iWay

iWay Application System Adapter for Amdocs ClarifyCRM User's Guide
Version 5 Release 5

Updated for J2EE CA 1.5

DN3501482.0305

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 Application System Adapter for Amdocs ClarifyCRM to develop online interconnections to ClarifyCRM.

How This Manual Is Organized

The following table lists and describes the chapters in this manual.

Chapter/Appendix		Contents
1	Introducing the iWay Application Systems Adapter for Amdocs ClarifyCRM	This section introduces the iWay Application 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	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.
4	Using Web Services Policy-Based Security	Describes how Web services policy-based security works and how to configure it.
5	Management and Monitoring	Describes how you can use managing and monitoring tools provided by the iