concurrent programs (using the `submit_request` element). If a table insertion fails, all insertions are rolled back; otherwise, they are committed, regardless of the concurrent program result.

Request Document Examples

The following are examples of request documents for Get, Insert, Update, and Delete services.

Example: GET Item Request Document

The following is an example of a request document to retrieve an item from the ClarifyCRM database.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Generated by the iBSE 2004-06-18T16:09:17Z
-->
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNam
  espace="urn:iwaysoftware:adapter:clarifyCBO:feb2004"
  xmlns:cl="urn:iwaysoftware:adapter:clarifyCBO:feb2004" element
  FormDefault="qualified">
  <xs:element name="sp">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="data"
          type="cl:dataType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="dataType">
    <xs:sequence>
      <xs:element name="cbo"
        type="cl:cboType1" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cboType1">
    <xs:sequence>
      <xs:element name="keys"
        type="cl:keysType" minOccurs="0" />
      <xs:element name="fields"
        type="cl:fieldsType" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"
      use="required" />
    <xs:attribute name="action" type="xs:string"
      fixed="get" />
  </xs:complexType>
  <xs:complexType name="cboType2">
    <xs:sequence>
      <xs:element name="fields"
        type="cl:fieldsType" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"
      use="required" />
    <xs:attribute name="relation" type="xs:string"
      use="required" />
  </xs:complexType>
</xs:schema>
```

```
</xs:complexType>
<xs:complexType name="fieldsType">
  <xs:sequence>
    <xs:element name="field"
      type="cl:fieldType"
      maxOccurs="unbounded" />
    <xs:element name="cbo"
      type="cl:cboType2" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="keysType">
  <xs:sequence>
    <xs:element name="field"
      type="cl:fieldType" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="fieldType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="name"
        type="xs:string" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>
```


Example: Insert Item Request Document

The following is an example of an XML request document to insert an item in the database.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Generated by the iBSE 2004-06-18T18:20:15Z
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNam
  espace="urn:iwaysoftware:adapter:clarifyCBO:feb2004"
  xmlns:cl="urn:iwaysoftware:adapter:clarifyCBO:feb2004" element
  FormDefault="qualified">
  <xs:element name="sp">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="data"
          type="cl:dataType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="dataType">
    <xs:sequence>
      <xs:element name="cbo"
        type="cl:cboType1" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cboType1">
    <xs:sequence>
      <xs:element name="fields"
        type="cl:fieldsType" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"
      use="required" />
    <xs:attribute name="action" type="xs:string"
      fixed="insert" />
  </xs:complexType>
  <xs:complexType name="fieldsType">
    <xs:sequence>
      <xs:element name="field"
        type="cl:fieldType"
        maxOccurs="unbounded" />
      <xs:element name="cbo"
        type="cl:cboType2" minOccurs="0" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cboType2">
    <xs:sequence>
      <xs:element name="fields"
        type="cl:fieldsType" />
    </xs:sequence>
  </xs:complexType>
```

```
</xs:sequence>
<xs:attribute name="name" type="xs:string"
  use="required" />
<xs:attribute name="relation" type="xs:string"
  use="required" />
</xs:complexType>
<xs:complexType name="fieldType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="name"
        type="xs:string" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>
```

Example: Update Item Request Document

The following is an example and an XML request document to update an item in the database.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Generated by the iBSE 2004-06-18T18:28:47Z
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetName
  space="urn:iwaysoftware:adapter:clarifyCBO:feb2004"
  xmlns:cl="urn:iwaysoftware:adapter:clarifyCBO:feb2004" element
  FormDefault="qualified">
  <xs:element name="sp">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="data"
          type="cl:dataType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="dataType">
    <xs:sequence>
      <xs:element name="cbo"
        type="cl:cboType1" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cboType1">
    <xs:sequence>
      <xs:element name="keys"
        type="cl:keysType" minOccurs="0" />
      <xs:element name="fields"
        type="cl:fieldsType" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"
      use="required" />
    <xs:attribute name="action" type="xs:string"
      fixed="update" />
  </xs:complexType>
  <xs:complexType name="cboType2">
    <xs:sequence>
      <xs:element name="fields"
        type="cl:fieldsType" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"
      use="required" />
    <xs:attribute name="relation" type="xs:string"
      use="required" />
  </xs:complexType>
```

```
<xs:complexType name="fieldsType">
  <xs:sequence>
    <xs:element name="field"
      type="cl:fieldType"
      maxOccurs="unbounded" />
    <xs:element name="cbo"
      type="cl:cboType2" minOccurs="0"
      maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="keysType">
  <xs:sequence>
    <xs:element name="field"
      type="cl:fieldType" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="fieldType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="name"
        type="xs:string" use="required" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>
```

Example: Delete Item Request Document

The following is an XML request document to delete an item in the database.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Generated by the iBSE 2004-06-18T18:36:10Z
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNam
  espace="urn:iwaysoftware:adapter:clarifyCBO:feb2004"
  xmlns:cl="urn:iwaysoftware:adapter:clarifyCBO:feb2004" element
  FormDefault="qualified">
  <xs:element name="sp">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="data"
          type="cl:dataType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="dataType">
    <xs:sequence>
      <xs:element name="cbo"
        type="cl:cboType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cboType">
    <xs:sequence>
      <xs:element name="keys"
        type="cl:keysType" minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"
      use="required" />
    <xs:attribute name="action" type="xs:string"
      fixed="delete" />
  </xs:complexType>
  <xs:complexType name="keysType">
    <xs:sequence>
      <xs:element name="field"
        type="cl:fieldType"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="fieldType">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="name"
          type="xs:string" use="required" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:schema>
```

```
        </xs:simpleContent>  
    </xs:complexType>  
</xs:schema>
```

Response Document Examples

The following are examples of response documents for Get, Insert, Update, and Delete services.

Example: Get Item Response Document

The following is an example of an XML response document in response to a Get request.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Generated by the iBSE 2004-06-18T18:47:48Z
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="urn:iwaysoftware:adapter:clarifyCBO:feb2004"
  xmlns:cl="urn:iwaysoftware:adapter:clarifyCBO:feb2004"
  elementFormDefault="qualified">
  <xs:element name="eda">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="response" type="cl:responseType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="timestamp"
        type="xs:string" />
      <xs:element name="cncresult"
        type="cl:cncresultType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cncresultType">
    <xs:sequence>
      <xs:element name="result"
        type="cl:resultType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="resultType">
    <xs:sequence>
      <xs:element name="resultset"
        type="cl:resultsetType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="resultsetType">
    <xs:sequence>
      <xs:element name="row" type="cl:rowType" maxOccurs="unbounded" />
    </xs:sequence>
  <xs:attribute name="name" type="xs:string"

```

```
        use="required" />
    </xs:complexType>
    <xs:complexType name="rowType">
        <xs:sequence>
            <xs:element name="column"
                type="cl:columnType" minOccurs="0" maxOccurs="unbounded" />
            <xs:element name="resultset"
                type="cl:resultsetType" minOccurs="0" maxOccurs="unbounded" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="columnType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="name"
                    type="xs:string" use="required" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```


Example: Insert Item Response Document

The following is an example of an XML response document in response to an insert request.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
Generated by the iBSE 2004-06-18T18:53:23Z
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace=
  "urn:iwaysoftware:adapter:clarifyCBO:feb2004" xmlns:
  cl="urn:iwaysoftware:adapter:clarifyCBO:feb2004" elementFormDefault=
  "qualified">
  <xs:element name="eda">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="response" type="cl:responseType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="timestamp"
        type="xs:string" />
      <xs:element name="cncresult"
        type="cl:cncresultType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cncresultType">
    <xs:sequence>
      <xs:element name="result"
        type="cl:resultType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="resultType">
    <xs:sequence>
      <xs:element name="edastatus"
        type="xs:string" />
      <xs:element name="resultset"
        type="cl:resultsetType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="resultsetType">
    <xs:sequence>
      <xs:element name="row"
        type="cl:rowType" />
    </xs:sequence>
  <xs:attribute name="name" type="xs:string"
    use="required" />
</xs:complexType>
```

```
<xs:complexType name="rowType">
  <xs:sequence>
    <xs:element name="column"
      type="cl:columnType" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="columnType">
  <xs:simpleContent>
    <xs:extension base="xs:int">
      <xs:attribute name="name"
        fixed="objid" />
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
</xs:schema>
```

Example: Update Item Response Document

The following is an example of an XML response document in response to an update request.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
  Generated by the iBSE 2004-06-18T18:59:38Z
-->
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace=
    "urn:iwaysoftware:adapter:clarifyCB0:feb2004"
  xmlns:cl="urn:iwaysoftware:adapter:clarifyCB0:feb2004"
  elementFormDefault="qualified">
  <xs:element name="eda">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="response" type="cl:responseType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="timestamp"
        type="xs:string" />
      <xs:element name="cncresult"
        type="cl:cncresultType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cncresultType">
    <xs:sequence>
      <xs:element name="result"
        type="cl:resultType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="resultType">
    <xs:sequence>
      <xs:element name="edastatus"
        type="xs:string" />
      <xs:element name="resultset"
        type="cl:resultsetType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="resultsetType">
    <xs:sequence>
      <xs:element name="row" type="cl:rowType" minOccurs="0"
        maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:attribute name="name"
```

```
        type="xs:string" use="required" />
    </xs:complexType>
    <xs:complexType name="rowType">
        <xs:sequence>
            <xs:element name="column"
                type="cl:columnType" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="columnType">
        <xs:simpleContent>
            <xs:extension base="xs:int">
                <xs:attribute name="name"
                    fixed="objid" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

Example: Delete Item Response Document

The following is an example of an XML response document in response to a delete request.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <!--
  Generated by the iBSE 2004-06-18T19:05:19Z
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace=
  "urn:iwaysoftware:adapter:clarifyCBO:feb2004" xmlns:cl=
  "urn:iwaysoftware:adapter:clarifyCBO:feb2004" elementFormDefault=
  "qualified">
  <xs:element name="eda">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="response" type="cl:responseType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="responseType">
    <xs:sequence>
      <xs:element name="timestamp"
        type="xs:string" />
      <xs:element name="cncresult"
        type="cl:cncresultType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="cncresultType">
    <xs:sequence>
      <xs:element name="result"
        type="cl:resultType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="resultType">
    <xs:sequence>
      <xs:element name="edastatus"
        type="xs:string" />
      <xs:element name="column"
        type="cl:columnType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="columnType">
    <xs:simpleContent>
      <xs:extension base="xs:int">
        <xs:attribute name="name"
          fixed="RecordsDeleted" />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

</xs:schema>

Generating a Business Service

You can generate a business service (also known as a Web service) for ClarifyCRM. To generate a business service, you must deploy the adapter in a business services environment using iWay Business Services Engine (iBSE). iBSE exposes functionality as Web services and serves as a gateway to heterogeneous back-end applications and databases.

A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered a “black box” that may require input and delivers a result. Web services can be integrated within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

You can make a Web service available to other services within a host server by generating WSDL (Web Services Description Language) from the Web service.

Ensure that the servlet iBSE is properly configured. For more information on installing and deploying iWay components, see the *iWay Installation and Configuration for BEA WebLogic* manual.

Note: In a J2EE Connector Architecture (JCA) implementation of iWay adapters, Web services are not available. When the adapters are deployed to use the iWay Connector for JCA, the Common Client Interface provides integration services using the iWay adapters. For more information, see the *iWay Installation and Configuration for BEA WebLogic* manual and the *iWay Connector for JCA for BEA WebLogic User's Guide*.

Procedure: How to Create a Business Service

To create a business service:

1. Connect to a ClarifyCRM target as described in *Opening a Connection to ClarifyCRM* on page 2-5.
2. In the left pane of Application Explorer, expand the target node to display its modules. The list includes all modules currently supported by ClarifyCRM.
3. Open the module containing the table for which you want to create a business service.
4. In the right pane, move the pointer over *Operations* and select *Create iWay Business Services*.

The Create Web Service pane opens on the right with two options to create a new or existing service, as shown in the following image.

Create Web Service for get

☒ Create a new service

☐ Use an existing service

Help < Back Next > Cancel

5. Select *Create a new service* and click *Next*.

The next Create a Web Service pane opens displaying descriptive fields for the service, as shown in the following image.

Create Web Service for get

Service Name:

Description:

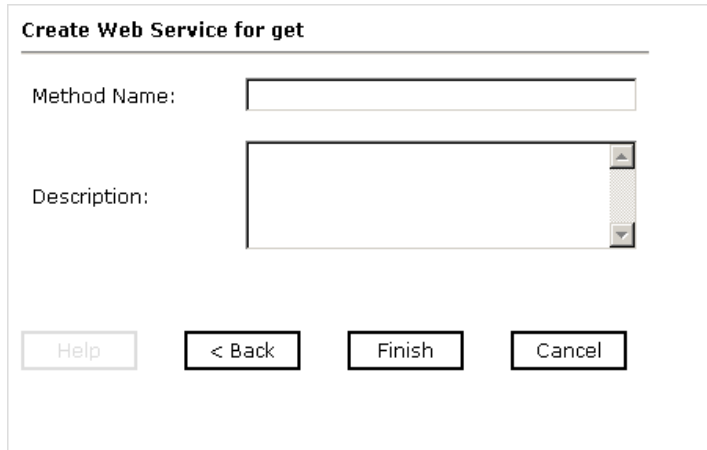
License:

Help < Back Next > Cancel

- a. In the Service Name field, type a descriptive name for the iWay Business Services.
- b. In the Description field, type a brief description of the iWay Business Services.
- c. From the License field list, select a license definition.

6. Click *Next*.

The next Create a Web Service pane opens displaying descriptive fields for the method that the service will use, as shown in the following image.



The image shows a dialog box titled "Create Web Service for get". It contains two input fields: "Method Name:" and "Description:". The "Method Name:" field is a single-line text box. The "Description:" field is a multi-line text box with a vertical scrollbar on the right. At the bottom of the dialog, there are four buttons: "Help", "< Back", "Finish", and "Cancel". The "Help" button is disabled (grayed out), while the other three are active.

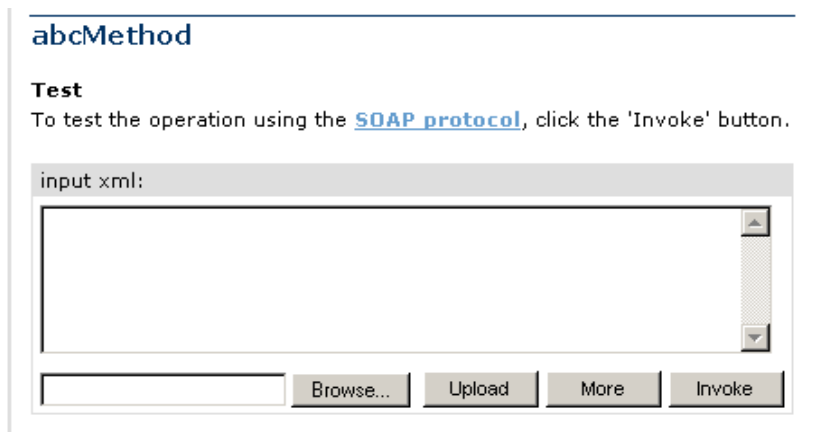
a. In the Method Name field, type a descriptive name for the method.

b. In the Description field, type a brief description of the method.

7. Click *Finish*.

The iWay Business Services tab opens. The left pane lists all the available services, including the one just created.

On the right, a test pane opens. The following image shows an example of this pane for a method named `abcMethod`. The pane provides a text field for inputting XML code, in the right pane under the **Test** headings. This pane provides a text field to paste input XML code or browse to a file that can be uploaded. Below the text field is a browse field, and four action buttons.



Testing a Business Service

iWay provides a test tool to test business services. When you create a new business service, test it to ensure it functions properly.

Procedure: How to Test a Business Service

To test a business service:

1. Access the Test pane.

If you just created the business service, the right pane automatically displays the test options.

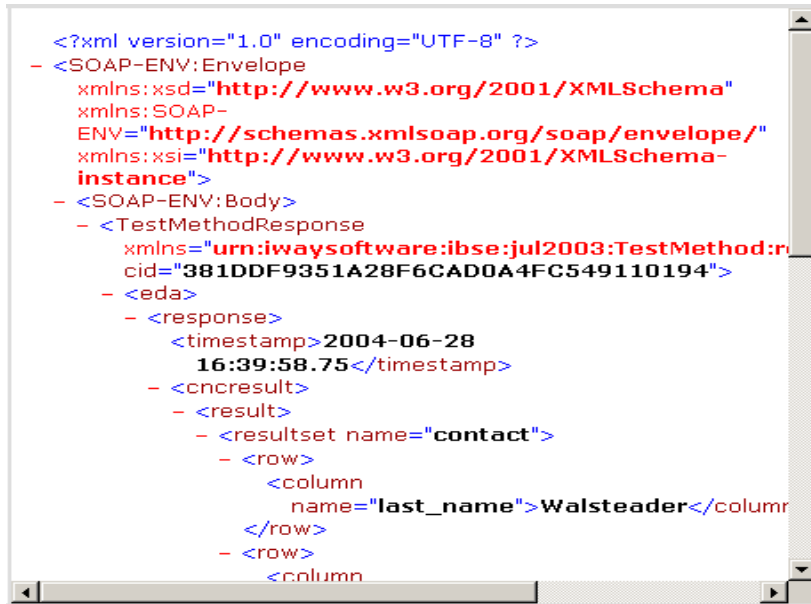
If you want to test an existing business service, click the *iWay Business Services* tab, expand the list of business services in the left pane, and select the business service you want to test. The test options appear in the right pane.

2. Enter a sample XML document that will query the service in the input xml field. Do this using copy and paste, or browsing to a file and uploading the content.

For sample input XML, see *iWay Business Services Input XML for Get* on page 2-37. and *iWay Business Services Input XML for Insert* on page 2-38.

3. Click *Invoke*.

The test results appear in the right pane. An example of test results is shown in the following image.



Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to ClarifyCRM Business Objects. The user name and password values that you provided for ClarifyCRM Business Objects during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service.

```
<SOAP-ENV:Header>
  <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m>Password>String</m>Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>
```

Note: You can remove the following tags from the SOAP header, since they are not required:

```
<m:disposition>String</m:disposition>
<m:language>String</m:language>
```

Example: iWay Business Services Input XML for Get

```
- <sp>
- <data>
- <cbo name="contact_role" action="get">
- <keys>
  <field name="objid">268435527</field>
</keys>
- <fields>
  <field name="role_name" data_type="string" />
  <field name="primary_site" data_type="string" />
- <cbo name="contact" relation="contact_role2contact">
- <fields>
  <field name="first_name" data_type="string" />
  <field name="last_name" data_type="string" />
  <field name="phone" data_type="string" />
</fields>
</cbo>
- <cbo name="site" relation="contact_role2site">
- <fields>
  <field name="site_id" data_type="string" />
  <field name="name" data_type="string" />
- <cbo name="address" relation="cust_primaddr2address">
- <fields>
  <field name="address" data_type="string" />
  <field name="city" data_type="string" />
  <field name="state" data_type="string" />
</fields>
</cbo>
</fields>
</cbo>
</fields>
</data>
</sp>
```

Example: iWay Business Services Input XML for Insert

```
- <sp>
- <data>
  - <cbo name="contact_role" action="insert">
    - <fields>
      <field name="role_name" data_type="string">Default</field>
    - <cbo name="contact" relation="contact_role2contact">
      - <fields>
        <field name="first_name" data_type="string">Harvey</field>
        <field name="last_name" data_type="string">Walsteader</field>
        <field name="phone" data_type="string">5165551212</field>
      </fields>
    </cbo>
  - <cbo name="site" relation="contact_role2site">
    - <fields>
      <field name="site_id" data_type="string">1288589</field>
      <field name="name" data_type="string">Infogain</field>
    - <cbo name="address" relation="cust_primaddr2address">
      - <fields>
        <field name="address" data_type="string">B-17,sec-58</field>
        <field name="city" data_type="string">Noida</field>
        <field name="state" data_type="string">U.P.</field>
      </fields>
    </cbo>
    - <cbo name="address" relation="cust_billaddr2address">
      - <fields>
        <field name="address" data_type="string">B-17,sec-58</field>
        <field name="city" data_type="string">Noida</field>
        <field name="state" data_type="string">U.P.</field>
      </fields>
    </cbo>
    - <cbo name="address" relation="cust_shipaddr2address">
      - <fields>
        <field name="address" data_type="string">B-17,sec-58</field>
        <field name="city" data_type="string">Noida</field>
        <field name="state" data_type="string">U.P.</field>
      </fields>
    </cbo>
  </fields>
</cbo>
</data>
</sp>
```

Deleting a Business Service

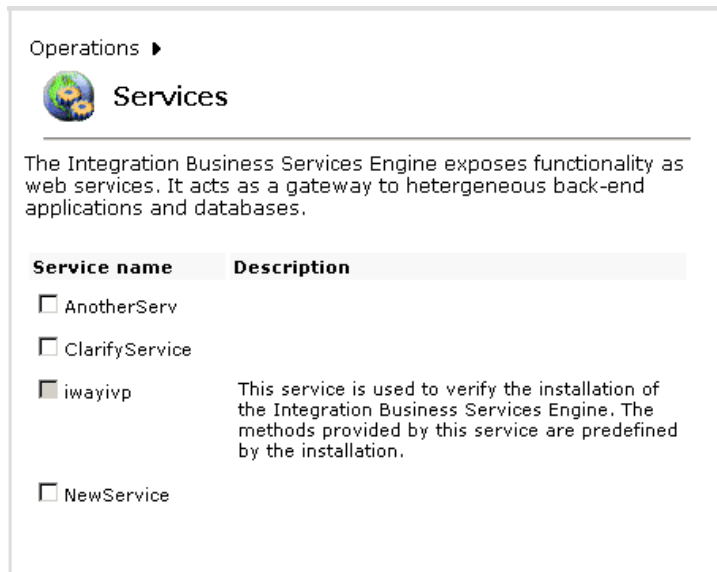
If a business service is no longer required you can delete it from the Services list.

Procedure: How to Delete a Business Service Method

To delete a business service method:

1. Click the *iWay Business Services* tab of Application Explorer.
The left pane contains an expandable list of business services.
2. In the left pane, expand the iWay Business Services node.
3. Select the *Services* node.

The Services pane opens on the right. This pane provides a list of available services with a description of each service. A check box appears to the left of each service. An example of a Services pane is shown in the following image.



4. In the right pane, select the check box next to the service(s) you want to delete.
5. In the right pane, move the pointer over *Operations* and select *Delete*.
A confirmation appears in the right pane.
6. Click *OK* to delete the service(s).
The service no longer appears in the Services pane.

CHAPTER 3

Creating XML Schemas and Web Services With ClarifyCRM ClearBasic

Topics:

- Generating Schemas and Business Services Overview
- Starting iWay Application Explorer
- Opening a Connection to ClarifyCRM
- Closing or Deleting a Target
- Modifying a Target
- Generating a Schema
- Request and Response Documents
- Generating a Business Service

This section describes how to use iWay Servlet Application Explorer to create XML schemas and Web services for ClarifyCRM using ClarifyCRM ClearBasic (CB).

Although this section describes the Java™ servlet implementation of Application Explorer, other implementations provide the same functionality by means of similar graphical user interfaces. For information on running Application Explorer in BEA WebLogic Workshop, see Appendix B, *Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services with ClarifyCRM ClearBasic*.

Generating Schemas and Business Services Overview

The iWay Applications System Adapter for Amdocs ClarifyCRM enables you to create and open connections to the ClarifyCRM database using the iWay Servlet Application Explorer. Through a connection to the database you can generate XML schemas to define request and response documents, and generate a business service.

1. **Start iWay Servlet Application Explorer and open a new or existing connection** to the ClarifyCRM database. See *Opening a Connection to ClarifyCRM* on page 3-4.
2. **Generate XML schemas** that define request and response documents for your SQL statements and stored procedures. See *Generating a Schema* on page 3-11.

You can use the schemas when you create request documents and when you develop logic to process responses.

3. **Create request documents** for each operation against each table and for each stored procedure. See *Request and Response Documents* on page 3-16.

You can use a third-party XML tool to generate a request document from the XML schema. For examples of request and response documents.

4. **Generate a business service** (also known as a Web service) for an SQL statement or stored procedure. See *Generating a Business Service* on page 3-16.

Starting iWay Application Explorer

Before you can use iWay Servlet Application Explorer, you must start BEA WebLogic Server.

Procedure: How to Start Application Explorer

1. Start BEA WebLogic Server, as follows:

On Windows:

Click the Start menu and select *Programs, BEA WebLogic Platform 8.1, User Projects*, then *your domain for iWay*, and then click *Start Server*.

On UNIX or from a command line:

Type the following at the prompt:

```
BEA_HOME\user_projects\domains\DOMAIN_NAME\startWebLogic.cmd
```

where:

`BEA_HOME`

Is the directory where BEA WebLogic is installed.

`DOMAIN_NAME`

Is the domain you are using for iWay.

2. After BEA WebLogic Server is running, enter the following URL in your browser window

<http://hostname:port/iwae/index.html>

where:

[hostname](#)

Is the machine where your application server is installed.

[port](#)

Is the port number on which the application server is listening.

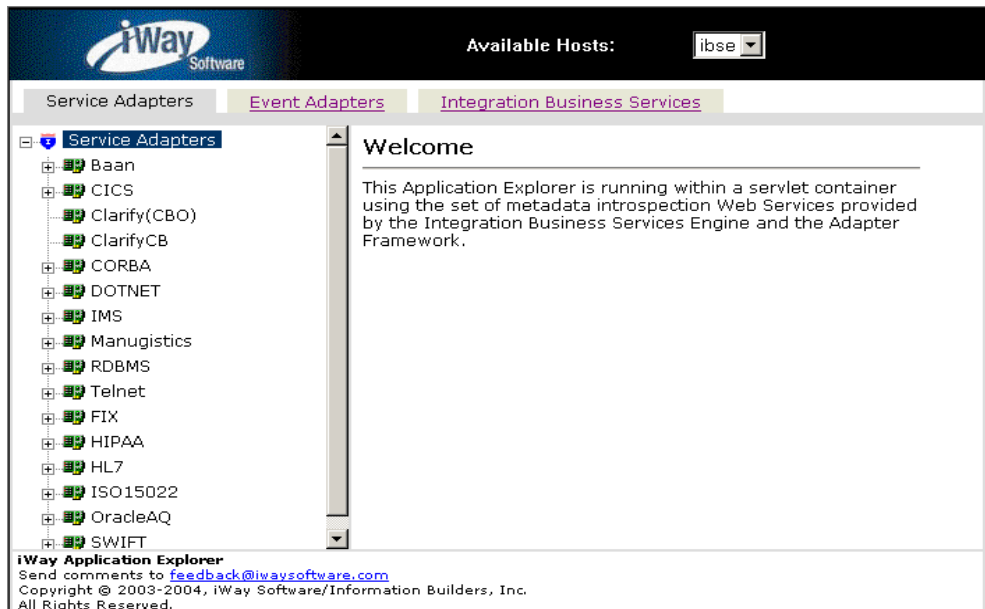
iWay Application Explorer opens, as shown in the following image. The Available Hosts drop-down list appears in the upper-right corner. Three tabs appear near the top of the Application Explorer window. From left to right they are:

Service Adapters, where you create and manage connections to the ClarifyCRM database.

Event Adapters, where you configure ClarifyCRM database event listening.

Integration Business Services, where you generate XML schemas and create and view business services.

The left pane of the window contains an expandable list of adapter nodes, events, or business services, depending on the tab you select. The right pane provides the details of the selected adapter, event, or business service and is the work area where you define and modify adapter functions and services.



Your selection in the Available Hosts drop-down list reflects the instances you can access. The default choices are iWay JCA Connector (JCA) and Servlet iBSE (ibse). For information about adding instances, see the *iWay Installation and Configuration Guide*.

3. Expand the list of adapters by clicking the *Service Adapters* node.

You are ready to add new targets to ClarifyCRM.

Opening a Connection to ClarifyCRM

To browse ClarifyCRM database tables, you must create targets for ClarifyCRM. These targets serve as your connection points. You must establish a connection to ClarifyCRM every time you start iWay Application Explorer or after you disconnect from the system.

You can connect to ClarifyCRM by:

- Creating a new ClarifyCRM CB target.
- Connecting to an existing ClarifyCRM CB target.

The left pane of Application Explorer displays the supported application systems, which are based on the iWay adapters you installed and are licensed to use.

Creating a New ClarifyCRM CB Target

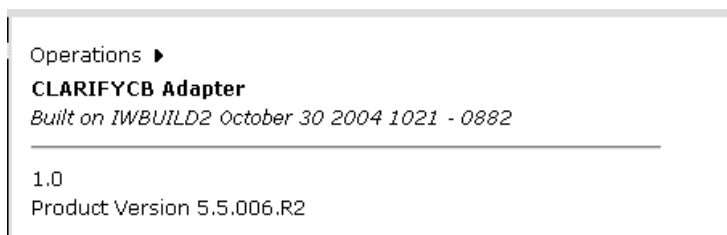
A target serves as the connection point to your Enterprise Information System (EIS) and is automatically saved after you create it. To connect to ClarifyCRM for the first time, you must create a new target from the Service Adapters tab.

Procedure: How to Create a New ClarifyCRM CB Target

To create a new ClarifyCRM CB target:

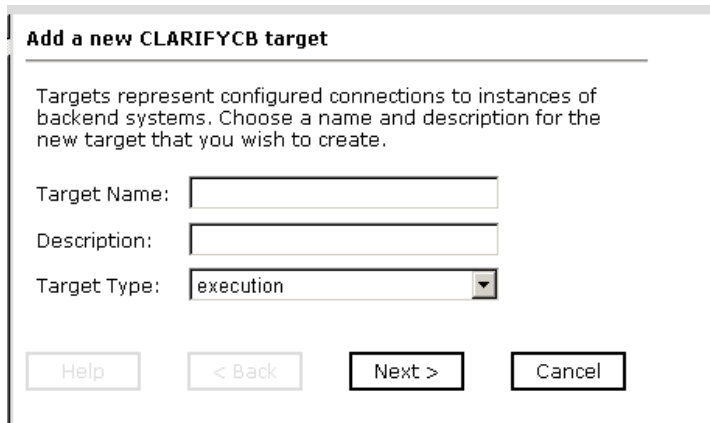
1. In the left pane, expand the *Service Adapters* node and select the *ClarifyCB* node.

Access to the Operations menu for the selected adapter appears on the right, as shown in the following image.



2. Move the pointer over *Operations* and select *Define a new target*.

The Add a new CLARIFYCB target pane opens in the right pane. This pane includes a general statement about targets, two fields for identifying the target, a Help button, and three action buttons, as shown in the following image.



The screenshot shows a dialog box titled "Add a new CLARIFYCB target". Inside the dialog, there is a paragraph of text: "Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create." Below this text are three input fields: "Target Name:" with a text box, "Description:" with a text box, and "Target Type:" with a drop-down menu showing "execution". At the bottom of the dialog are four buttons: "Help", "< Back", "Next >", and "Cancel".

- a. In the Target Name field, type a descriptive name for the target, for example, NewCBTarget.
 - b. In the Description field, type a brief description of the connection.
 - c. From the Target Type drop-down menu, select *execution*.
3. Click *Next*.

The Set connection info pane opens on the right displaying six fields with target connection parameter values, a Help button, and three action buttons. An example of the Set connection info pane is shown in the following image.

Set connection info

Flag to indicate whether to use the TPNOTRAN flag: ☒

Flag to indicate whether to use the Occurrence attribute: ☒

XML element for processing Tuxedo service requests:

JNDI Name for Tuxedo Adapter EJB:

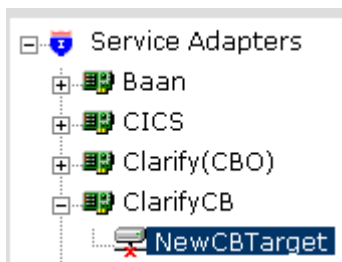
Access mode:

URL to WebLogic server running Tuxedo Connector:

Note: The ClarifyCRM connection parameters are consistent with those found in your ClarifyCRM system. For more information on parameter values that are specific to your ClarifyCRM configuration, consult your ClarifyCRM system administrator.

- a. Select the following check boxes, as required:
 - Flag to indicate whether to use the TPNOTRAN flag
 - Flag to indicate whether to use the Occurrence attribute
 - b. Type an XML element for processing Tuxedo service requests.
 - c. Type a JNDI name for the Tuxedo Adapter EJB.
 - d. Select LOCAL, REMOTE_HTTP, or REMOTE_RMI from the Access mode drop-down list.
 - e. Type a URL in the URL to WebLogic server running Tuxedo Connector field.
4. Click *Finish*.

In the left pane, the new target appears beneath the ClarifyCB node. The following image shows NewCBTarget as the new target under the ClarifyCB node. This target node displays a red X to indicate it is disconnected.



You are ready to connect to the application target you created.

Connecting to an Existing Target

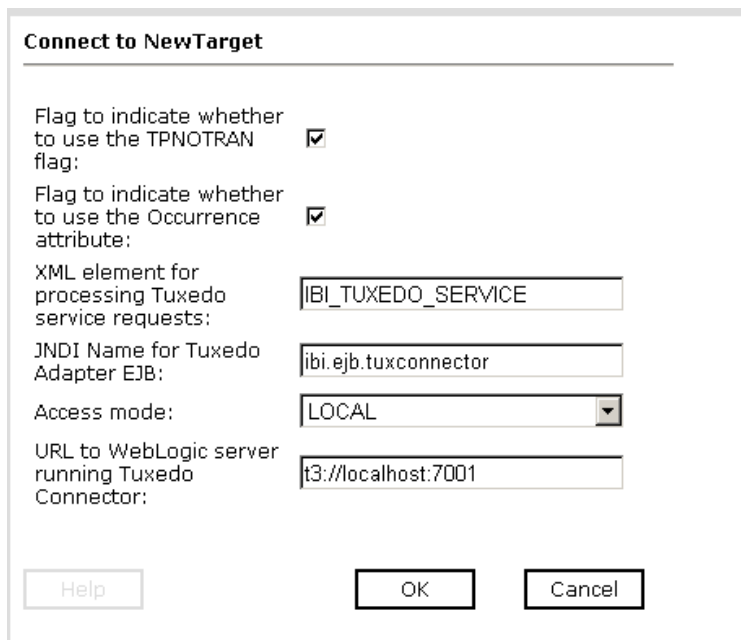
You can use existing targets to connect to instances of ClarifyCRM.

Procedure: How to Connect to an Existing ClarifyCRM CB Target

To connect to an existing ClarifyCRM CB target:

1. In the left pane, expand the *ClarifyCB* node and select the target to which you want to connect.
2. In the right pane, move the pointer over *Operations* and select *Connect*.

The Connect to *targetname* pane opens on the right. This pane includes four fields populated with values for the selected target, followed by a help button and two action buttons, as shown in the following image.



The image shows a dialog box titled "Connect to NewTarget". It contains the following fields and controls:

- Flag to indicate whether to use the TPNOTRAN flag: ☒
- Flag to indicate whether to use the Occurrence attribute: ☒
- XML element for processing Tuxedo service requests:
- JNDI Name for Tuxedo Adapter EJB:
- Access mode:
- URL to WebLogic server running Tuxedo Connector:
- Buttons: Help, OK, Cancel

3. Click OK.

If the parameters are correct and the ClarifyCRM or server component is available, the selected target under the ClarifyCB node displays a plus sign to indicate a connection exists and the Operations menu becomes available on the right, as shown in the following image. If not, a message appears in the right pane.



Closing or Deleting a Target

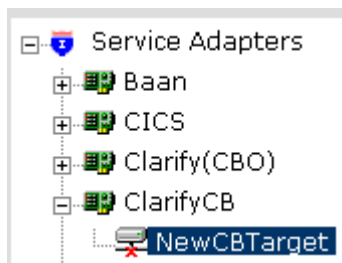
Although you can maintain multiple open connections to different application systems, we recommend that you close connections when they are not in use.

Procedure: How to Disconnect From a Target

To close, or disconnect from, a target:

1. Select the ClarifyCRM target from which you want to disconnect.
2. In the right pane, move the pointer over *Operations* and select *Disconnect*.

Disconnecting from the ClarifyCRM target drops the connection, but the target definition and its node remain visible in the left pane, as shown in the following image.



The node icon displays a red X to reflect that the connection is closed.

Procedure: How to Delete a Target

You can delete a connection whether or not it is closed; if open, it automatically closes before it is deleted.

To delete a target:

1. In the left pane, select the ClarifyCRM target you want to delete.
2. In the right pane, move the pointer over *Operations* and select *Delete*.

A confirmation dialog box opens.

3. Click *OK*.

The target node disappears from the left pane.

Modifying a Target

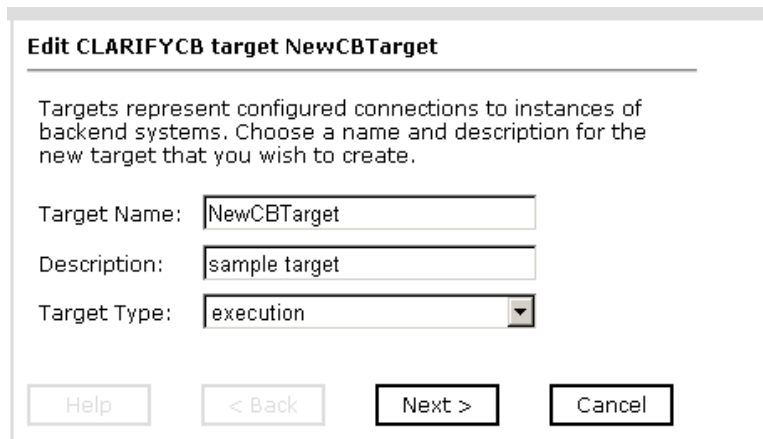
After a target exists in iWay Servlet Application Explorer, you can edit the target name, description, and connection parameters. The Edit function is available only for a disconnected target.

Procedure: How to Edit a Target

To edit a target in Application Explorer:

1. Be sure that the node you want to edit is disconnected. See *Closing or Deleting a Target* on page 3-9.
2. In the left pane, select the target you want to edit.
3. In the right pane, move the pointer over *Operations* and select *Edit*.

The Edit CLARIFYCB target *name* pane opens on the right. This pane provides a general statement about targets followed by three fields that describe the target (Target Name, Description, and Target Type), a help button, and three action buttons. An example of an Edit pane is shown in the following image.



Edit CLARIFYCB target NewCBTarget

Targets represent configured connections to instances of backend systems. Choose a name and description for the new target that you wish to create.

Target Name:

Description:

Target Type:

4. Modify the target name, description, and target type as required.
5. Click *Next*.

The Set connection info pane opens. This pane is described in *Create a New ClarifyCRM CB Target* on page 3-4.

6. Edit the parameters in the Set connection info pane as required.
7. Click *Finish*.

Generating a Schema

XML schemas are used to define service request documents and the corresponding response documents.

When you deploy the adapter in a business services environment, you are not required to generate a schema. For more information, see *Generating a Business Service* on page 3-16.

You can create XML schemas for ClarifyCRM CB:

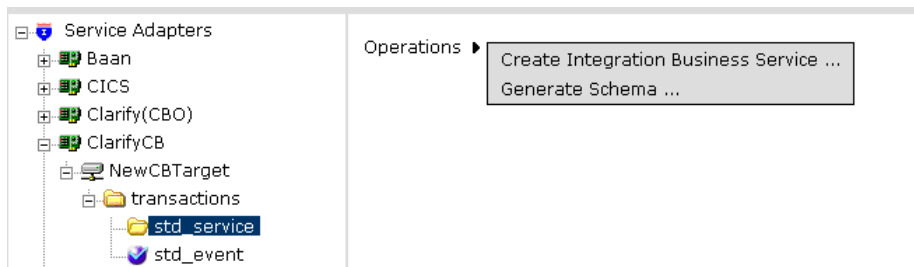
- services
- events

After you create the schema, you can view it or save it and use the schema to create request and response documents.

Procedure: How to Create an XML Schema for a Service

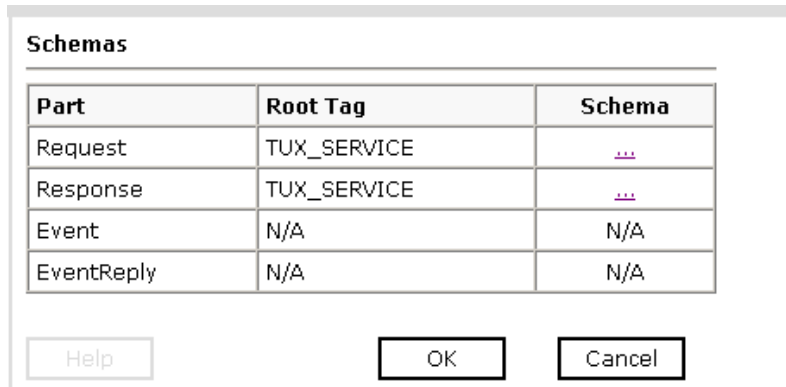
To create a schema for a ClarifyCRM table:

1. Connect to the ClarifyCB target.
2. In the left pane, expand the ClarifyCB target node and the *transactions* node.
3. Under the transactions node, select the *std_service* node.



4. In the right pane, move the pointer over *Operations* and select *Generate Schema*.

Application Explorer builds the schema and displays a Schemas table, a help button, and two action buttons in the right pane. The table provides three columns labeled Part, Root Tag, and Schema. The Schema column provides hyperlinks to the different schemas. The following image is an example of a service schema table.



Schemas

Part	Root Tag	Schema
Request	TUX_SERVICE	...
Response	TUX_SERVICE	...
Event	N/A	N/A
EventReply	N/A	N/A

Help OK Cancel

5. In the Schema column of the table, click the browse hyperlink ([...](#)) associated with the type of schema (request or response) you want to view.

The XML schema appears in the right pane, as shown in the following image.



6. To save the schema to a directory other than the default (see *Schema Location* on page 3-15 for the default location):
 - a. Right-click the right pane.
 - b. Select *View Source*.

The source XML schema opens in a text editor.

- c. Save the schema or copy and paste it to another location.

Use the *Back* button on the Web browser to return to the Schemas table.

To return to the Operations menu from the Schemas table, click *OK*.

Procedure: How to Create an XML Schema for an Event

To create a schema for a ClarifyCRM table:

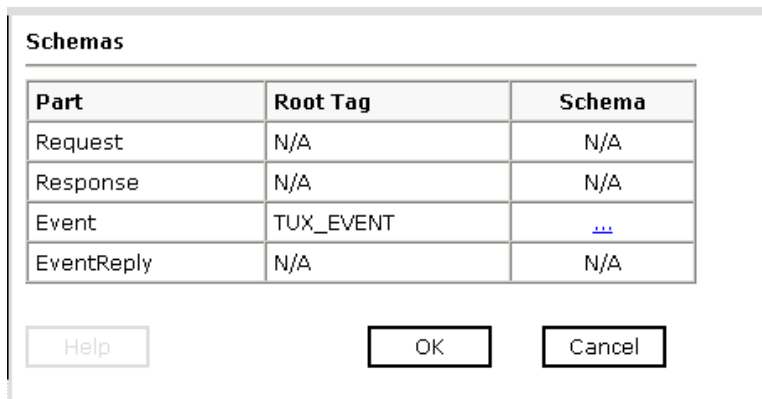
1. Connect to the ClarifyCB target.
2. In the left pane, expand the ClarifyCB target node and the *transactions* node.

3. Under the transactions node, select the *std_event* node.



4. In the right pane, move the pointer over *Operations* and select *Generate Schema*.

Application Explorer builds the schema and displays a Schemas table, a help button, and two action buttons in the right pane. The table provides three columns labeled Part, Root Tag, and Schema. The Schema column provides a hyperlink to the event schema. The following image is an example of an event schema table.



5. In the Schema column of the table, click the hyperlink ([...](#)) to view the event schema.
The XML schema appears in the right pane.
6. To save the schema to a directory other than the default (see *Schema Location* on page 3-15 for the default location):
 - a. Right-click the right pane.
 - b. Select *View Source*.

The source XML schema opens in a text editor.

- c. Save the schema or copy and paste it to another location.

Use the *Back* button on the Web browser to return to the Schemas table.

To return to the Operations menu from the Schemas table, click *OK*.

Schema Location

Application Explorer stores the schemas it creates in subdirectories under the iWay home directory of the machine where it is installed. The exact location of the schemas differs depending on whether you deploy Application Explorer with an iBSE or a JCA configuration.

- When using the adapter with an iBSE configuration, the schemas are stored under a \schemas subdirectory of the iWay home directory, for example,

`C:\Program Files\iway55\bea\ibse\wsdl\schemas\service\ClarifyCB\NewCBTarget`

where:

`NewCBTarget`

Is the name of the connection to the ClarifyCRM system as defined in Application Explorer. Under this directory, Application Explorer creates subdirectories containing schemas.

- When using the adapter with a JCA configuration, the schemas are stored under a \schemas subdirectory of the iWay home directory, for example,

`C:\Program Files\iWay55\config\base\schemas\ClarifyCB\NewCBTarget`

where:

`NewCBTarget`

Is the name of the connection to the ClarifyCRM system as defined in Application Explorer. Application Explorer stores the schemas in this directory.

Request and Response Documents

You can generate request and response document schemas using Application Explorer and then use a third-party XML tool to generate a request document instance from the schema.

For services, the request and response documents are for:

- tables
- concurrent programs

When processing a request document, the adapter performs all table insertions before it runs any concurrent programs (using the `submit_request` element). If a table insertion fails, all insertions are rolled back; otherwise, they are committed, regardless of the concurrent program result.

Generating a Business Service

You can generate a business service (also known as a Web service) for a ClarifyCRM table. To generate a business service, you must deploy the adapter in a business services environment using Integration Business Services Engine (iBSE). iBSE exposes functionality as Web services and serves as a gateway to heterogeneous back-end applications and databases.

A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered a “black box” that may require input and delivers a result. Web services can be integrated within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

You can make a Web service available to other services within a host server by generating WSDL (Web Services Description Language) from the Web service.

Ensure that the servlet iBSE is properly configured. For more information on installing and deploying iWay components, see the *iWay Installation and Configuration for BEA WebLogic* manual.

Note: In a J2EE Connector Architecture (JCA) implementation of iWay adapters, Web services are not available. When the adapters are deployed to use the iWay Connector for JCA, the Common Client Interface provides integration services using the iWay adapters. For more information, see the *iWay Installation and Configuration for BEA WebLogic* manual and the *iWay Connector for JCA for BEA WebLogic User's Guide*.

Procedure: How to Create a Business Service

To create a business service:

1. Connect to a ClarifyCB target.
2. In the left pane, expand the target node and the *transactions* node.
3. Under the transactions node, select the *std_service* node.
4. In the right pane, move the pointer over *Operations* and select *Create Integration Business Service*.

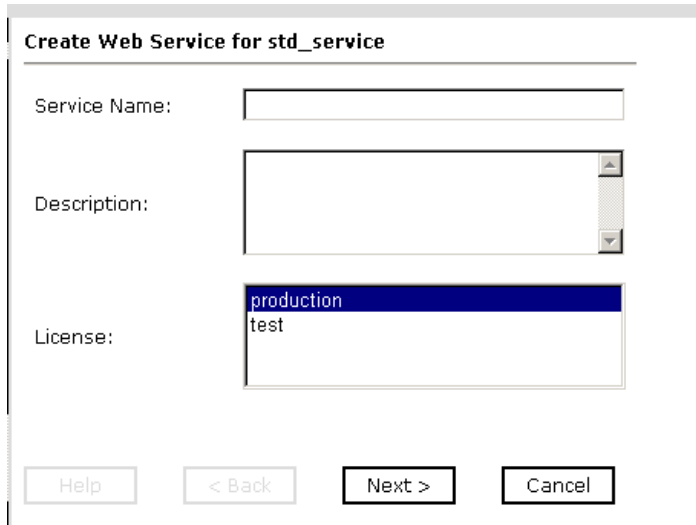
The Create Web Service for *std_service* pane opens. This pane provides two option buttons that enable you to create a new service or use an existing service, a help button, and three action buttons, as shown in the following image.



The screenshot shows a dialog box titled "Create Web Service for std_service". Inside the dialog, there are two radio button options: "Create a new service" (which is selected) and "Use an existing service". At the bottom of the dialog, there are four buttons: "Help", "< Back", "Next >", and "Cancel". The "Next >" button is highlighted with a black border.

5. Select *Create a new service* and click *Next*.

The Create Web Service for std_service pane opens on the right, as shown in the following image. This pane provides three field to identify the new business service, a help button, and three action buttons.



The image shows a dialog box titled "Create Web Service for std_service". It contains three input fields: "Service Name:" with a single-line text box, "Description:" with a multi-line text box, and "License:" with a list box. The list box contains two items: "production" (highlighted in blue) and "test". At the bottom of the dialog are four buttons: "Help", "< Back", "Next >", and "Cancel".

- a. In the Service Name field, type a descriptive name for the Integration Business Service.
 - b. In the Description field, type a brief description of the Integration Business Service.
 - c. From the License field list, select a license definition.
6. Click *Next*.

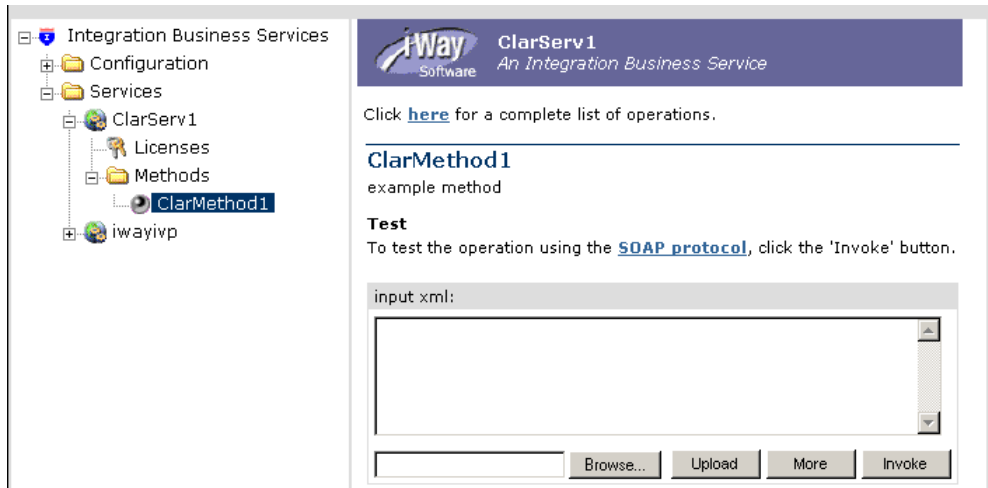
The right pane displays the next Create Web Service for `std_service` pane that prompts you for information about the method of the service, as shown in the following image. It includes two fields to identify the method, a help button, and three action buttons.



The image shows a dialog box titled "Create Web Service for std_service". It contains two input fields: "Method Name:" with a single-line text box, and "Description:" with a multi-line text box. Below the input fields are four buttons: "Help", "< Back", "Finish", and "Cancel". The "Finish" button is highlighted with a black border.

- a. In the Method Name field, type a descriptive name for the method.
 - b. In the Description field, type a brief description of the method.
7. Click *Finish*.

The Integration Business Services tab opens. The new business service appears on the left and a test pane for the service method appears on the right, as shown in the following image. The test pane provides a text field in which to paste the XML input or browse to a file that can be uploaded. Below the text field is the browse field and three action buttons.



Testing a Business Service

iWay provides a test tool to test business services. When you create a new business service, test it to ensure it functions properly.

Procedure: How to Test a Business Service

To test a business service:

1. Access the Test pane.

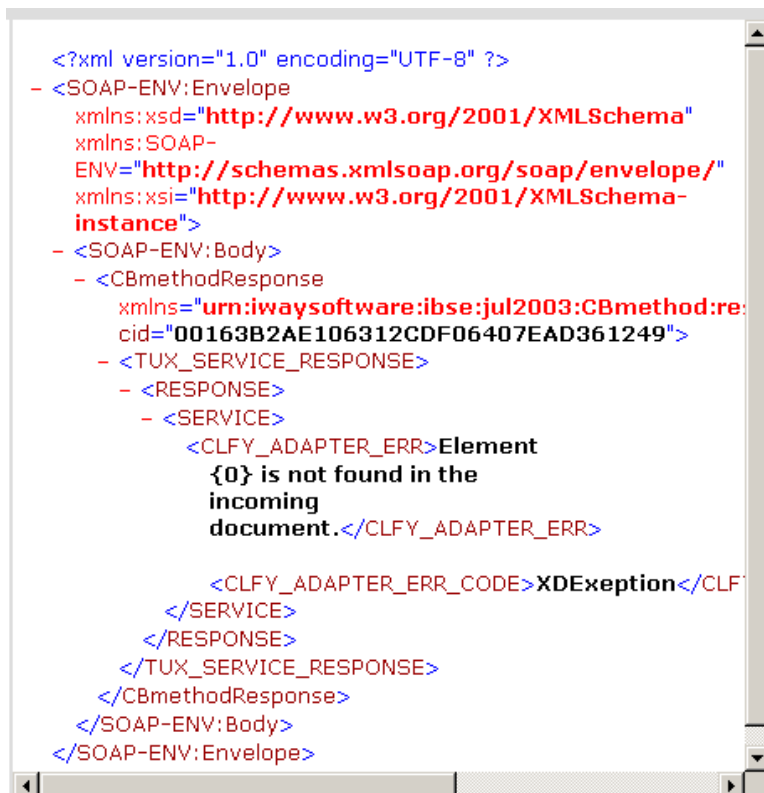
If you just created the business service, the right pane automatically displays the test options.

If you want to test an existing business service, click the *Integration Business Services* tab, expand the list of business services in the left pane, and select the business service you want to test. The test options appear in the right pane.

2. Enter a sample XML document that will query the service in the input xml field. Do this using copy and paste, or browsing to a file and uploading the content.

3. Click *Invoke*.

The test results appear in the right pane. An example of test results appear in the following image.



Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to ClarifyCRM ClearBasic. The user name and password values that you provided for ClarifyCRM ClearBasic during target creation using Application Explorer are overwritten for this Web service request.

The following is a sample SOAP header that is included in the WSDL file for a Web service.

```
<SOAP-ENV:Header>
  <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m>Password>String</m>Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>
```

Note: You can remove the following tags from the SOAP header, since they are not required:

```
<m:disposition>String</m:disposition>
<m:language>String</m:language>
```

Deleting a Business Service

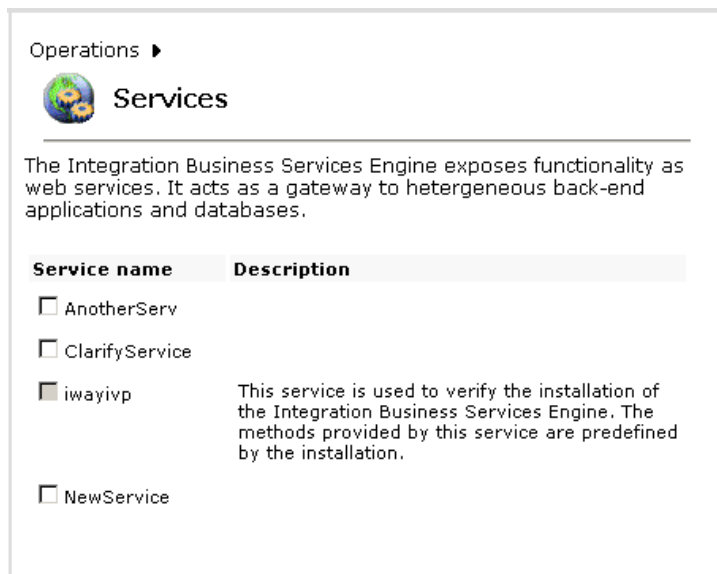
If a business service is no longer required you can delete it from the Services list.

Procedure: How to Delete a Business Service Method

To delete a business service method:

1. Click the *Integration Business Services* tab of Application Explorer.
2. In the left pane, expand the *Integration Business Services* node.
3. Select the *Services* node.

The left pane of the window contains an expandable list of business services. The right pane displays a list of available services with a description of each service. A check box appears next to each service. The following image is an example of the Services pane.



4. In the right pane, select the check box next to the service(s) you want to delete.
5. In the right pane, move the pointer over *Operations* and select *Delete*.

A confirmation appears in the right pane.

6. Click *OK* to delete the service(s).

The service no longer appears in the service name list.

CHAPTER 4

Listening for Events Using ClarifyCRM Business Objects

Topics:

- Understanding iWay Event Functionality
- Creating an Event Port
- Editing or Deleting an Event Port
- Creating a Channel
- Editing or Deleting a Channel

This section describes how to use the iWay Applications System Adapter for Amdocs ClarifyCRM deployed to an application server, such as BEA WebLogic Server, to listen for events in a ClarifyCRM database table using ClarifyCRM Business Objects (CBOs). Several listening techniques are available, enabling you to choose the technique that best suits your requirements.

Although this section describes the Java™ servlet implementation of Application Explorer, other implementations provide the same functionality by means of similar graphical user interfaces. For information on running Application Explorer in BEA WebLogic Workshop, see Appendix C, *Using Application Explorer in BEA WebLogic Workshop for Event Handling with CBOs*.

Understanding iWay Event Functionality

Events are generated as a result of ClarifyCRM activity. You can use these events to trigger an action in your application.

After you create a connection to your application system, you can add events using iWay Servlet Application Explorer. To create an iWay event, you must create a port and a channel.

- **Port**

A port is a logical definition of how to direct event data. It includes information on where to store the information and where to place it after it is processed. It associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption.

For example, you can use a Java Message Service (JMS) protocol to route the result of polling a table to a JMS queue hosted by BEA WebLogic Server, a J2EE application server. For more information, see *Creating an Event Port* on page 4-3.

- **Channel**

A channel defines the listening capability that detects events in the target Enterprise Information System (EIS) or database. It represents configured connections to particular instances of back-end or other types of systems. A channel binds one or more event ports to a particular listener managed by an adapter. For more information, see *Creating a Channel* on page 4-17.

There are several techniques that you can employ when listening for ClarifyCRM events, depending upon your requirements.

Creating an Event Port

You can create an event port in Application Explorer from either the Service Adapters tab or the Event Adapters tab. There are slight variations in the two methods, but both result in a new port, therefore, you can use either of the following procedures.

- *Create an Event Port from the Service Adapters Tab* on page 4-3
- *Create an Event Port from the Event Adapters Tab* on page 4-6

Several dispositions are available when defining a port. When you use Application Explorer with an iWay Business Services Engine (iBSE) implementation, the following port dispositions are available:

- File
- iBSE
- MSMQ
- JMS
- SOAP
- HTTP
- MQ Series
- MAIL

Note: The MAIL disposition option will be supported in a future release.

When you use Application Explorer with a JCA implementation, the following port dispositions are available:

- File
- JMS
- MQ Series
- HTTP

Additional information about each port disposition appears in this section after the procedures to create a port.

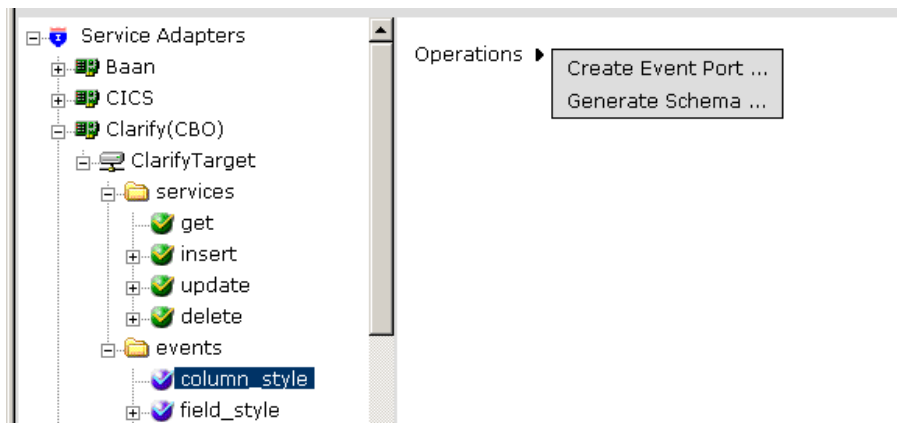
Procedure: How to Create an Event Port from the Service Adapters Tab

To create an event port from the Service Adapters tab:

1. Click the *Service Adapters* tab.
2. In the left pane, expand the *Service Adapters* node, the *Clarify(CBO)* node, and the target node for the event.

3. Expand the *events* node and select an event component.

The following image shows the expanded events node in the left pane, and the available options in the Operations menu in the right pane.



4. In the right pane, move the pointer over *Operations* and select *Create Event Port*.

The Create Event Port pane opens on the right, as shown in the following image. This pane includes a general statement about event ports and parameter fields to identify the new event port.

Create Event Port

The Event Port is used to route events generated by external systems. An event port consists of an event schema bound to an event disposition and assigned to an event channel. Channels are configured by transport and you can assign multiple ports to each channel.

Event Port Name:

Event Port Description:

Disposition Protocol:

FILE

Help

< Back

Next >

Cancel

- a. In the Event Port Name field, type a name for the event port.
 - b. In the Event Port Description field, type a brief description of the port.
 - c. From the Disposition Protocol drop-down list, select a protocol.
5. Click *Next*.

The Specify (selected protocol) Disposition pane opens on the right. This pane includes a brief description of the disposition type selected in the previous step and the default URL to the document that will contain event information. An example of the Specify FILE Disposition pane is shown in the following image.

Specify FILE Disposition

Disposition type File uses an iWay file url to specify the destination filename or directory in which the event document is stored. During run-time, the destination file name may need to be indexed to avoid overwriting. It supports an optional errorTo port or url.

Disposition Url:

6. In the Disposition URL field, keep the default URL or enter a new URL to the event document. To find the URL format for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page 4-7.
7. Click *Finish*.

The Event Adapters tab opens. The left pane shows the new event port under the target *ports* node and the right pane shows summary information associated with the new event port. To see an image of a port node for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page 4-7.

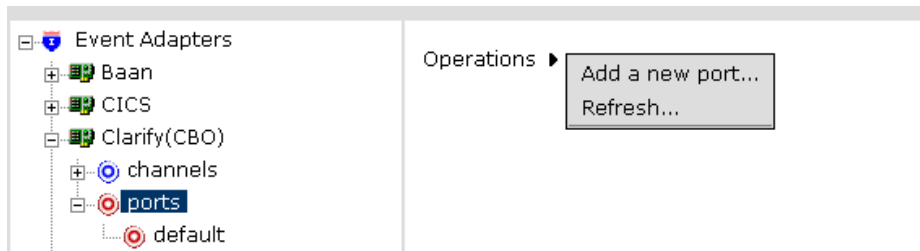
You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 4-17.

Procedure: How to Create an Event Port from the Event Adapters Tab

To create an event port from the Event Adapters tab:

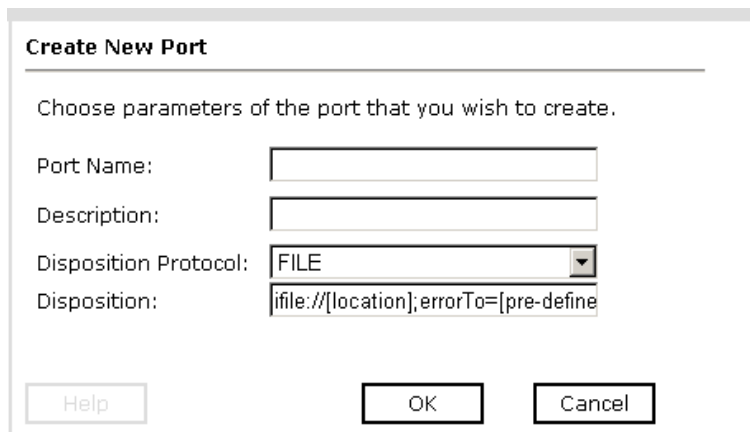
1. Click the Event Adapters tab.
2. In the left pane, expand the Event Adapters node and the *Clarify(CBO)* node.
3. Select the *ports* node.

The following image shows an expanded Event Adapters node in the left pane, and the available option in the Operations menu in the right pane.



4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create Event Port pane opens on the right, as shown in the following image. This pane provides event port parameter fields to identify the new event port.



- a. In the Port Name field, type a name for the event port.
- b. In the Description field, type a brief description of the port.
- c. From the Disposition Protocol drop-down list, select a protocol.

- d. In the Disposition field, specify a URL to the destination file that will hold the event data. To find the URL format for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page 4-7.

5. Click OK.

The left pane shows the new event port under the target ports node. The right pane shows summary information associated with the new event port. To see an image of a port node for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page 4-7.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 4-17.

FILE Disposition URL Format

The format of the URL to the FILE disposition event document is:

- **When pointing Application Explorer to an iBSE deployment**, specify the destination file using the following format:

```
ifile://location[;errorTo=errorDest]
```

- **When pointing Application Explorer to a JCA deployment**, specify the destination file using the following format:

```
location
```

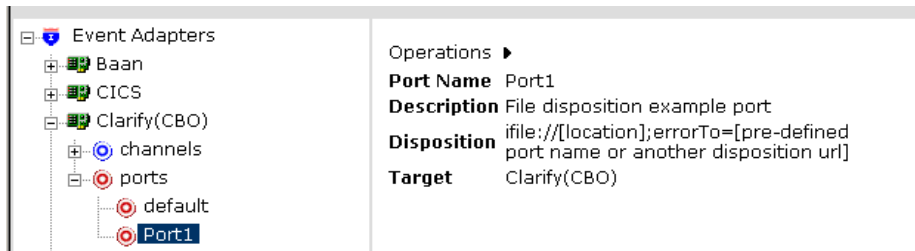
The following table lists and describes the FILE disposition parameters.

Parameter	Description
<code>location</code>	Full directory path and file name to which the data is written.
<code>errorDest</code>	Location to which error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

For example:

```
ifile://c:\temp\ClarifyEvent.txt;errorTo=ifile://c:\temp\error
```

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears on the right. The following image shows a listing and description for a FILE event port.



iBSE Disposition URL Format

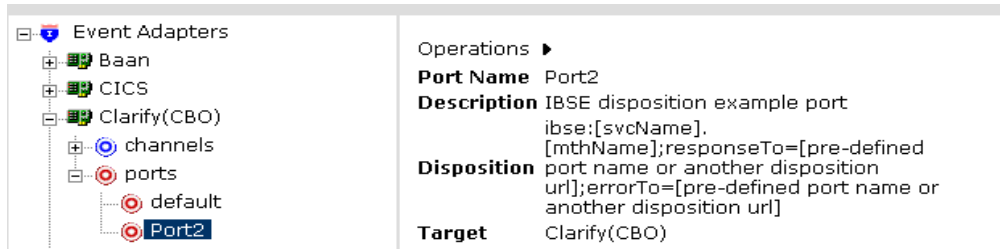
The format of the URL to the iBSE disposition event document is:

`ibse: / svcName.methName[; responseTo=respDest] [; errorTo=errorDest]`

The following table lists and describes the iBSE disposition parameters.

Parameter	Description
svcName	Name of the service created with iBSE.
methName	Name of the method created for the Web service.
respDest	Location where responses to the Web service are posted. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.
errorDest	Location where error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears on the right. The following image shows a listing and description for an iBSE event port.



MSMQ Disposition URL Format

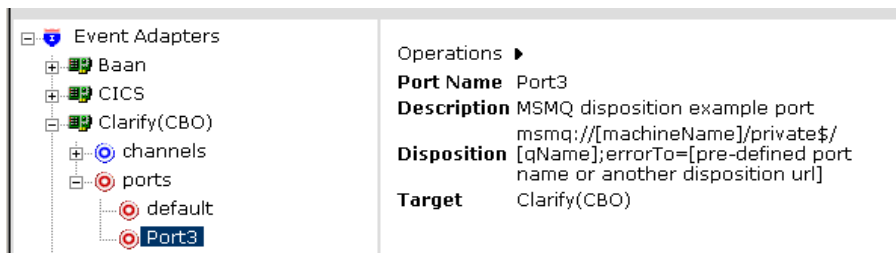
The format of the URL to the MSMQ disposition event document is:

`msmq: /host/queueType/queueName[;errorTo=errorDest]`

The following table lists and describes the MSMQ disposition parameters.

Parameter	Description
host	Name of the host on which the Microsoft Queuing system runs.
queueType	Type of queue. For private queues, enter <i>Private\$</i> . Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
queueName	Name of the queue where messages are placed.
errorDest	Location where error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears on the right. The following image shows a listing and description for an MSMQ event port.



JMS Disposition URL Format

The format of the URL to the JMS disposition event document is:

- When pointing Application Explorer to an **ibSE** deployment, specify the JMS destination file using the following URL format:

```
jmsq:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory;
user=userID;password=pass[;errorTo=errorDest]
```

- When pointing Application Explorer to a **JCA** deployment, specify the JMS destination file using the following URL format:

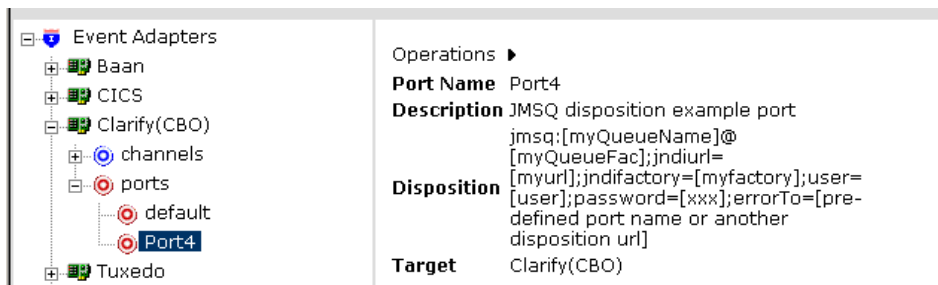
```
jms:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory
```

The following table lists and describes the JMS disposition parameters.

Parameter	Description
queue	Name of a queue to which events are emitted.
conn_factory	The connection factory, a resource that contains information about the JMS Server. The BEA WebLogic connection factory is: <code>javax.jms.QueueConnectionFactory</code>

Parameter	Description
jndi_url	<p>The URL to use to contact the JNDI provider. The syntax of this URL depends on which JNDI provider is being used. This value corresponds to the standard JNDI property:</p> <pre>java.naming.provider.url</pre> <p>For BEA WebLogic Server, this is:</p> <pre>t3://host:port</pre> <p>where:</p> <p><i>host</i></p> <p>Is the machine name where BEA WebLogic Server is installed.</p> <p><i>port</i></p> <p>Is the port on which BEA WebLogic Server is listening. The default port if not changed at installation is 7001.</p>
jndi_factory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For BEA WebLogic Server, the BEA WebLogic factory is weblogic.jndi.WLInitialContextFactory.
userID	User ID associated with this queue.
pass	Password associated with the user ID.
errorDest	<p>Location to which error logs are sent. Optional.</p> <p>A predefined port name or another disposition URL. The URL must be complete, including the protocol.</p>

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears on the right. The following image shows a listing and description for an JMS event port.



SOAP Disposition URL Format

The format of the URL to the SOAP disposition event document is:

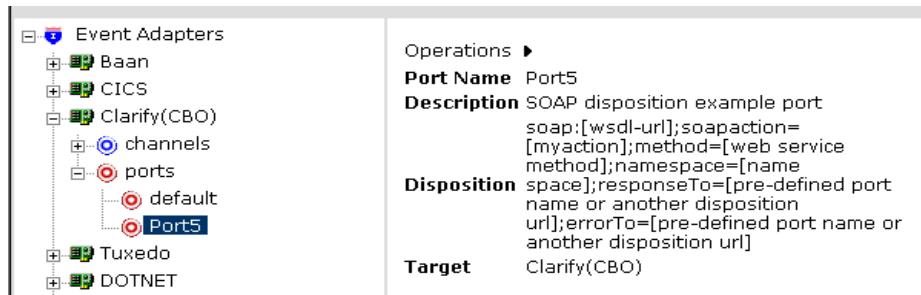
```
soap:[wsdl-url];soapaction=[myaction];method=[web service
method];namespace=[namespace];responseTo=[pre-defined port name or
another disposition URL];errorTo=[pre-defined port name or another
disposition url]
```

The following table lists and describes the SOAP disposition parameters.

Parameter	Description
wsdl-url	<p>The URL to the WSDL file that is required to create the SOAP message. For example,</p> <pre>http://localhost:7001/ibse/IBSEServlet/test/ webservice.ibs?wsdl</pre> <p>where:</p> <pre>webservice</pre> <p>Is the name of the Web service you created using Application Explorer.</p> <p>To find this value, navigate to the iWay Business Services tab, expand the <i>Services</i> node, select the service you created, and click the <i>Service Description</i> option on the right. The WSDL URL appears in the Address field of the window that opens.</p> <p>You can also open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.</p>
soapaction	The method that will be called by the SOAP disposition. This value can be found in the WSDL file.
method	The Web service method you are using. This value is found in the WSDL file.
namespace	The XML namespace you are using. This value is found in the WSDL file.
responseTo	The location to which responses are posted. A predefined port name or another full URL. Optional. The URL must be complete, including the protocol.

Parameter	Description
errorTo	The location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears on the right. The following image shows a listing and description for an SOAP event port.



HTTP Disposition URL Format

The format of the URL to the HTTP disposition event document is:

- When pointing Application Explorer to an **IBSE** deployment, specify the HTTP destination file using the following format:

`ihttp://url;responseTo=respDest`

- When pointing Application Explorer to a **JCA** deployment, specify the HTTP destination file using the following format:

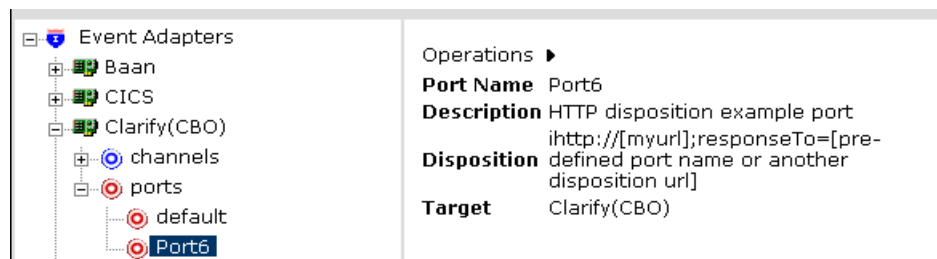
`http://host:port/uri`

The following table describes the HTTP disposition parameters.

Parameter	Description
url	The URL target for the post operation.
respDest	Location where responses are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
host	Name of the machine on which the Web server resides.

Parameter	Description
port	Port number on which the Web server is listening.
uri	Universal resource identifier that completes the url specification.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears on the right. The following image shows a listing and description for an HTTP event port.



MQSeries Disposition URL Format

The format of the URL to the MQSeries disposition event document is:

- When pointing Application Explorer to an **IBSE** deployment, specify the MQSeries destination file using the following format:

```
mqseries:/qManager/qName;host=hostName;port=portNum;
channel=chanName[;errorTo=errorDest]
```

- When pointing Application Explorer to a **JCA** deployment, use the following format:

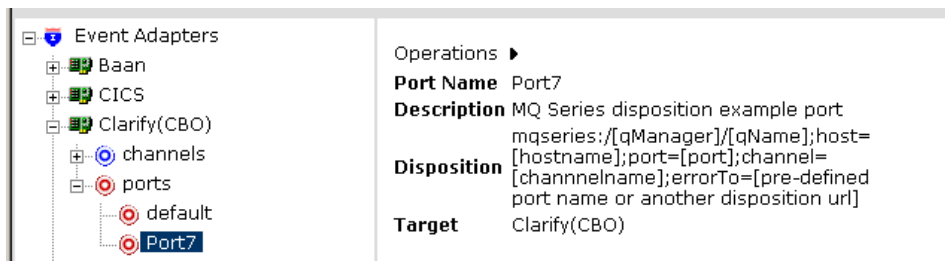
```
mq:qmanager@respqueue;host=;port=;channel=
```

The following table lists and describes the MQSeries disposition parameters.

Parameter	Description
qManager	Name of the queue manager to which the server must connect.
qName or respqueue	Name of the queue where messages are placed.
host	Name of the machine on which MQSeries resides (MQ client only).
port	Port number for connecting to an MQ server (queue manager). MQ client only.

Parameter	Description
chan	The case-sensitive name of the channel that connects with the remote MQ server (queue manager). MQ client only. The default MQSeries channel name is SYSTEM.DEF.SVRCONN.
errorTo	Location where error documents are sent. Can be a predefined port name or another full URL. Optional.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears on the right. The following image shows a listing and description for an MQSeries event port.



Editing or Deleting an Event Port

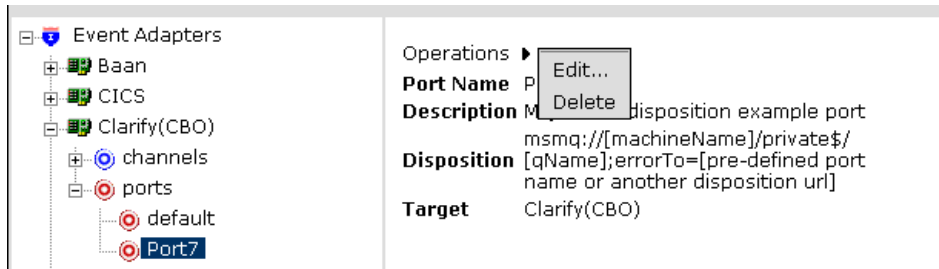
The following procedures describe how to edit and delete an event port using Application Explorer.

Procedure: How to Edit an Event Port

To edit an existing event port using Application Explorer:

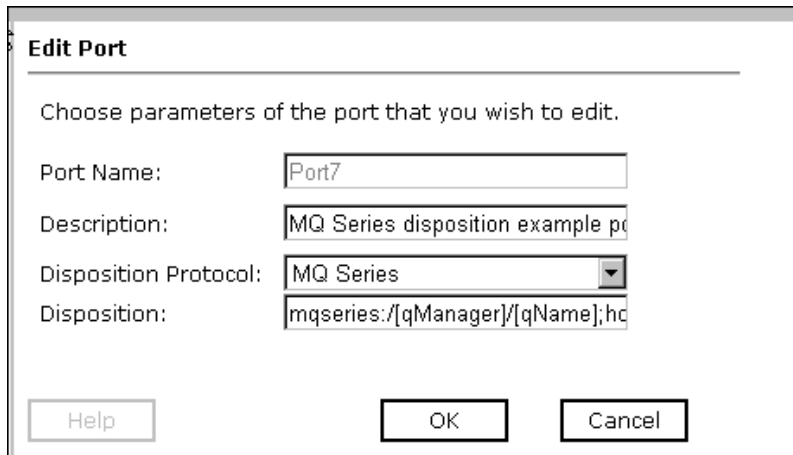
1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *Clarify(CBO)* node, and the *ports* node.
3. Select the event port you want to edit.

The following image shows an expanded Event Adapters node in the left pane and the available Edit and Delete options are shown in the Operations menu in the right pane.



4. In the right pane, move the pointer over *Operations* and select *Edit*.

The Edit Port pane opens on the right. This pane displays the port parameter fields that can be changed. An example of this pane is shown in the following image.



You cannot change the name of the port in the Edit Port pane.

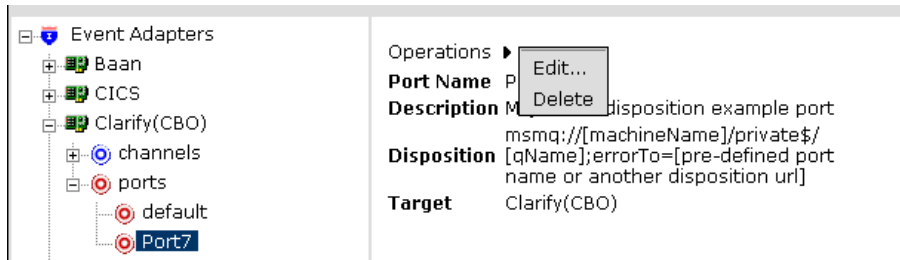
5. Make the required changes to the Description, Disposition Protocol, or Disposition fields, and click *OK*.

Procedure: How to Delete an Event Port

To delete an existing event port using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *Clarify(CBO)* node, and the *ports* node.
3. Select the event port you want to delete.

The following image shows an expanded Event Adapters node on the left and the Operations menu on the right.



4. In the right pane, move the pointer over *Operations* and select *Delete*.

A confirmation dialog box opens.

5. Click *OK*.

The event port disappears from the list in the left pane.

Creating a Channel

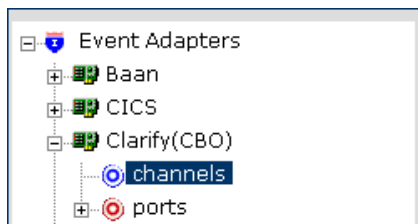
The following procedure describes how to create a channel for a ClarifyCRM event. All defined event ports must be associated with a channel.

Procedure: How to Create a Channel

To create a channel using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node and the *Clarify(CBO)* node.

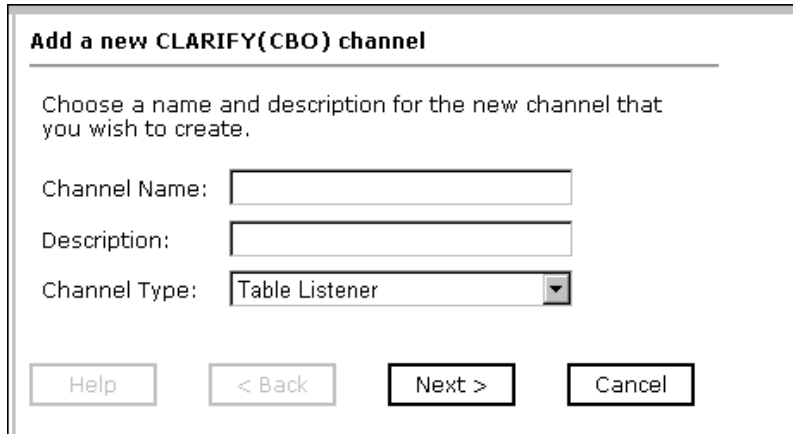
The following image shows the channels node highlighted in the left pane.



3. Select the *channels* node.

4. In the right pane, move the pointer over *Operations* and select *Add a new channel*.

The Add a new CLARIFY(CBO) channel pane opens on the right. This pane, shown in the following image, provides three parameter fields to define a channel.



The image shows a dialog box titled "Add a new CLARIFY(CBO) channel". Inside the dialog, there is a text instruction: "Choose a name and description for the new channel that you wish to create." Below this instruction are three input fields: "Channel Name:" followed by a text box, "Description:" followed by a text box, and "Channel Type:" followed by a drop-down menu currently showing "Table Listener". At the bottom of the dialog are four buttons: "Help", "< Back", "Next >", and "Cancel".

- a. In the Channel Name field, enter a name for the channel.
 - b. In the Description field, enter a brief description of the channel.
 - c. From the Channel Type drop-down list, accept the default channel type, *Table Listener*.
5. Click *Next*.

The Edit channels pane opens on the right. This pane, shown in the following image, provides tabs related to the available listeners. Each tab displays the parameter fields related to that specific listener.

Note: Only the Oracle Parameters and SQL Server Parameters tabs apply to Clarify(CBO).

Edit channels

JDBC-ODBC Bridge Parameters Oracle Parameters SQL Server Parameters EDA Server Parameters

Data Source:

User:

Password:

Polling Interval:

SQL Query:

Post Query:

Delete Keys:

- a. Select a listener parameters tab.
- b. Enter the system information that is specific to the database to which you will listen.

The following table lists and describes the parameters found in the listener tabs.

Field	Description
Host	Name of the server where the ClarifyCRM database instance resides.
Port	Port number where the database is listening.
SID	A unique name of the database service, chosen by the database administrator or the person who installed ClarifyCRM.

Field	Description
Database Name	Name of the database.
User	The ClarifyCRM database user ID to access the ClarifyCRM database underlying the ClarifyCRM system. The user ID must have database access to the tables you want to access.
Password	Password associated with the specified user ID.
Polling Interval	A value, in seconds, at which to check for new input.
SQL Query	An SQL query, for example: <code>select * from iw_events</code>

Field	Description
Post Query*	<p>One or more SQL statements that are executed after each new record is read from the table. Separate multiple statements with a semicolon (;).</p> <p>Case sensitive: the case used to specify the column names must match the case used in the SELECT statement that polled the table. If the SQL Query property was omitted so that a default SELECT statement polled the table, the case used to specify the column names must match the case used to define the columns in the DBMS native schema.</p> <p>When you specify a value for this parameter, the table data is retained after it is read.</p> <p>If you do not specify a value for Post Query, each record read from the table is deleted after it is read, depending on whether you specify the Delete Keys property. If you:</p> <ul style="list-style-type: none"> • Specify the Delete Keys property, by default the adapter issues a DELETE statement with a WHERE clause containing every key column specified for the Delete Keys property. <p>At run-time this will be faster than if you had not specified the Delete Keys property if there is an index on the key, or if there are fewer key columns than there are columns in the SELECT statement that polled the table.</p> <ul style="list-style-type: none"> • Do not specify the Delete Keys property, by default the adapter issues a DELETE statement with a WHERE clause that specifies every column from the SELECT statement that polled the table. <p>* The SQL Post-query and Delete Keys parameters are mutually exclusive, as Delete Keys applies to the default DELETE statement, and SQL Post-query overrides the default DELETE statement. You can provide a value for one or the other, but not for both.</p> <p>There are two field operators, ? and ^, that you can use in a post-query SQL statement; for more information, see <i>Post Query Parameter Operators</i> on page 4-23.</p>

Field	Description
Delete Keys*	<p>Comma-separated list of key columns used in the default DELETE statement. DELETE operates on keys, therefore specify the table key columns.</p> <p>Case sensitive: the case used to specify the column names must match the case used in the SELECT statement that polled the table. If the SQL Query property was omitted so that a default SELECT statement polled the table, the case used to specify the column names must match the case used to define the columns in the DBMS native schema.</p> <p>* The Delete Keys and SQL Post Query parameters are mutually exclusive. Delete Keys applies to the default DELETE statement and SQL Post Query overrides the default DELETE statement. You can provide a value for one or the other, but not for both. For more information, see the description of the Post Query parameter in this table.</p>

6. Click Next.

The Select Ports pane opens on the right, as shown in the following image. This pane provides a list of Available ports and Current ports, and a series of buttons that allow you move ports from one list to the other.

Select Ports

Available

Test
abcTest

<<

<

>

>>

Current

Help

< Back

Finish

Cancel

7. Associate one or more ports with this channel.

To associate one port, select a port from the list of Available ports and click the single right arrow button to transfer the port to the list of Current ports. Repeat this step to associate additional ports.

To associate all ports in the Available list, click the double right arrow button.

8. Click *Finish*.

The summary window opens. The new channel appears under the channels node in the left pane. The right pane displays the channel description, channel status, and the available ports.

An X over the channel icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

Procedure: How to Start a Channel

To start a channel:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *Clarify(CBO)* node, and the *channels* node.
3. Select the channel you want to start.
4. In the right pane, move the pointer over *Operations* and select *Start the channel*.

The channel becomes active.

In the left pane, the X that was over the icon disappears.

To stop the channel at any time, move the pointer over *Operations* and select *Stop the channel*.

Post Query Parameter Operators

When you configure a Table Listener, you can use two special field operators, ? and ^, with the SQL Post Query parameter. Both of these operators dynamically substitute database values in the SQL post-query statement at run time.

- ?fieldname is evaluated at run time as field = value.

The ? operator is useful in UPDATE statements:

```
UPDATE table WHERE ?field
```

For example, the following statement

```
UPDATE Stock_Prices_Temp WHERE ?RIC
```

might be evaluated at run time as:

```
UPDATE Stock_Prices_Temp WHERE RIC = 'PG'
```

- ^fieldname is evaluated at run time as value

The ^ operator is useful in INSERT statements:

```
INSERT INTO table VALUES (^field1, ^field2, ^field3, ... )
```

For example, the following statement

```
INSERT INTO Stock_Prices_Temp VALUES (^RIC, ^Price, ^Updated)
```

might be evaluated at run time as:

```
INSERT INTO Stock_Prices_Temp VALUES ('PG', 88.62, '2003-03-18  
16:24:00.0')
```

Editing or Deleting a Channel

This section explains how to edit the properties of a channel and how to delete a channel from an event port.

Procedure: How to Edit a Channel

To edit an existing channel:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *Clarify(CBO)* node, and the *channels* node.
3. Select the channel you want to edit.
4. Stop the channel if it is started.
5. In the right pane, move the pointer over *Operations* and select *Edit*.

The Edit CLARIFY(CBO) channel pane opens.

6. Make the required changes in this parameters pane and click *Next* to continue to the next parameters pane.
7. When you complete changes to the last parameters pane, Selected Ports, click *Finish*.

Procedure: How to Delete a Channel

To delete an existing channel:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *Clarify(CBO)* node, and the *channels* node.

- 3.** Select the channel you want to delete.
- 4.** Stop the channel if it is started.
- 5.** In the right pane, move the pointer over *Operations* and select *Delete*.
A confirmation dialog box opens.
- 6.** To delete the channel you selected, click *OK*.
The channel disappears from the list in the left pane.

CHAPTER 5

Listening for Events Using ClarifyCRM ClearBasic

Topics:

- Understanding iWay Event Functionality
- Creating an Event Port
- Editing or Deleting an Event Port
- Creating a Channel
- Editing or Deleting a Channel

This section describes how to use the iWay Application Systems Adapter for Amdocs ClarifyCRM deployed to an application server, such as BEA WebLogic Server, to listen for events in a ClarifyCRM database table through ClarifyCRM ClearBasic (CB). Several listening techniques are available, enabling you to choose the technique that best suits your requirements.

Although this section describes the Java™ servlet implementation of Application Explorer, other implementations provide the same functionality by means of similar graphical user interfaces. For information on running Application Explorer in WebLogic Workshop, see Appendix D, *Using Application Explorer in BEA WebLogic Workshop for Event Handling with ClarifyCRM ClearBasic*.

Understanding iWay Event Functionality

Events are generated as a result of ClarifyCRM activity. You can use events to trigger an action in your application.

After you create a connection to your application system, you can add events using iWay Servlet Application Explorer. To create an iWay event, you must create a port and a channel.

- **Port**

A port is a logical definition of how to direct event data. It includes information on where to store the information and where to place it after it is processed. It associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption.

For example, you can use a JMS protocol to route the result of polling a table to a JMS queue hosted by BEA WebLogic Server, a J2EE application server.

- **Channel**

A channel defines the listening capability that detects events in the target Enterprise Information System (EIS) or database. It represents configured connections to particular instances of back-end or other types of systems. A channel binds one or more event ports to a particular listener managed by an adapter.

Creating an Event Port

You can create an event port in Application Explorer from either the Service Adapters tab or the Event Adapters tab. There are slight variations in the two methods, but both result in a new port, therefore, you can use either of the following procedures.

- *Create an Event Port from the Service Adapters Tab on page 5-3*
- *Create an Event Port from the Event Adapters Tab on page 5-5*

Several dispositions are available when defining a port. When you use Application Explorer with an Integration Business Services Engine (iBSE) implementation, the following port dispositions are available:

- File
- iBSE
- MSMQ
- JMS
- SOAP

- HTTP
- MQ Series
- MAIL

Note: The MAIL disposition option will be supported in a future release.

When you use Application Explorer with a JCA implementation, the following port dispositions are available:

- File
- JMS
- MQ Series
- HTTP

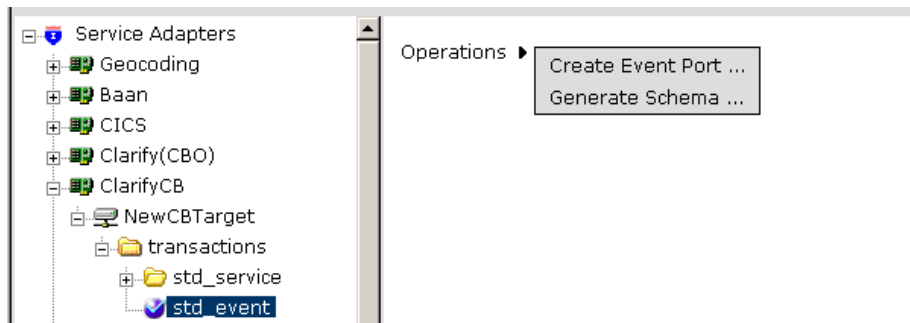
Additional information about each port disposition appears in this section after the procedures to create a port.

Procedure: How to Create an Event Port from the Service Adapters Tab

To create an event port from the Service Adapters tab:

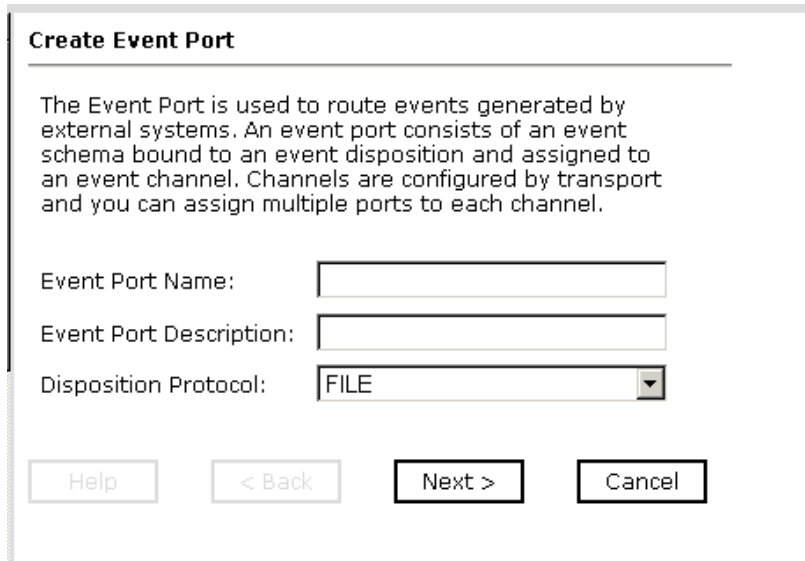
1. Click the *Services Adapters* tab.
2. In the left pane, expand the *Service Adapters* node and the *ClarifyCB* node.
3. Connect to the ClarifyCB target for which you want to create the port.
4. Under the ClarifyCB target, expand the *transactions* node and select the *std_event* node.

The following image shows an expanded transactions node on the left, and the port Operations menu on the right.



5. In the right pane, move the pointer over *Operations* and select *Create Event Port*.

The Create Event Port pane opens on the right, as shown in the following image. This pane provides three parameter fields to identify the new port, a help button, and three action buttons.



Create Event Port

The Event Port is used to route events generated by external systems. An event port consists of an event schema bound to an event disposition and assigned to an event channel. Channels are configured by transport and you can assign multiple ports to each channel.

Event Port Name:

Event Port Description:

Disposition Protocol:

- a. In the Event Port Name field, type a name for the event port.
 - b. In the Event Port Description field, type a brief description of the port.
 - c. From the Disposition Protocol drop-down list, select a protocol.
6. Click *Next*.

The Specify *protocol* Disposition pane opens on the right. The pane includes a brief description of the disposition type that was selected in Step 4, and provides a field with a default URL to the document that will contain event information, a Help button, and three action buttons. An example of the Specify FILE Disposition pane is shown in the following image.

7. In the Disposition URL field, keep the default URL or enter a new URL to the event document. To find the URL format for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page 5-7.
8. Click *Finish*.

The Event Adapters tab opens. The left pane shows the new event port under the target *ports* node and the right pane shows summary information associated with the new event port. To see an image of a port node for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page 5-7.

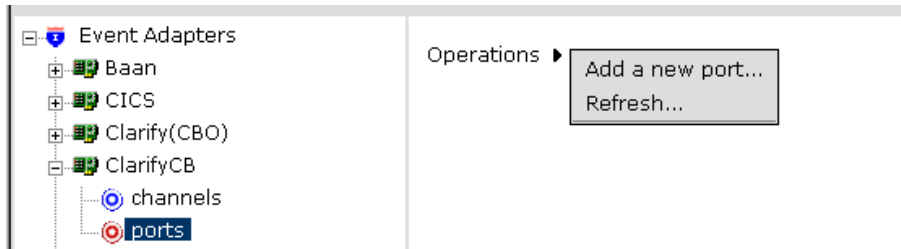
You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 5-16.

Procedure: How to Create an Event Port from the Event Adapters Tab

To create an event port from the Event Adapters tab:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node and the *ClarifyCB* node.
3. Select the *ports* node.

The following image shows an expanded Event Adapters node on the left, and the port Operations menu on the right.



4. In the right pane, move the pointer over *Operations* and select *Add a new port*.

The Create New Port pane opens on the right, as shown in the following image. This pane provides four parameter fields to identify the new port, a help button, and two action buttons.

A screenshot of a 'Create New Port' dialog box. The title bar says 'Create New Port'. Below the title bar is a horizontal line. The text 'Choose parameters of the port that you wish to create.' is displayed. There are four input fields: 'Port Name:', 'Description:', 'Disposition Protocol:', and 'Disposition:'. The 'Disposition Protocol:' field has a dropdown menu with 'FILE' selected. The 'Disposition:' field contains the text 'ifile:///location;errorTo=pre-define'. At the bottom, there are three buttons: 'Help', 'OK', and 'Cancel'.

- a. In the Port Name field, type a name for the event port.
 - b. In the Description field, type a brief description of the port.
 - c. From the Disposition Protocol drop-down list, select *FILE*.
 - d. In the Disposition field, specify a destination file to which the event data will be written. To find the URL format for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page 5-7.
5. Click *OK*.

The left pane shows the new event port under the target *ports* node. The right pane shows summary information associated with the new event port. To see an image of a port node for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page 5-7.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 5-16.

FILE Disposition URL Format

The format of the URL to the FILE disposition event document is:

- **When pointing Application Explorer to an iBSE deployment**, specify the destination file using the following format:

```
ifile://location[;errorTo=errorDest]
```

- **When pointing Application Explorer to a JCA deployment**, specify the destination file using the following format:

```
location
```

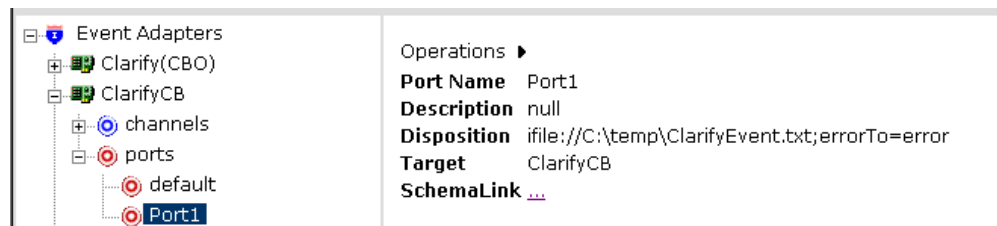
The following table describes the FILE disposition parameters.

Parameter	Description
location	Full directory path and file name to which the data is written.
errorDest	Location to which error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

For example:

```
ifile://c:\temp\ClarifyEvent.txt;errorTo=ifile://c:\temp\error
```

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears in the right pane. The following image shows a listing of an event port with a FILE disposition.



iBSE Disposition URL Format

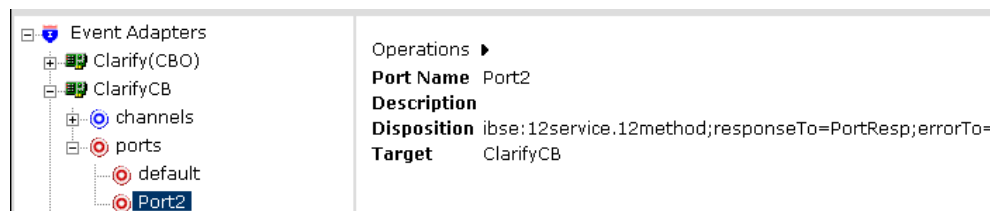
The format of the URL to the iBSE disposition event document is:

`ibse:/svcName.methName[;responseTo=respDest][;errorTo=errorDest]`

The following table describes the disposition parameters.

Parameter	Description
svcName	Name of the service created with iBSE.
methName	Name of the method created for the Web service.
respDest	Location where responses to the Web service are posted. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.
errorDest	Location where error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears in the right pane. The following image shows a listing of an event port with a iBSE disposition.



MSMQ Disposition URL Format

The format of the URL to the MSMQ disposition event document is:

`msmq:/host/queueType/queueName[;errorTo=errorDest]`

The following table defines the MSMQ disposition parameters.

Parameter	Description
host	Name of the host on which the Microsoft Queuing system runs.

Parameter	Description
queueType	Type of queue. For private queues, enter <i>Private\$</i> . Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
queueName	Name of the queue where messages are placed.
errorDest	Location where error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears in the right pane. The following image shows a listing of an event port with an MSMQ disposition.



JMS Disposition URL Format

The format of the URL to the JMS disposition event document is:

- When pointing Application Explorer to an **ibSE** deployment, specify the JMS destination file using the following URL format:

```
jmsq:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory;
user=userID;password=pass[;errorTo=errorDest]
```

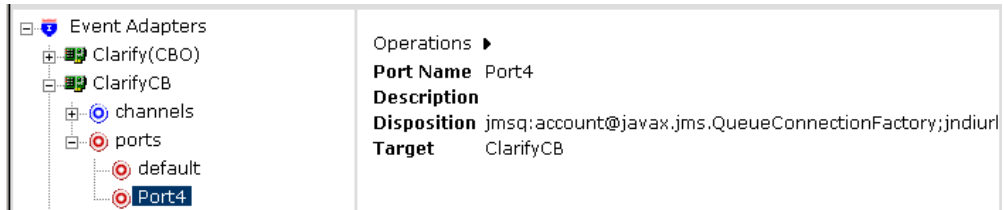
- When pointing Application Explorer to a **JCA** deployment, specify the JMS destination file using the following URL format:

```
jms:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory
```

The following table describes the disposition parameters.

Parameter	Description
queue	Name of a queue to which events are emitted.
conn_factory	The connection factory, a resource that contains information about the JMS Server. The BEA WebLogic connection factory is: <code>javax.jms.QueueConnectionFactory</code>
jndi_url	The URL to use to contact the JNDI provider. The syntax of this URL depends on which JNDI provider is being used. This value corresponds to the standard JNDI property: <code>java.naming.provider.url</code> For BEA WebLogic Server, this is: <code>t3://host:port</code> where: <code>host</code> Is the machine name where BEA WebLogic Server is installed. <code>port</code> Is the port on which BEA WebLogic Server is listening. The default port if not changed at installation is 7001.
jndi_factory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For BEA WebLogic Server, the BEA WebLogic factory is weblogic.jndi.WLInitialContextFactory.
userID	User ID associated with this queue.
pass	Password associated with the user ID.
errorDest	Location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears in the right pane. The following image shows a listing of an event port with an JMS disposition.



SOAP Disposition URL Format

The format of the URL to the SOAP disposition event document is:

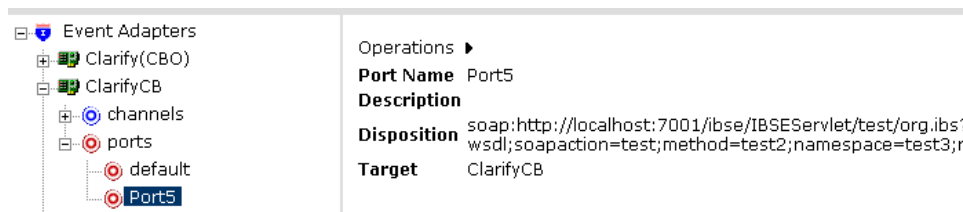
```
soap:[wsdl-url];soapaction=[myaction];method=[web service
method];namespace=[namespace];responseTo=[pre-defined port name or
another disposition URL];errorTo=[pre-defined port name or another
disposition url]
```

The following table defines the disposition parameters.

Parameter	Description
wsdl-url	<p>The URL to the WSDL file that is required to create the SOAP message. For example,</p> <pre>http://localhost:7001/ibse/IBSEServlet/test/ webservice.ibs?wsdl</pre> <p>where:</p> <pre>webservice</pre> <p>Is the name of the Web service you created using Application Explorer.</p> <p>To find this value, navigate to the Integration Business Services tab, expand the <i>Services</i> node, select the service you created, and click the <i>Service Description</i> link on the right. The WSDL URL appears in the Address field of the window that opens.</p> <p>You can also open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.</p>
soapaction	The method that will be called by the SOAP disposition. This value can be found in the WSDL file.

Parameter	Description
method	The Web service method you are using. This value is found in the WSDL file.
namespace	The XML namespace you are using. This value is found in the WSDL file.
responseTo	The location to which responses are posted. A predefined port name or another full URL. Optional. The URL must be complete, including the protocol.
errorTo	The location to which error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears in the right pane. The following image shows a listing of an event port with an SOAP disposition.



HTTP Disposition URL Format

The format of the URL to the HTTP disposition event document is:

- When pointing Application Explorer to an **ibse** deployment, specify the HTTP destination file using the following format:

http://url;responseTo=respDest

- When pointing Application Explorer to a **JCA** deployment, specify the HTTP destination file using the following format:

http://host:port/uri

The following table describes the disposition parameters.

Parameter	Description
url	The URL target for the post operation.

Parameter	Description
respDest	Location where responses are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
host	Name of the machine on which the Web server resides.
port	Port number on which the Web server is listening.
uri	Universal resource identifier that completes the url specification.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears in the right pane. The following image shows a listing of an event port with an HTTP disposition.



MQSeries Disposition URL Format

The format of the URL to the MQSeries disposition event document is:

- When pointing Application Explorer to an **IBSE** deployment, specify the MQSeries destination file using the following format:

```
mqseries:/qManager/qName;host=hostName;port=portNum;
channel=chanName[;errorTo=errorDest]
```

- When pointing Application Explorer to a **JCA** deployment, use the following format:

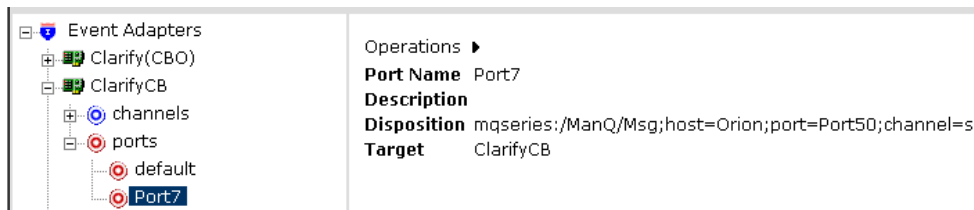
```
mq:qmanager@respqueue;host=;port=;channel=
```

The following table lists and describes the disposition parameters for MQSeries.

Parameter	Description
qManager	Name of the queue manager to which the server must connect.
qName or respqueue	Name of the queue where messages are placed.

Parameter	Description
host	Name of the machine on which MQSeries resides (MQ client only).
port	Port number for connecting to an MQ server (queue manager). MQ client only.
chan	The case-sensitive name of the channel that connects with the remote MQ server (queue manager). MQ client only. The default MQSeries channel name is SYSTEM.DEF.SVRCONN.
errorTo	Location where error documents are sent. Can be a predefined port name or another full URL. Optional.

Once an event port is created, it appears in the left pane under the target *ports* node and summary information associated with the event port appears in the right pane. The following image shows a listing of an event port with an MQSeries disposition.



Editing or Deleting an Event Port

The following procedures describe how to edit and delete an event port using Application Explorer.

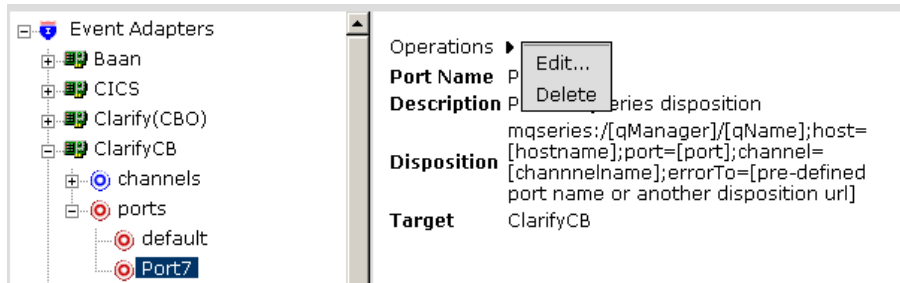
Procedure: How to Edit an Event Port

To edit an existing event port using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *ClarifyCB* node, and the *ports* node.

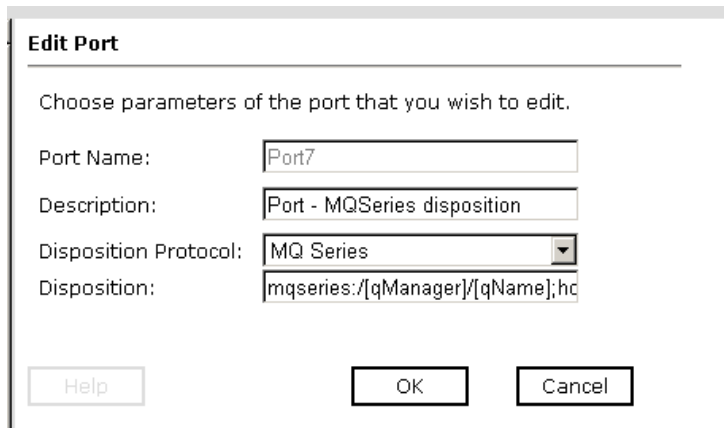
3. Select the event port you want to edit.

The following image shows an expanded Event Adapters node on the left, and the port Operations menu on the right.



4. In the right pane, move the pointer over *Operations* and select *Edit*.

The Edit Port pane, shown in the following image, opens on the right. This pane provides four port parameter fields, a help button, and two action buttons.



You cannot change the name of the port in the Edit Port pane.

5. Make the required changes to the Description, Disposition Protocol, or Disposition fields, and click OK.

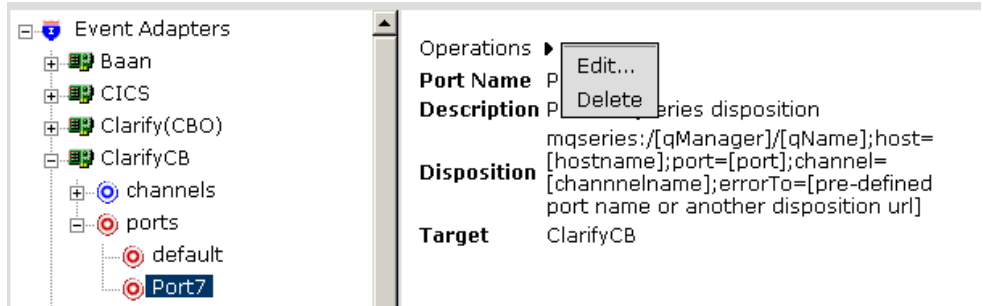
Procedure: How to Delete an Event Port

To delete an existing event port using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *ClarifyCB* node, and the *ports* node.

3. Select the event port you want to delete.

The following image shows an expanded Event Adapters node on the left, and the port Operations menu on the right.



4. In the right pane, move the pointer over *Operations* and select *Delete*.

A confirmation dialog box opens.

5. Click *OK*.

The event port disappears from the list in the left pane.

Creating a Channel

The following procedure describes how to create a channel for a ClarifyCRM event. All defined event ports must be associated with a channel.

Procedure: How to Create a Channel

To create a channel using Application Explorer:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node and the *ClarifyCB* node.
3. Select the *channels* node.

The following image shows the channel node on the left and the Operations menu on the right.



4. In the right pane, move the pointer over *Operations* and select *Add a new channel*.

On the right, the Add a new CLARIFYCB channel pane opens. This pane, shown in the following image, provides three fields to identify the channel, a help button, and three action buttons.

Add a new CLARIFYCB channel

Choose a name and description for the new channel that you wish to create.

Channel Name:

Description:

Channel Type:

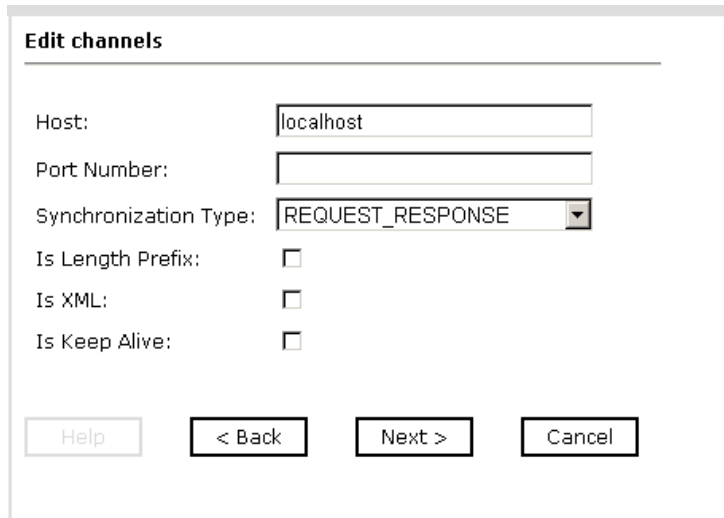
- a. In the Channel Name field, enter a name for the channel.
 - b. In the Description field, enter a brief description of the channel.
 - c. Choose a listener from the Channel Type drop-down list.
5. Click *Next*.

If you chose the JNDI Listener as the channel type, the following Edit channels pane opens on the right.

Edit channels

JNDI Name:

If you chose the TCP Listener as the channel type, the following Edit channels pane opens on the right.



The screenshot shows a dialog box titled "Edit channels". It contains the following fields and controls:

- Host:** A text input field containing the value "localhost".
- Port Number:** An empty text input field.
- Synchronization Type:** A drop-down menu with "REQUEST_RESPONSE" selected.
- Is Length Prefix:** An unchecked checkbox.
- Is XML:** An unchecked checkbox.
- Is Keep Alive:** An unchecked checkbox.
- Buttons:** At the bottom, there are four buttons: "Help" (disabled), "< Back", "Next >", and "Cancel".

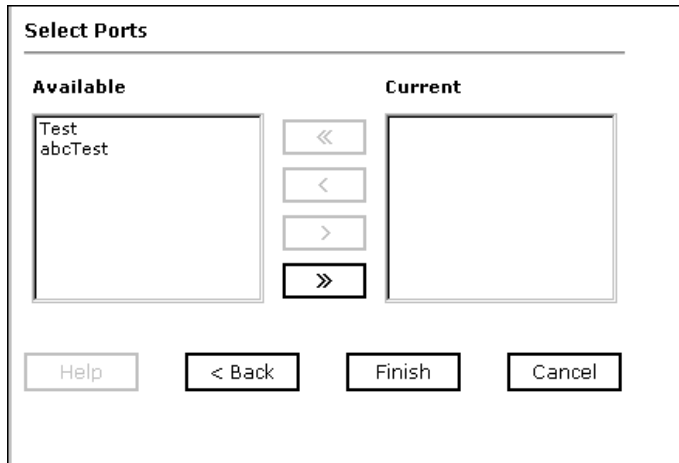
6. Type or select the appropriate channel values in the Edit channels pane.

In the JNDI Listener Edit channels pane, edit the JNDI name value, if necessary.

In the TCP Listener Edit channels pane, edit the Host name, if necessary; type a value in the Port Number field, select a synchronization type (REQUEST_RESPONSE, REQUEST_ACK, or REQUEST) from the drop-down list; and select one or more of the check boxes (Is Length Prefix, Is XML, Is Keep Alive).

7. Click *Next*.

The Select Ports pane opens on the right, as shown in the following image. A list of available ports appear in the Available box on the left, and the ports that are currently associated appear in the Current box on the right. This pane also contains a series of buttons to move ports from one box to another, a help button, and three action buttons.



8. Associate one or more ports with this channel.

To associate one port, select a port from the list of Available ports and click the single right arrow (>) button to transfer the port to the list of Current ports. Repeat this step to associate additional ports.

To associate all ports in the Available list, click the double right arrow (>>) button.

9. Click *Finish*.

On the left, the new channel appears under the channels node. On the right, a display of the channel description, channel status, and the available ports appears.

An X over the channel icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

Procedure: How to Start a Channel

To start a channel:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *ClarifyCB* node, and the *channels* node.
3. Select the channel you want to start.
4. In the right pane, move the pointer over *Operations* and select *Start the channel*.

The channel becomes active.

In the left pane, the X that was over the icon disappears.

To stop the channel at any time, move the pointer over *Operations* and select *Stop the channel*.

Editing or Deleting a Channel

This section explains how to edit the properties of a channel and how to delete a channel from an event port.

Procedure: How to Edit a Channel

To edit an existing channel:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *ClarifyCB* node, and the *channels* node.
3. Select the channel you want to edit.
4. Stop the channel if it is started.
5. In the right pane, move the pointer over *Operations* and select *Edit*.
The Edit channels pane opens.
6. Make the required changes in this parameters pane and click *Next* to continue to the next parameters pane. When you complete changes to the last parameters pane, Selected Ports, click *Finish*.

Procedure: How to Delete a Channel

To delete an existing channel:

1. Click the *Event Adapters* tab.
2. In the left pane, expand the *Event Adapters* node, the *ClarifyCB* node, and the *channels* node.
3. Select the channel you want to delete.
4. Stop the channel if it is started.
5. In the right pane, move the pointer over *Operations* and select *Delete*.

A confirmation dialog box opens.

6. To delete the channel you selected, click *OK*.

The channel disappears from the list in the left pane.

CHAPTER 6

Using Web Services Policy-Based Security

Topics:

- iWay Business Services Policy-Based Security
- Configuring iWay Business Services Policy-Based Security

Servlet Application Explorer provides a security feature called iWay Business Services policy-based security. The following topics describe how this feature works and how to configure it.

Note: For the iWay 5.5 RG2 Release, it is recommended that policy-based security not be enabled.

iWay Business Services Policy-Based Security

iWay Business Services provide a layer of abstraction between the back-end business logic they invoke and the user or application running the business service. This enables easy application integration but raises the issue of controlling the use and execution of critical and sensitive business logic that is run as a business service.

Servlet Application Explorer controls the use of business services that use adapters with a feature called policy-based security. This feature enables an administrator to apply *policies* to iWay Business Services (iBS) to deny or permit their execution.

A *policy* is a set of privileges associated with the execution of a business service that can be applied to an existing or new iBS. When you assign specific rights or privileges inside a policy, you need not recreate privileges for every iBS that has security issues in common with other iWay Business Services. Instead, you can use one policy for many iWay Business Services.

The goal is to secure requests at both the transport and the SOAP request level that is transmitted on the wire. Some policies do not deal with security issues directly but affect the run-time behavior of the business services to which they are applied.

The iBSE administrator creates an instance of a policy type, names it, associates individual users and/or groups (a collection of users), and then applies the policy to one or more business services.

You can assign a policy to an iBS or to a method within an iBS. If a policy is applied only to a method, other methods in that iBS are not governed by it. However, if a policy is applied to the iBS, all methods are governed by it. At run time, the user ID and password that are sent to iBSE in the SOAP request message are checked against the list of users for all policies applied to the specific iBS. The Resource Execution policy type is supported and dictates who can or cannot execute the iBS.

When a policy is not applied, the default value for an iBS is to “grant all.” For example, anyone can execute the iBS until the Resource Execution policy is associated to the iBS. At that time, only users granted execution permission, or those who do not belong to a group that was denied execution permissions, have access to the iBS.

Configuring iWay Business Services Policy-Based Security

Before you create instances of policies, you must have a minimum of one user or one group to associate to an instance. You can create users and groups using Servlet Application Explorer. For more information, see *Create a User to Associate With a Policy* on page 6-3 or *Create a Group to Associate With a Policy* on page 6-5.

An execution policy governs who can execute the business service to which the policy is applied. For more information, see *Create an Execution Policy* on page 6-7.

You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to iBSE and therefore, need not be applied to an individual business service. You need not create a policy, however, you must enable the Security Policy option in Servlet Application Explorer. For more information, see *Configure IP and Domain Restrictions* on page 6-10.

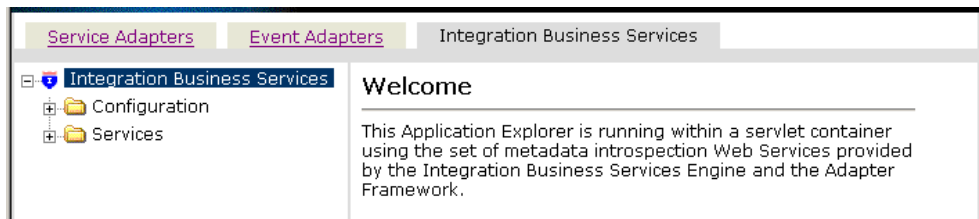
Note: For the iWay 5.5 RG2 Release, it is recommended that policy-based security not be enabled.

Procedure: How to Create a User to Associate With a Policy

To create a user to associate with a policy:

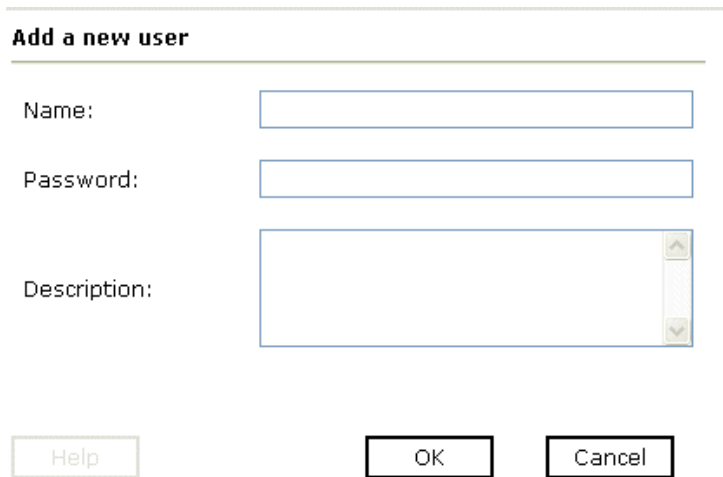
1. Open *Servlet Application Explorer*.

The following image shows the window that opens and includes three tabs corresponding to Service Adapters, Event Adapters, and iWay Business Services. The iWay Business Services tab is active and displays a Welcome screen on the right. The image shows the iWay Business Services node expanded in the left pane.



- a. Click the *iWay Business Services* tab.
 - b. Expand the *Configuration* node.
 - c. Expand the *Security* node.
 - d. Expand the *Users and Groups* node.
 - e. Select *Users*.
2. In the right pane, move the pointer over *Operations* and select *Add*.

The following image shows the Add a new user pane that opens and includes fields where you enter a user name, a password, and a description of the user. The pane includes a Help button, an OK button to instruct the system to accept inputs, and a Cancel button to escape from the pane.



Add a new user

Name:

Password:

Description:

- a. In the Name field, type a user ID.
 - b. In the Password field, type the password associated with the user ID.
 - c. In the Description field, type a description of the user (optional).
3. Click **OK**.

The following image opens and shows a new user added to the configuration. It includes a definition of a user and a user ID and description.

Operations ▸



Users

A user is an object that can be granted or denied permissions to run Integration Business Services. A user can belong to one or more groups. Policies that specify particular rights can be associated with user.

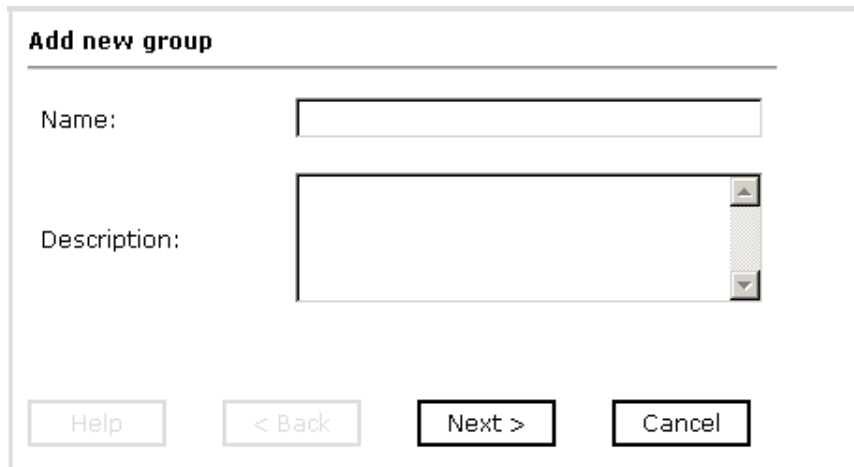
User Id	Description
<input type="checkbox"/> bse1	

Procedure: How to Create a Group to Associate With a Policy

To create a group to associate with a policy:

1. Open *Servlet Application Explorer*.
 - a. Click the *iWay Business Services* tab.
 - b. Expand the *Configuration* node.
 - c. Expand the *Security* node.
 - d. Expand the *Users and Groups* node.
 - e. Select *Groups*.
2. In the right pane, move the pointer over *Operations* and click *Add*.

The following image shows the Add new group pane that opens with fields where you enter a name and a description for the group. To continue after typing inputs, click the *Next* button. The pane also includes a *Help* button, a *Back* button to return to the previous screen, and a *Cancel* button to escape from the pane.



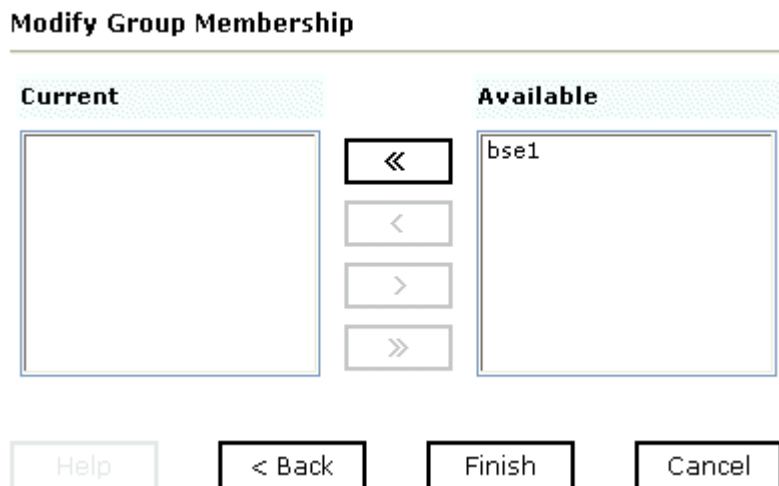
Add new group

Name:

Description:

- a. In the Name field, type a name for the group.
 - b. In the Description field, type a description for the group (optional).
3. Click **Next**.

The following image shows the Modify Group Membership pane where you can move users to or from a group using the arrow keys to move them between the Current and Available lists and then clicking the Finish button. The pane includes a Help button, a Back button to return to the previous screen, and a Cancel button to escape from the pane.



Modify Group Membership

Current		Available
<input type="text"/>	<input type="button" value="«"/>	<input type="text" value="bse1"/>
	<input type="button" value=" <"/>	
	<input type="button" value=" >"/>	
	<input type="button" value="»"/>	

You can either highlight a single user in the list of available users and add it to the current list by clicking the left arrow, or you can click the double left arrow to add all users in the list of available users to the group.

4. After you select a minimum of one user, click *Finish*.

The new group is added.

The following image shows a pane with a new group added to the configuration. It includes a definition of a group and the group name and description.

Operations ►



Groups

A group is an object that can be granted or denied permissions to run Integration Business Services. A group is used as a container for one or more users. Policies that specify particular rights can be associated with a group.

Group name	Description
<input type="checkbox"/> newgroup	

Procedure: How to Create an Execution Policy

To create an execution policy:

1. Open *Servlet Application Explorer*.
 - a. Click the *iWay Business Services* tab.
 - b. Expand the *Configuration* node.
 - c. Select *Policies*.

The following image shows the Policies pane on the right where you apply a policy. The Operations menu becomes available with three options, Build/Rebuild, Add, and Refresh.



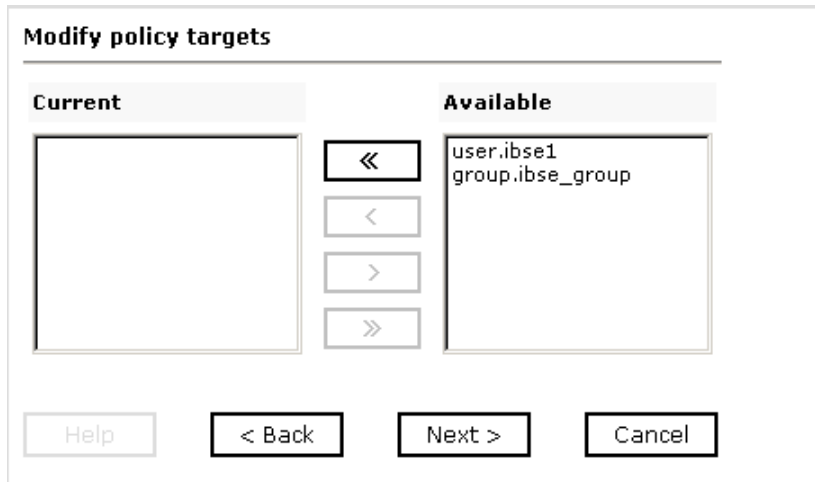
2. Move the pointer over *Operations* and click *Add*.

The following image shows the Add a new policy pane that opens with fields for entering the name, type, and description of the policy. To continue, click the *Next* button. The pane includes a *Help* button, a *Back* button to return to the previous screen, and a *Cancel* button to escape from the pane.

The dialog box is titled 'Add a new policy'. It contains three input fields: 'Name:' with a text box, 'Type:' with a drop-down menu showing 'Execution', and 'Description:' with a text area. At the bottom, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'. The 'Next >' button is highlighted with a black border.

- a. In the Name field, type a name for the policy.
 - b. From the Type drop-down list, select *Execution*.
 - c. In the Description field, type a description for the policy (optional).
3. Click *Next*.

The following image shows the Modify policy targets pane that opens and includes a list of current and available targets and arrow buttons to move targets from one list to the other. The pane also includes a Help button, a Back button to return to the previous screen, a Next button to continue to the next screen, and a Cancel button to escape from the pane.



4. Select a minimum of one user or group from the Available pane.

Note: This user ID is checked against the value in the user ID element of the SOAP header sent to iBSE in a SOAP request.

5. Click *Next*.

The following image shows the Modify policy permissions pane that opens and includes drop-down lists where you can select to grant or deny permission to members and then click a button to finish. The pane also includes a Help button, a Back button to return to the previous screen, and a Cancel button to escape from the pane.

Member Id	Permission
user.ibse1	Deny
group.ibse_group	Deny

Buttons: Help, < Back, Finish, Cancel

6. To assign whether users or groups may execute the iBSE, select *Grant* to permit execution or *Deny* to restrict execution from a Permission drop-down list.
7. Click *Finish*.

The following image shows the pane that summarizes your configuration. It includes a definition of policies and the name, type, and description of the policies.

Operations ▶

Policies

You can configure policies for the Integration Business Services Engine to manage resource execution, service routing, data restrictions and failover/recovery actions.

Name	Type	Description
<input type="checkbox"/> ibse_policy	Execution	

Procedure: How to Configure IP and Domain Restrictions

To configure IP and domain restrictions:

1. Open *Servlet Application Explorer*.
 - a. Select the *iWay Business Services* tab.
 - b. Expand the *Configuration* node.
 - c. Expand the *Security* node.
 - d. Select *IP and Domain*.
2. In the right pane, move the pointer over *Operations* and click *Add*.

The following image shows the Add a new IP/Domain pane that opens where you enter information for the IP/Domain in four fields. You must select a type of restriction from a drop-down list before you can enter information in the IP(Mask)/Domain field. The pane also includes a Help button, an OK button to instruct the system to accept inputs, and a Cancel button to escape from the pane.

Add a new IP/Domain

IP(Mask)/Domain:

Type:

Access Control:

Description:

- a. From the Type drop-down list, select the type of restriction.
- b. In the IP(Mask)/Domain field, type the IP or domain name using the following guidelines.

If you select Single (Computer) from the Type drop-down list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click *DNS Lookup* to obtain the IP Address based on the DNS name.

If you select Group (of Computers), you must provide the IP address and subnet mask for the computer group.

If you select Domain, you must provide the domain name, for example, yahoo.com.

3. From the Access Control drop-down list, select *Grant* to permit access or *Deny* to restrict access for the IP addresses and domain names you are adding.
4. Click OK.

The following image shows the pane that opens and summarizes your configuration including the domain name, whether access is granted or denied, and a description (optional).

Operations ►



IP and Domain

You can configure the Integration Business Services Engine to use policies that control access from a single IP address, a group of IP addresses, or all addresses within a particular domain.

IP(Mask) / Domain	Access	Description
<input type="checkbox"/> test	Deny	

CHAPTER 7

Management and Monitoring

Topics:

- Managing and Monitoring Services and Events Using iBSE
- Managing and Monitoring Services and Events Using the JCA Test Tool
- Setting Engine Log Levels
- Configuring Connection Pool Sizes
- Migrating Repositories
- Exporting or Importing Targets
- Retrieving or Updating Web Service Method Connection Information
- Starting or Stopping a Channel Programmatically

After you create services and events using Servlet Application Explorer, you can use managing and monitoring tools provided by the Integration Business Services Engine (iBSE) and the iWay Connector for JCA to measure the performance of your run-time environment. This section describes how to configure and use these features.

Managing and Monitoring Services and Events Using iBSE

Integration Business Services Engine (iBSE) provides a console to manage and monitor services and events currently in use and to display resource usage and invocation statistics. These indicators can help you adjust your environment for optimum efficiency.

The following monitoring levels are available for services:

- System
- Service
- Method

The following monitoring levels are available for events:

- System
- Channel
- Port

Procedure: How to Configure Monitoring Settings

To configure monitoring settings:

1. Ensure that your BEA WebLogic Server is started.
2. To access the monitoring console, enter the following URL in your Web browser:

`http://localhost:port/ibse/IBSEConfig`

where:

`localhost`

Is the machine where the application server is running.

`port`

Is the HTTP port for the application server.

The following image shows the iBSE Settings window that opens. It lists property names and includes fields where you can enter values for each property. To configure system settings, the System pane contains drop-down lists for selecting language, encoding, the debug level, and the number of asynchronous processors. It also contains a field where you can enter a path to the adapters lib directory.

To configure security settings, the Security pane contains fields for typing the Admin User name and the associated password and a check box for specifying policy.

To configure repository settings, the Repository pane contains a drop-down list for selecting the repository type, fields to type information for the repository URL, driver, user, and password, and a check box where you can enable repository pooling. In the upper and lower right of the window is a Save button. In the lower left of the window is an option to click to access more configuration settings.

iBSE Settings:
Save

Property Name	Property Value
System	
Language	English
Adapter Lib Directory	C:\Program Files\iWay55\lib
Encoding	UTF-8
Debug Level	NONE
Number of Async. Processors	0
Security	
Admin User	iway
Admin Password	****
Policy	<input type="checkbox"/>
Repository	
Repository Type	File System
Repository Url	file://C:\Program Files\iWay55\bea\ibse
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	<input type="checkbox"/>
More configuration... Save	

3. Click *More configuration*.

Tip: To access the monitoring console directly, enter the following URL in your Web browser:

<http://localhost:port/ibse/IBSEStatus>

where:

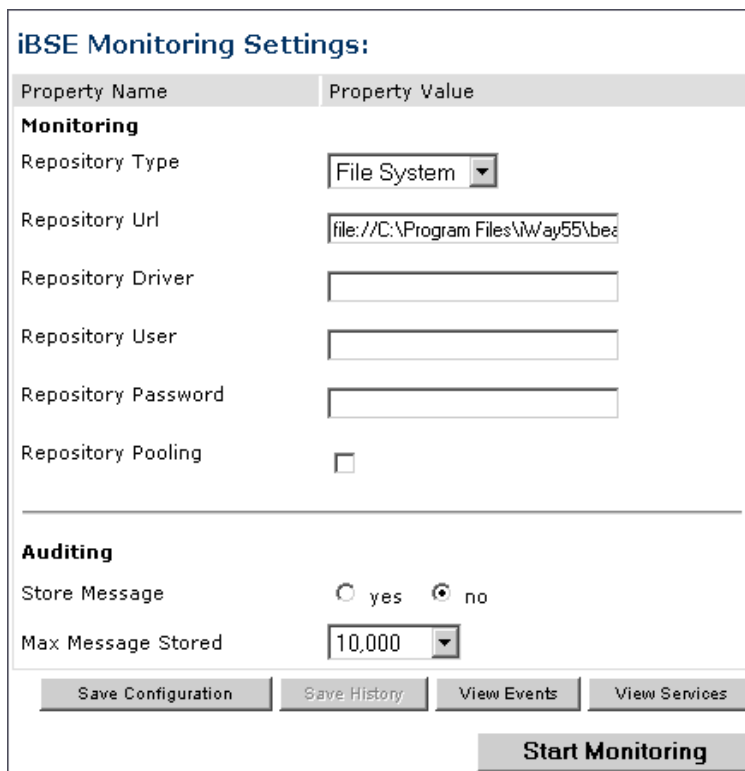
localhost

Is the machine where the application server is running.

port

Is the HTTP port for the application server.

The following image shows the iBSE Monitoring Settings window that opens. It lists property names and includes a corresponding field where you can enter values for each property. The Monitoring pane contains a drop-down list for selecting the repository type, fields to type information for the repository URL, driver, user, and password, and a check box where you can enable repository pooling. The Auditing pane contains an option button to click to specify whether to store a message and a drop-down list where you can select the maximum messages to store. At the bottom of the window is a row of buttons that you can click to save your configuration, view events, or view services. The Save History button is inactive. After you enter properties and choose whether to save or view, you can click the Start Monitoring button.



The image shows a window titled "iBSE Monitoring Settings:". It is divided into two main sections: "Monitoring" and "Auditing".

Monitoring Section:

- Property Name:** Repository Type
- Property Value:** File System (selected in a dropdown)
- Property Name:** Repository Url
- Property Value:** file:///C:/Program Files/iWay55/bes (text input)
- Property Name:** Repository Driver
- Property Value:** (empty text input)
- Property Name:** Repository User
- Property Value:** (empty text input)
- Property Name:** Repository Password
- Property Value:** (empty text input)
- Property Name:** Repository Pooling
- Property Value:** ☐

Auditing Section:

- Property Name:** Store Message
- Property Value:** ☐ yes ☒ no
- Property Name:** Max Message Stored
- Property Value:** 10,000 (selected in a dropdown)

Buttons:

- Save Configuration
- Save History (disabled)
- View Events
- View Services
- Start Monitoring

- a. In the Monitoring pane, from the Repository Type drop-down list, select the type of repository you are using.
- b. To connect to the database in the Repository Url field, type a JDBC URL.
- c. To connect to the database in the Repository Driver field, type a JDBC Class.
- d. To access the monitoring repository database, type a user ID and password.
- e. To enable pooling, click the *Repository Pooling* check box.
- f. In the Auditing pane, select *yes* if you want to store messages.

This option is disabled by default.

Note: You must start and then, stop monitoring to enable this option.

- g. Select the maximum number of messages you want to store.

By default, 10,000 is selected.

Note: Depending on your environment and the number of messages that are exchanged, storing a large number of messages may affect system performance. If you need more information about your system resources, consult your system administrator.

- h. Click *Save Configuration*.

4. Click *Start Monitoring*.

iBSE begins to monitor all services and events currently in use. If you selected the option to store messages, iBSE stores messages.

5. To stop monitoring, click *Stop Monitoring*.

Procedure: How to Monitor Services

To monitor services:

1. Ensure that your BEA WebLogic Server is started.
2. From the iBSE Monitoring Settings window, click *Start Monitoring*.
3. Click *View Services*.

The following image shows the System Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list where you select a service. On the right, space is reserved for a drop-down list of methods that will appear. The Statistics pane contains a table with a summary of service statistics and two drop-down lists where you can select a successful or failed invocation to view more information about that service. At the bottom of the window is a home button to click to return to the iBSE Monitoring Settings window.

The screenshot shows a window titled "Service Statistics". It is divided into two main sections: "Web Service Methods" and "Statistics".

Web Service Methods

Service	Method
all	

Statistics

Total Time	55 min
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	828 ms
Average Back End Time	530 ms
Last Back End Time	765 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

At the bottom right of the window is a button labeled "< home".

The system level summary provides services statistics at a system level.

The following table consists of two columns, one that lists the name of each statistic and the other that describes the corresponding service statistic.

Statistic	Description
Total Time	Total amount of time iBSE monitors services. The time starts after you click Start Monitoring in the iBSE Monitoring Settings window.
Total Request Count	Total number of services requests that were made during the monitoring session.
Total Success Count	Total number of successful service executions.
Total Error Count	Total number of errors that were encountered.
Average Request Size	Average size of an available service request.
Average Response Size	Average size of an available service response size.
Average Execution Time	Average execution time for a service.
Last Execution Time	Last execution time for a service.
Average Back End Time	Average back end time for a service.
Last Back End Time	Last back end time for a service.
Successful Invocations	A list of successful services arranged by correlation ID. To retrieve more information for a service, you can select the service from the drop-down list.
Failed Invocations	A list of failed services arranged by correlation ID. To retrieve more information for a service, you can select the service from the drop-down list.

4. Select a service from the drop-down list.

The following image shows the System Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list on the left where you select a service and a drop-down list on the right where you select a service method. The Statistics pane contains a table with a summary of service statistics and two drop-down lists. To view more information about that service, you can select it from the Successful Invocations or Failed Invocations drop-down list. To suspend or resume a service, you can click a button in the lower right. To return to the iBSE Monitoring Settings window, you click the home button (also located in the lower right).

The screenshot shows a window titled "Service Statistics". It has two main sections: "Web Service Methods" and "Statistics".

Web Service Methods

Service:

Method:

Statistics

Total Time	1 hrs
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	656 ms
Average Back End Time	530 ms
Last Back End Time	530 ms
Successful Invocations	<input type="text" value="select a correlation id"/>
Failed Invocations	<input type="text" value="select a correlation id"/>

- a. To stop a service at any time, click *Suspend Service*.
 - b. To restart the service, click *Resume Service*.
5. Select a method for the service from the Method drop-down list.

The following image shows the Method Level Summary (Service Statistics) window that opens. The Web Service Methods pane contains a drop-down list on the left where you select a service and a drop-down list on the right where you select a service method. The Statistics pane contains a table with a summary of service statistics and two drop-down lists. To view more information about that service, you can select it from the Successful Invocations or Failed Invocations drop-down list. To suspend or resume a service, you can click a button in the lower right. To return to the iBSE Monitoring Settings window, you click the home button (also located in the lower right).

Service Statistics

Web Service Methods

Service
Method

B0100033
GetEffectiveAddress

Statistics

Total Time	1 hrs
Total Request Count	1
Total Success Count	1
Total Error Count	0
Average Request Size	409.0 bytes
Average Response Size	665.0 bytes
Average Execution Time	656 ms
Last Execution Time	656 ms
Average Back End Time	530 ms
Last Back End Time	530 ms
Successful Invocations	select a correlation id
Failed Invocations	select a correlation id

Suspend Service
< home

- For additional information about a successful service and its method, select a service based on its correlation ID from the Successful Invocation drop-down list.

The following image shows the Invocation Level Statistics window that opens. The Message Information pane contains a table of information about the message. The Client Information pane contains a table of information about the client. The Detail pane contains a table that shows the size of the request and response messages, with options to click to view the respective XML documents. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.

The screenshot shows a web application window titled "Invocation Statistics". It is divided into three main sections: "Message Information", "Client Information", and "Detail".

Message Information

Received	2004-09-14 12:04:16.312
Sent to adapter	2004-09-14 12:04:16.406
Received from adapter	2004-09-14 12:04:16.936
Responded	2004-09-14 12:04:16.968
Status	SUCCESS

Client Information

Client IP	127.0.0.1
Client Host Name	127.0.0.1
User Name	

Detail

Message	Size
Request Message	409 bytes
Response Message	665 bytes

In the bottom right corner, there is a button labeled "< home".

7. To view the XML request document in your Web browser, click *Request Message*.
You can also view the XML response document for the service.
8. To return to the iBSE Monitoring Settings window, click *home*.

Procedure: How to Monitor Events

To monitor events:

1. Ensure that your BEA WebLogic Server is started.
2. In the iBSE Monitoring Settings window, click *Start Monitoring*.
3. Click *View Events*.

The following image shows the System Level Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel. On the right, space is reserved for a drop-down list of ports that will appear. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.

Channel Statistics

Channels

Channels Ports

all ▼

Statistics

Total Event Count	4
Total Success Count	3
Total Error Count	1
Average Event Size	337.0 bytes
Average Event Reply Size	na
Average Delivery Time	1274.0 ms
Last Delivery Time	250 ms
Successful Events	select a correlation id ▼
Failed Events	select a correlation id ▼

< home

The system level summary provides event statistics at a system level.

The following table consists of two columns, one that lists the name of each statistic and the other that describes the corresponding event statistic.

Statistic	Description
Total Event Count	Total number of events.
Total Success Count	Total number of successful event executions.
Total Error Count	Total number of errors that were encountered.
Average Event Size	Average size of an available event request.
Average Event Reply Size	Average size of an available event response.
Average Delivery Time	Average delivery time for an event.
Last Delivery Time	Last delivery time for an event.
Successful Events	List of successful events arranged by correlation ID. To retrieve more information for an event, select the event from the drop-down list.
Failed Events	List of failed events arranged by correlation ID. To retrieve more information for an event, select the event from the drop-down list.

4. Select a channel from the drop-down list.

The following image shows the Channel Level Event Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel and a drop-down list on the right where you select a port. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a button to click to suspend or resume a channel and a home button to click to return to the iBSE Monitoring Settings window.

Channel Statistics

Channels

Channels: TestChan Ports: all

Statistics

Total Event Count	3
Total Success Count	2
Total Error Count	1
Average Event Size	401.0 bytes
Average Event Reply Size	na
Average Delivery Time	1542.0 ms
Last Delivery Time	250 ms
Successful Events	select a correlation id
Failed Events	select a correlation id

Suspend Channel Start Channel

< home

- a. To stop a channel at any time, click *Suspend Channel*.
 - b. To start the channel, click *Start Channel*.
5. From the Ports drop-down list, select a port for the channel.

The following image shows the Port Level Event Summary (Channel Statistics) window that opens. The Channels pane contains a drop-down list on the left where you select a channel and a drop-down list on the right where you select a port. The Statistics pane contains a table with a summary of event statistics and two drop-down lists where you can select a successful or failed event to view more information about that event. In the lower right of the window is a button to click to suspend or resume a channel and a home button to click to return to the iBSE Monitoring Settings window.

The image shows a software window titled "Channel Statistics". It is divided into two main sections: "Channels" and "Statistics".

Channels Section:

- Contains two labels: "Channels" and "Ports".
- Under "Channels" is a drop-down menu showing "TestChan".
- Under "Ports" is a drop-down menu showing "TestPort".

Statistics Section:

Total Event Count	2
Total Success Count	2
Total Error Count	0
Average Event Size	446.0 bytes
Average Event Reply Size	na
Average Delivery Time	2189.0 ms
Last Delivery Time	na
Successful Events	select a correlation id
Failed Events	select a correlation id

At the bottom right of the window, there are two buttons: "Suspend Channel" and "Start Channel". Below these buttons is a button labeled "< home".

6. For more information about a successful event and its port, select an event based on its correlation ID from the Successful Events drop-down list.

The following image shows the Event Level Statistics (Message Statistics) window that opens. The Message Information pane contains a table of information pertaining to the event message. The Messages pane contains a table that shows the size of the event and reply messages, with an option to view an XML document of the event message. In the lower right of the window is a home button to click to return to the iBSE Monitoring Settings window.

Message Statistics

Message Information

Received At	2004-09-14 12:18:20.842
Disposed At	• TestPort
Delivered At	2004-09-14 12:18:23.562

Messages

Detail	size
Event Message	446 bytes
Reply Message	na

< home

- a. To view the XML event document in your Web browser, click *Event Message*.
- b. To return to the iBSE Monitoring Settings window, click *home*.

Managing and Monitoring Services and Events Using the JCA Test Tool

The JCA Test Tool, which is also known as the JCA Installation Verification Program (IVP), provides a console to manage and monitor services and events currently in use and to display resource usage and invocation statistics. These indicators can help you adjust your environment for optimum efficiency.

Procedure: How to Manage and Monitor Services Using the JCA Test Tool

To manage and monitor services using the JCA Test Tool:

1. Open a Web browser to:

<http://localhost:port/iwjcaivp>

where:

[localhost](#)

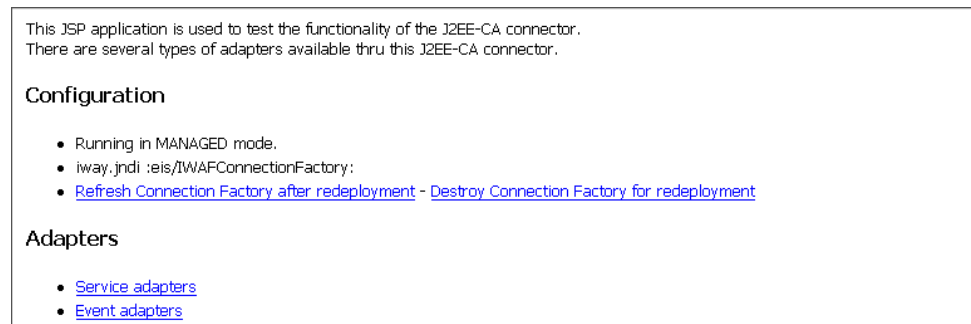
Is the name of the machine where your application server is running.

[port](#)

Is the port for the domain you are using. The port for the default domain is 7001.,for example:

<http://localhost:7001/iwjcaivp>

The following image shows the JCA Test Tool page that opens. The page contains a description of the function of the tool and configuration information, including options to change your connection settings. It also provides options for viewing service or event adapters.



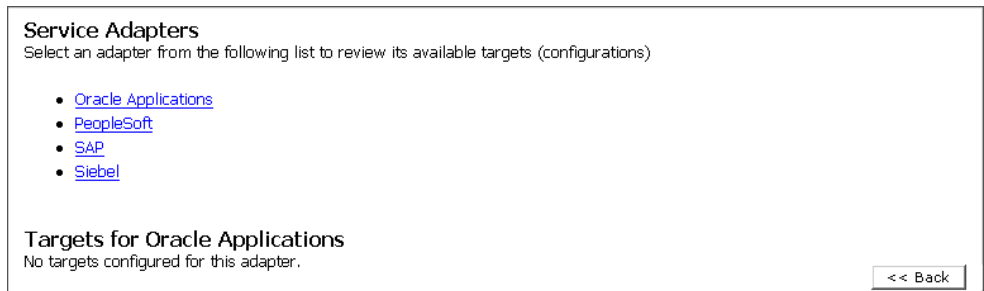
The JCA Test Tool runs in managed mode by default.

2. Perform the following steps to monitor the latest service adapter configuration.

Note: You must perform these steps for every new adapter target that is created using a JCA implementation of Application Explorer. In addition, you also must perform these steps for every new JCA configuration that is created using Application Explorer.

- a. Click *Destroy Connection Factory* for redeployment.
 - b. Redeploy the JCA connector module using the BEA WebLogic Server console.
 - c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.
3. Click *Service adapters*.

The following image shows the Service Adapters page that opens. The page provides a live list of available service adapters and a list of targets configured for a specific adapter. In the lower right is a Back button to click to return to the previous page.



4. Select a service adapter to monitor.

The following image shows the page that opens. The left side provides a live list of available service adapters and a list of any targets configured for a specific adapter. The upper right side shows statistics for a selected target. The middle right has a User field and a Password field. The lower right contains a box where you type or paste an input document. Below the input box is a Send button to click to send a request for a test service and a Reset button to click to reset the fields. In the lower right is a Back button to click to return to the previous page.

The screenshot displays the JCA Test Tool interface with the following sections:

- Service Adapters**
Select an adapter from the following list to review its available targets (configurations)
 - [Oracle Applications](#)
 - [PeopleSoft](#)
 - [SAP](#)
 - [Siebel](#)
- Targets for Siebel**
 - [TestService](#)
- Statistics for Siebel target TestService**

TotalRequestCount	: 0
TotalSuccessCount	: 0
TotalErrorCount	: 0
AverageExecutionTime	: 0 msec.
LastExecutionTime	: 0 msec.
- Request for Siebel target TestService**

Enter the data for this interaction. The configured user/password will be used if the User name is not provided.

User:

Password:

Input Doc:

- a. Click the desired target for your service adapter.
 - b. In the Request area, enter a user name and password.
 - c. In the Input Doc area, enter a request document that was created from the request schema for your service.
5. Click *Send*.

The following image shows the updated statistics that appear for your service if the request is successful. The statistics include the total number of requests, successes, and errors and the average and last execution time in milliseconds.

TotalRequestCount	: 0
TotalSuccessCount	: 0
TotalErrorCount	: 0
AverageExcecutionTime	: 0 msec.
LastExcecutionTime	: 0 msec.

Procedure: How to Manage and Monitor Events Using the JCA Test Tool

To manage and monitor events using the JCA Test Tool:

1. Open a Web browser to:

<http://localhost:port/iwjcaivp>

where:

[localhost](#)

Is the name of the machine where your application server is running.

[port](#)

Is the port for the domain you are using. The port for the default domain is 7001, for example:

<http://localhost:7001/iwjcaivp>

The following image shows the JCA Test Tool page that opens. The page contains a description of the function of the tool and configuration information, including options to change your connection settings. It also provides options for viewing service or event adapters.

This JSP application is used to test the functionality of the J2EE-CA connector.
There are several types of adapters available thru this J2EE-CA connector.

Configuration

- Running in MANAGED mode.
- `iway.jndi :eis/IWAFConnectionFactory:`
- [Refresh Connection Factory after redeployment](#) - [Destroy Connection Factory for redeployment](#)

Adapters

- [Service adapters](#)
- [Event adapters](#)

The JCA Test Tool runs in managed mode by default.

2. Perform the following steps to monitor the latest event adapter configuration.

Note: You must perform these steps for every new adapter target that is created using a JCA implementation of Application Explorer. In addition, you must also perform these steps for every new JCA configuration that is created using Application Explorer.

- a. Click *Destroy Connection Factory for redeployment*.
 - b. Redeploy the JCA connector module using the BEA WebLogic Server console.
 - c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.
3. Click *Event adapters*.

The Event Adapters page opens.

4. Select the event adapter to monitor.
5. Click the desired channel for your event adapter.
6. Click *start*.

The following image shows the updated statistics for your channel and the port. The statistics include the total number of requests, successes, and errors and the average and last execution time in milliseconds. There are options to click in the upper right of the page to start or refresh the channel.

Current channel Statistics	
Commands: start refresh	
Active: false	
TotalRequestCount	: 0
TotalSuccessCount	: 0
TotalErrorCount	: 0
AverageExcecutionTime	: 0 msec.
LastExcecutionTime	: 0 msec.
Statistics for port 'fileIN'	
TotalRequestCount	: 0
TotalSuccessCount	: 0
TotalErrorCount	: 0
AverageExcecutionTime	: 0 msec.
LastExcecutionTime	: 0 msec.

Setting Engine Log Levels

The following section describes how to set engine log levels for Servlet iBSE and JCA. For more information, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

Procedure: How to Enable Tracing for Servlet iBSE

To enable tracing for Servlet iBSE:

1. Open the Servlet iBSE configuration page at:

`http://localhost:port/ibse/IBSEConfig`

where:

`localhost`

Is the name of the machine where your application server is running.

`port`

Is the port for the domain you are using. The port for the default domain is 7001, for example:

`http://localhost:7001/ibse/IBSEConfig`

2. In the System pane, from the Debug drop-down list, select the level of tracing.
3. Click *Save*.

The default location for the trace information on Windows is:

`C:\Program Files\bea\ibse\ibselogs`

Procedure: How to Enable Tracing for JCA

To enable tracing for JCA:

1. Open the extracted ra.xml file in a text editor.
2. Locate and change the following setting:

LogLevel. This setting can be set to DEBUG, INFO, or ERROR.

```
<context-param>
<config-property>
  <config-property-name>LogLevel</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value></config-property-value>
</config-property>
```

For example:

```
<config-property-value>DEBUG</config-property-value>
```

A directory in the configuration directory contains the logs.

- a. Review the logs generated by your application server.
 - b. Leave the remainder of the previous file unchanged.
3. Save the file and exit the editor.
4. Redeploy the connector.

Configuring Connection Pool Sizes

The following topic describes how to configure connection pool sizes for the JCA connector.

Procedure: How to Configure Connection Pool Sizes

To configure connection pool sizes:

1. Open the extracted ra.xml file in a text editor.
2. Locate and change the following setting:

pool-params. The JCA Resource Connector has an initial capacity value of 0 by default and cannot be changed. The maximum capacity value is 10 by default and can be changed to a higher value.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE weblogic-connection-factory-dd (View Source for full
doctype...)>
- <weblogic-connection-factory-dd>
  <connection-factory-name>IWAFJCA</connection-factory-name>
  <jndi-name>eis/IWAFConnectionFactory</jndi-name>
  - <pool-params>
    <initial-capacity>0</initial-capacity>
    <max-capacity>10</max-capacity>
    <capacity-increment>1</capacity-increment>
    <shrinking-enabled>>false</shrinking-enabled>
    <shrink-period-minutes>200</shrink-period-minutes>
  </pool-params>
  <security-principal-map />
</weblogic-connection-factory-dd>
```

3. Save the file and exit the editor.
4. Redeploy the connector.

Migrating Repositories

During design time, a repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. For more information on configuring repositories, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

The information in the repository also is referenced at run time. For management purposes, you can migrate iBSE and JCA repositories to new destinations without affecting your existing configuration. For example, you may want to migrate a repository from a development environment to a production environment. The BEA WebLogic Server must be restarted to detect new repository changes.

File Repositories

If you want to migrate a File repository to another destination, copy the `ibserrepo.xml` file from the following path:

```
drive:\Program Files\iWay55\bea\ibse\ibserrepo.xml
```

where:

drive

Is the location of your iWay 5.5 installation.

You can place the `ibserrepo.xml` file in a new location that is a root directory of the iBSE Web application, for example:

```
drive:\ProductionConfig\bea\ibse\ibserrepo.xml
```

iBSE Repositories

The following topic describes how to migrate an iBSE repository that is configured for Oracle. You can follow the same procedure if you want to migrate an iBSE repository that is configured for Microsoft SQL Server 2000, Sybase, or DB2. However, when you are configuring a new environment, you must execute the script that creates the repository tables for your database. In addition, verify that all required files and drivers for your database are in the class path. For more information on configuring repositories, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

Note: The following procedure allows you to migrate only Web services. If migrating event handling information is one of your requirements, you must migrate at the database level. For more information, see *Migrating Event Handling Configurations* on page 7-28.

Procedure: How to Migrate an iBSE Repository Configured for Oracle

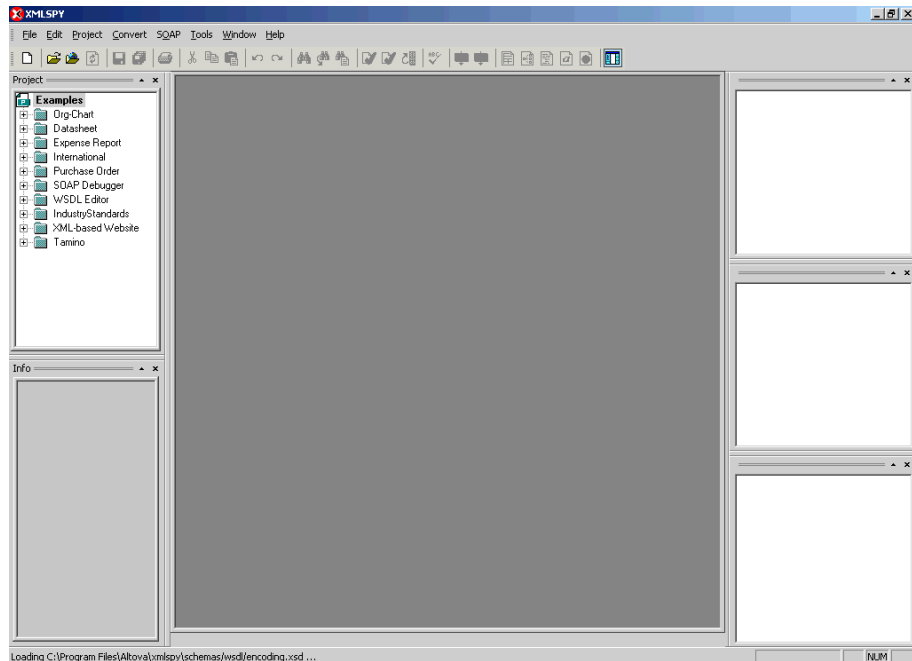
To migrate an iBSE repository that is configured for Oracle:

1. Copy the iBSE configuration service URL, for example:

<http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl>

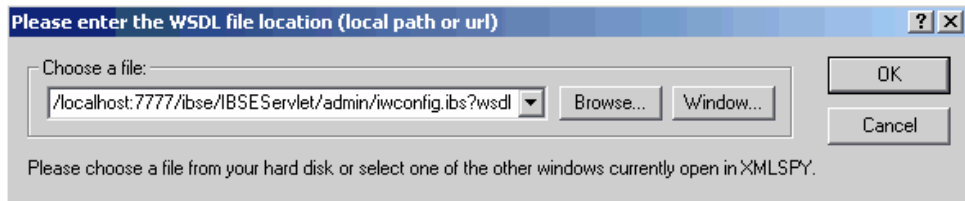
2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



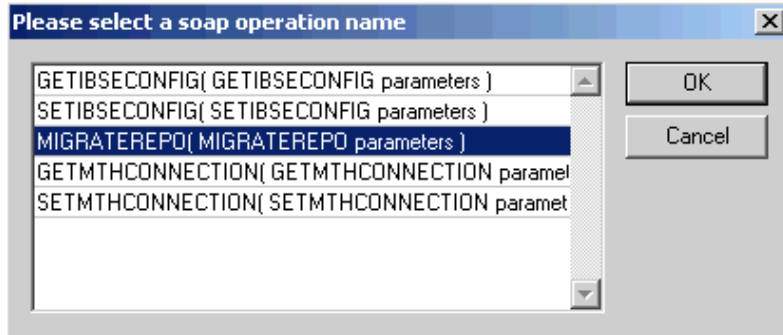
3. From the SOAP menu, select *Create new SOAP request*.

The following image shows the WSDL file location dialog box that opens, where you enter a local path or URL. The dialog includes Browse, Window, OK, and Cancel buttons.



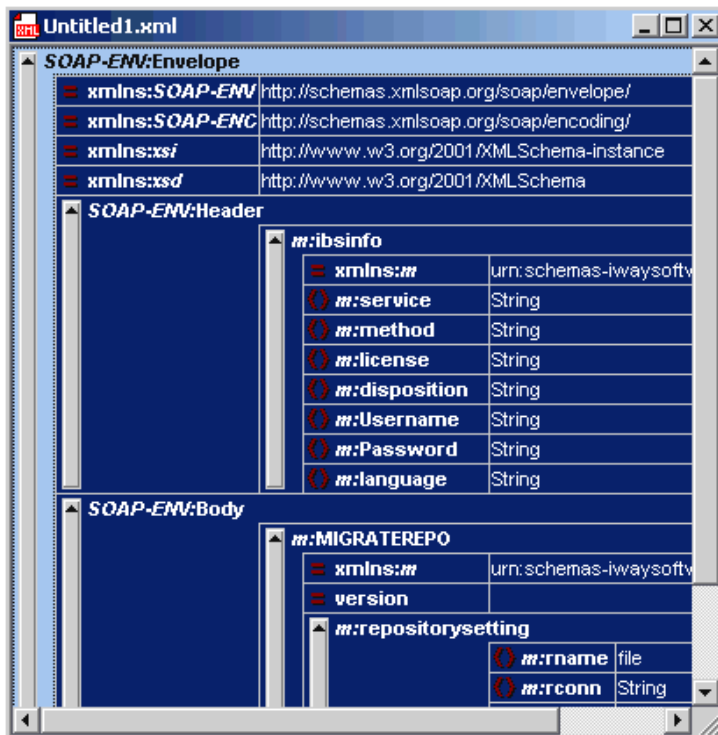
4. In the Choose a file field, paste the iBSE configuration service URL.
5. Click OK.

The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select from the list and click OK or to escape from the dialog box, you can click Cancel.



6. Select the *MIGRATEREPO(MIGRATEREPO parameters)* control method and click OK.

The following image shows a portion of the window that opens with the structure of the SOAP envelope. It includes information about location and schemas.



7. Locate the *Text view* icon in the tool bar.

In the following image, the pointer points to the *Text view* icon.



8. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

```
<m:MIGRATEREPO
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config" version="">
<m:repositorysetting>
<m:rname>oracle</m:rname>
<m:rconn>String</m:rconn>
<m:rdriver>String</m:rdriver>
<m:ruser>String</m:ruser>
<m:rpwd>String</m:rpwd>
</m:repositorysetting>
<m:servicename>String</m:servicename>
</m:MIGRATEREPO>
```

- a. For the <m:rconn> tag, replace the String placeholder with the repository URL where you want to migrate your existing iBSE repository.

For example, the Oracle repository URL has the following format:

```
jdbc:oracle:thin:@[host]:[port]:[sid]
```

- b. For the <m:rdriver> tag, replace the String placeholder with the location of your Oracle driver.

Note: This is an optional tag. If you do not specify a value, the default Oracle JDBC driver is used.

- c. For the <m:ruser> tag, replace the String placeholder with a valid user name to access the Oracle repository.
- d. For the <m:rpwd> tag, replace the String placeholder with a valid password to access the Oracle repository.

10. Perform one of the following migration options.

If you want to migrate a **single** Web service from the current iBSE repository, enter the Web service name in the <m:servicename> tag, for example:

```
<m:servicename>Service1</m:servicename>
```

If you want to migrate **multiple** Web services from the current iBSE repository, duplicate the <m:servicename> tag for each Web service, for example:

```
<m:servicename>Service1</m:servicename>
<m:servicename>Service2</m:servicename>
```

If you want to migrate **all** Web services from the current iBSE repository, remove the <m:servicename> tag.

11. From the SOAP menu, select *Send request to server*.

Your iBSE repository and the Web services you specified migrate to the new Oracle repository URL that you specified.

JCA Repositories

The following procedure describes how to migrate a JCA repository. For more information on configuring JCA repositories, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

Procedure: How to Migrate a JCA Repository

To migrate a JCA repository:

1. Navigate to the location of your JCA configuration directory where the repository schemas and other information is stored, for example:
`C:\Program Files\iWay55\config\base`
2. Locate and copy the *repository.xml* file.
3. Place this file in a new JCA configuration directory to migrate the existing repository.

Your JCA repository migrates to the new JCA configuration directory.

Migrating Event Handling Configurations

This topic describes how to migrate your iBSE repositories at a database level for Microsoft SQL Server 2000, Oracle, Sybase, or DB2. You can use this information to migrate event handling information, for example, port or channel configurations.

Procedure How to Migrate a Microsoft SQL Server 2000 Repository

To migrate a Microsoft SQL Server 2000 repository:

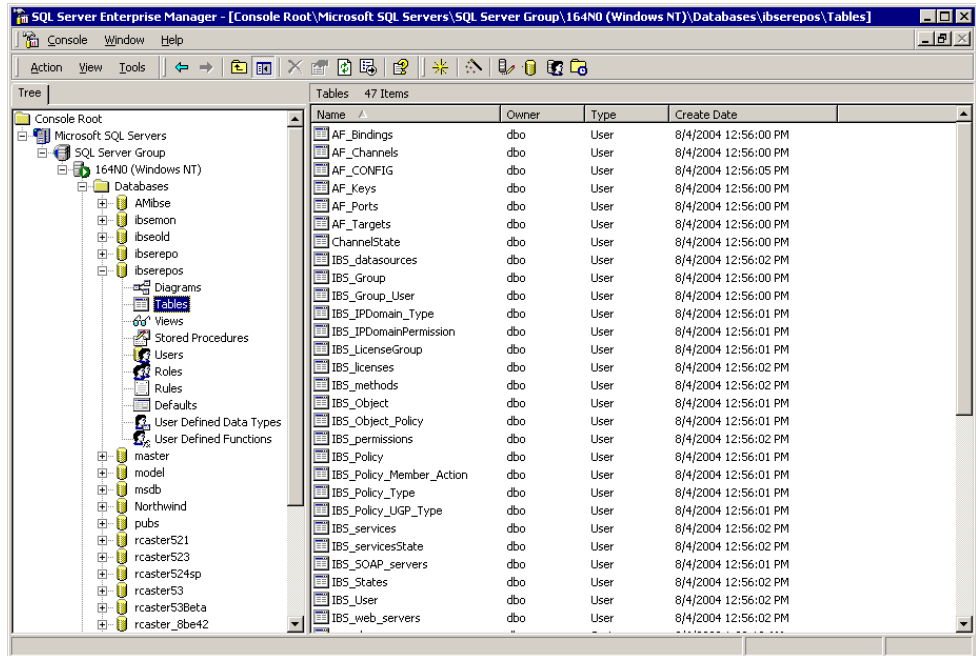
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

`C:\Program Files\iWay55\etc\setup`

This directory contains SQL to create the repository tables in the following file:

`iwse.sql`

You can use `iwse.sql` to create the database tables that are used by iBSE. For example, the following image shows the tree in the left pane and tables in the right pane. The tables are listed by name in one column with corresponding columns for information about owner, type, and the date the table was created.



For more information on configuring the Microsoft SQL Server 2000 repository, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

2. To migrate the tables that were created by the `iwse.sql` script for iBSE, use your Microsoft SQL Server 2000 database tool set. For more information, consult your database administrator.

Procedure How to Migrate an Oracle Repository

To migrate an Oracle repository:

1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

`C:\Program Files\iWay55\etc\setup`

This directory contains SQL to create the repository tables in the following files:

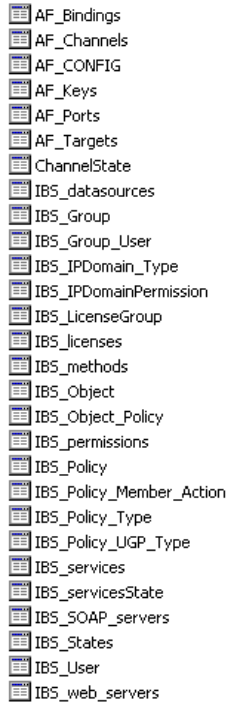
For Oracle 8:

`iwse.ora`

For Oracle 9:

[iwse.ora9](#)

2. To create the Oracle database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



A screenshot of a list of Oracle database tables. Each table name is preceded by a small icon of a document with a list. The tables listed are:

- AF_Bindings
- AF_Channels
- AF_CONFIG
- AF_Keys
- AF_Ports
- AF_Targets
- ChannelState
- IB5_datasources
- IB5_Group
- IB5_Group_User
- IB5_IPDomain_Type
- IB5_IPDomainPermission
- IB5_LicenseGroup
- IB5_licenses
- IB5_methods
- IB5_Object
- IB5_Object_Policy
- IB5_permissions
- IB5_Policy
- IB5_Policy_Member_Action
- IB5_Policy_Type
- IB5_Policy_UGP_Type
- IB5_services
- IB5_servicesState
- IB5_SOAP_servers
- IB5_States
- IB5_User
- IB5_web_servers

For more information on configuring the Oracle repository, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

3. To migrate the tables that were created by the SQL script for iBSE, use your Oracle database tool set. For more information, consult your database administrator.

Procedure How to Migrate a Sybase Repository

To migrate a Sybase repository:

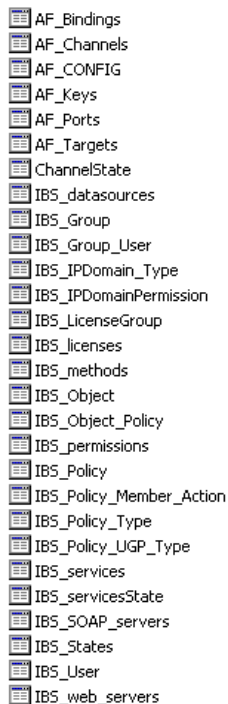
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

[C:\Program Files\iWay55\etc\setup](#)

This directory contains SQL to create the repository tables in the following file:

[sybase-iwse.sql](#)

2. To create the Sybase database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



A screenshot of a database table list, likely from a Sybase Enterprise Manager or similar tool. The list contains the following tables:

- AF_Bindings
- AF_Channels
- AF_CONFIG
- AF_Keys
- AF_Ports
- AF_Targets
- ChannelState
- IB5_datasources
- IB5_Group
- IB5_Group_User
- IB5_IPDomain_Type
- IB5_IPDomainPermission
- IB5_LicenseGroup
- IB5_Licenses
- IB5_methods
- IB5_Object
- IB5_Object_Policy
- IB5_permissions
- IB5_Policy
- IB5_Policy_Member_Action
- IB5_Policy_Type
- IB5_Policy_UGP_Type
- IB5_services
- IB5_servicesState
- IB5_SOAP_servers
- IB5_States
- IB5_User
- IB5_web_servers

For more information on configuring the Sybase repository, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

3. To migrate the tables that were created by the SQL script for iBSE, use your Sybase database tool set. For more information, consult your database administrator.

Procedure How to Migrate a DB2 Repository

To migrate a DB2 repository:

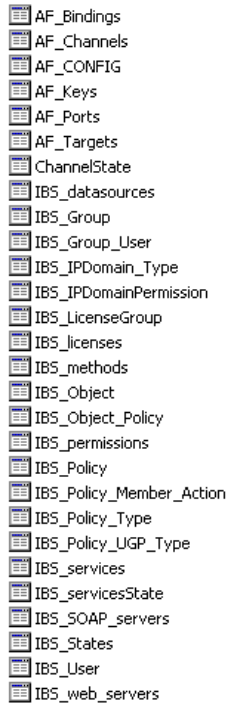
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

`C:\Program Files\iWay55\etc\setup`

This directory contains SQL to create the repository tables in the following file:

`db2-iwse.sql`

2. To create the DB2 database tables that are used by iBSE, use the SQL script as shown in the example in the following image that shows a list of tables.



AF_Bindings
AF_Channels
AF_CONFIG
AF_Keys
AF_Ports
AF_Targets
ChannelState
IB5_datasources
IB5_Group
IB5_Group_User
IB5_IPDomain_Type
IB5_IPDomainPermission
IB5_LicenseGroup
IB5_licenses
IB5_methods
IB5_Object
IB5_Object_Policy
IB5_permissions
IB5_Policy
IB5_Policy_Member_Action
IB5_Policy_Type
IB5_Policy_UGP_Type
IB5_services
IB5_servicesState
IB5_SOAP_servers
IB5_States
IB5_User
IB5_web_servers

For more information on configuring the DB2 repository, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

You can migrate the tables that were created by the SQL script for iBSE using your DB2 database toolset. For more information, consult your database administrator.

Exporting or Importing Targets

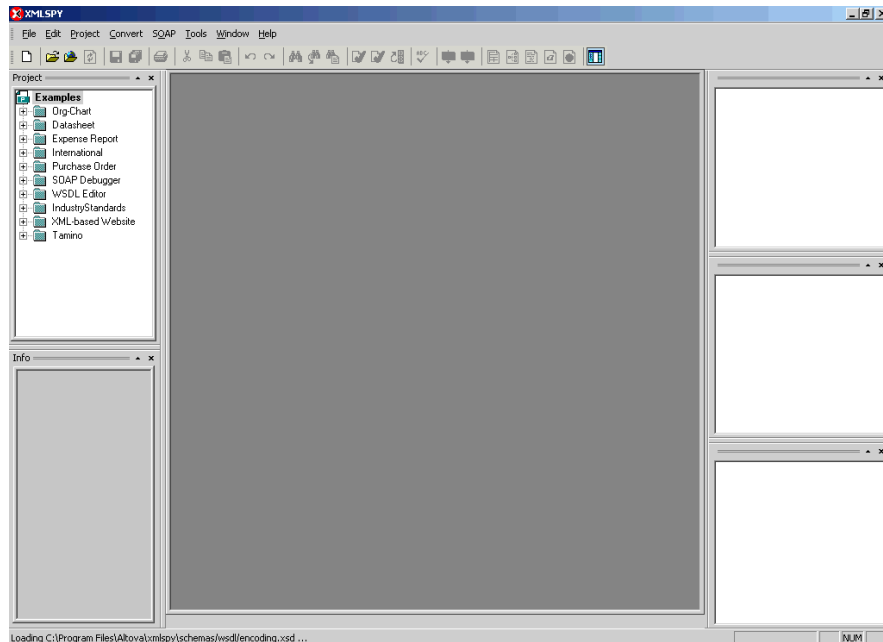
After you migrate your repository, you can export or import targets with their connection information and persistent data between repositories.

Procedure: How to Export a Target

To export a target:

1. Copy the iBSE administrative services for Application Explorer URL, for example:
<http://localhost:7777/ibse/IBSEServlet/admin/iwae.ibs?wsdl>
2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE administrative services for Application Explorer URL.
5. Click OK.

The soap operation name dialog box opens and lists the available control methods.

6. Select the *EXPORTTARGET(EXPORTTARGET parameters)* control method and click OK.

A window opens that shows the structure of the SOAP envelope.

7. Locate the *Text view* icon in the tool bar.

In the following image, the pointer points to the Text view icon.



8. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

```
<m:EXPORTTARGET  
xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:af">  
<m:target>String</m:target>  
<m:name>String</m:name>  
</m:EXPORTTARGET>
```

- a. For the <m:target> tag, replace the String placeholder with the EIS target system name as it appears in Application Explorer and verify whether this value is case sensitive.
 - b. For the <m:name> tag, replace the String placeholder with the name of the target you want to export.
10. From the SOAP menu, select *Send request to server*.

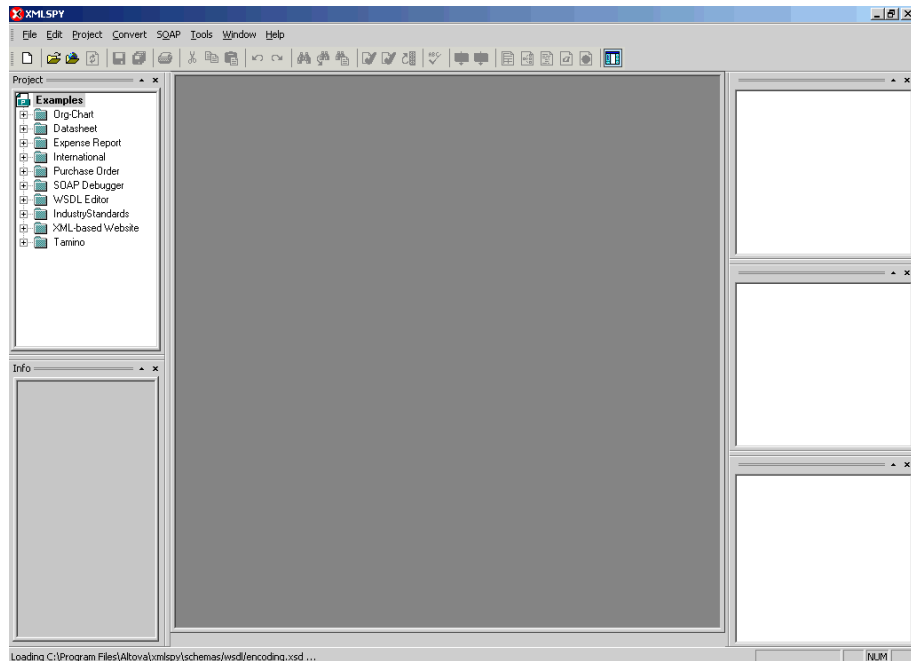
A response is returned that contains the <m: exporttime> and <m: contents> elements. You must use these elements when importing your target.

Procedure: How to Import a Target

To import a target:

1. Copy the iBSE administrative services for Application Explorer URL, for example:
<http://localhost:7777/ibse/IBSEServlet/admin/iwae.ibs?wsdl>
2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE administrative services for Application Explorer URL and click *OK*.

The soap operation name dialog box opens and lists the available control methods.

5. Select the *IMPORTTARGET(IMPORTTARGET parameters)* control method and click *OK*.

A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:IMPORTTARGET
xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:af">
<m:targetinstance>
<m:target>String</m:target>
<m:name>String</m:name>
<m:description>String</m:description>
<m:repositoryid>String</m:repositoryid>
<m:exporttime>2001-12-17T09:30:47-05:00</m:exporttime>
<m:contents>R01GODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b</m:contents>
</m:targetinstance>
</m:IMPORTTARGET>
```

- a. For the <m:target> tag, replace the String placeholder with the EIS target system name.
 - b. For the <m:name> tag, replace the String placeholder with the new name of the target you want to import.
 - c. For the <m:description> tag, replace the String placeholder with a description of the target.
 - d. For the <m:repositoryid> tag, copy and paste the contents of the <m:repositoryid> tag that was returned when you exported your target.
 - e. For the <m: exporttime> tag, copy and paste the contents of the <m: exporttime> tag that was returned when you exported your target.
 - f. For the <m: contents> tag, copy and paste the contents of the <m: contents> tag that was returned when you exported your target.
- 9.** From the SOAP menu, select *Send request to server*.

Retrieving or Updating Web Service Method Connection Information

After you migrate your repository, you can retrieve or update connection information for your Web service methods.

Procedure: How to Retrieve Web Service Method Connection Information

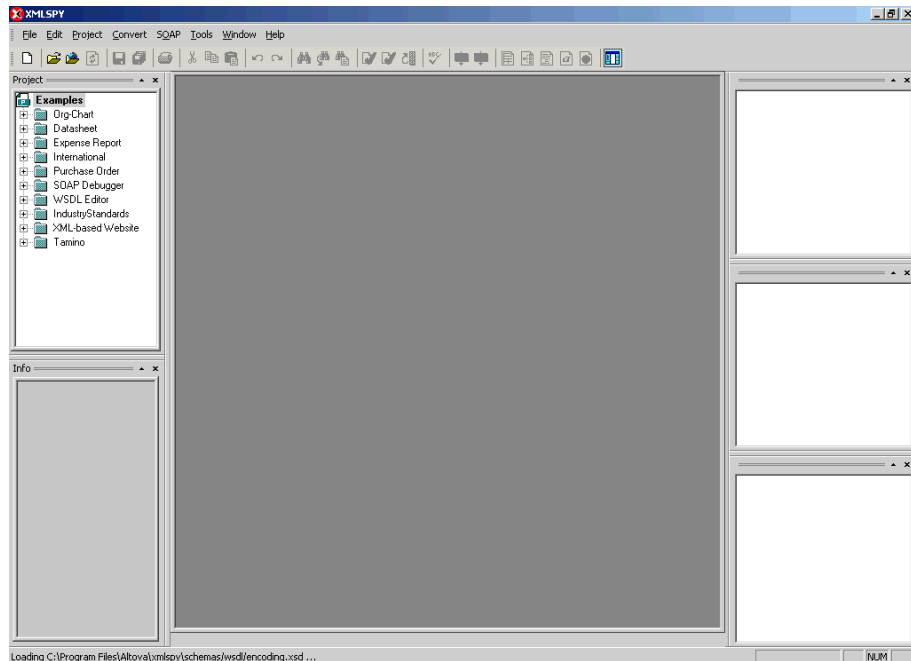
To retrieve Web service method connection information:

1. Copy the iBSE configuration service URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl
```

2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE configuration service URL, and click *OK*.

The soap operation name dialog box opens and lists the available control methods.

5. Select the *GETMTHCONNECTION(GETMTHCONNECTION parameters)* control method and click *OK*.

A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:GETMTHCONNECTION  
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config">  
<m:serviceName>String</m:serviceName>  
<m:methodName>String</m:methodName>  
</m:GETMTHCONNECTION>
```

- a. For the <m:serviceName> tag, replace the String placeholder with the name of the Web service.
 - b. For the <m:methodName> tag, replace the String placeholder with name of the Web service method.
9. From the SOAP menu, select *Send request to server*.

A response is returned that contains the <m: descriptor> element. You must use this element when updating your Web service method.

Procedure: How to Update Web Service Method Connection Information

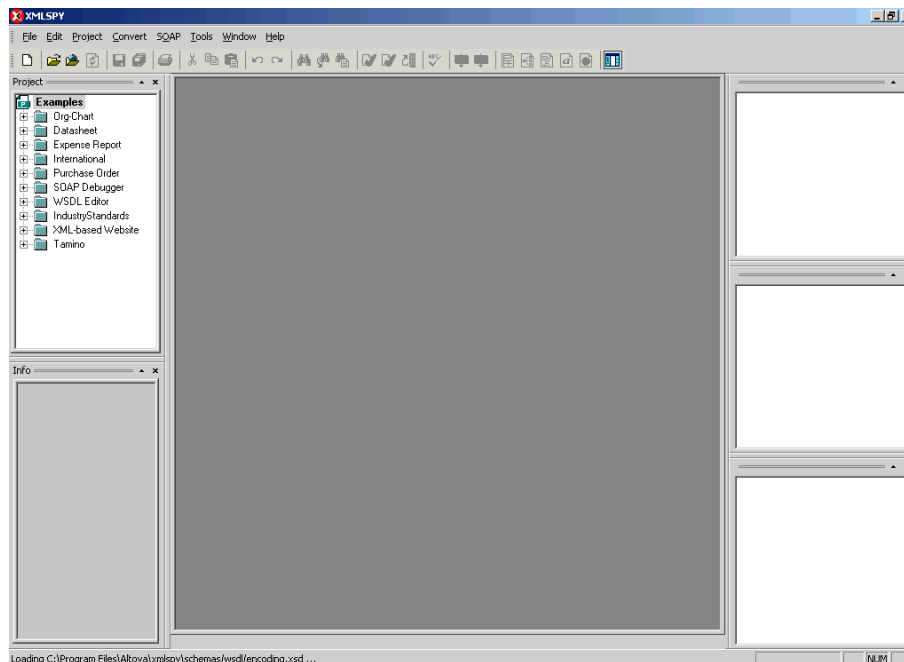
To update Web service method connection information:

1. Copy the iBSE configuration service URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwconfig.ibs?wsdl
```

2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.



3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE configuration service URL, and click *OK*.

The soap operation name dialog box opens and lists the available control methods.

5. Select the *SETMTHCONNECTION(SETMTHCONNECTION parameters)* control method and click *OK*.

A window opens that shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The <SOAP-ENV:Header> tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<m:SETMTHCONNECTION
xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config">
<m:servicename>String</m:servicename>
<m:methodname>String</m:methodname>
<m:descriptor format=" " channel=" ">
    <m:option title=" ">
        <m:group title=" ">
            <m:param/>
        </m:group>
    </m:option>
</m:descriptor>
</m:SETMTHCONNECTION>
```

- a. For the <m:servicename> tag, replace the String placeholder with the name of the Web service.
 - b. For the <m:methodname> tag, replace the String placeholder with the name of the Web service method.
 - c. For the <m: descriptor> tag, copy and paste the contents of the <m: descriptor> tag that was returned when you retrieved Web Service method connection information.
- 9.** Modify the contents of the <m: descriptor> tag to change the existing Web Service method connection information.
- 10.** From the SOAP menu, select *Send request to server*.

Starting or Stopping a Channel Programmatically

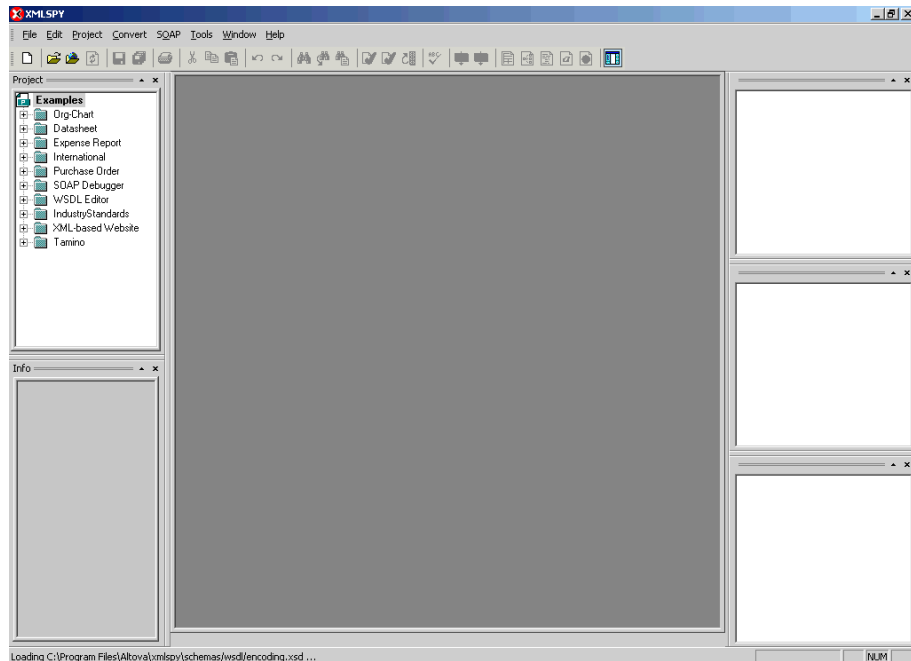
The following topic describes how to start or stop a channel programmatically.

Procedure: How to Start a Channel Programmatically

To start a channel programmatically:

1. Copy the iBSE control event URL, for example:
<http://localhost:7777/ibse/IBSEServlet/admin/iwevent.ibs?wsdl>
2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.

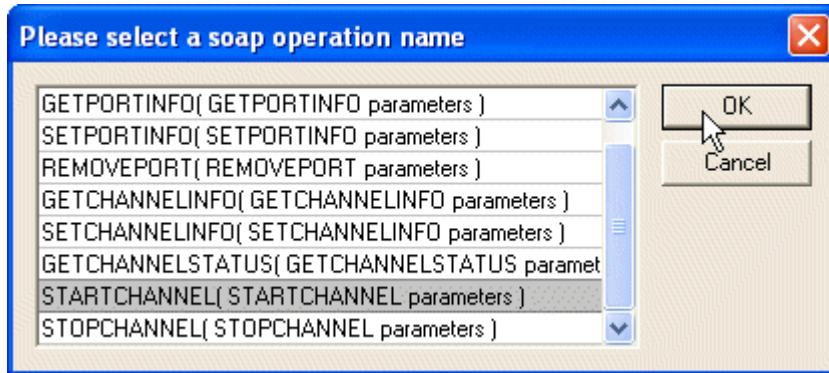


3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE control event URL, and click *OK*.

The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select one and click OK or to escape from the dialog box, you can click Cancel.



5. Select the *STARTCHANNEL*(*STARTCHANNEL parameters*) control method and click *OK*.

A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the *Text view* icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The `<SOAP-ENV:Header>` tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STARTCHANNEL
    xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
    <m:channel>String</m:channel>
  </m:STARTCHANNEL>
</SOAP-ENV:Body>
```

9. For the `<m:channel>` tag, replace the *String* placeholder with the name of the Channel you want to start.
10. From the SOAP menu, select *Send request to server*.

Procedure: How to Stop a Channel Programmatically

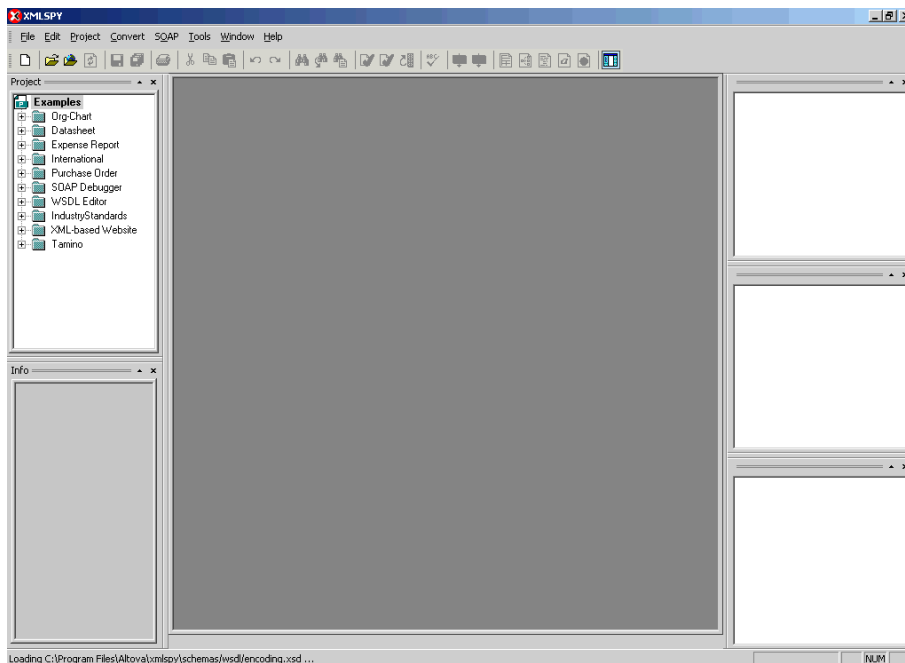
To stop a channel programmatically:

1. Copy the iBSE control event URL, for example:

<http://localhost:7777/ibse/IBSEServlet/admin/iwevent.ibs?wsdl>

2. Open a third party XML editor, for example, XMLSPY.

The following image shows the XMLSPY window. The upper left has a Project pane that contains a tree of sample files, and the lower left has a blank Info pane. The middle pane is blank. The right side is divided into three blank panes.

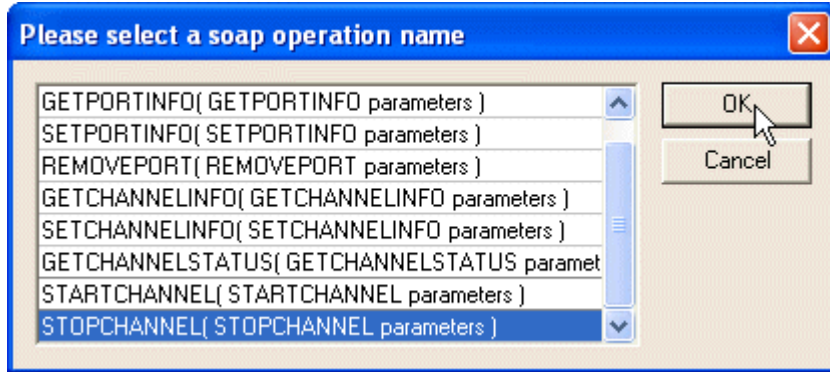


3. From the SOAP menu, select *Create new SOAP request*.

The WSDL file location dialog box opens.

4. In the Choose a file field, paste the iBSE control event URL, and click *OK*.

The following image shows the soap operation name dialog box that opens with a list of available control methods. You can select one and click OK or to escape from the dialog box, you can click Cancel.



5. Select the *STOPCHANNEL(STOPCHANNEL parameters)* control method and click *OK*.

A window opens, which shows the structure of the SOAP envelope.

6. Locate the *Text view* icon in the toolbar.

In the following image, the pointer points to the Text view icon.



7. To display the structure of the SOAP envelope as text, click the *Text view* icon.

The `<SOAP-ENV:Header>` tag is not required and can be deleted from the SOAP envelope.

8. Locate the following section:

```
<SOAP-ENV:Body>
  <m:STOPCHANNEL
    xmlns:m="urn:schemas-iwaysoftware-com:dec2002:iwse:event">
    <m:channel>String</m:channel>
  </m:STOPCHANNEL>
</SOAP-ENV:Body>
```

9. For the `<m:channel>` tag, replace the String placeholder with the name of the Channel you want to stop.
10. From the SOAP menu, select *Send request to server*.

APPENDIX A

Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services With CBO

Topics:

- Starting Application Explorer in BEA WebLogic Workshop
- Creating a New Configuration
- Connecting to and Managing a Target
- Creating an XML Schema
- Creating an Integration Business Service
- Adding a Control for an iWay Resource in BEA WebLogic Workshop
- Adding an Extensible CCI Control

This section describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas and Web services with ClarifyCRM Business Objects (CBO).

In addition, this section explains how to add a control to a Web service or BEA WebLogic Workshop application.

Starting Application Explorer in BEA WebLogic Workshop

The server must be started where iWay Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic Server.

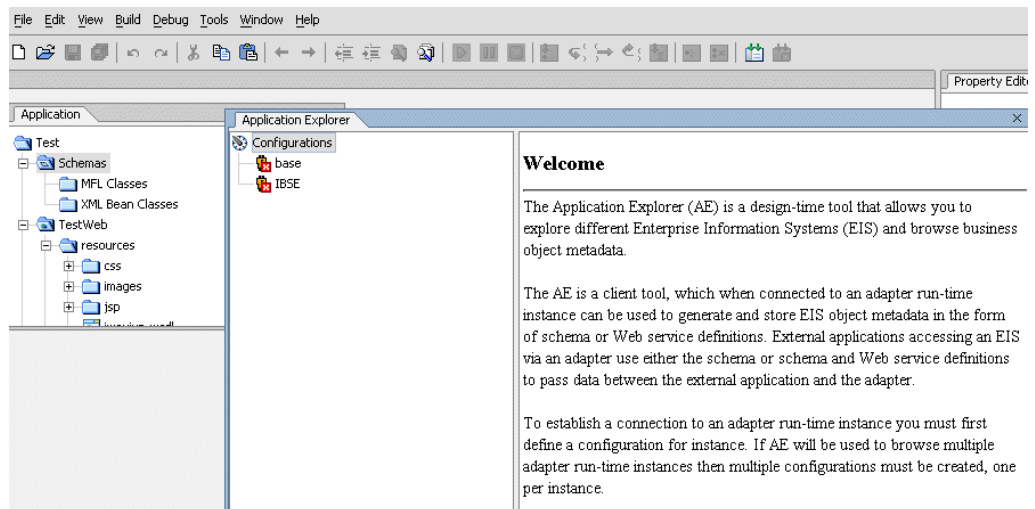
You can run Application Explorer in BEA WebLogic Workshop using an Integration Business Services Engine (IBSE) configuration or J2EE Connector Architecture (JCA) configuration.

Procedure: How to Start Application Explorer in BEA WebLogic Workshop

To start Application Explorer in BEA WebLogic Workshop:

1. Ensure that BEA WebLogic Server is running.
2. Start BEA WebLogic Workshop.
3. From BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer*.

Application Explorer opens in BEA WebLogic Workshop, as shown in the following image. This image also shows the Application Explorer welcome statement that appears on the right when you select the Configurations node on the left.



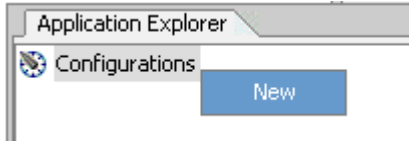
You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

Creating a New Configuration

Before you can start using Application Explorer, you must define a new configuration for iBSE or JCA.

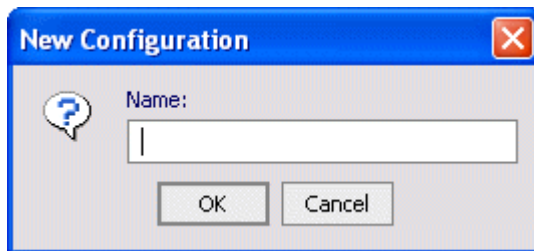
Procedure: How to Create a New Configuration for iBSE or JCA

To create a new configuration:



1. Right-click *Configurations* and select *New*.

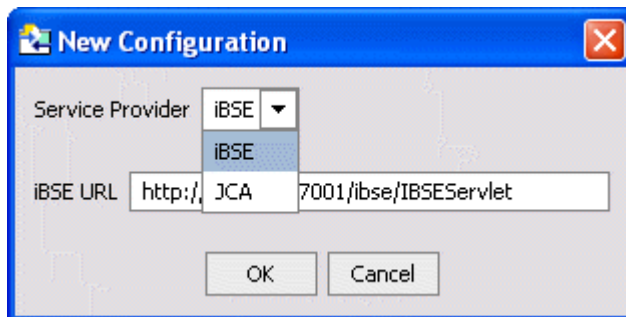
The New Configuration dialog box opens, as shown in the following image.



2. Type the name of the new configuration and click *OK*.

Note: If you are creating a new JCA configuration, type *base* in the name field. You must use this value if you are pointing to the default iWay configuration.

The following dialog box opens, as shown in the following image.



3. From the Service Provider drop-down list, select *iBSE* or *JCA*.

If you select iBSE, type the URL for iBSE, for example,

<http://localhost:7001/ibse/IBSEServlet>

where:

[localhost](#)

Is where your application server is running.

If you select JCA, enter the full path to the directory where iWay 5.5 is installed, for example,

[C:\Program Files\iWay55](#)

where:

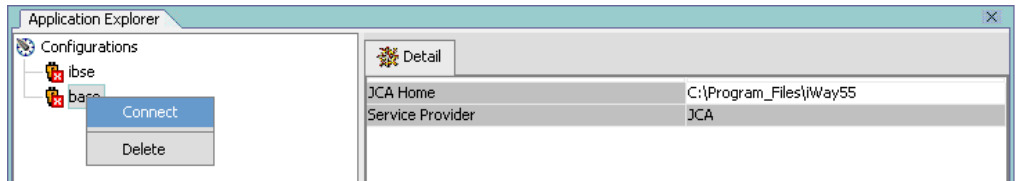
[iWay55](#)

Is the full path to your iWay installation.

4. Click OK.

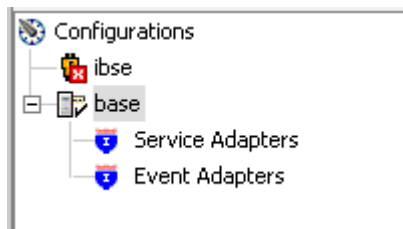
A node representing the new configuration appears under the Configurations node. The right pane displays details of the configuration you created.

After you add your configuration, you must connect to it.

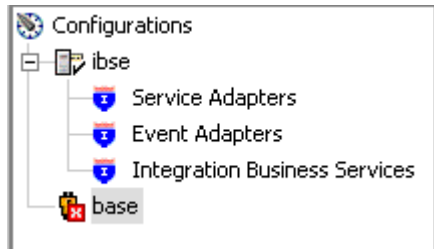


5. Right-click the configuration to which you want to connect and select *Connect*.

When you connect to base, the Service Adapters and Event Adapters nodes appear, as shown in the following image.



When you connect to iBSE, the Service Adapters, Event Adapters, and Integration Business Services nodes appear, as shown in the following image.



6. To display the service and event adapters that are installed, expand each node.

The Service Adapters list includes a ClarifyCRM node that enables you to connect to ClarifyCRM metadata and create XML request and response schemas to use to listen for events or create Web services. For more information, see *Creating an Integration Business Service* on page A-12.

The Event Adapters list includes a ClarifyCRM node that enables you to create ports and channels for ClarifyCRM event handling. For more information, see Appendix C, *Using Application Explorer in BEA WebLogic Workshop for Event Handling with CBOs*.

Connecting to and Managing a Target

To browse ClarifyCRM, you must create a ClarifyCRM target and connect to it. The target serves as your connection point. You must establish a connection to Clarify every time you start iWay Application Explorer or after you disconnect from ClarifyCRM.

The left pane displays the application systems supported by Application Explorer. These are based on the adapters you installed and are licensed to use.

Creating and Connecting to a Target

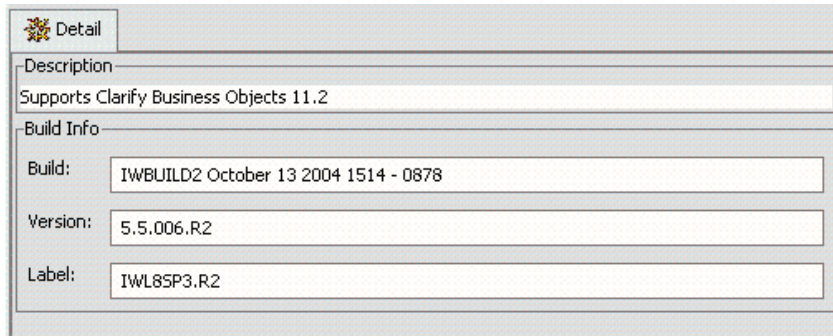
To connect to ClarifyCRM for the first time, you must create a new target. The target is automatically saved after it is created.

Procedure: How to Create a New Target

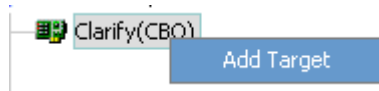
To create a new target:

1. In the left pane, expand *Service Adapters* and click the *Clarify(CBO)* node.

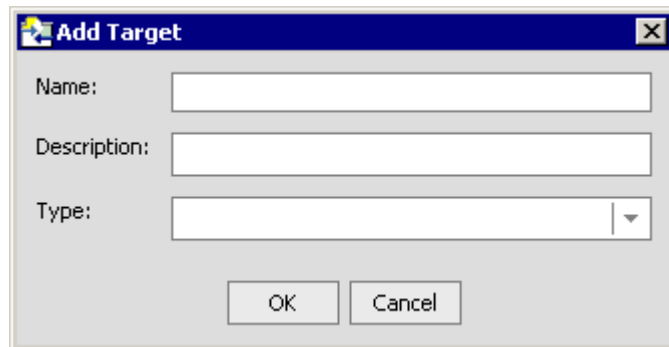
Descriptive information (for example, build and version) for the adapter appears in the right pane. The following image is an example of the details for the *Clarify(CBO)* node.



2. Right-click the *Clarify(CBO)* node and select *Add Target*.



The Add target dialog box opens. The dialog box provides three fields to define the target and two action buttons.



- a. In the Name field, type a descriptive name for the target, for example, ClarifyTarget.
- b. In the Description field, type a brief description of the target.

The Type field is not currently available.

3. Click *OK*.

The Basic dialog box opens in which you must specify connection information for Clarify and the application server that is hosting ClarifyCRM.

The image shows a 'Basic' dialog box with four input fields, each followed by an asterisk to indicate they are required:

- Clarify Server Name*
- Clarify Database Name*
- User Name*
- User Password*

At the bottom of the dialog are 'OK' and 'Cancel' buttons. Below the dialog, a red text label states: 'Fields marked with * are required.'

Note: The ClarifyCRM connection parameters are consistent with those found in your ClarifyCRM system. For more information on parameter values that are specific to your ClarifyCRM configuration, consult your ClarifyCRM system administrator.

- a. In the Clarify Server Name field, type the name of the ClarifyCRM server.
 - b. In the Clarify Database Name field, type the name of the ClarifyCRM database.
 - c. In the User Name field, type your user ID for the server.
 - d. In the User Password, type the user password for the server.
4. Click OK.

In the left pane, the new target (ClarifyTarget) appears below the Clarify(CBO) node.

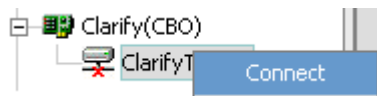


You can now connect to the target you defined.

Procedure: How to Connect to a Target

To connect to a Clarify(CBO) target:

1. In the left pane, expand the *Clarify(CBO)* node and select the target to which you want to connect.



2. Right-click the target and select *Connect*.

The ClarifyTarget node in the left pane changes to reflect that a connection was made.

3. Expand the target node to reveal the list of interfaces. The following image shows an example of an expanded Clarify(CBO) target.



Managing a Target

Although you can maintain multiple open connections to different application systems, we recommend that you close connections when they are not in use. After you disconnect, you can modify an existing target.

You can modify the connection parameters when your system properties change. You also can delete a target. The following procedures describe how to disconnect from a target, edit a target, and delete a target.

Procedure: How to Disconnect From a Target

To disconnect from a target:



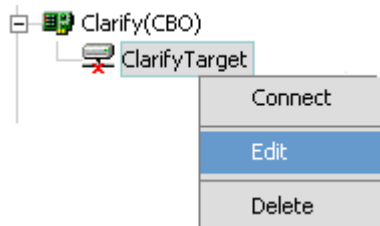
1. Right-click the target from which you want to disconnect.
2. Select *Disconnect*.

Disconnecting from the application system drops the connection, but the node remains. The target node appearance in the left pane changes to reflect that you disconnected from the target.

Procedure: How to Edit a Target

To edit a target:

1. Ensure that the target you want to edit is disconnected.



2. In the left pane, right-click the target and select *Edit*.

The Basic window opens, displaying connection information. Enter your password and click *OK*.

The following dialog box opens.

Clarify Server Name*	<input type="text" value="clarifyserver"/>
Clarify Database Name*	<input type="text" value="clarify"/>
User Name*	<input type="text" value="cl"/>
User Password*	<input type="password" value="**"/>

3. Change the properties in the dialog box as required and click *OK*.

Procedure: How to Delete a Target

To delete a target:

1. Right-click the target.
2. Select *Delete*.

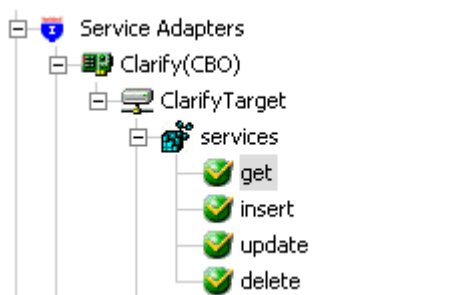
The target node disappears from the left pane.

Creating an XML Schema

After you create a new configuration and connect to the target, Application Explorer enables you to create a request or response schema.

Procedure: How to Create a Request and Response Schema

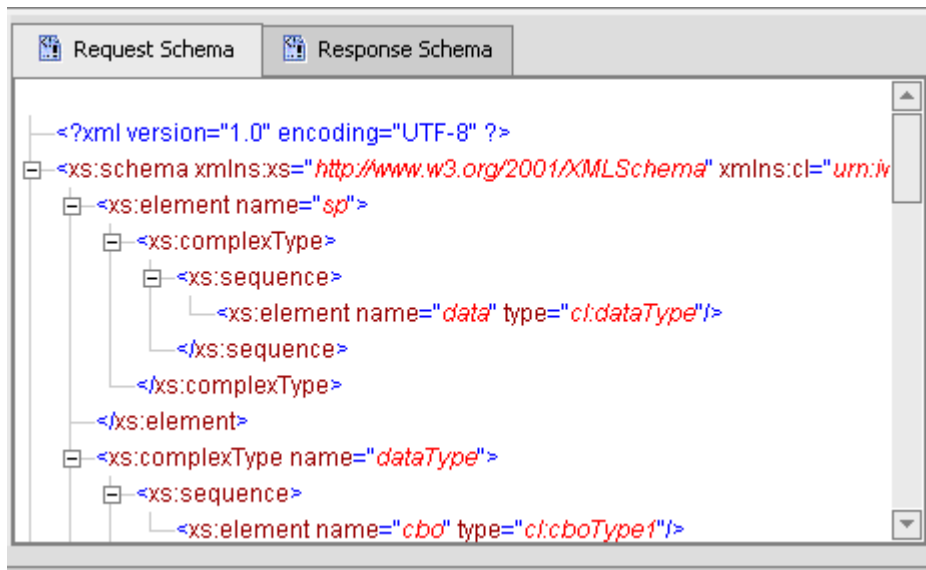
To create a request and response schema:



1. Under Service Adapters, expand the *Clarify(CBO)* node, the target node, and the *services* node, and select the service for which you want to create the schema.

The following XML schemas appear for the interface:

- Request
- Response

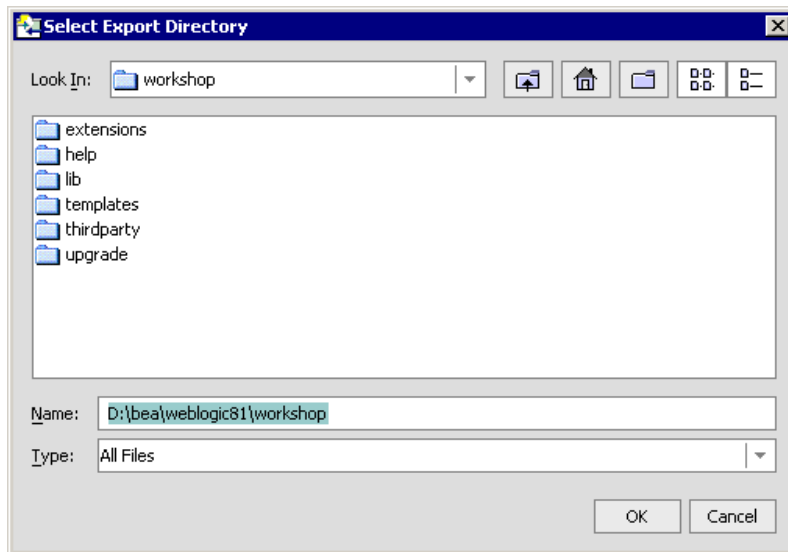


2. To view the appropriate schema in the right pane, click the *Request Schema* or the *Response Schema* tab.

The schema you select appears in the pane.

3. To save the schema:
 - a. In the left pane, right-click the service to which the schema belongs and select *Export Schema(s)* from the drop-down list.

The Select Export Directory window opens, as shown in the following image.



- b. Browse to the directory where you want the schema to reside and type a name for the schema file.
 - c. Click OK.

Schema Location

After you browse the Component Interfaces and make a selection, the request and response XML schemas are automatically created for that Component Interface and stored in the repository you created, for example,

`drive:\Program Files\iWay55\bea\ibse\wsdl\schemas\service\ClarifyCRM\ClarifyTarget\SA45280C`

where:

ClarifyTarget

Is the name of the Clarify target.

SA45280C

Is a randomly generated folder name indicating where the schemas are stored.

Creating an Integration Business Service

You can create an integration business service (also known as a Web service) for objects you want to use with your adapter. To generate a business service, you must deploy the adapter using the Integration Business Services Engine (iBSE). iBSE exposes functionality as Web services and serves as a gateway to heterogeneous back-end applications and databases.

A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered as a “black box” that may require input and delivers a result. Web services integrate within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

You can make a Web service available to other services within a host server by generating WSDL (Web Services Description Language) from the Web service.

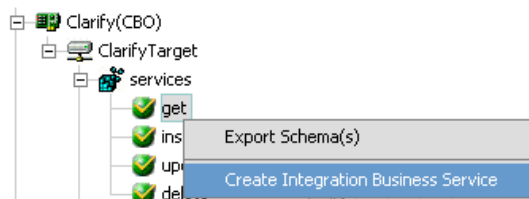
Because Application Explorer runs within BEA WebLogic Workshop, you can easily incorporate Web services into BEA WebLogic Workflows. To enable BEA WebLogic Workshop to use Web services, export the WSDL to a directory accessible to BEA WebLogic Workshop.

Note: In a J2EE Connector Architecture (JCA) implementation of iWay adapters, Web services are not available. When the adapters are deployed to use the iWay Connector for JCA, the Common Client Interface provides integration services using the iWay adapters. For more information, see the *iWay Installation and Configuration for BEA WebLogic* manual and the *iWay Connector for JCA for BEA WebLogic Server User's Guide*.

Procedure: How to Create an Integration Business Service

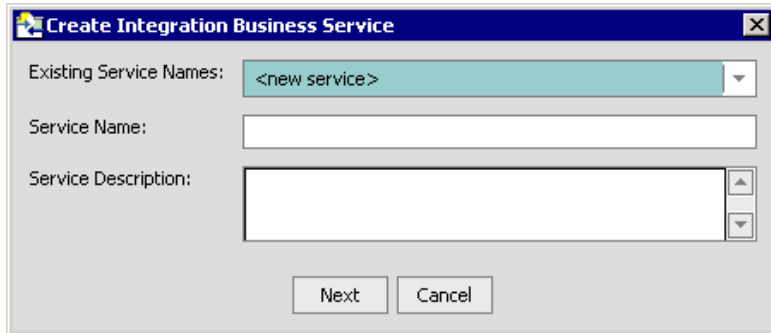
To create an Integration Business Service:

1. Expand the target node and the *services* node, and select the interface for which you want to create a business service.



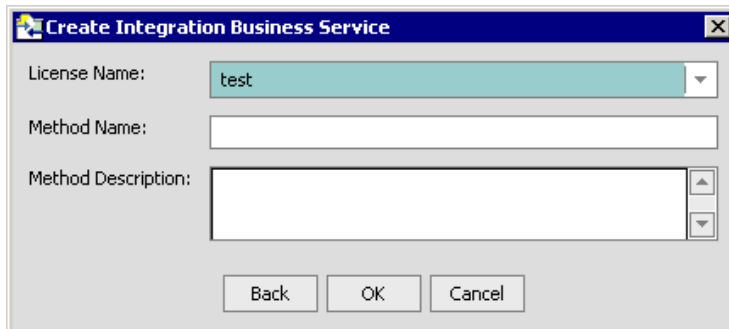
2. Right-click the service and select *Create Integration Business Service*.

The Create Integration Business Service dialog box opens.

The dialog box is titled "Create Integration Business Service". It contains three input fields: "Existing Service Names:" with a drop-down menu showing "<new service>", "Service Name:" with a text box, and "Service Description:" with a text box and vertical scroll bars. At the bottom are "Next" and "Cancel" buttons.

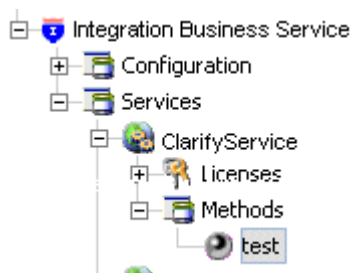
- a. From the Existing Service Names drop-down list, select whether you want to create a new service name or use an existing service name.
 - b. In the Service Name field, type a name if it is a new business service, for example, ClarifyService.
 - c. In the Service Description field, type a brief description if it is a new business service.
3. Click *Next*.

The Create Integration Business Service dialog box displays additional fields.

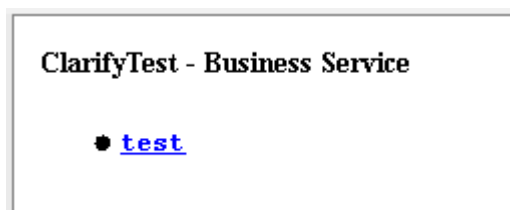
The dialog box is titled "Create Integration Business Service". It contains three input fields: "License Name:" with a drop-down menu showing "test", "Method Name:" with a text box, and "Method Description:" with a text box and vertical scroll bars. At the bottom are "Back", "OK", and "Cancel" buttons.

- a. From the License Name drop-down list, select a license.
 - b. In the Method Name field, type a name for the method.
 - c. In the Method Description field, type a brief description for the method.
4. Click *OK*.

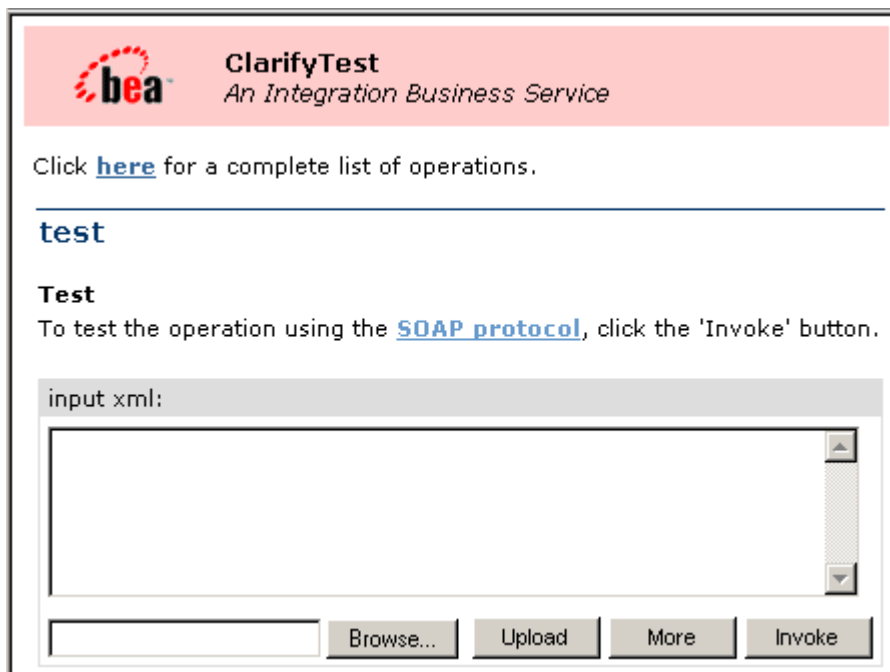
The business service and method appear below the Integration Business Service node.



5. In the right pane, click the method name, for example, test.



On the right, the test pane opens. An example of the test pane is shown in the following image.



6. To invoke the service, enter a sample XML document in the input xml field.

For sample input XML, see *Testing a Business Service on page A-15*.

7. Click *Invoke*.

The result appears in the right pane.

Example: Testing a Business Service

The following is a sample XML document that you can use to test the business service:

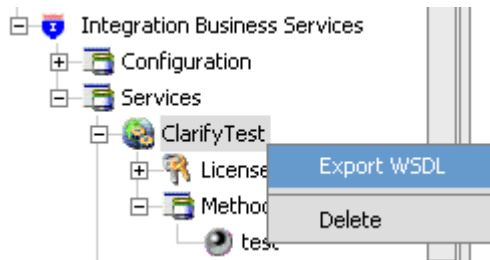
```
<?xml version="1.0" encoding="UTF-8"
?><FIXMLMessage><Header><Sender><CompID>
.....56
.....</CompID>
      </Sender><Target><CompID>
.....B2B
.....</CompID>
      </Target><SendingTime>
.....20031119-22:32:58
.....</SendingTime>
      </Header><ApplicationMessage><Order><ClOrdID>
.....BUY1069281178615
.....</ClOrdID><HandInst Value="1"/><Instrument><Symbol>
.....C
.....</Symbol>
      </Instrument><Side Value="1"/><OrderQuantity><OrderQty>
.....100
.....</OrderQty>
      </OrderQuantity><OrderType><MarketOrder Value="1"/>
      </OrderType><OrderDuration><TimeInForce Value="0"/>
      </OrderDuration><Currency
Value="USD"/></Order></ApplicationMessage></FIXMLMessage>
```

Exporting iWay WSDL for Use in BEA WebLogic Workshop Workflows

Because Application Explorer runs within BEA WebLogic Workshop, you can easily incorporate Web services into BEA WebLogic Workflows. To enable BEA WebLogic Workshop to use Web services, you simply export the WSDL to a directory accessible to BEA WebLogic Workshop.

Procedure: How to Export iWay WSDL for Use in BEA WebLogic Workshop Workflows

To export WSDL to a directory accessible to BEA WebLogic Workshop:

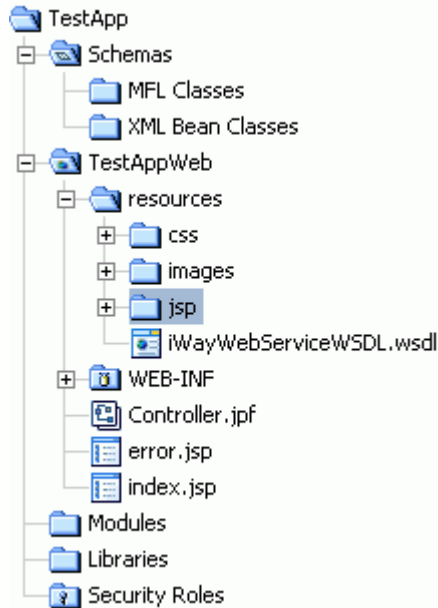


1. After you create a Web service, right-click the Web service name and select *Export WSDL*.

The Save dialog box appears.

2. Save the WSDL to a directory accessible to BEA WebLogic Workshop, for example, the \resources directory in your BEA WebLogic Workshop Web application directory structure.

The WSDL file appears under the resources folder of your Web application:



Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to ClarifyCRM Business Objects. The user name and password values that you provided for ClarifyCRM Business Objects during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service.

```
<SOAP-ENV:Header>
  <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m>Password>String</m>Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>
```

Note: You can remove the following tags from the SOAP header, since they are not required:

```
<m:disposition>String</m:disposition>  
<m:language>String</m:language>
```

Adding a Control for an iWay Resource in BEA WebLogic Workshop

Java controls provide a convenient way to incorporate access to iWay resources. You can add controls in BEA WebLogic Workshop to use Web services created by Application Explorer, or you can add controls that enable you to take advantage of the JCA resources of Application Explorer.

Adding a Web Service Control to a BEA WebLogic Workshop Application

After you create an iWay Web service using Application Explorer and export the WSDL file, you can create a control for the Web service.

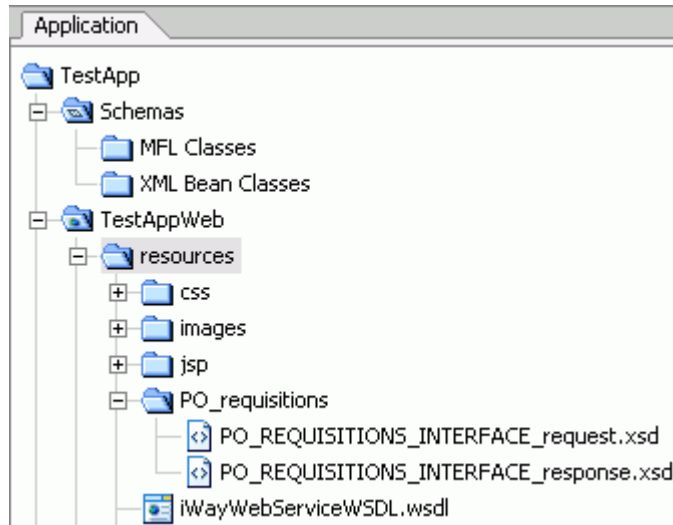
For more information on exporting a WSDL file, see *Export iWay WSDL for Use in BEA WebLogic Workshop Workflows* on page A-16.

Procedure: How to Add a Web Service Control

To add a Web service control:

1. After exporting the WSDL file from Application Explorer, locate the file in the Application tab of your BEA WebLogic Workshop application.

For example, a WSDL file saved to the \resources directory in your BEA WebLogic Workshop Web application directory structure appears as follows:



2. Right-click the *WSDL* file and select *Generate Service Control*.

The control for the WSDL appears below the WSDL file in the resources tree.



Adding an Extensible CCI Control

An iWay control enables access to resources provided by Application Explorer when it is used in conjunction with a JCA deployment. You must add an iWay control before using it in a BEA WebLogic Workshop application workflow.

The following topic describes the enhanced CCI control, which is extensible and provides JCX with typed inputs and outputs for JCA in BEA WebLogic Workshop.

Overview

The extensible iWay CCI control provides:

- **Method and tag validation.** BEA WebLogic Workshop provides warnings regarding invalid methods and tags.
- **Improved error handling.**

You can define new methods that rely on the generic *service* and *authService* methods. For example, you can define a JCX with a new method without writing casting code or explicit transformations such as the following:

```
public ResponseDataType MethodName(RequestDataType VariableName) throws  
Exception;
```

where:

ResponseDataType

Is the XML Bean Class value that is generated from the response schema.

MethodName

Is the method name used by the extensible CCI control.

RequestDataType

Is the XML Bean Class value that is generated from the request schema.

VariableName

Is the request variable that stores the request document, which is used as input by the extensible CCI control.

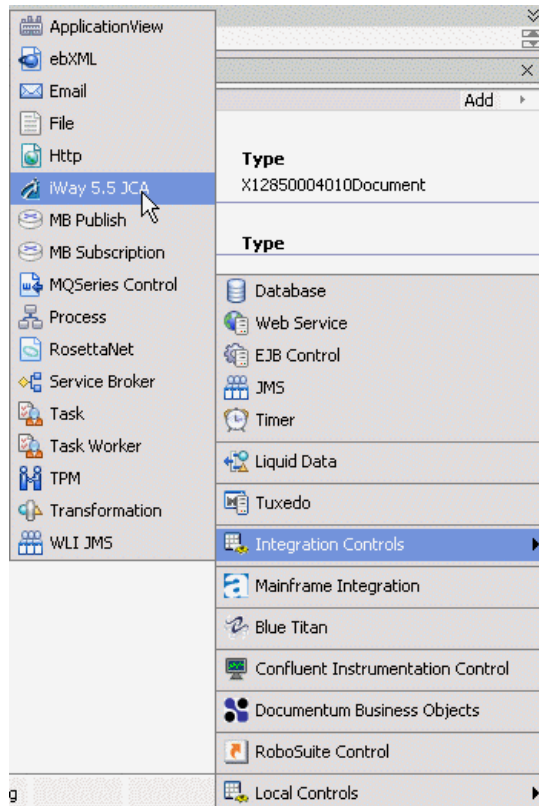
In addition, the extensible CCI control now generates a JCX file to which you can add your own methods. For more information, see *Defining a Control Using the Extensible CCI Control* on page A-20.

You can also use dynamic class casting to specify schema-based input or output XmlObjects to be casted into a pure XmlObject as a service method, which is expected by the CCI control. For more information, see *Using Dynamic Class Casting* on page A-28.

Example: Defining a Control Using the Extensible CCI Control

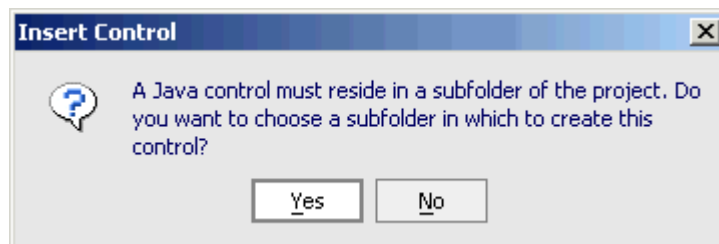
The following sample JCX demonstrates how to define a control for Clarify using the extensible CCI control in BEA WebLogic Workshop.

1. Start BEA WebLogic Workshop and create a new project.



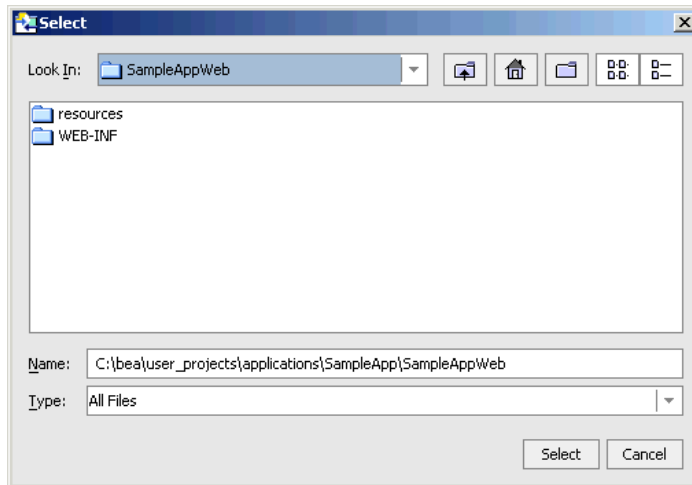
2. Click **Add** from the Controls section in the Data Palette tab, select *Integration Controls*, and click *iWay 5.5 JCA*.

The Insert Control message box opens.



3. Click **Yes**.

The Select dialog box opens.



4. Choose a subfolder for the CCI control and click *Select*.

The Insert Control - iWay 5.5 JCA dialog box opens.

Insert Control - iWay 5.5 JCA

STEP 1 Variable name for this control:

STEP 2 I would like to :

☐ Use an iWay 5.5 JCA control already defined by a JCX file

JCX file:

☒ Create a new iWay 5.5 JCA control to use.

New JCX name:

☐ Make this a control factory that can create multiple instances at runtime

STEP 3

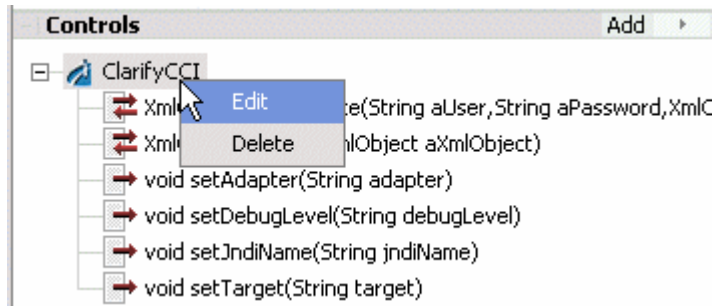
Adapter Name:

Target Name:

Debug Level: ▼

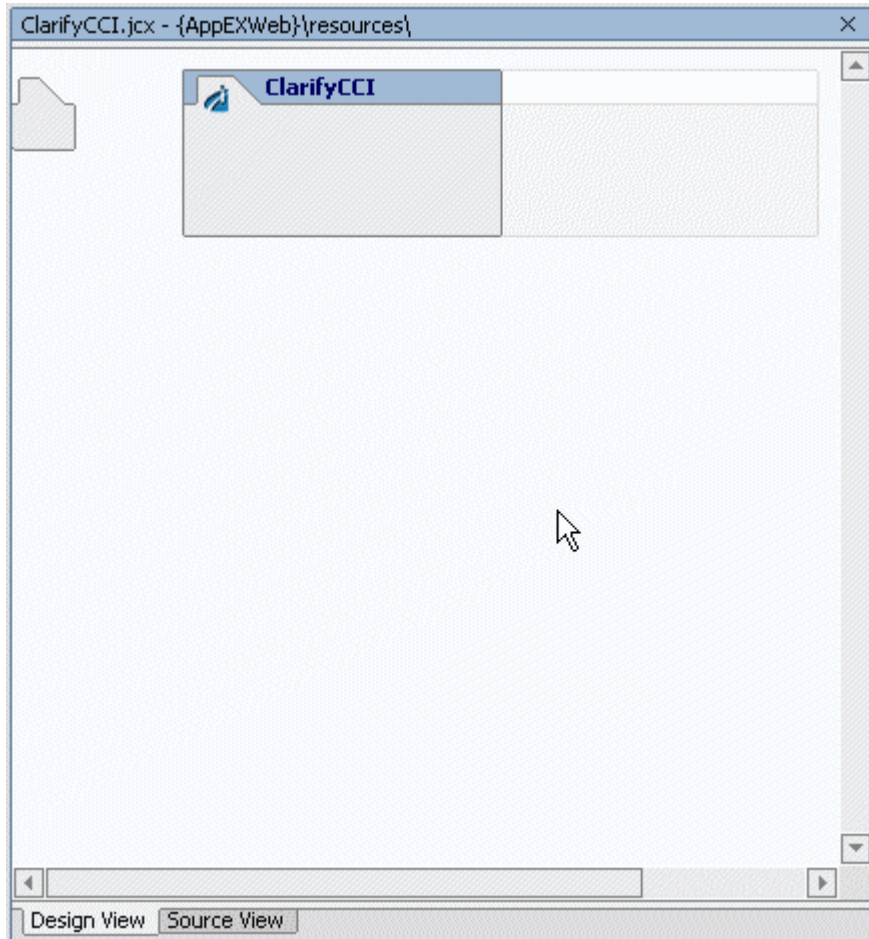
- a. Provide a variable name for the control.
 - b. Click *Create a new iWay 5.5 JCA control to use* and provide a new JCX name.
 - c. Enter the adapter name, target name, and select a debug level from the drop-down list.
5. Click *Create*.

A new JCX file is created.



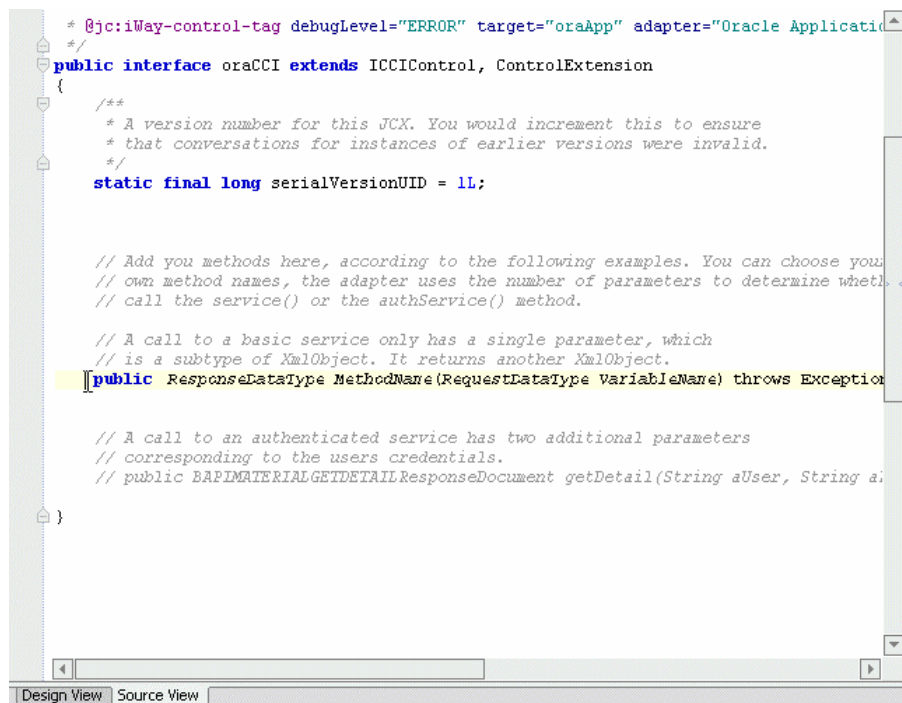
6. Right-click the control, for example, ClarifyCCI, and select *Edit*.

The Design View for the control opens.



7. Click the *Source View* tab.

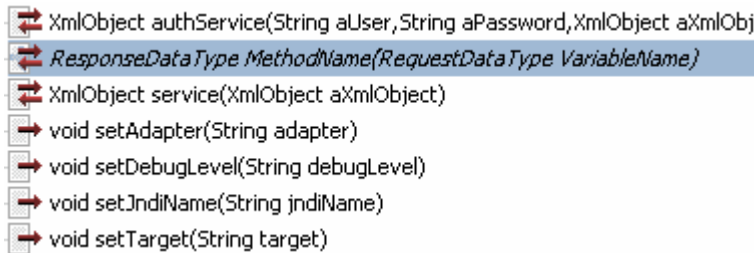
The Source View for the control opens.



Perform the following steps:

- Uncomment the public class definition.
- Change the existing response data type to match your response data type that is generated from your Clarify response schema.
- Change the existing method name to match your method.
- Change the existing request data type to match your request data type that is generated from your Clarify request schema.

The following control is now available in BEA WebLogic Workshop and can be added to a workflow:



Note: You can view available data types under the *XML Bean Classes* folder in the *Application* tab, which are added once you import your XML request or response schemas from Application Explorer.

These data types are case sensitive and must be entered exactly as shown.

Using the Extensible CCI Control

The extensible CCI control functions much like a database control since it generates JCX files to which you can add your own methods.

Your own methods can use the correct input and output types rather than the generic `XmlObject` types that the JCA control uses. Since the control is just a proxy that uses a reflection to call the relevant method, it handles the casting for you. You are no longer required to write custom code that does the cast or transformations that are cast between an `XmlObject`.

For example, instead of the generic `XmlObject`:

```
XmlObject service(XmlObject input) throws java.lang.Exception;
```

you call:

```
public ResponseDataType MethodName(RequestDataType VariableName) throws
Exception;
```

where:

ResponseDataType

Is the XML Bean Class value that is generated from the response schema.

MethodName

Is the method name used by the extensible CCI control.

RequestDataType

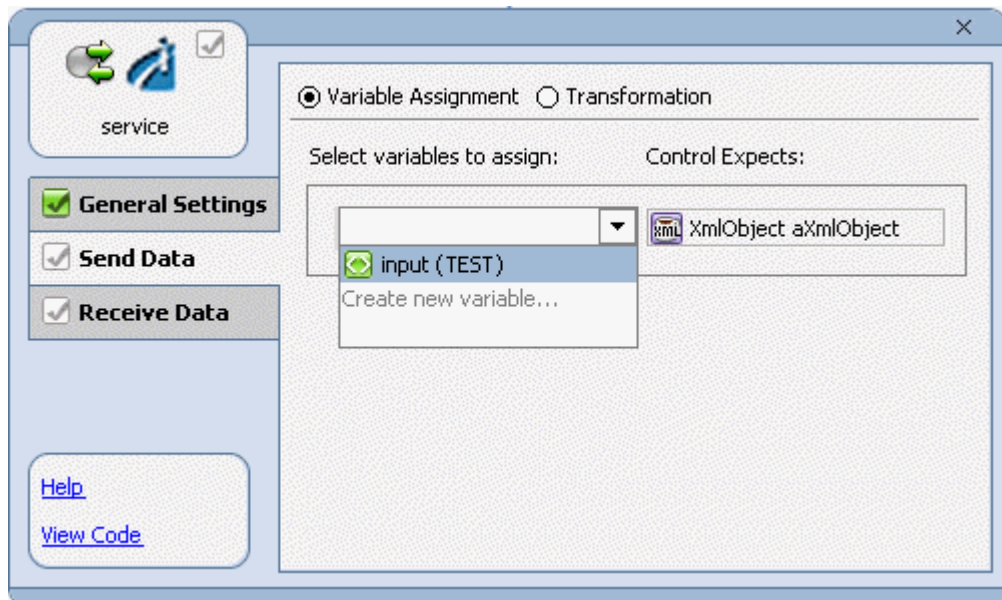
Is the XML Bean Class value that is generated from the request schema.

VariableName

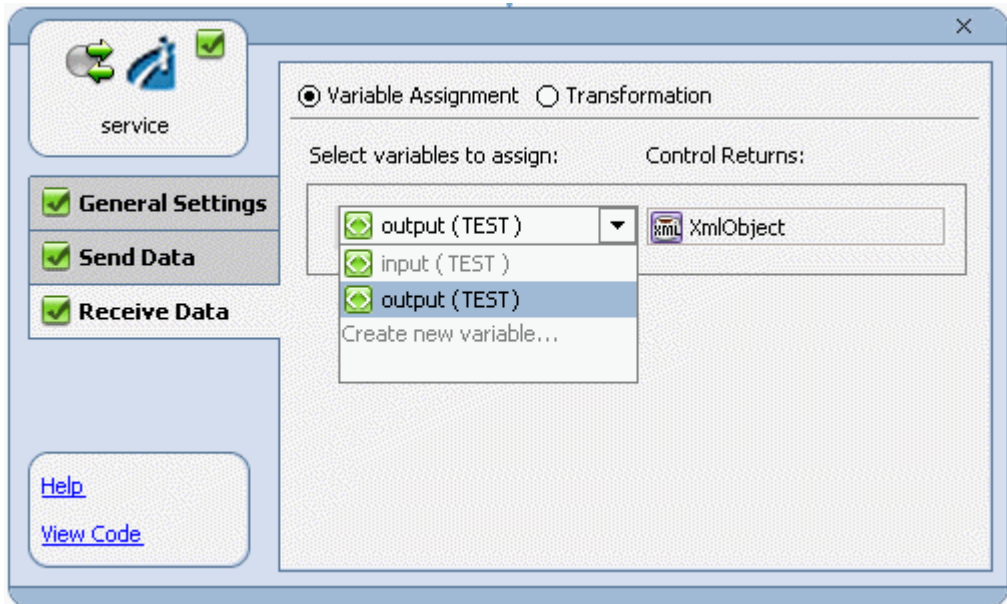
Is the request variable that stores the request document, which is used as input by the extensible CCI control.

Example: Using Dynamic Class Casting

The following example uses dynamic class casting to specify a schema-based input XmlObject to be casted into a pure XmlObject as a service method, which is expected by the CCI control.



The following example uses dynamic class casting where the CCI control returns a pure XmlObject, which is casted dynamically into a schema-based output XmlObject.



APPENDIX B

Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services with ClarifyCRM ClearBasic

Topics:

- Starting Application Explorer in BEA WebLogic Workshop
- Creating a New Configuration
- Connecting to and Managing a Target
- Creating an XML Schema
- Creating an Integration Business Service
- Adding a Control for an iWay Resource in BEA WebLogic Workshop
- Adding an Extensible CCI Control

This section describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop to create XML schemas and Web services with ClarifyCRM ClearBasic (CB).

In addition, this section explains how to add a control to a Web service or BEA WebLogic Workshop application.

Starting Application Explorer in BEA WebLogic Workshop

The server must be started where iWay Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic Server.

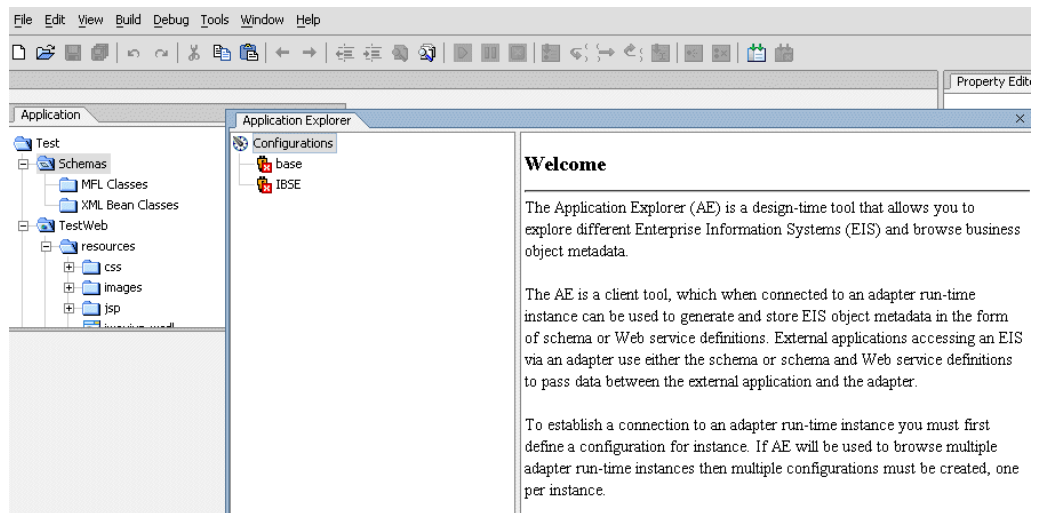
You can run Application Explorer in BEA WebLogic Workshop using an Integration Business Services Engine (IBSE) configuration or J2EE Connector Architecture (JCA) configuration.

Procedure: How to Start Application Explorer in BEA WebLogic Workshop

To start Application Explorer in BEA WebLogic Workshop:

1. Ensure that BEA WebLogic Server is running.
2. Start BEA WebLogic Workshop.
3. From the BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer*.

Application Explorer opens in BEA WebLogic Workshop. This image also shows the Application Explorer welcome statement that appears on the right when you select the Configurations node on the left.



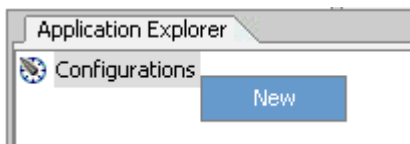
You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

Creating a New Configuration

Before you can start using Application Explorer, you must define a new configuration for iBSE or JCA.

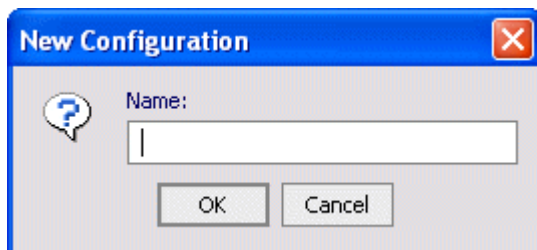
Procedure: How to Create a New Configuration for iBSE or JCA

To create a new configuration:



1. Right-click *Configurations* and select *New*.

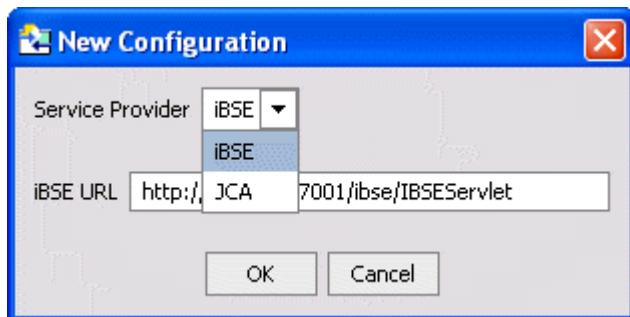
The New Configuration dialog box opens, as shown in the following image.



2. Type the name of the new configuration and click *OK*.

Note: If you are creating a new JCA configuration, type *base* in the name field. You must use this value if you are pointing to the default iWay configuration.

The following dialog box opens, as shown in the following image.



3. From the Service Provider drop-down list, select *iBSE* or *JCA*.

If you select iBSE, type the URL for iBSE, for example,

<http://localhost:7001/ibse/IBSEServlet>

where:

[localhost](#)

Is where your application server is running.

If you select JCA, enter the full path to the directory where iWay 5.5 is installed, for example,

[C:\Program Files\iWay55](#)

where:

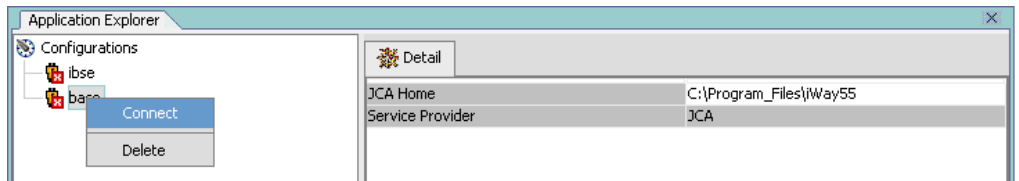
[iWay55](#)

Is the full path to your iWay installation.

4. Click OK.

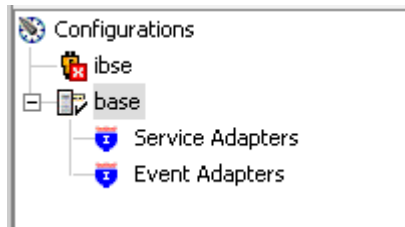
A node representing the new configuration appears under the Configurations node. The right pane displays details of a selected configuration.

After you add your configuration, you must connect to it.

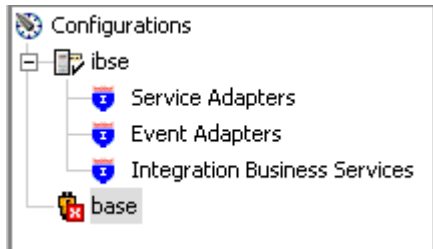


5. Right-click the configuration to which you want to connect and select *Connect*.

When you connect to base, the Service Adapters and Event Adapters nodes appear.



When you connect to iBSE, the Service Adapters, Event Adapters, and Integration Business Services nodes appear.



6. To display the service and event adapters that are installed, expand each node.

The Service Adapters list includes a ClarifyCRM node that enables you to connect to ClarifyCRM metadata and create XML request and response schemas to use to listen for events or create Web services. For more information, see *Creating an Integration Business Service* on page B-12.

The Event Adapters list includes a ClarifyCRM node that enables you to create ports and channels for ClarifyCRM event handling. For more information, see Appendix D, *Using Application Explorer in BEA WebLogic Workshop for Event Handling with ClarifyCRM ClearBasic*.

Connecting to and Managing a Target

To browse ClarifyCRM, you must create a ClarifyCRM target and connect to it. The target serves as your connection point. You must establish a connection to Clarify every time you start iWay Application Explorer or after you disconnect from ClarifyCRM.

The left pane displays the application systems supported by Application Explorer. These are based on the iWay adapters you installed and are licensed to use.

Creating and Connecting to a Target

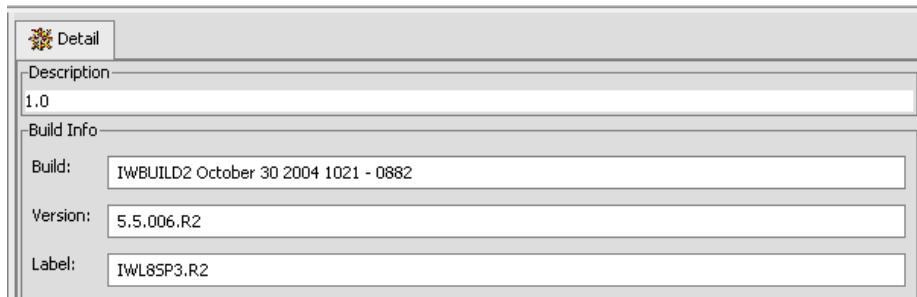
To connect to ClarifyCRM for the first time, you must create a new target. The target is automatically saved after it is created.

Procedure: How to Create a New Target

To create a new target:

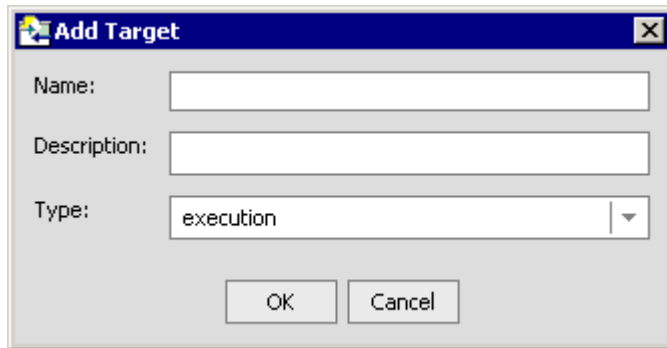
1. In the left pane, expand *Service Adapters* and click the *ClarifyCB* node.

Descriptive information (for example, build and version) for the adapter appears in the right pane.



2. Right-click the *ClarifyCB* node and select *Add Target*.

The Add target dialog box opens.



- a. In the Name field, type a descriptive name for the target, for example, *ClarifyTarget*.
 - b. In the Description field, type a brief description of the target.
 - c. From the Target Type drop-down menu, select *execution*.
3. Click *OK*.

Target connection information opens on the right displaying six fields with connection parameter values and two action buttons.

The screenshot shows a dialog box titled "execution" with a close button (X) in the top right corner. Inside the dialog, there are six fields and two buttons. The first two fields are checkboxes, both of which are checked. The next four fields are text boxes or a dropdown menu. The last two buttons are "OK" and "Cancel". At the bottom of the dialog, there is a red text label: "Fields marked with * are required."

<input checked="" type="checkbox"/> Flag to indicate whether to use the TPNOTRAN flag	
<input checked="" type="checkbox"/> Flag to indicate whether to use the Occurrence attribute	
XML element for processing Tuxedo service requests*	IBI_TUXEDO_SERVICE
JNDI Name for Tuxedo Adapter EJB*	ibi.ejb.tuxconnector
Access mode	LOCAL
URL to WebLogic server running Tuxedo Connector*	t3://localhost:7001

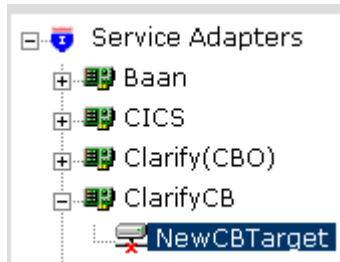
OK Cancel

Fields marked with * are required.

Note: The ClarifyCRM connection parameters are consistent with those found in your ClarifyCRM system. For more information on parameter values that are specific to your ClarifyCRM configuration, consult your ClarifyCRM system administrator.

- a. Select the following check boxes, as required:
 - Flag to indicate whether to use the TPNOTRAN flag
 - Flag to indicate whether to use the Occurrence attribute
 - b. Type an XML element for processing Tuxedo service requests.
 - c. Type a JNDI name for the Tuxedo Adapter EJB.
 - d. Select LOCAL, REMOTE_HTTP, or REMOTE_RMI from the Access mode drop-down list.
 - e. Type a URL in the URL to WebLogic server running Tuxedo Connector field.
4. Click OK.

In the left pane, the new target appears beneath the ClarifyCB node. The following image shows a the new target, NewCBTarget under the ClarifyCB node. This target node displays a red X to indicate it is disconnected.



You are ready to connect to the application target you created.

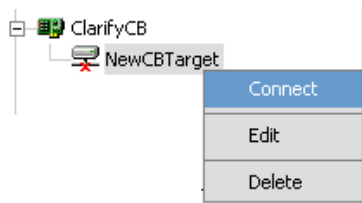
Connecting to an Existing Target

You can use existing targets to connect to instances of ClarifyCRM.

Procedure: How to Connect to an Existing ClarifyCB Target

To connect to an existing ClarifyCB target:

1. In the left pane, expand the *ClarifyCB* node and select the target to which you want to connect.



2. Right-click the target and select *Connect*.

The ClarifyTarget node in the left pane changes to reflect that a connection was made.

3. Expand the target node to reveal the transactions node, as shown in the following image.



Managing a Target

Although you can maintain multiple open connections to different application systems, we recommend that you close connections when they are not in use. After you disconnect, you can modify an existing target.

You can modify the connection parameters when your system properties change. You also can delete a target. The following procedures describe how to disconnect from a target, edit a target, and delete a target.

Procedure: How to Disconnect From a Target

To disconnect from a target:



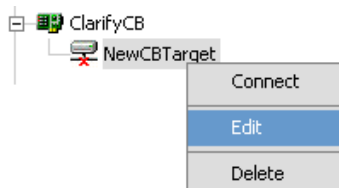
1. Right-click the target from which you want to disconnect.
2. Select *Disconnect*.

Disconnecting from the application system drops the connection, but the node remains. The target node appearance in the left pane changes to reflect that you disconnected from the target.

Procedure: How to Edit a Target

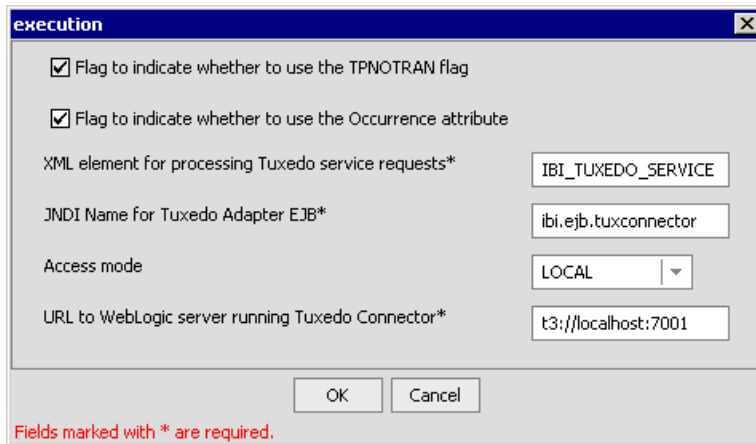
To edit a target:

1. Ensure that the target you want to edit is disconnected.



2. In the left pane, right-click the target and select *Edit*.

The dialog box with target parameters opens, as shown in the following image. This is the same dialog box that appears when you create a target.



The screenshot shows a dialog box titled "execution" with a close button (X) in the top right corner. It contains several configuration options:

- ☒ Flag to indicate whether to use the TPNOTRAN flag
- ☒ Flag to indicate whether to use the Occurrence attribute
- XML element for processing Tuxedo service requests*: IBI_TUXEDO_SERVICE
- JNDI Name for Tuxedo Adapter EJB*: ibi.ejb.tuxconnector
- Access mode: LOCAL (dropdown menu)
- URL to WebLogic server running Tuxedo Connector*: t3://localhost:7001

At the bottom, there are "OK" and "Cancel" buttons. Below the buttons, a red text label reads: "Fields marked with * are required."

3. Change the properties in the dialog box as required and click OK.

Procedure: How to Delete a Target

To delete a target:

1. In the left pane, right-click the target you want to delete.
2. Select *Delete*.

The target node disappears from the left pane.

Creating an XML Schema

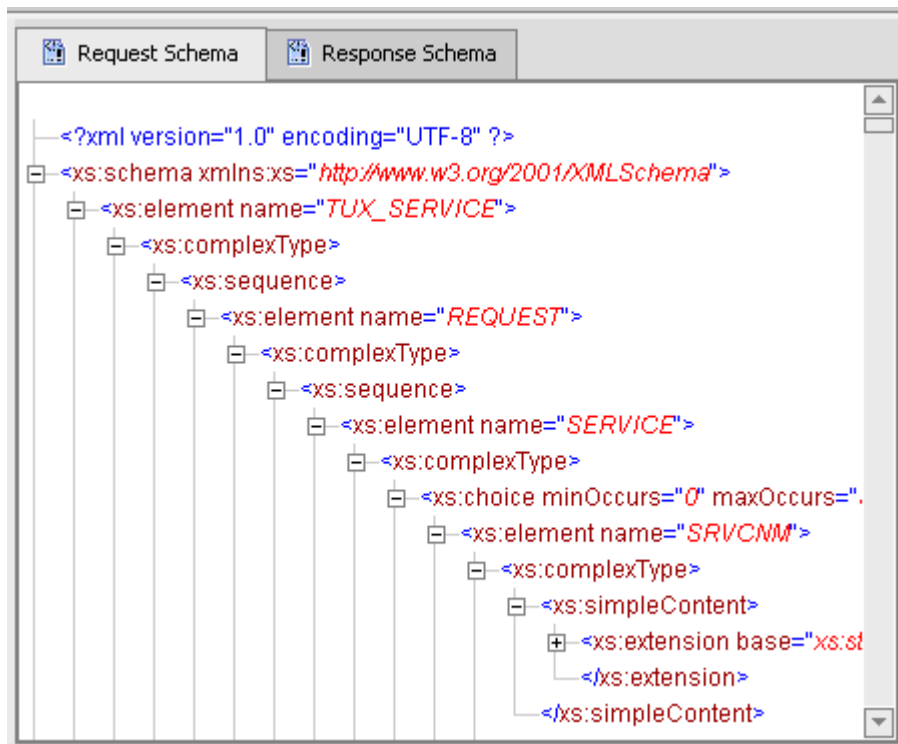
After you create a new configuration and connect to the target, Application Explorer enables you to create a request or response schema.

Procedure: How to Create XML Schema

To create XML schema:

1. Connect to the ClarifyCB target.
2. In the left pane, expand the target node and the *transactions* node.

3. Under the transactions node, select either the *std_service* node or the *std_event* node.

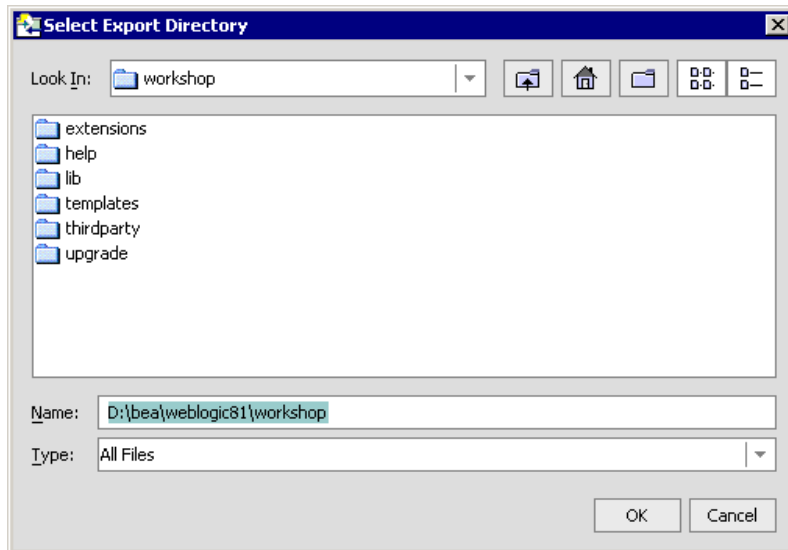


4. Click the appropriate schema tab to view a schema.

The schema you select appears on the right. The previous image shows an example of a *std_service* node schema with a Request Schema tab and a Response Schema tab.

5. To save the schema:
 - a. Right-click the *std_service* or *std_event* node and select *Export Schema(s)* from the drop-down list.

The Select Export Directory window opens, as shown in the following image.



- b. Browse to the directory where you want the schema to reside and type a name for the schema file.
- c. Click *OK*.

Creating an Integration Business Service

You can create an integration business service (also known as a Web service) for objects you want to use with your adapter. To generate a business service, you must deploy the adapter using the Integration Business Services Engine (iBSE). iBSE exposes functionality as Web services and serves as a gateway to heterogeneous back-end applications and databases.

A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity. For the caller or sender, a Web service can be considered as a “black box” that may require input and delivers a result. Web services integrate within an enterprise as well as across enterprises on any communication technology stack, whether asynchronous or synchronous, in any format.

You can make a Web service available to other services within a host server by generating WSDL (Web Services Description Language) from the Web service.

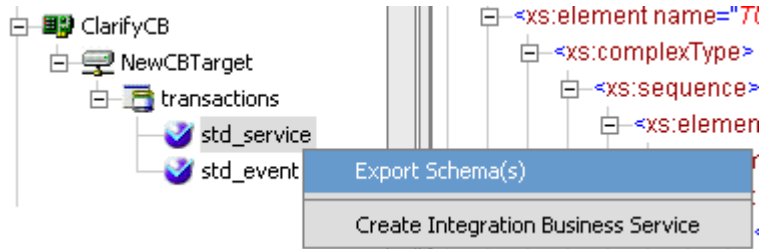
Because Application Explorer runs within BEA WebLogic Workshop, you can easily incorporate Web services into BEA WebLogic Workflows. To enable BEA WebLogic Workshop to use Web services, export the WSDL to a directory accessible to BEA WebLogic Workshop.

Note: In a J2EE Connector Architecture (JCA) implementation of iWay adapters, Web services are not available. When the adapters are deployed to use the iWay Connector for JCA, the Common Client Interface provides integration services using the iWay adapters. For more information, see the *iWay Installation and Configuration for BEA WebLogic* manual and the *iWay Connector for JCA for BEA WebLogic Server User's Guide*.

Procedure: How to Create an Integration Business Service

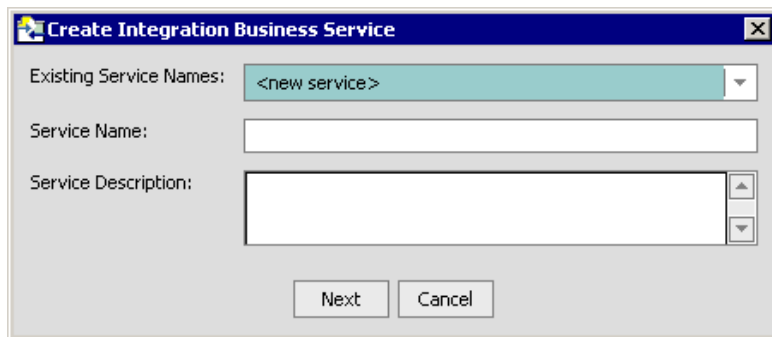
To create an Integration Business Service:

1. Expand the target node and the *transactions* node, and select the *std_service* node.



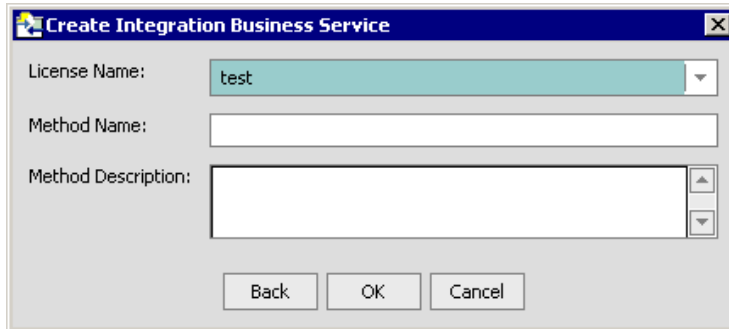
2. Right-click the *std_service* node and select *Create Integration Business Service*.

The Create Integration Business Service dialog box opens, as shown in the following image.



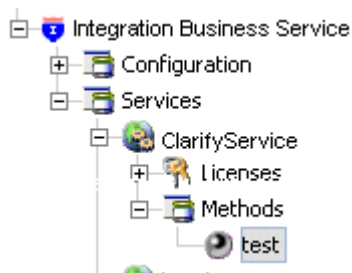
- a. From the Existing Service Names drop-down list, select whether you want to create a new service name or use an existing service name.
 - b. In the Service Name field, type a name if it is a new business service, for example, ClarifyService.
 - c. In the Service Description field, type a brief description if it is a new business service.
3. Click *Next*.

The Create Integration Business Service dialog box displays additional fields.

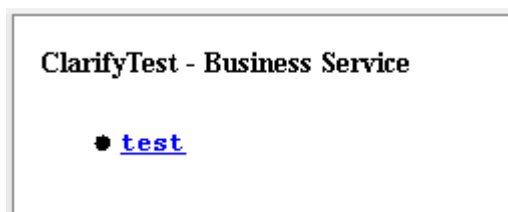
A dialog box titled "Create Integration Business Service" with a close button (X) in the top right corner. It contains three input fields: "License Name:" with a dropdown menu showing "test", "Method Name:" with an empty text box, and "Method Description:" with a larger empty text box and vertical scrollbars. At the bottom are three buttons: "Back", "OK", and "Cancel".

- a. From the License Name drop-down list, select a license.
 - b. In the Method Name field, type a name for the method.
 - c. In the Method Description field, type a brief description for the method.
4. Click OK.

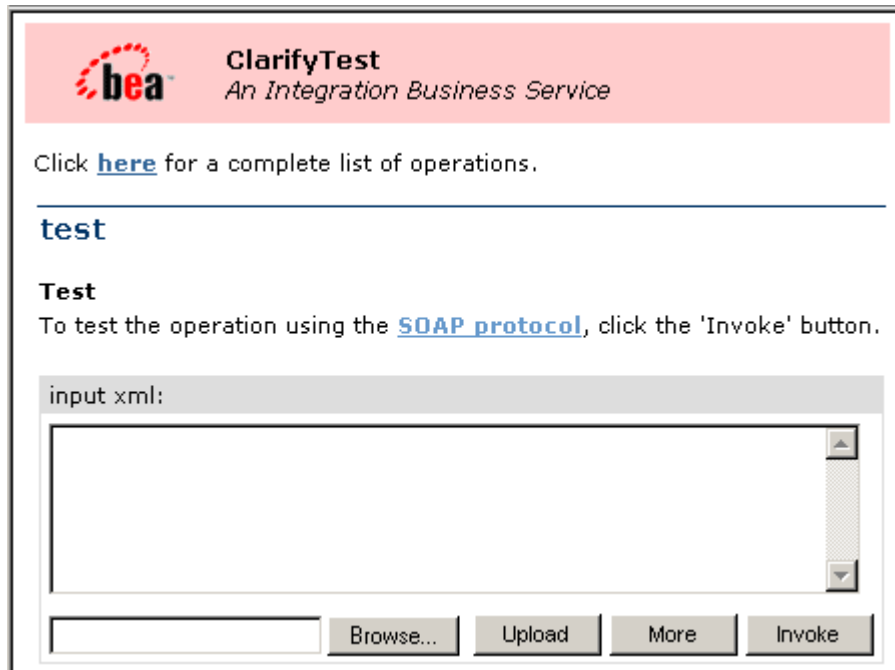
The business service and method appear below the Integration Business Service, Services node.



5. In the right pane, click the name of the method, for example, test, as shown in the following image.



The test pane opens on the right. An example of a test pane is shown in the following image.



The screenshot shows a web interface for 'ClarifyTest', described as 'An Integration Business Service'. The interface has a pink header with the BEA logo. Below the header, there is a link 'here' for a complete list of operations. The main section is titled 'test' and contains a 'Test' section. This section instructs the user to click the 'Invoke' button to test the operation using the 'SOAP protocol'. Below the instruction is a large text area labeled 'input xml:'. At the bottom of the interface, there are four buttons: 'Browse...', 'Upload', 'More', and 'Invoke'.

6. To invoke the service, enter a sample XML document in the input xml field.
7. Click *Invoke*.

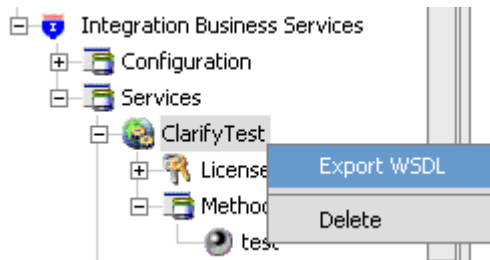
The result appears in the right pane.

Exporting iWay WSDL for Use in BEA WebLogic Workshop Workflows

Because Application Explorer runs within BEA WebLogic Workshop, you can easily incorporate Web services into BEA WebLogic Workflows. To enable BEA WebLogic Workshop to use Web services, you simply export the WSDL to a directory accessible to BEA WebLogic Workshop.

Procedure: How to Export iWay WSDL for Use in BEA WebLogic Workshop Workflows

To export WSDL to a directory accessible to BEA WebLogic Workshop:

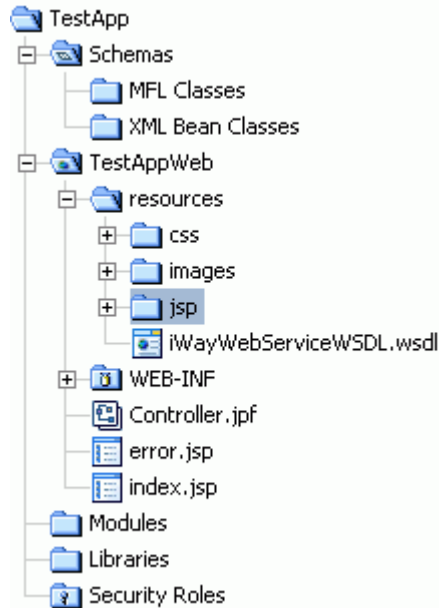


1. After you create a Web service, right-click the Web service name and select *Export WSDL*.

The Save dialog box appears.

2. Save the WSDL to a directory accessible to BEA WebLogic Workshop, for example, the \resources directory in your BEA WebLogic Workshop Web application directory structure.

The WSDL file appears under the resources folder of your Web application:



Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to ClarifyCRM ClearBasic. The user name and password values that you provided for ClarifyCRM ClearBasic during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service:

```
<SOAP-ENV:Header>
  <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m>Password>String</m>Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>
```

Note: You can remove the following tags from the SOAP header, since they are not required:

```
<m:disposition>String</m:disposition>  
<m:language>String</m:language>
```

Adding a Control for an iWay Resource in BEA WebLogic Workshop

Java controls provide a convenient way to incorporate access to iWay resources. You can add controls in BEA WebLogic Workshop to use Web services created by Application Explorer, or you can add controls that enable you to take advantage of the JCA resources of Application Explorer.

Adding a Web Service Control to a BEA WebLogic Workshop Application

After you create an iWay Web service using Application Explorer and export the WSDL file, you can create a control for the Web service.

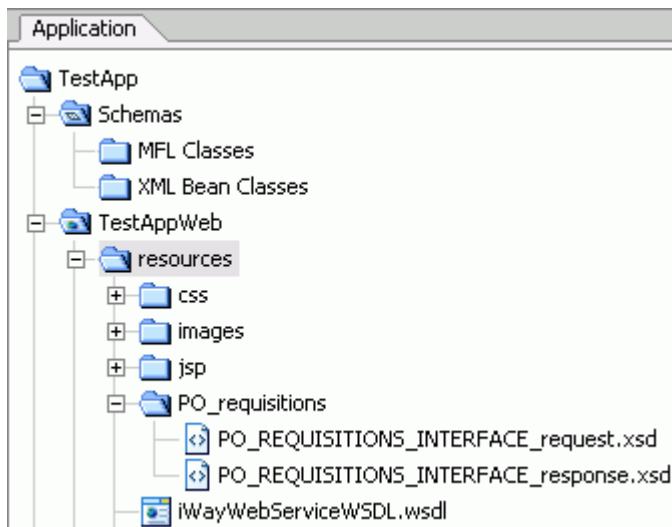
For more information on exporting a WSDL file, see *Export iWay WSDL for Use in BEA WebLogic Workshop Workflows* on page B-16.

Procedure: How to Add a Web Service Control

To add a Web service control:

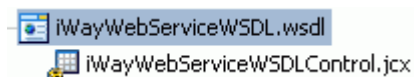
1. After exporting the WSDL file from Application Explorer, locate the file in the Application tab of your BEA WebLogic Workshop application.

For example, a WSDL file saved to the \resources directory in your BEA WebLogic Workshop Web application directory structure appears as shown in the following image.



2. Right-click the *WSDL* file and select *Generate Service Control*.

The control for the WSDL appears below the WSDL file in the resources tree.



Adding an Extensible CCI Control

An iWay control enables access to resources provided by Application Explorer when it is used in conjunction with a JCA deployment. You must add an iWay control before using it in a BEA WebLogic Workshop application workflow.

The following topic describes the enhanced CCI control, which is extensible and provides JCX with typed inputs and outputs for JCA in BEA WebLogic Workshop.

Overview

The extensible iWay CCI control provides:

- **Method and tag validation.** BEA WebLogic Workshop provides warnings regarding invalid methods and tags.
- **Improved error handling.**

You can define new methods that rely on the generic *service* and *authService* methods. For example, you can define a JCX with a new method without writing casting code or explicit transformations such as the following:

```
public ResponseDataType MethodName(RequestDataType VariableName) throws  
Exception;
```

where:

ResponseDataType

Is the XML Bean Class value that is generated from the response schema.

MethodName

Is the method name used by the extensible CCI control.

RequestDataType

Is the XML Bean Class value that is generated from the request schema.

VariableName

Is the request variable that stores the request document, which is used as input by the extensible CCI control.

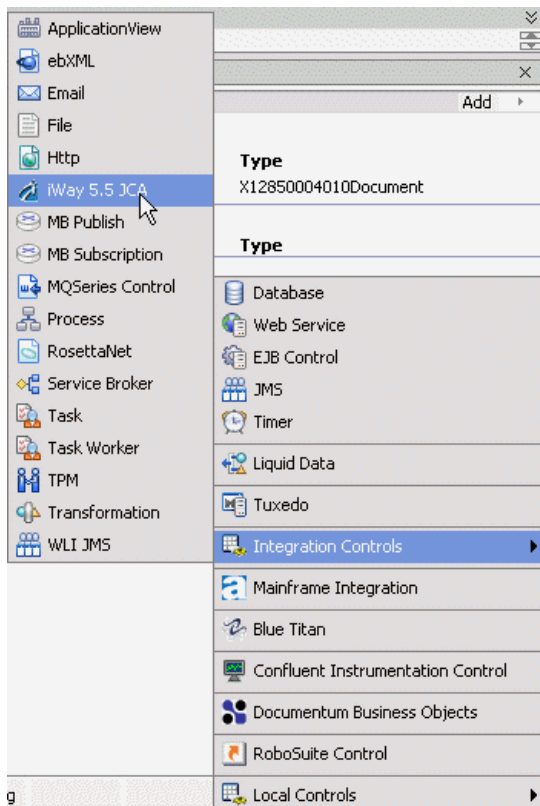
In addition, the extensible CCI control now generates a JCX file to which you can add your own methods. For more information, see *Defining a Control Using the Extensible CCI Control* on page B-20.

You can also use dynamic class casting to specify schema-based input or output XmlObjects to be casted into a pure XmlObject as a service method, which is expected by the CCI control. For more information, see *Using Dynamic Class Casting* on page B-28.

Example: Defining a Control Using the Extensible CCI Control

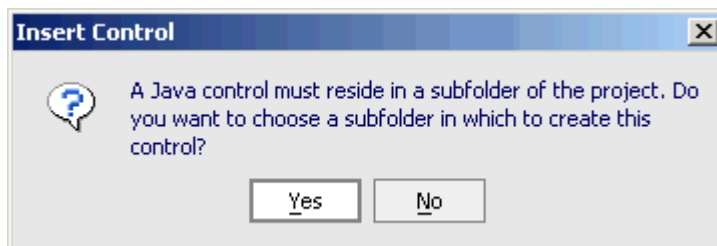
The following sample JCX demonstrates how to define a control for Clarify using the extensible CCI control in BEA WebLogic Workshop.

1. Start BEA WebLogic Workshop and create a new project.



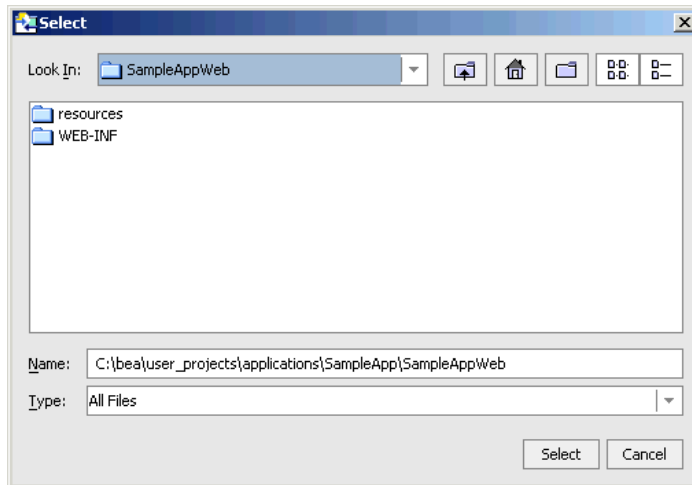
2. Click **Add** from the Controls section in the Data Palette tab, select *Integration Controls*, and click *iWay 5.5 JCA*.

The Insert Control message box opens.



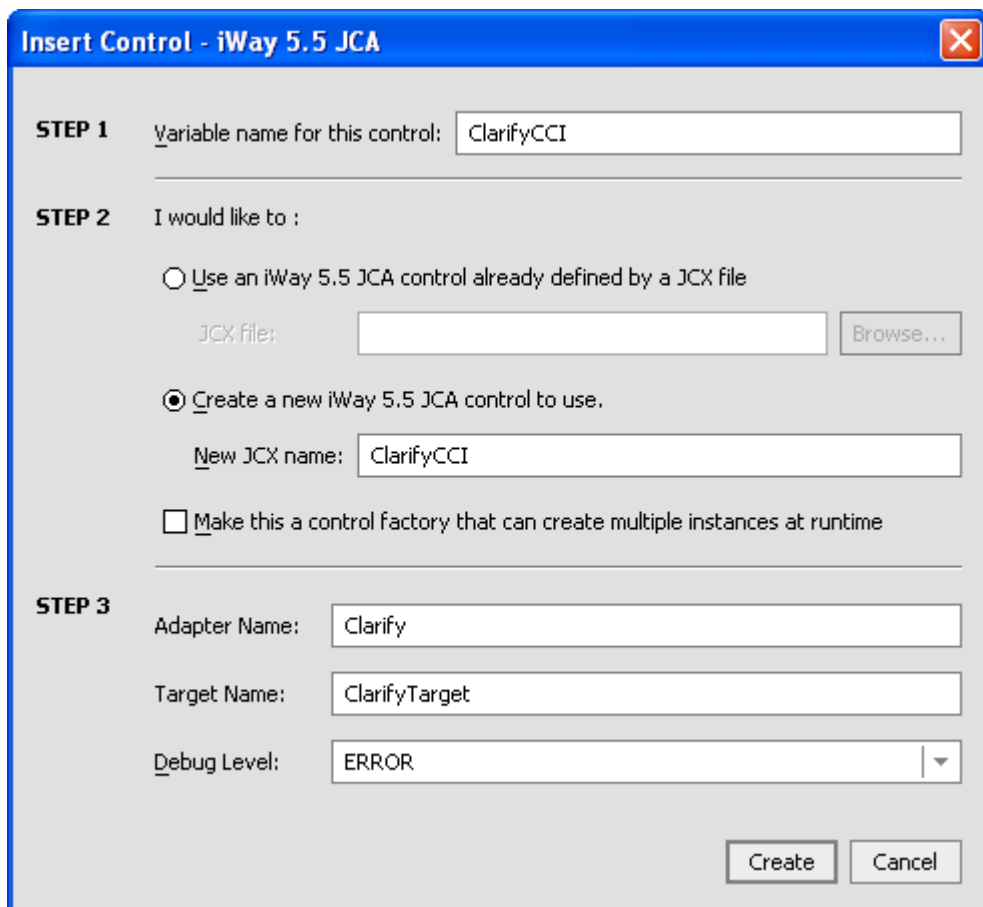
3. Click **Yes**.

The Select dialog box opens.



4. Choose a subfolder for the CCI control and click *Select*.

The Insert Control - iWay 5.5 JCA dialog box opens.



The dialog box is titled "Insert Control - iWay 5.5 JCA" and contains three steps for configuring a JCA control.

STEP 1 Variable name for this control:

STEP 2 I would like to :

☐ Use an iWay 5.5 JCA control already defined by a JCX file

JCX file:

☒ Create a new iWay 5.5 JCA control to use.

New JCX name:

☐ Make this a control factory that can create multiple instances at runtime

STEP 3

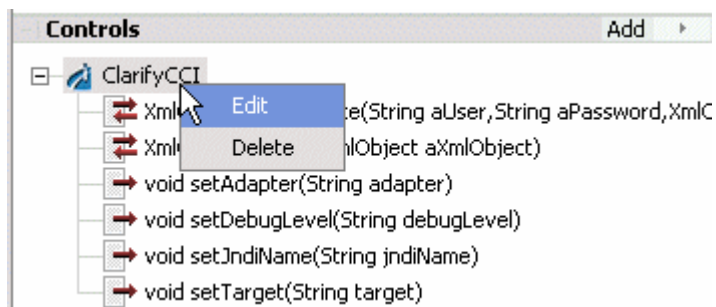
Adapter Name:

Target Name:

Debug Level:

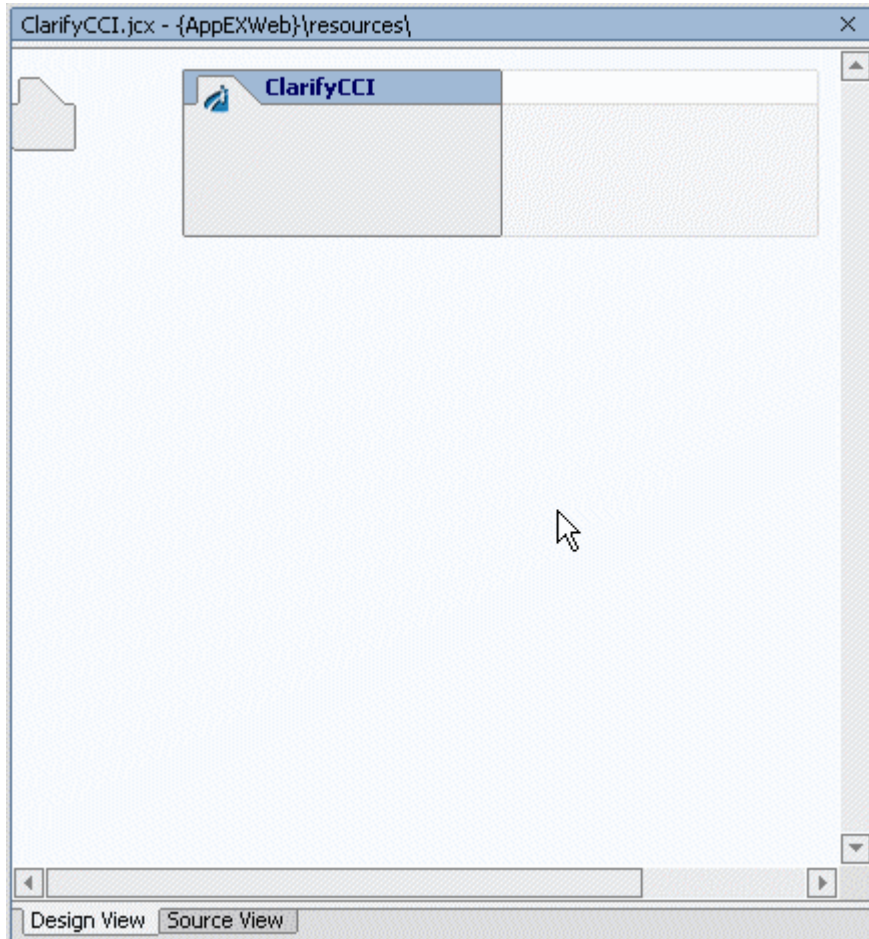
- a. Provide a variable name for the control.
 - b. Click *Create a new iWay 5.5 JCA control to use* and provide a new JCX name.
 - c. Enter the adapter name, target name, and select a debug level from the drop-down list.
5. Click *Create*.

A new JCX file is created.



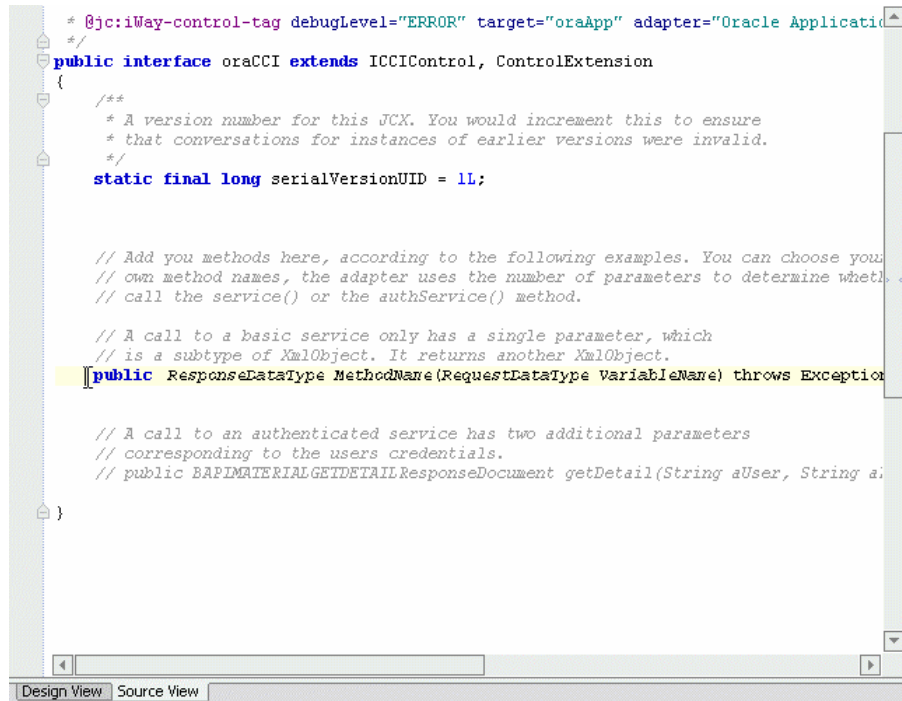
6. Right-click the control, for example, ClarifyCCI, and select *Edit*.

The Design View for the control opens.



7. Click the *Source View* tab.

The Source View for the control opens.



```

* @jcc: iWay-control-tag debugLevel="ERROR" target="oraApp" adapter="Oracle Applicatio
*/
public interface oraCCI extends ICCIControl, ControlExtension
{
    /**
     * A version number for this JCC. You would increment this to ensure
     * that conversations for instances of earlier versions were invalid.
     */
    static final long serialVersionUID = 1L;

    // Add your methods here, according to the following examples. You can choose your
    // own method names, the adapter uses the number of parameters to determine wheth
    // call the service() or the authService() method.

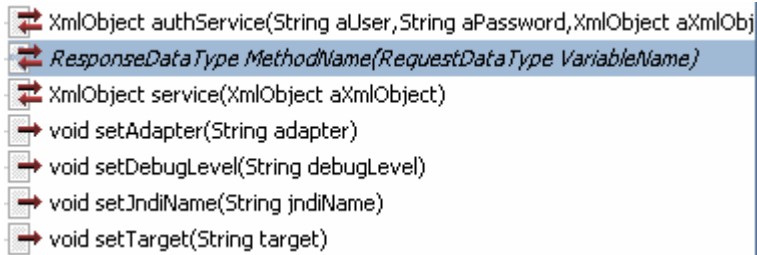
    // A call to a basic service only has a single parameter, which
    // is a subtype of XmlObject. It returns another XmlObject.
    public ResponseDataType MethodName(RequestDataType VariableName) throws Exception

    // A call to an authenticated service has two additional parameters
    // corresponding to the users credentials.
    // public BAPIMATERIALGETDETAILResponseDocument getDetail(String aUser, String a
}
    
```

Perform the following steps:

- a. Uncomment the public class definition.
- b. Change the existing response data type to match your response data type that is generated from your Clarify response schema.
- c. Change the existing method name to match your method.
- d. Change the existing request data type to match your request data type that is generated from your Clarify request schema.

The following control is now available in BEA WebLogic Workshop and can be added to a workflow:



Note: You can view available data types under the *XML Bean Classes* folder in the *Application* tab, which are added once you import your XML request or response schemas from Application Explorer.

These data types are case sensitive and must be entered exactly as shown.

Using the Extensible CCI Control

The extensible CCI control functions much like a database control since it generates JCX files to which you can add your own methods.

Your own methods can use the correct input and output types rather than the generic `XmlObject` types that the JCA control uses. Since the control is just a proxy that uses a reflection to call the relevant method, it handles the casting for you. You are no longer required to write custom code that does the cast or transformations that are cast between an `XmlObject`.

For example, instead of the generic `XmlObject`:

```
XmlObject service(XmlObject input) throws java.lang.Exception;
```

you call:

```
public ResponseDataType MethodName(RequestDataType VariableName) throws  
Exception;
```

where:

ResponseDataType

Is the XML Bean Class value that is generated from the response schema.

MethodName

Is the method name used by the extensible CCI control.

RequestDataType

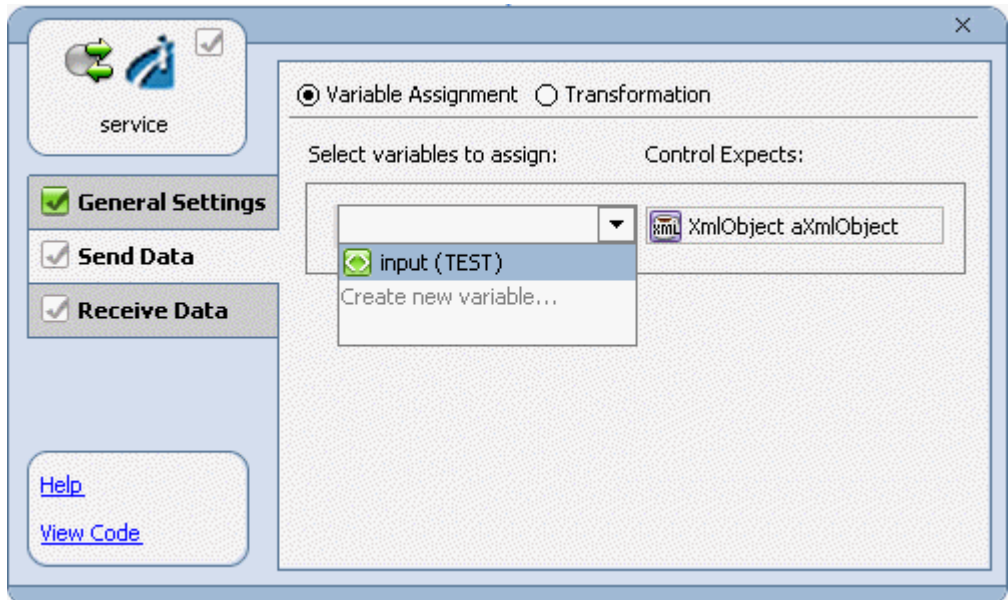
Is the XML Bean Class value that is generated from the request schema.

VariableName

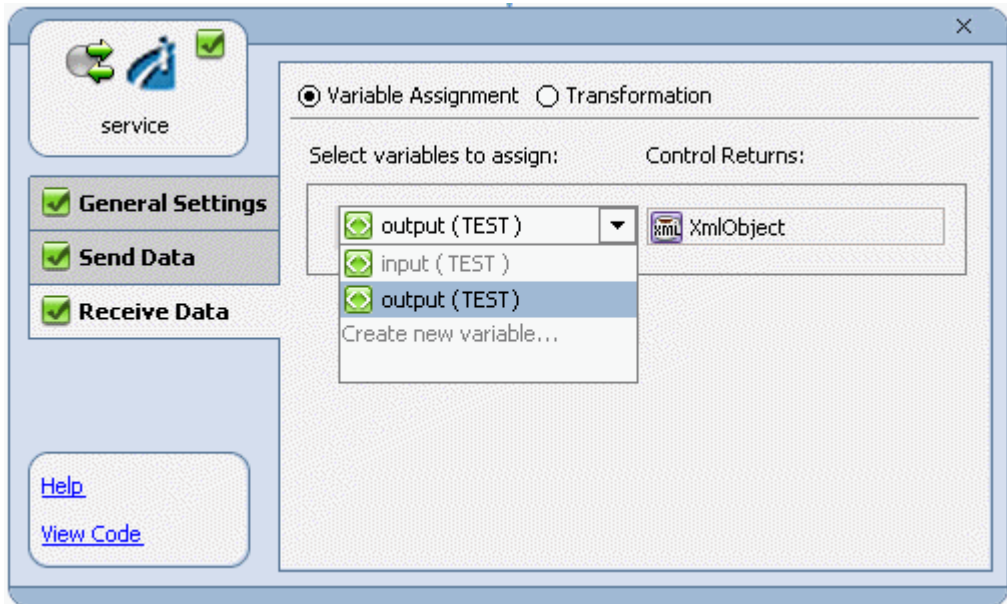
Is the request variable that stores the request document, which is used as input by the extensible CCI control.

Example: Using Dynamic Class Casting

The following example uses dynamic class casting to specify a schema-based input XmlObject to be casted into a pure XmlObject as a service method, which is expected by the CCI control.



The following example uses dynamic class casting where the CCI control returns a pure XmlObject, which is casted dynamically into a schema-based output XmlObject.



APPENDIX C

Using Application Explorer in BEA WebLogic Workshop for Event Handling with CBOs

Topics:

- Starting Application Explorer in BEA WebLogic Workshop
- Understanding iWay Event Functionality
- Creating an Event Port
- Modifying an Event Port
- Creating a Channel
- Modifying a Channel
- Deploying iWay Components in a Clustered BEA WebLogic Environment

This section describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop to listen for events in ClarifyCRM database tables through ClarifyCRM Business Objects (CBOs).

Starting Application Explorer in BEA WebLogic Workshop

The server must be started where iWay Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic Server.

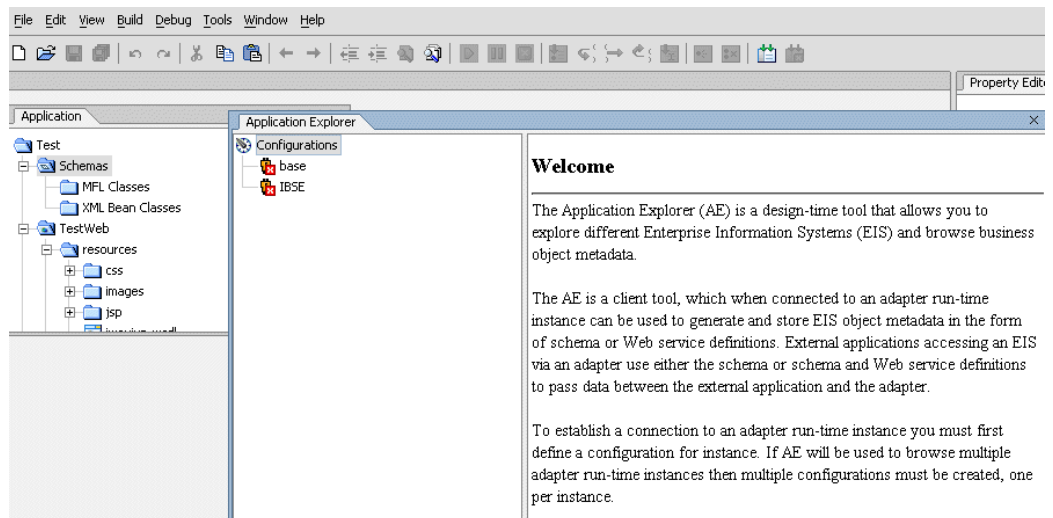
You can run Application Explorer in BEA WebLogic Workshop using an Integration Business Services Engine (IBSE) configuration or J2EE Connector Architecture (JCA) configuration.

Procedure: How to Start Application Explorer in BEA WebLogic Workshop

To start Application Explorer in BEA WebLogic Workshop:

1. Ensure that BEA WebLogic Server is running.
2. Start BEA WebLogic Workshop.
3. From the BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer*.

Application Explorer opens in BEA WebLogic Workshop, as shown in the following image. This image also shows the Application Explorer welcome statement that appears on the right when you select the Configurations node on the left.



You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

Understanding iWay Event Functionality

Events are generated as a result of activity in an application system. You can use events to trigger an action in your application. For example, ClarifyCRM may generate an event when customer information is updated. If your application must perform in response to activity, your application is a consumer of this event.

After you create a connection to your application system, you can add events using Application Explorer. To define a ClarifyCRM event, you must create a port and a channel.

- Port

A port associates a particular business object exposed by the adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption. For more information, see *Creating an Event Port on page C-3*.

- Channel

A channel represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by the adapter. For more information, see *Creating a Channel on page C-16*.

Creating an Event Port

You can create an event port in Application Explorer from either the Service Adapters node or the Event Adapters node. There are slight variations in the two methods, but both result in a new port; therefore, you can use either of the following procedures.

- *Create an Event Port from the Service Adapters Node on page C-4*
- *Create an Event Port from the Event Adapters Node on page C-6*

Several dispositions are available when defining a port. When you use Application Explorer with an Integration Business Services Engine (iBSE) implementation, the following port dispositions are available:

- File
- iBSE
- MSMQ
- JMSQ
- SOAP
- HTTP
- MQSeries

- Mail

Note: The Mail disposition will be supported in a future release.

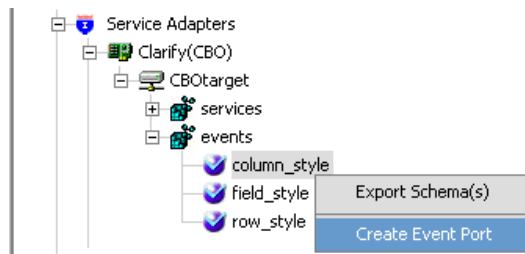
When you use Application Explorer with a JCA implementation, the following port dispositions are available:

- File
- JMSQ
- MQ Series
- HTTP

Procedure: How to Create an Event Port from the Service Adapters Node

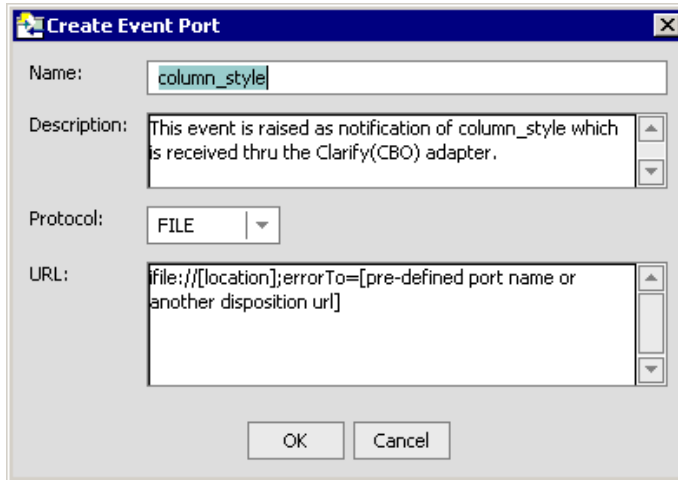
To create an event port from the Service Adapters node:

1. In the left pane of Application Explorer, expand the *Clarify(CBO)* node under *Service Adapters*.
2. Connect to the Clarify(CBO) target for which you want to create the event port.
3. Expand the target node and the *events* node, and select an event item (*column_style*, *field_style*, or *row_style*).



4. Right-click the event item and select *Create Event Port*.

The Create Event Port window opens. The fields that describe the event port are populated with the information associated with the event port item you selected. An example of this window with the `column_style` event item is shown in the following image.



- a. In the Name field, keep the default or type a new name for the event port.
 - b. In the Description field, keep the default or type a brief description of the port.
 - c. From the Disposition Protocol drop-down list, select a protocol.
 - d. In the URL field, keep the default or specify a URL to the destination file that will hold the event data. To find the URL format for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page C-7.
5. Click **OK**.

In the left pane, the Event Adapters node opens and the new port appears under the Ports node of the target for which you created the port.

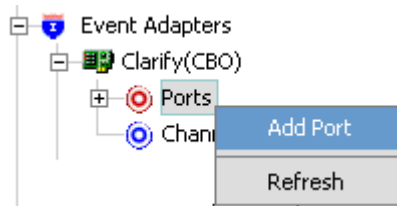
To see an example of a port node listing, go to *FILE Disposition URL Format* on page C-7.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page C-16.

Procedure: How to Create an Event Port from the Event Adapters Node

To create an event port from the Event Adapters node:

1. In the left pane of Application Explorer, expand the *Clarify(CBO)* node under *Event Adapters*.



2. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box opens.

A screenshot of the 'Add Port' dialog box. The dialog has a title bar with a close button. It contains four fields: 'Name' (a text box), 'Description' (a text box with vertical scrollbars), 'Protocol' (a drop-down menu showing 'FILE'), and 'URL' (a text box with vertical scrollbars). The URL field contains the text: 'file:///location];errorTo=[pre-defined port name or another disposition url]'. At the bottom are 'OK' and 'Cancel' buttons.

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description of the port.
- c. From the Protocol drop-down list, select the protocol.
- d. In the URL field, type a destination file to which the event data will be written. To find the URL format for the protocol you select, see the subsequent sections of disposition URL formats that begin with *FILE Disposition URL Format* on page C-7.

3. Click OK.

The new port appears under the Ports node of the target for which you created the port. To see an example of a port node listing, go to *FILE Disposition URL Format* on page C-7.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page C-16.

FILE Disposition URL Format

The format of the URL to the FILE disposition event document is:

- **When pointing Application Explorer to an iBSE deployment**, specify the destination file using the following format:

```
ifile://location[;errorTo=errorDest]
```

- **When pointing Application Explorer to a JCA deployment**, specify the destination file using the following format:

```
location
```

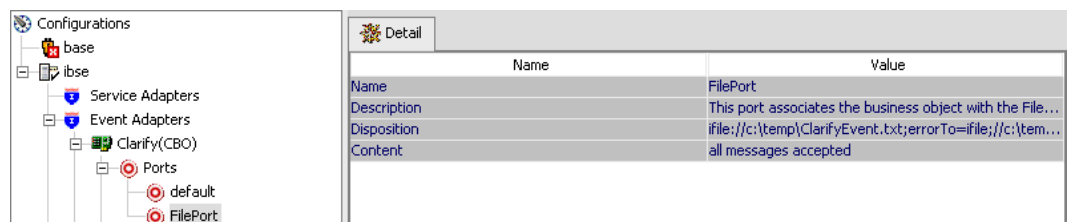
The following table describes the FILE disposition parameters.

Parameter	Description
<i>location</i>	Full directory path and file name to which the data is written.
<i>errorDest</i>	Location to which error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

For example:

```
ifile://c:\temp\ClarifyEvent.txt;errorTo=ifile://c:\temp\error
```

Once an event port is created, it appears in the left pane under the target ports node. Summary information associated with a selected event port appears in the right pane. The following image shows a listing of an event port with a FILE disposition.



iBSE Disposition URL Format

The format of the URL to the iBSE disposition event document is:

`ibse:/svcName.mthName[;responseTo=respDest][;errorTo=errorDest]`

The following table describes the disposition parameters.

Parameter	Description
<code>svcName</code>	Name of the service created with iBSE.
<code>mthName</code>	Name of the method created for the Web service.
<code>respDest</code>	Location where responses to the Web service are posted. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.
<code>errorDest</code>	Location where error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target ports node. Summary information associated with a selected event port appears in the right pane. To see an example of a port node listing, go to *FILE Disposition URL Format* on page C-7.

MSMQ Disposition URL Format

The format of the URL to the MSMQ disposition event document is:

`msmq:/host/queueType/queueName[;errorTo=errorDest]`

The following table defines the MSMQ disposition parameters.

Parameter	Description
<code>host</code>	Name of the host on which the Microsoft Queuing system runs.
<code>queueType</code>	Type of queue. For private queues, enter <i>Private\$</i> . Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
<code>queueName</code>	Name of the queue where messages are placed.

Parameter	Description
<code>errorDest</code>	Location where error logs are sent. Optional. Can be a pre-defined port name or another disposition URL. The URL must be complete, including the protocol.

Once an event port is created, it appears in the left pane under the target ports node. Summary information associated with a selected event port appears in the right pane. To see an example of a port node listing, go to *FILE Disposition URL Format* on page C-7.

JMS Disposition URL Format

The format of the URL to the JMS disposition event document is:

- When pointing Application Explorer to an **ibSE** deployment, specify the JMS destination file using the following URL format:

```
jmsq:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory;
user=userID;password=pass[;errorTo=errorDest]
```

- When pointing Application Explorer to a **JCA** deployment, specify the JMS destination file using the following URL format:

```
jms:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory
```

The following table describes the disposition parameters.

Parameter	Description
queue	Name of a queue to which events are emitted.
conn_factory	The connection factory, a resource that contains information about the JMS Server. The BEA WebLogic connection factory is: <code>javax.jms.QueueConnectionFactory</code>

Parameter	Description
jndi_url	<p>The URL to use to contact the JNDI provider. The syntax of this URL depends on which JNDI provider is being used. This value corresponds to the standard JNDI property:</p> <pre>java.naming.provider.url</pre> <p>For BEA WebLogic Server, this is</p> <pre>t3://host:port</pre> <p>where:</p> <pre>host</pre> <p>Is the machine name where BEA WebLogic Server is installed.</p> <pre>port</pre> <p>Is the port on which BEA WebLogic Server is listening. The default port if not changed at installation is 7001.</p>
jndi_factory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For BEA WebLogic Server, the BEA WebLogic factory is weblogic.jndi.WLInitialContextFactory.
userID	User ID associated with this queue.
pass	Password associated with the user ID.
errorDest	<p>Location to which error logs are sent. Optional.</p> <p>A predefined port name or another disposition URL. The URL must be complete, including the protocol.</p>

Once an event port is created, it appears in the left pane under the target ports node. Summary information associated with a selected event port appears in the right pane. To see an example of a port node listing, go to *FILE Disposition URL Format* on page C-7.

SOAP Disposition URL Format

The format of the URL to the SOAP disposition event document is:

```
soap:[wsdl-url];soapaction=[myaction];method=[web service
method];namespace=[namespace];responseTo=[pre-defined port name or
another disposition URL];errorTo=[pre-defined port name or another
disposition url]
```

The following table defines the parameters for the disposition.

Parameter	Description
wSDL-url	<p>The URL to the WSDL file that is required to create the SOAP message. For example:</p> <p>http://localhost:7001/ibse/IBSEServlet/test/webservice.ibs?wsdl</p> <p>where:</p> <p>webservice</p> <p>Is the name of the Web service you created using Application Explorer.</p> <p>To find this value, navigate to the Integration Business Services tab, expand the <i>Services</i> node, select the service you created, and click the <i>Service Description</i> link on the right. The WSDL URL appears in the Address field of the window that opens.</p> <p>You can also open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.</p>
soapaction	The method that will be called by the SOAP disposition. This value can be found in the WSDL file.
method	The Web service method you are using. This value is found in the WSDL file.
namespace	The XML namespace you are using. This value is found in the WSDL file.
responseTo	The location to which responses are posted. A predefined port name or another full URL. Optional. The URL must be complete, including the protocol.
errorTo	<p>The location to which error logs are sent. Optional.</p> <p>A predefined port name or another disposition URL. The URL must be complete, including the protocol.</p>

Once an event port is created, it appears in the left pane under the target ports node. Summary information associated with a selected event port appears in the right pane. To see an example of a port node listing, go to *FILE Disposition URL Format* on page C-7.

HTTP Disposition URL Format

The format of the URL to the HTTP disposition event document is:

- When pointing Application Explorer to an **ibSE** deployment, specify the HTTP destination file using the following format:

```
ihhttp://url[;responseTo=respDest]
```

- When pointing Application Explorer to a **JCA** deployment, specify the HTTP destination file using the following format:

```
http://host:port/uri
```

The following table describes the disposition parameters.

Parameter	Description
url	The URL target for the post operation.
respDest	Location where responses are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
host	Name of the machine on which the Web server resides.
port	Port number on which the Web server is listening.
uri	Universal resource identifier that completes the url specification.

Once an event port is created, it appears in the left pane under the target ports node. Summary information associated with a selected event port appears in the right pane. To see an example of a port node listing, go to *FILE Disposition URL Format* on page C-7.

MQSeries Disposition URL Format

The format of the URL to the MQSeries disposition event document is:

- When pointing Application Explorer to an **ibSE** deployment, specify the MQSeries destination file using the following format:

```
mqseries:/qManager/qName;host=hostName;port=portNum;  
channel=chanName[;errorTo=errorDest]
```

- When pointing Application Explorer to a **JCA** deployment, use the following format:

```
mq:qmanager@respqueue;host=;port=;channel=
```

The following table lists and describes the disposition parameters for MQSeries.

Parameter	Description
qManager	Name of the queue manager to which the server must connect.
qName or respqueue	Name of the queue where messages are placed.
host	Name of the machine on which MQSeries resides (MQ client only).
port	Port number for connecting to an MQ server (queue manager). MQ client only.
channel	The case-sensitive name of the channel that connects with the remote MQ server (queue manager). MQ client only. The default MQSeries channel name is SYSTEM.DEF.SVRCONN.
errorTo	Location where error documents are sent. Can be a predefined port name or another full URL. Optional.

Once an event port is created, it appears in the left pane under the target ports node. Summary information associated with a selected event port appears in the right pane. To see an example of a port node listing, go to *FILE Disposition URL Format* on page C-7.

Post Query Parameter Operators

When you configure a Table Listener, you can use two special field operators, ? and ^, with the SQL Post Query parameter. Both of these operators dynamically substitute database values in the SQL post-query statement at run time.

- ?fieldname is evaluated at run time as field = value.

The ? operator is useful in UPDATE statements:

```
UPDATE table WHERE ?field
```

For example, the following statement

```
UPDATE Stock_Prices_Temp WHERE ?RIC
```

might be evaluated at run time as:

```
UPDATE Stock_Prices_Temp WHERE RIC = 'PG'
```

- ^fieldname is evaluated at run time as value

The ^ operator is useful in INSERT statements:

```
INSERT INTO table VALUES (^field1, ^field2, ^field3, ... )
```

For example, the following statement

```
INSERT INTO Stock_Prices_Temp VALUES (^RIC, ^Price, ^Updated)
```

might be evaluated at run time as:

```
INSERT INTO Stock_Prices_Temp VALUES ('PG', 88.62, '2003-03-18  
16:24:00.0')
```

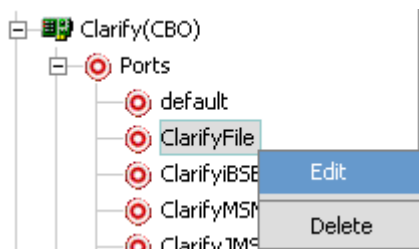
Modifying an Event Port

The following procedures describe how to edit and delete an event port using Application Explorer. If you want to review the port settings, select the port name. The right pane displays summary information associated with that event port.

Procedure: How to Edit an Event Port

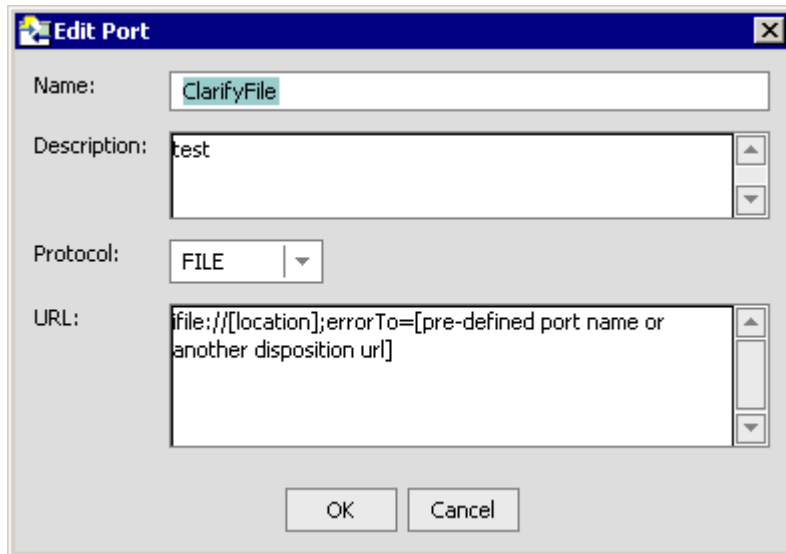
To edit an event port:

1. To view the available ports, expand the *Event Adapters* node, the *Clarify(CBO)* node, and then the *Ports* node.



2. Right-click the port you want to edit, and select *Edit*.

The Edit Port dialog box opens.

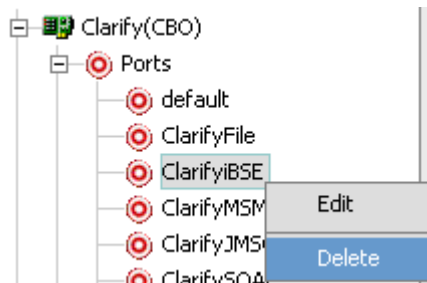


3. Make the required changes and click *OK*.

Procedure: How to Delete an Event Port

To delete an existing event port:

1. To view the available ports, expand the *Event Adapters* node, the *Clarify(CBO)* node, and then the *Ports* node.



2. Right-click the port you want to remove, and select *Delete*.

The event port node disappears from the ports list in the left pane.

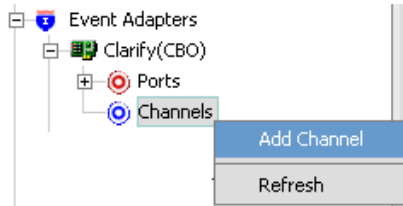
Creating a Channel

The following procedure describes how to create a channel for a Clarify(CBO) event. All defined event ports must be associated with a channel.

Procedure: How to Create a Channel

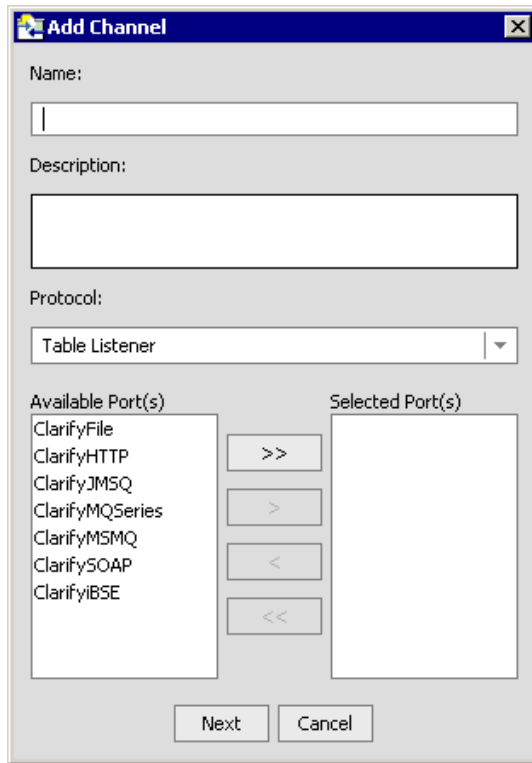
To create a channel:

1. In the left pane of Application Explorer, expand the *Clarify(CBO)* node under Event Adapters.



2. Right-click the *Channels* node and select *Add Channel*.

The Add Channel dialog box opens.

The image shows a Windows-style dialog box titled "Add Channel". It has a standard title bar with a close button. The dialog contains several fields: a "Name:" text box, a "Description:" text box, and a "Protocol:" drop-down menu currently showing "Table Listener". Below these are two list boxes: "Available Port(s)" on the left and "Selected Port(s)" on the right. The "Available Port(s)" list contains: ClarifyFile, ClarifyHTTP, ClarifyJMSQ, ClarifyMQSeries, ClarifyMSMQ, ClarifySOAP, and ClarifyBSE. Between the two list boxes are four arrow buttons: a double right arrow (>>), a single right arrow (>), a single left arrow (<), and a double left arrow (<<). At the bottom of the dialog are "Next" and "Cancel" buttons.

- a. In the Name field, type a name for the channel.
 - b. In the Description field, type a brief description of the channel.
 - c. From the Protocol drop-down list, select *Table Listener*.
 - d. To associate one or more available ports with this channel, select the port in the Available Port(s) box and click the single right arrow button to move it to the Selected Port(s) box. To move all ports listed in the Available Port(s) box to the Selected Port(s) box, click the double right arrow. Use the single or double left arrows to return items to the Available Port(s) box.
3. Click *Next*.

The Table Listener dialog box opens. This dialog box provides four tabs that represent the available listener parameters. The tabs are EDA Server Parameters, SQL Server Parameters, Oracle Parameters, and JDBC-ODBC Bridge Parameters. Each tab, when selected, reveals its associated parameters.

Note: Only the SQL Server Parameters and Oracle Parameters tabs apply to Clarify(CBO).

EDA Server Parameters

SQL Server Parameters

Oracle Parameters

JDBC-ODBC Bridge Parameters

Data Source

User

Password

Polling Interval

SQL Query

Post Query

Delete Keys

OK Cancel

Fields marked with * are required.

4. Enter values for the Table Listener parameters. The following table lists and defines the parameters found in the four tabs of the Table Listener dialog box.

Parameter	Description
Host	Name of the server where the ClarifyCRM database instance resides.
Port	Port number where the database is listening.

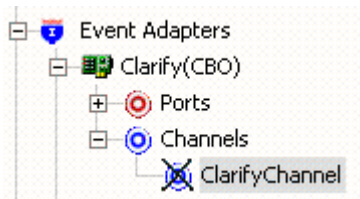
Parameter	Description
SID	A unique name of the database service, chosen by the database administrator or the person who installed ClarifyCRM.
Database Name	Name of the database.
User	The ClarifyCRM database user ID to access the ClarifyCRM database underlying the ClarifyCRM system. The user ID must have database access to the tables you want to access.
Password	Password associated with the specified user ID.
Polling Interval	A value, in seconds, at which to check for new input.
SQL Query	An SQL query, for example: <code>select * from iw_events</code>

Parameter	Description
Post Query*	<p>One or more SQL statements that are executed after each new record is read from the table. Separate multiple statements with a semicolon (;).</p> <p>Case sensitive: the case used to specify the column names must match the case used in the SELECT statement that polled the table. If the SQL Query property was omitted so that a default SELECT statement polled the table, the case used to specify the column names must match the case used to define the columns in the DBMS native schema.</p> <p>When you specify a value for this parameter, the table data is retained after it is read.</p> <p>If you do not specify a value for Post Query, each record read from the table is deleted after it is read, depending on whether you specify the Delete Keys property. If you:</p> <ul style="list-style-type: none"> • Specify the Delete Keys property, by default the adapter issues a DELETE statement with a WHERE clause containing every key column specified for the Delete Keys property. At run-time this will be faster than if you had not specified the Delete Keys property if there is an index on the key, or if there are fewer key columns than there are columns in the SELECT statement that polled the table. • Do not specify the Delete Keys property, by default the adapter issues a DELETE statement with a WHERE clause that specifies every column from the SELECT statement that polled the table. <p>* The SQL Post-query and Delete Keys parameters are mutually exclusive, as Delete Keys applies to the default DELETE statement, and SQL Post-query overrides the default DELETE statement. You can provide a value for one or the other, but not for both.</p> <p>There are two field operators, ? and ^, that you can use in a post-query SQL statement; for more information, see <i>Post Query Parameter Operators</i> on page C-13.</p>


Parameter	Description
Delete Keys*	<p>Comma-separated list of key columns used in the default DELETE statement. DELETE operates on keys, therefore specify the table key columns.</p> <p>Case sensitive: the case used to specify the column names must match the case used in the SELECT statement that polled the table. If the SQL Query property was omitted so that a default SELECT statement polled the table, the case used to specify the column names must match the case used to define the columns in the DBMS native schema.</p> <p>* The Delete Keys and SQL Post Query parameters are mutually exclusive. Delete Keys applies to the default DELETE statement and SQL Post Query overrides the default DELETE statement. You can provide a value for one or the other, but not for both. For more information, see the description of the Post Query parameter in this table.</p>

5. Click OK.

The channel appears below the Channels node in the left pane.

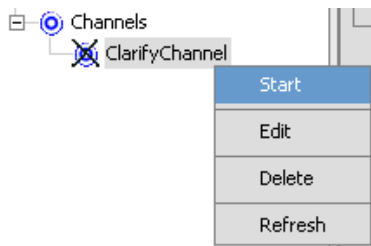


When you select a channel listed under the Channels node, summary information appears under the Detail tab on the right.

JDBC Parameters		Advanced	
Oracle Parameters	SQL Server Parameters	EDA Server Parameters	
 Detail		JDBC-ODBC Bridge Parameters	
Name	<input type="text" value="ClarifyChannel"/>		
Description	<input type="text" value="sample channel"/>		
Type	<input type="text" value="Table Listener"/>		
ports			
Name 0	<input type="text" value="EventPort1"/>		
Name 1	<input type="text" value="EventPort2"/>		

The ports field on the lower area of the Detail tab displays the name of the event port you assigned to this channel. The assigned parameters related to the table listeners are also provided through tabs located at the top of this pane.

You are ready to start your channel to listen for events.



- To activate your event configuration, right-click the channel node and select *Start*. To stop the channel at any time, right-click the channel and select *Stop*.

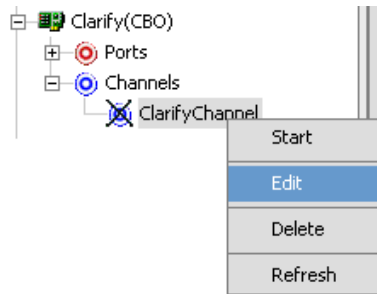
Modifying a Channel

The following procedures describe how to edit and delete a channel using Application Explorer. To review the channel settings, select the channel from the *Channels* list. The right pane displays summary information associated with the channel you created.

Procedure: How to Edit a Channel

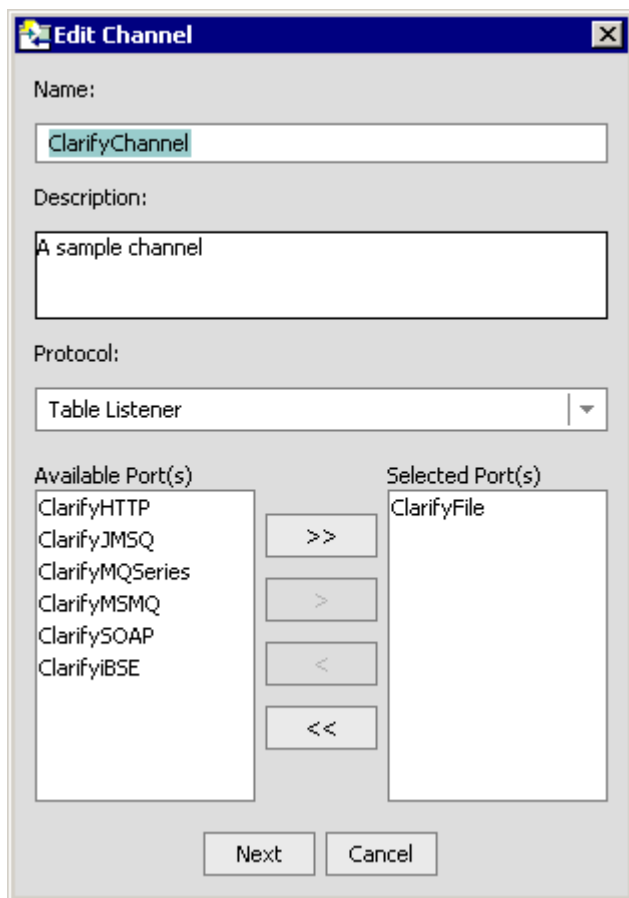
To edit a channel:

1. To view the available channels, select the *Channels* node under the target node in the left pane.



2. Right-click the channel you want to edit and select *Edit*.

The Edit Channel dialog box opens.



The image shows a Windows-style dialog box titled "Edit Channel". It contains the following fields and controls:

- Name:** A text box containing "ClarifyChannel".
- Description:** A text box containing "A sample channel".
- Protocol:** A dropdown menu currently showing "Table Listener".
- Available Port(s):** A list box containing "ClarifyHTTP", "ClarifyJMSQ", "ClarifyMQSeries", "ClarifyMSMQ", "ClarifySOAP", and "ClarifyIBSE".
- Selected Port(s):** A list box containing "ClarifyFile".
- Navigation Buttons:** Four buttons between the port lists: ">>", ">", "<", and "<<".
- Action Buttons:** "Next" and "Cancel" buttons at the bottom.

3. Make the required changes to the channel configuration.
4. Click Next.

The Edit Channel dialog box opens, as shown in the following image. This dialog box enables you to edit the table listener parameters.

Note: Only the SQL Server Parameters and Oracle Parameters tabs apply to Clarify(CBO).

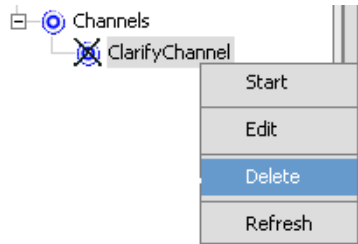
The image shows a dialog box titled "Edit Channel" with a close button (X) in the top right corner. The dialog box contains several tabs: "JDBC-ODBC Bridge Parameters", "Advanced", "JDBC Parameters", "EDA Server Parameters", "SQL Server Parameters", and "Oracle Parameters". The "Oracle Parameters" tab is currently selected. This tab contains a list of parameters, each with a text input field: "Host", "Port", "SID", "User", "Password", "Polling Interval", "SQL Query", "Post Query", and "Delete Keys". At the bottom of the dialog box, there are "OK" and "Cancel" buttons. Below the buttons, a red text label reads: "Fields marked with * are required."

5. Make the required changes and click **OK**.

Procedure: How to Delete a Channel

To delete an existing channel:

1. In the left pane, right-click the channel under the *Channels* list that you want to delete.



2. Select *Delete*.

The channel disappears from the Channels list.

Deploying iWay Components in a Clustered BEA WebLogic Environment

Events can be configured in a clustered BEA WebLogic environment. You can deploy iBSE or JCA to this environment. This topic uses iBSE as an example, but you can follow the same procedures when deploying JCA. The only difference is that you need to deploy the JCA connector .RAR file to the clustered environment.

A cluster consists of multiple server instances running simultaneously, yet appears to clients to be a single server instance. The server instances that contain a cluster can be run on one machine, but are usually run on multiple machines.

Clustering provides the following benefits:

- Load balancing
- High availability

Service requests are processed through the HTTP router and routed to an available managed server.

Events are server-specific and are not processed through the HTTP router. You must configure each server separately.

Procedure: How to Deploy iWay Components in a Clustered Environment

To deploy iWay components in a clustered environment:

1. Using the BEA Configuration Wizard:
 - a. Configure an administrative server to manage the managed servers.
 - b. Add and configure as many managed servers as required.

- c. Add and configure an HTTP router. This does not have to be a part of BEA WebLogic and can be an outside component.
- d. If you configure the HTTP router within BEA WebLogic, start it by entering the following command:

```
StartManagedWebLogic HTTPROUTER http://localhost:7001
```

where:

`HTTPROUTER`

Is the name of the server on which the HTTP router is running.

`http://localhost:7001`

Is the location of the admin console.

- e. Add the managed servers to your cluster/clusters.

For more information on configuring BEA WebLogic Integration for deployment in a clustered environment, see *Deploying WebLogic Integration Solutions*.

2. Start BEA WebLogic Server and open WebLogic Server Console.
3. Deploy iBSE to the cluster by selecting *Web Application Modules* from the Domain Configurations section, and clicking *Deploy a new Web Application Module*.

A page appears for you to specify where the Web application is located.

4. To deploy iBSE, select the option button next to the ibse directory and then click *Target Module*.

Deploy a Web Application Module

Select the archive for this Web application module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, [your file\(s\)](#) and/or confirm your Web application module contains valid descriptors.

Location: [localhost](#) \ [C:](#) \ [iWay55](#) \ bea

<input checked="" type="radio"/>	ibse
<input type="radio"/>	iwa
<input type="radio"/>	iwjcaivp

5. To deploy servlet Application Explorer, select the option button next to the `iwae` directory and then click *Target Module*.

If you are using servlet Application Explorer, deploy it only on the admin server or one of the managed servers.

Deploy a Web Application Module

Select the archive for this Web application module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should [upload your file\(s\)](#) and/or confirm your Web application module contains valid descriptors.

Location: `localhost \ C: \ Program Files \ iWay55 \ bea`

<input type="radio"/>	ibse
<input checked="" type="radio"/>	iwae
<input type="radio"/>	iwjcaivp

Target Module

The following window opens.

Select targets for this Web application module

Select the servers and/or clusters on which you want to deploy your new Web Application module

Independent Servers

☐ AdminServer
 ☐ HTTPROUTER

Clusters

☒ MYCluster

☒ All servers in the cluster
 ☐ Part of the cluster

☐ MS1
 ☐ MS2

6. Select the servers and/or clusters on which you want to deploy the application and click *Continue*.

The following window opens.

Source Accessibility

During runtime, a targeted server must be able to access this Web Application module's files. This access can be accomplished by either copying the Web Application module onto every server, or by defining a single location where the files exist.

How should the source files be made accessible?

- ☐ **Copy this Web Application module onto every target for me.**

During deployment, the files in this Web Application module will be copied automatically to each of the targeted locations.

- ☒ **I will make the Web Application module accessible from the following location:**

C:\iWay55\bea\ibse

Provide the location from where all targets will access this Web Application module's files. You must ensure the Web Application module's files exist in this location and that each target can reach the location.

7. Select the *I will make the Web Application module accessible from the following location* option button and provide the location from which all targets will access iBSE.

iWay Software recommends that you use a single instance of iBSE, rather than copying iBSE onto every target.

Note: iBSE must use a database repository (SQL or Oracle). Do not use a file repository. You can select this in the Repository Type drop-down list in the iBSE monitoring page. After configuring a database repository, you must restart all of the managed servers.

<http://hostname:port/ibse/IBSEConfig/>

where:

[hostname](#)

Is where your application server is running. Use the IP address or machine name in the URL; do not use localhost.

[port](#)

Is the port specific to each server, since you deploy iBSE to an entire cluster. For example, 8001, 8002, or any other port that is specified for each managed node.

8. Click *Deploy*.

Procedure: How to Configure Ports and Channels in a Clustered Environment

You can use Swing Application Explorer deployed in BEA WebLogic Workshop or Servlet Application Explorer to configure ports and channels in a clustered environment.

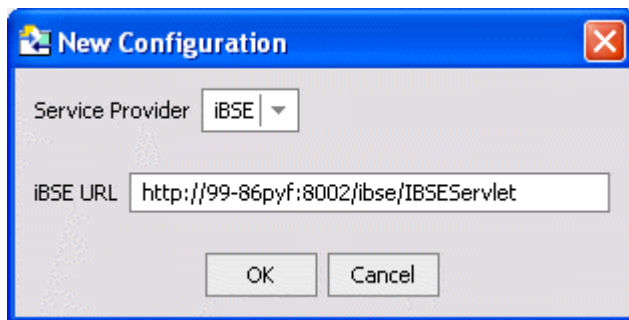
Note: Before using Servlet Application Explorer in a clustered environment, you must edit the web.xml file and specify the correct URL to your iBSE deployment. The default location on Windows is:

`C:\Program Files\iWay55\bea\iwaee\WEB-INF\web.xml`

For more information on configuring the web.xml file for Servlet Application Explorer, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

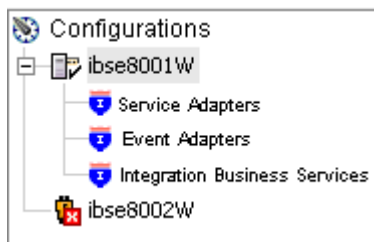
To configure ports and channels in a clustered environment:

1. Open Swing Application Explorer in BEA WebLogic Workshop.
2. Create a new connection to the iBSE instance. For information on creating a new configuration, see *Appendix A, Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services With CBO*.



Note: Use the IP address or machine name in the URL; do not use localhost.

3. Connect to the new configuration and select the *Event Adapters* node in the left pane of Application Explorer.

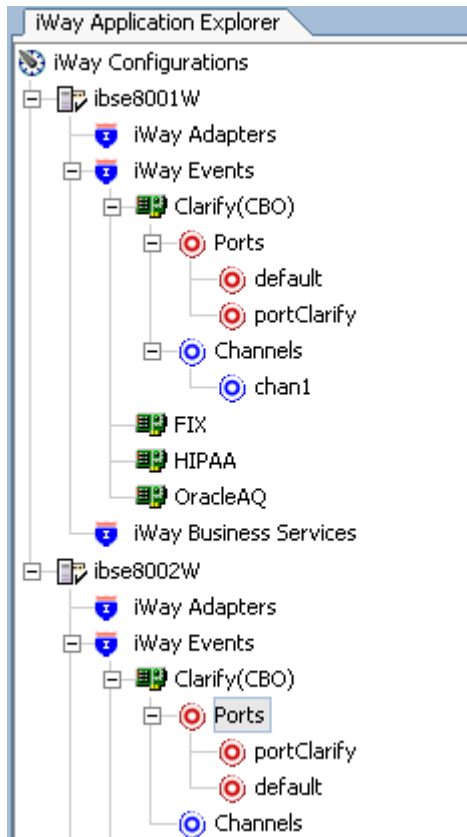


4. Add a new port for the ClarifyCRM adapter. For more information, see *Creating an Event Port* on page C-3.
5. Create a channel and add the port you created. For more information, see *Creating a Channel* on page C-16.
6. Click *Next* and enter the application server parameters.

7. Start the channel.
8. Create a new configuration and connect to the second iBSE instance.

The connection to iBSE must be configured to each instance of the managed server.

The following image shows two configurations.



The following operations performed on one managed server will be replicated on all other managed servers:

- Create port and channel: Creates the channel and port under all available servers.
- Delete port and channel. Deletes the port and channel under all available servers.

The following operations must be performed on each server:

- Start channel. Starts the channel for the specific server.
- Stop channel. Stops the channel for the specific server.

APPENDIX D

Using Application Explorer in BEA WebLogic Workshop for Event Handling with ClarifyCRM ClearBasic

Topics:

- Starting Application Explorer in BEA WebLogic Workshop
- Understanding iWay Event Functionality
- Creating an Event Port
- Modifying an Event Port
- Creating a Channel
- Modifying a Channel
- Deploying iWay Components in a Clustered BEA WebLogic Environment

This section describes how to use iWay Java Swing Application Explorer running in BEA WebLogic Workshop to listen for events in ClarifyCRM database tables with ClarifyCRM ClearBasic (CB).

Starting Application Explorer in BEA WebLogic Workshop

The server must be started where iWay Application Explorer is running. Before you can use Application Explorer, you must start BEA WebLogic Server.

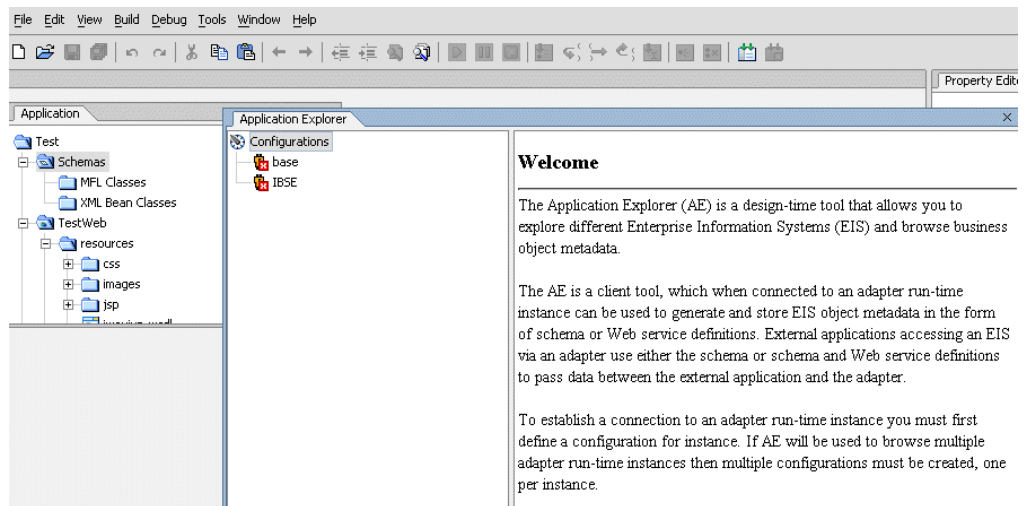
You can run Application Explorer in BEA WebLogic Workshop using an Integration Business Services Engine (IBSE) configuration or J2EE Connector Architecture (JCA) configuration.

Procedure: How to Start Application Explorer in BEA WebLogic Workshop

To start Application Explorer in BEA WebLogic Workshop:

1. Ensure that BEA WebLogic Server is running.
2. Start BEA WebLogic Workshop.
3. From BEA WebLogic Workshop View menu, select *Windows* and then, *Application Explorer*.

Application Explorer opens in BEA WebLogic Workshop, as shown in the following image. This image also shows the Application Explorer welcome statement that appears on the right when you select the Configurations node on the left.



You can resize and drag-and-drop the Application Explorer window within BEA WebLogic Workshop. For example, you can drag it to the upper part of BEA WebLogic Workshop.

Understanding iWay Event Functionality

Events are generated as a result of activity in an application system. You can use events to trigger an action in your application. For example, ClarifyCRM may generate an event when customer information is updated. If your application must perform in response to activity, your application is a consumer of this event.

After you create a connection to your application system, you can add events using Application Explorer. To define a ClarifyCRM event, you must create a port and a channel.

- Port

A port associates a particular business object exposed by the adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption. For more information, see *Creating an Event Port on page D-3*.

- Channel

A channel represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by the adapter. For more information, see *Creating a Channel on page D-24*.

Creating an Event Port

The following procedures describe how to create an event port using Application Explorer. The following port dispositions are available:

- File
- iBSE
- MSMQ
- JMSQ
- SOAP
- HTTP
- MQSeries
- Mail

Note: The Mail disposition will be supported in a future release.

With a JCA implementation, the following port dispositions are available:

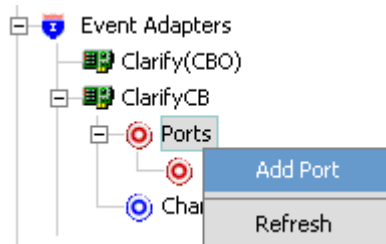
- File
- JMSQ

- MQ Series
- HTTP

Procedure: How to Create an Event Port for File

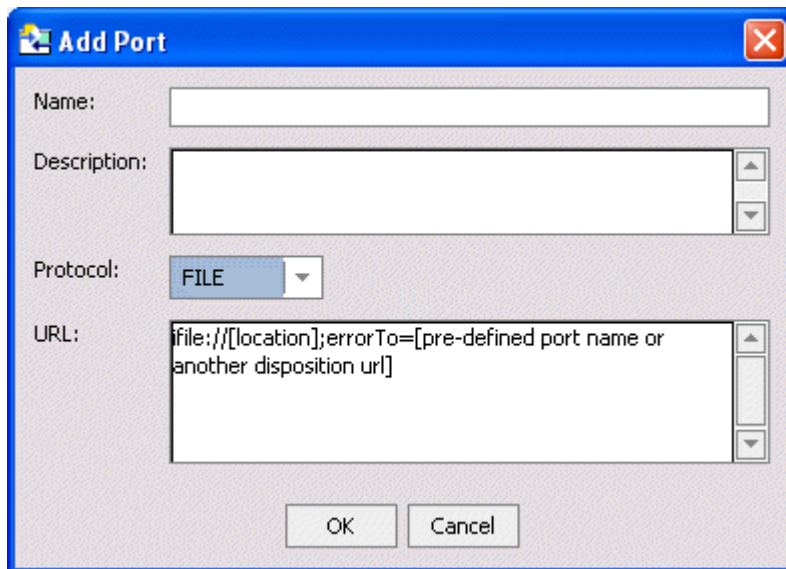
To create an event port for File:

1. In the left pane of Application Explorer, expand the *ClarifyCB* node under Event Adapters.



2. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box opens, as shown in the following image.



- a. In the Name field, type a name for the event port, for example, ClarifyFile.
- b. In the Description field, type a brief description of the port.
- c. From the Protocol drop-down list, select *FILE*.

- d. In the URL field, type a destination file to which the event data is written.

When pointing Application Explorer to an iBSE deployment, specify the destination file using the following format:

```
file://location[;errorTo=errorDest]
```

When pointing Application Explorer to a JCA deployment, specify the destination file using the following format:

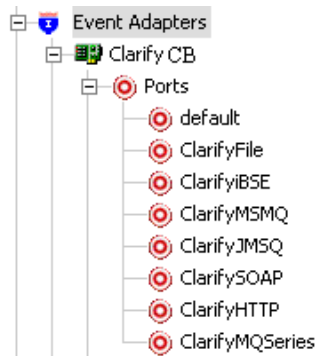
```
location
```

The following table describes the URL parameters.

Parameter	Description
<i>location</i>	The full directory path and file name to which the data is written.
<i>errorTo</i>	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click OK.

In the left pane, the event port appears below the Ports node.



4. To review the port settings, select the port name.

In the right pane, a table appears that summarizes the information associated with the event port you created.

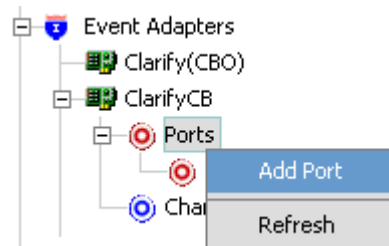
Detail	
Name	Value
Name	ClarifyFile
Description	test
Disposition	ifile:///location];errorTo=[pre-defined port ...
Content	all messages accepted

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page D-24.

Procedure: How to Create an Event Port for iBSE

To create an event port for iBSE:

1. In the left pane of Application Explorer, expand the *ClarifyCB* node under Event Adapters.



2. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box opens.

Add Port

Name:

Description:

Protocol: **IBSE**

URL:

OK Cancel

- In the Name field, type a name for the event port, for example, ClarifyiBSE.
- In the Description field, type a brief description of the port.
- From the Protocol drop-down list, select *IBSE*.
- In the URL field, enter an iBSE destination using the following format:

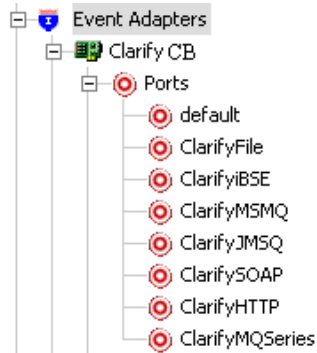
`ibse:/svcName.mthName[;responseTo=respDest][;errorTo=errorDest]`

The following table describes the disposition parameters.

Parameter	Description
<i>svcName</i>	Name of the service created with iBSE.
<i>mthName</i>	Name of the method created for the Web service.
<i>respDest</i>	Location where responses to the Web service are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.
<i>errorDest</i>	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click **OK**.

In the left pane, the event port appears below the Ports node.



4. To review the port settings, select the port name.

In the right pane, a table appears that summarizes the information associated with the event port you created.

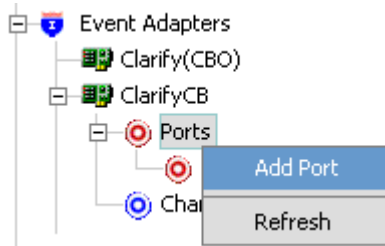
Detail	
Name	Value
Name	ClarifyIBSE
Description	
Disposition	ibse:[svcName].[mthName];responseTo=[p...
Content	all messages accepted

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page D-24.

Procedure: How to Create an Event Port for MSMQ

To create an event port for a Microsoft Message Queuing (MSMQ) queue:

1. In the left pane of Application Explorer, expand the *ClarifyCB* node under Event Adapters.



2. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box opens.

- a. In the Name field, type a name for the event port, for example, *ClarifyMSMQ*.
- b. In the Description field, type a brief description of the port.
- c. From the Protocol drop-down list, select *MSMQ*.
- d. In the URL field, enter an MSMQ destination in the following format:

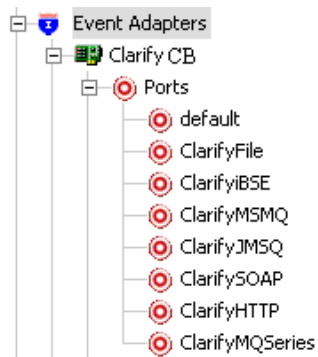
`msmq: / host / queueType / queueName [; errorTo=errorDest]`

The following table defines the disposition parameters.

Parameter	Description
<i>host</i>	Name of the machine on which the Microsoft Queuing system runs.
<i>queueType</i>	The type of queue. For private queues, enter <i>Private\$</i> . Private queues are queues that are not published in Active Directory. They appear only on the local computer that contains them. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
<i>queueName</i>	Name of the queue where messages are placed.
<i>errorDest</i>	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click OK.

In the left pane, the event port appears below the Ports node.



4. To review the port settings, select the port name.

In the right pane, a table appears that summarizes the information associated with the port you created.

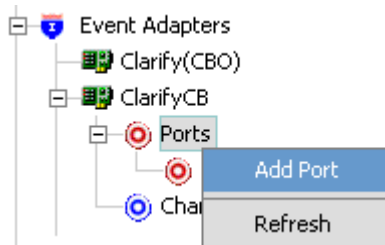
Detail	
Name	Value
Name	ClarifyMSMQ
Description	
Disposition	msmq://[machineName]/private\$/[qName];e...
Content	all messages accepted

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page D-24.

Procedure: How to Create a Port for JMS

To create a port for a JMS queue:

1. In the left pane of Application Explorer, expand the *ClarifyCB* node under Event Adapters.



2. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box opens.

- a. In the Name field, type a name for the event port, for example, ClarifyJMSQ.
- b. In the Description field, type a brief description of the port.
- c. From the Protocol drop-down list, select *JMSQ*.
- d. In the URL field, type a destination file to which the event data is written.

When pointing Application Explorer to an iBSE deployment, specify the JMS destination file using the following URL format:

```
jmsq:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory;
user=userID;password=pass[;errorTo=errorDest]
```

When pointing Application Explorer to a JCA deployment, specify the destination file using the following format:

```
jms:queue@conn_factory;jndiurl=jndi_url;jndifactory=jndi_factory
```

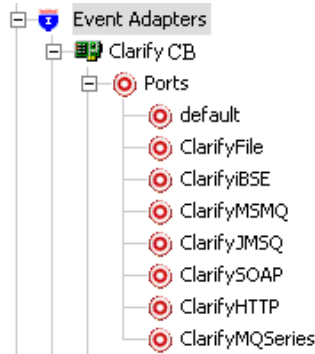
The following table describes the URL parameters.

Parameter	Description
<i>queue</i>	Name of a queue to which events are emitted.

Parameter	Description
<i>conn_factory</i>	The connection factory, a resource that contains information about the JMS Server. The BEA WebLogic connection factory is: <i>javax.jms.QueueConnectionFactory</i>
<i>jndi_url</i>	The URL of the application server. For BEA WebLogic Server, the URL is <i>t3://host:port</i> where: <i>host</i> Is the machine name where BEA WebLogic Server resides. <i>port</i> Is the port on which BEA WebLogic Server is listening. The default port, if not changed at installation, is 7001.
<i>jndi_factory</i>	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For BEA WebLogic Server, the BEA WebLogic factory is <i>weblogic.jndi.WLInitialContextFactory</i> .
<i>userID</i>	User ID associated with this queue.
<i>pass</i>	Password associated with this user ID.
<i>errorDest</i>	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click OK.

The event port appears below the Ports node in the left pane.



4. To review the port settings, select the port name.

In the right pane, a table appears that summarizes the information associated with the event port you created.

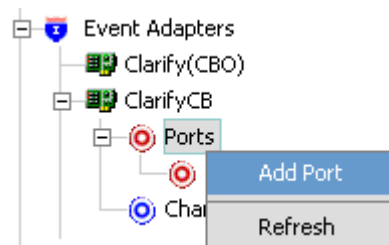
Detail	
Name	Value
Name	ClarifyJMSQ
Description	
Disposition	json:[myQueueName]@[myQueueFac];jndi...
Content	all messages accepted

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page D-24.

Procedure: How to Create a Port for the SOAP Disposition

To create a port for a SOAP disposition:

1. In the left pane of Application Explorer, expand the *ClarifyCB* node under Event Adapters.



2. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box opens.

- a. In the Name field, type a name for the event port.
- b. In the Description field, type a brief description of the port.
- c. From the Protocol drop-down list, select *SOAP*.
- d. In the URL field, type a destination file to which the event data is written, using the following format:

```
soap:[wsdl-url];soapaction=[myaction];method=[web service
method];namespace=[namespace];responseTo=[pre-defined port name or
another disposition URL];errorTo=[pre-defined port name or another
disposition url]
```

The following table defines the parameters for the disposition.

Parameter	Description
<code>wSDL-url</code>	<p>The URL to the WSDL file that is required to create the SOAP message. For example:</p> <p><code>http://localhost:7001/ibse/IBSEServlet/test/webservice.ibs?wSDL</code></p> <p>where:</p> <p><code>webservice</code></p> <p>Is the name of the Web service you created using Application Explorer.</p> <p>To find this value, navigate to the Integration Business Services tab, expand the <i>Services</i> node, select the service you created, and click the <i>Service Description</i> link on the right. The WSDL URL appears in the Address field of the window that opens.</p> <p>You can also open the WSDL file in a third party XML editor (for example, XMLSPY) and view the SOAP request settings to find this value.</p>
<code>soapaction</code>	The method that will be called by the SOAP disposition. This value can be found in the WSDL file.
<code>method</code>	The Web service method you are using. This value is found in the WSDL file.
<code>namespace</code>	The XML namespace you are using. This value is found in the WSDL file.
<code>responseTo</code>	The location to which responses are posted. A predefined port name or another full URL. Optional. The URL must be complete, including the protocol.
<code>errorTo</code>	<p>The location to which error logs are sent. Optional.</p> <p>A predefined port name or another disposition URL. The URL must be complete, including the protocol.</p>

3. Click OK.

The event port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

Detail	
Name	Value
Name	ClarifySOAP
Description	
Disposition	soap:[wsdl-url];soapaction=[myaction];met...
Content	all messages accepted

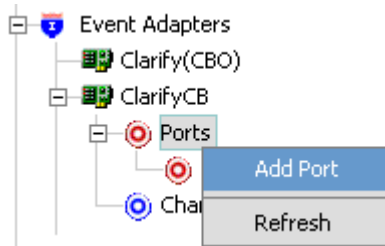
You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page D-24.

Procedure: How to Create an Event Port for HTTP

The HTTP disposition uses an HTTP URL to specify an HTTP end point to which the event document is posted.

To create an event port for HTTP disposition:

1. In the left pane of Application Explorer, expand the *ClarifyCB* node under Event Adapters.



2. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box opens.

- a. In the Name field, type a name for the event port, for example, ClarifyHTTP.
- b. In the Description field, type a brief description of the port.
- c. From the Protocol drop-down list, select *HTTP*.
- d. In the URL field, enter an HTTP destination:

When pointing Application Explorer to an iBSE deployment, specify the destination file using the following format:

`http://url[;responseTo=respDest]`

When pointing Application Explorer to a JCA deployment, specify the destination file using the following format:

`http://host:port/uri`

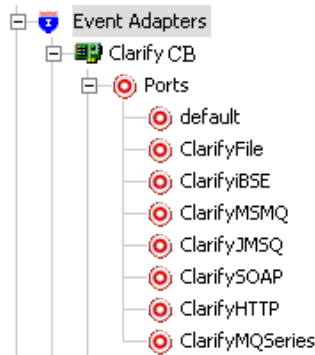
The following table describes the URL parameters.

Parameter	Description
<code>url</code>	The URL target for the post operation.
<code>respDest</code>	Location where responses are posted. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

Parameter	Description
<i>host</i>	Name of the host on which the Web server resides.
<i>port</i>	Port number on which the Web server is listening.

3. Click **OK**.

The event port appears below the Ports node in the left pane.



4. To review the port settings, select the port name.

In the right pane, a table appears that summarizes the information associated with the event port you created.

Detail	
Name	Value
Name	ClarifyHTTP
Description	
Disposition	http://[myurl];responseTo=[pre-defined po...
Content	all messages accepted

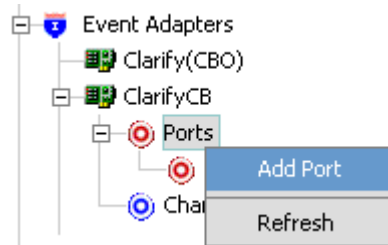
You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page D-24.

Procedure: How to Create an Event Port for MQSeries

The MQSeries disposition allows an event to be enqueued to an MQSeries queue. You can specify both queue manager and queue name.

To create a port for an MQSeries queue:

1. In the left pane of Application Explorer, expand the *ClarifyCB* node under Event Adapters.



2. Right-click the *Ports* node and select *Add Port*.

The Add Port dialog box opens.

The 'Add Port' dialog box is shown with the following fields and options:

- Name:** A text input field.
- Description:** A text area with up and down arrow buttons on the right.
- Protocol:** A drop-down menu currently showing 'MQ Series'.
- URL:** A text area containing the template: `mqseries:/[qManager]/[qName];host=[hostname];port=[port];channel=[channelname];errorTo=[pre-defined port name or another disposition url]`. It has up and down arrow buttons on the right.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

- a. In the Name field, type a name for the event port, for example, *ClarifyMQSeries*.
- b. In the Description field, type a brief description of the port.
- c. From the Protocol drop-down list, select *MQSeries*.

- d. In the URL field, enter an MQSeries destination.

When pointing Application Explorer to an iBSE deployment, specify the destination file using the following format:

```
mqseries:/qManager/qName;host=hostName;port=portNum;  
channel=chanName[;errorTo=errorDest]
```

When pointing Application Explorer to a JCA deployment, use the following format:

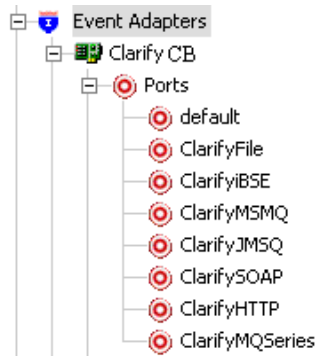
```
mq:qmanager@respqueue;host=;port=;channel=
```

The following table describes the URL parameters.

Parameter	Description
<i>qManager</i>	Name of queue manager to which the server must connect.
<i>qName</i>	Name of the queue where messages are placed.
<i>hostName</i>	Name of the host on which MQSeries resides (MQ client only).
<i>portNum</i>	Port number for connecting to an MQ Server queue manager (MQ client only).
<i>chanName</i>	Case-sensitive name of the channel that connects with the remote MQ Server queue manager (MQ client only). The default MQSeries channel name is SYSTEM.DEF.SVRCONN.
<i>errorDest</i>	Location where error logs are sent. Optional. A predefined port name or another disposition URL. The URL must be complete, including the protocol.

3. Click OK.

The event port appears below the Ports node in the left pane.



4. To review the port settings, select the port name.

In the right pane, a table appears that summarizes the information associated with the event port you created.

Detail	
Name	Value
Name	ClarifyMQSeries
Description	
Disposition	mqseries://[qManager]/[qName];host=[host...
Content	all messages accepted

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page D-24.

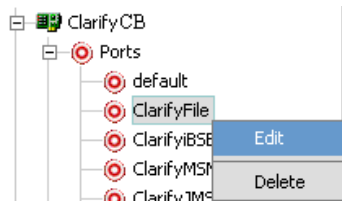
Modifying an Event Port

The following procedures describe how to edit and delete an event port using Application Explorer. If you want to review the port settings, select the port name. The right pane displays summary information associated with that event port.

Procedure: How to Edit an Event Port

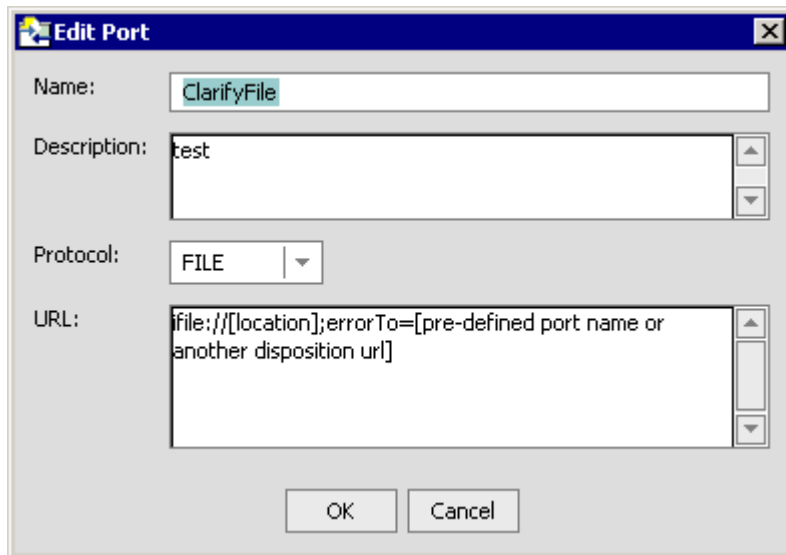
To edit an event port:

1. To view the available ports, expand the *Event Adapters* node, the *ClarifyCB* node, and then the *Ports* node.



2. Right-click the port you want to edit, and select *Edit*.

The Edit Port dialog box opens.

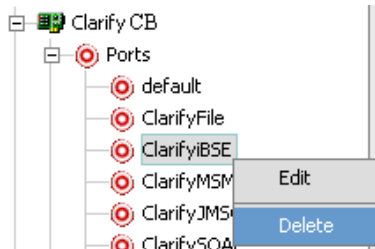


3. Make the required changes and click *OK*.

Procedure: How to Delete an Event Port

To delete an existing event port:

1. To view the available ports, expand the *Event Adapters* node, the *ClarifyCB* node, and then the *Ports* node.



2. Right-click the port you want to remove, and select *Delete*.

The event port node disappears from the ports list in the left pane.

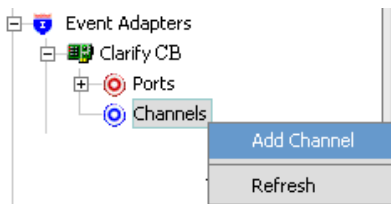
Creating a Channel

The following procedure describes how to create a channel for a ClarifyCB event. All defined event ports must be associated with a channel.

Procedure: How to Create a Channel

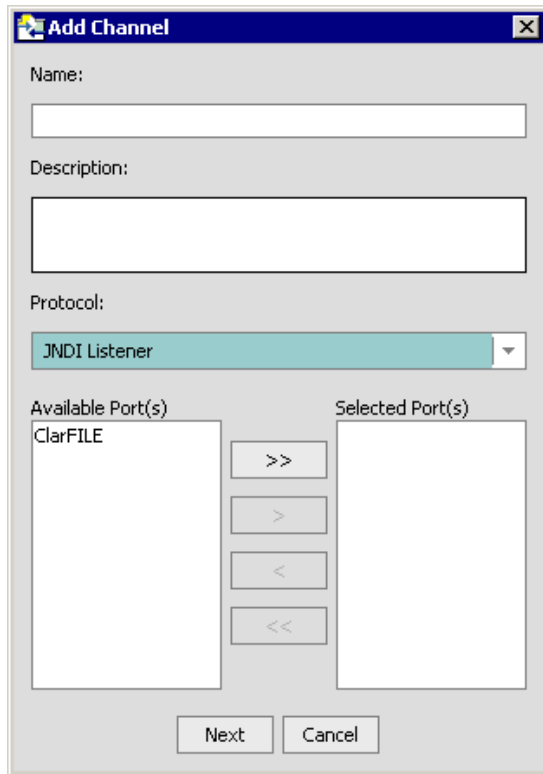
To create a channel:

1. In the left pane of Application Explorer, expand the *ClarifyCB* node under Event Adapters.



2. Right-click the *Channels* node and select *Add Channel*.

The Add Channel dialog box opens.



The image shows a Windows-style dialog box titled "Add Channel". It contains the following fields and controls:

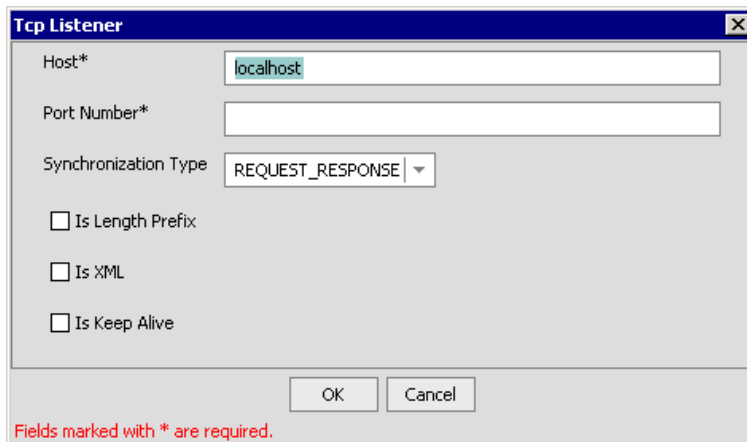
- Name:** A text input field.
- Description:** A larger text input field.
- Protocol:** A drop-down menu currently showing "JNDI Listener".
- Available Port(s):** A list box containing the text "ClarFILE".
- Selected Port(s):** An empty list box.
- Navigation Buttons:** Four buttons between the port lists: a double right arrow (>>), a single right arrow (>), a single left arrow (<), and a double left arrow (<<).
- Footer Buttons:** "Next" and "Cancel" buttons at the bottom.

- a. In the Name field, type a name for the channel.
 - b. In the Description field, type a brief description of the channel.
 - c. From the Protocol drop-down list, select either *JNDI Listener* or *TCP Listener*.
 - d. To associate one or more available ports with this channel, select the port in the Available Port(s) box and click the single right arrow button to move it to the Selected Port(s) box. To move all ports listed in the Available Port(s) box to the Selected Port(s) box, click the double right arrow. Use the single or double left arrows to return items to the Available Port(s) box.
- 3.** Click *Next*.

If you chose the JNDI Listener as the channel type, the following window opens on the right.

A dialog box titled "JNDI Listener" with a close button (X) in the top right corner. It contains a text field labeled "JNDI Name*" with the value "IW_JNDI_LISTENER" entered. Below the text field are "OK" and "Cancel" buttons. At the bottom, a red text label reads "Fields marked with * are required."

If you chose the TCP Listener as the channel type, the following window opens on the right.

A dialog box titled "Tcp Listener" with a close button (X) in the top right corner. It contains several fields: "Host*" with the value "localhost", "Port Number*" (empty), and "Synchronization Type" with a dropdown menu showing "REQUEST_RESPONSE". Below these are three checkboxes: "Is Length Prefix", "Is XML", and "Is Keep Alive", all of which are unchecked. At the bottom are "OK" and "Cancel" buttons. A red text label at the bottom reads "Fields marked with * are required."

4. Type or select the appropriate channel values in the JNDI or TCP Listener window.

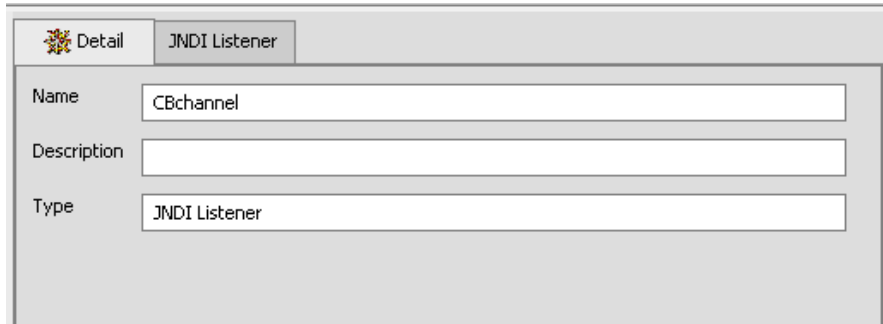
In the JNDI Listener window, edit the JNDI name value, if necessary.

In the TCP Listener window, edit the Host name, if necessary; type a value in the Port Number field, select a synchronization type (REQUEST_RESPONSE, REQUEST_ACK, or REQUEST) from the drop-down list; and select one or more of the check boxes (Is Length Prefix, Is XML, Is Keep Alive).

5. Click OK.

The new channel appears under the Channels node on the left. An X over the channel icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

When you select a channel listed under the Channels node, summary information appears in the right pane.



Procedure: How to Start a Channel

To start a channel:

1. In the left pane, expand the *Event Adapters* node, the *ClarifyCB* node, and the *Channels* node.
2. Right-click the channel you want to start and select *Start*.

The channel becomes active. To indicate a channel is active, the channel icon on the left no longer displays an X. In the following image, CBchannel is active, while CBchannel2 is not.



To stop the channel at any time, right-click the channel and select *Stop*.

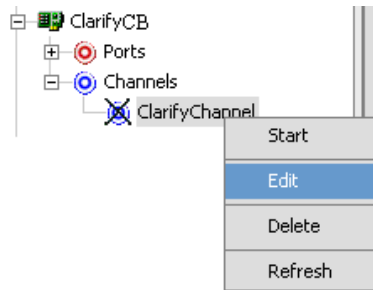
Modifying a Channel

The following procedures describe how to edit and delete a channel using Application Explorer. To review the channel settings, select the channel from the *Channels* list. The right pane displays summary information associated with the channel you created.

Procedure: How to Edit a Channel

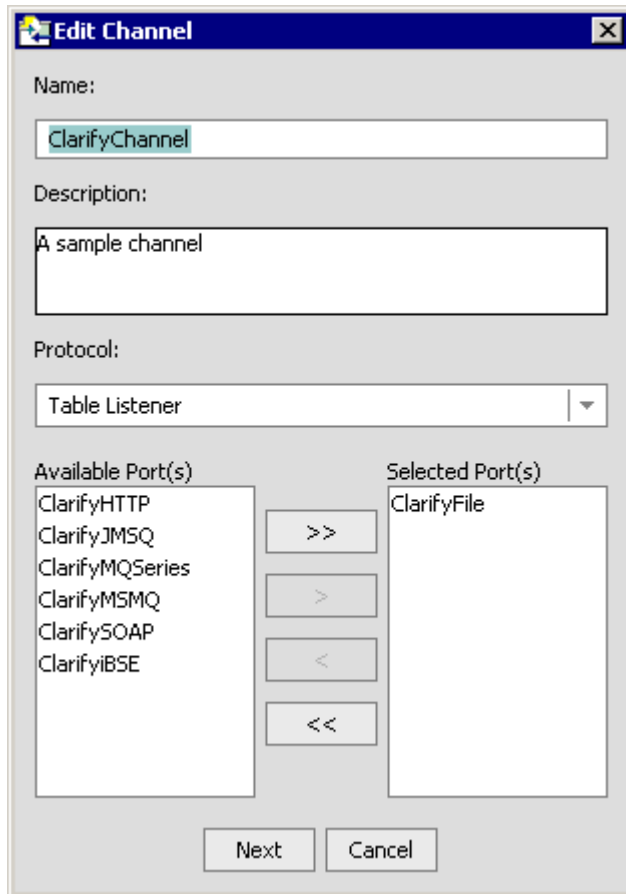
To edit a channel:

1. To view the available channels, select the *Channels* node under the target node in the left pane.




2. Stop the channel if it is active.
3. Right-click the channel you want to edit and select *Edit*.

The Edit Channel dialog box opens, as shown in the following image.



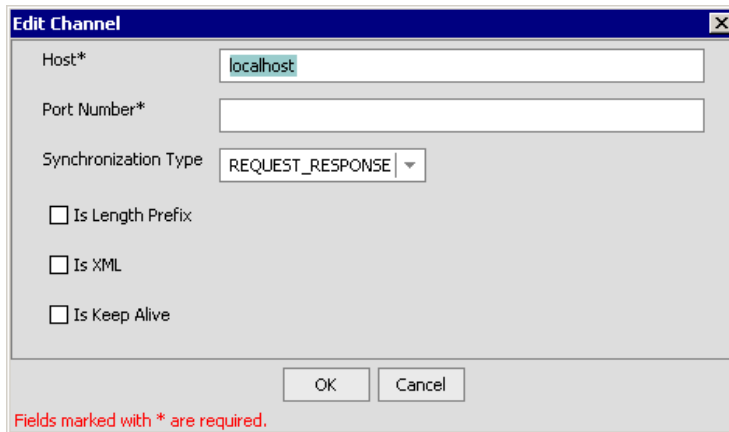
4. Make the required changes to the channel configuration.
5. Click Next.

If the channel type is JNDI Listener, the following window opens on the right.



The 'Edit Channel' dialog box for a JNDI Listener. It has a title bar with a close button. The main area contains a text field labeled 'JNDI Name*' with the value 'IW_JNDI_LISTENER'. Below the text field are 'OK' and 'Cancel' buttons. At the bottom, a red note states 'Fields marked with * are required.'

If the channel type is a TCP Listener, the following window opens on the right.



The 'Edit Channel' dialog box for a TCP Listener. It has a title bar with a close button. The main area contains several fields: 'Host*' with the value 'localhost', 'Port Number*' (empty), 'Synchronization Type' with a dropdown menu showing 'REQUEST_RESPONSE', and three checkboxes: 'Is Length Prefix', 'Is XML', and 'Is Keep Alive'. Below these fields are 'OK' and 'Cancel' buttons. At the bottom, a red note states 'Fields marked with * are required.'

6. Type or select the appropriate channel values in the JNDI or TCP Listener window.

In the JNDI Listener window, edit the JNDI name value, if necessary.

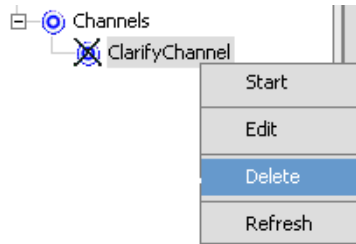
In the TCP Listener window, edit the Host name, if necessary; type a value in the Port Number field, select a synchronization type (REQUEST_RESPONSE, REQUEST_ACK, or REQUEST) from the drop-down list; and select one or more of the check boxes (Is Length Prefix, Is XML, Is Keep Alive).

7. Click OK.

Procedure: How to Delete a Channel

To delete an existing channel:

1. In the left pane, right-click the channel under the *Channels* list that you want to delete.



2. Select *Delete*.

The channel disappears from the Channels list.

Deploying iWay Components in a Clustered BEA WebLogic Environment

iWay events can be configured in a clustered BEA WebLogic environment.

A cluster consists of multiple server instances running simultaneously, yet appears to clients to be a single server instance. The server instances that contain a cluster can be run on one machine, but are usually run on multiple machines.

Clustering provides the following benefits:

- Load balancing
- High availability

Service requests are processed through the HTTP router and routed to an available managed server.

Events are server-specific and are not processed through the HTTP router. You must configure each server separately.

Procedure: How to Deploy iWay Components in a Clustered Environment

To deploy iWay components in a clustered environment:

1. Using the BEA Configuration Wizard:
 - a. Configure an administrative server to manage the managed servers.
 - b. Add and configure as many managed servers as required.
 - c. Add and configure an HTTP router. This does not have to be a part of BEA WebLogic and can be an outside component.

- d. If you configure the HTTP router within BEA WebLogic, start it by entering the following command:

```
StartManagedWebLogic HTTPROUTER http://localhost:7001
```

where:

HTTPROUTER

Is the name of the server on which the HTTP router is running.

http://localhost:7001

Is the location of the admin console.

- e. Add the managed servers to your cluster/clusters.

For more information on configuring BEA WebLogic Integration for deployment in a clustered environment, see *Deploying WebLogic Integration Solutions*.

2. Start BEA WebLogic Server and open WebLogic Server Console.
3. Deploy iBSE to the cluster by selecting *Web Application Modules* from the Domain Configurations section, and clicking *Deploy a new Web Application Module*.

A page appears for you to specify where the Web application is located.

Note: You can deploy JCA to a cluster, but you can only point it to one directory, and to the machine on which it is installed.

4. To deploy iBSE, select the option button next to the ibse directory and then click *Target Module*.

Deploy a Web Application Module

Select the archive for this Web application module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, [your file\(s\)](#) and/or confirm your Web application module contains valid descriptors.

Location: [localhost](#) \ [C:](#) \ [iWay55](#) \ bea

	ibse
	iwa
	iwjcaivp

5. To deploy servlet Application Explorer, select the option button next to the iwae directory and then click *Target Module*.

If you are using servlet Application Explorer, deploy it only on the admin server or one of the managed servers.

Deploy a Web Application Module

Select the archive for this Web application module

Select the file path that represents your archive or exploded archive directory.

Note: Only valid file paths are shown below. If you do not find what you are looking for, you should [upload your file\(s\)](#) and/or confirm your Web application module contains valid descriptors.

Location: [localhost](#) \ [C:](#) \ [Program Files](#) \ [iWay55](#) \ bea

<input type="radio"/>	ibse
<input checked="" type="radio"/>	iwae
<input type="radio"/>	iwjcaivp

Target Module

The following window opens.

Select targets for this Web application module

Select the servers and/or clusters on which you want to deploy your new Web Application module

Independent Servers

<input type="checkbox"/>	AdminServer
<input type="checkbox"/>	HTTPROUTER

Clusters

<input checked="" type="checkbox"/>	MYCluster
<input checked="" type="radio"/>	All servers in the cluster
<input type="radio"/>	Part of the cluster
<input type="checkbox"/>	MS1
<input type="checkbox"/>	MS2

6. Select the servers and/or clusters on which you want to deploy the application and click *Continue*.

The following window opens.

Source Accessibility

During runtime, a targeted server must be able to access this Web Application module's files. This access can be accomplished by either copying the Web Application module onto every server, or by defining a single location where the files exist.

How should the source files be made accessible?

- ☐ **Copy this Web Application module onto every target for me.**

During deployment, the files in this Web Application module will be copied automatically to each of the targeted locations.

- ☒ **I will make the Web Application module accessible from the following location:**

C:\Way55\bea\ibse

Provide the location from where all targets will access this Web Application module's files. You must ensure the Web Application module's files exist in this location and that each target can reach the location.

7. Select the *I will make the Web Application module accessible from the following location* option button and provide the location from which all targets will access iBSE.

iWay Software recommends that you use a single instance of iBSE, rather than copying iBSE onto every target.

Note: iBSE must use a database repository (SQL or Oracle). Do not use a file repository. You can select this in the Repository Type drop-down list in the iBSE monitoring page. After configuring a database repository, you must restart all of the managed servers.

<http://hostname:port/ibse/IBSEConfig/>

where:

[hostname](#)

Is where your application server is running. Use the IP address or machine name in the URL; do not use localhost.

[port](#)

Is the port specific to each server, since you deploy iBSE to an entire cluster. For example, 8001, 8002, or any other port that is specified for each managed node.

8. Click *Deploy*.

Procedure: How to Configure Ports and Channels in a Clustered Environment

You can use Swing Application Explorer deployed in BEA WebLogic WorkShop or Servlet Application Explorer to configure ports and channels in a clustered environment.

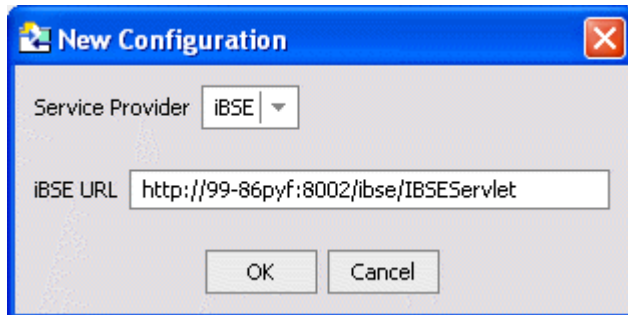
Note: Before using Servlet Application Explorer in a clustered environment, you must edit the web.xml file and specify the correct URL to your iBSE deployment. The default location on Windows is:

`C:\Program Files\iWay55\bea\iwaee\WEB-INF\web.xml`

For more information on configuring the web.xml file for Servlet Application Explorer, see the *iWay Installation and Configuration for BEA WebLogic* documentation.

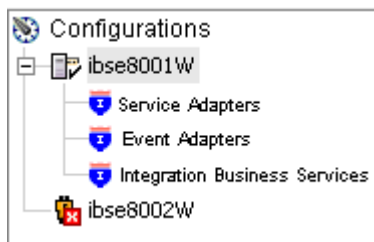
To configure ports and channels in a clustered environment:

1. Open Swing Application Explorer in BEA WebLogic Workshop.
2. Create a new connection to the iBSE instance. For information on creating a new configuration, see *Appendix B, Using Application Explorer in BEA WebLogic Workshop to Create XML Schemas and Web Services with ClarifyCRM ClearBasic*.



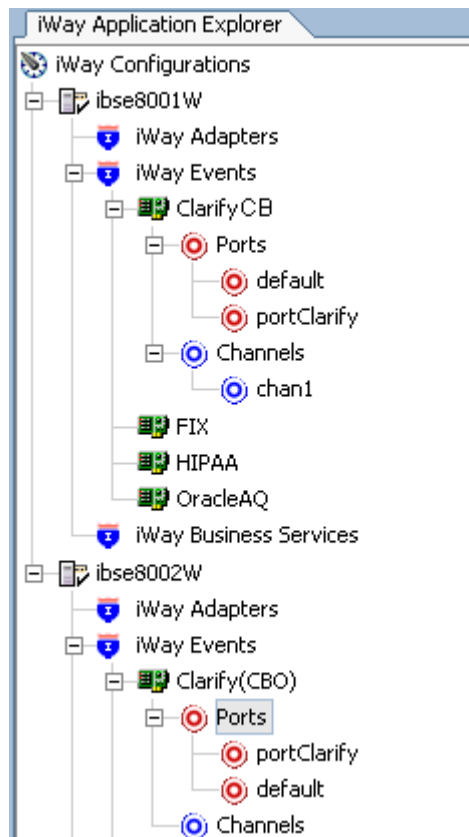
Note: Use the IP address or machine name in the URL; do not use localhost.

3. Connect to the new configuration and select the iWay Events node in the left pane of Application Explorer.



4. Add a new port for the ClarifyCRM adapter. For more information, see *Creating an Event Port* on page D-3.
5. Create a channel and add the port you created. For more information, see *Creating a Channel* on page D-24.
6. Click *Next* and enter the application server parameters.
7. Start the channel.
8. Create a new configuration and connect to the second iBSE instance.

The connection to iBSE must be configured to each instance of the managed server. The following image shows two configurations.



The following operations performed on one managed server will be replicated on all other managed servers:

- Create port and channel: Creates the channel and port under all available servers.

- Delete port and channel. Deletes the port and channel under all available servers.

The following operations must be performed on each server:

- Start channel. Starts the channel for the specific server.
- Stop channel. Stops the channel for the specific server.

Reader Comments

In an ongoing effort to produce effective documentation, the Documentation Services staff at Information Builders welcomes any opinion you can offer regarding this manual.

Please use this form to relay suggestions for improving this publication or to alert us to corrections. Identify specific pages where applicable. You can contact us through the following methods:

Mail: Documentation Services - Customer Support
Information Builders, Inc.
Two Penn Plaza
New York, NY 10121-2898

Fax: (212) 967-0460

E-mail: books_info@ibi.com

Web form: <http://www.informationbuilders.com/bookstore/derf.html>

Name: _____

Company: _____

Address: _____

Telephone: _____ Date: _____

E-mail: _____

Comments:

Reader Comments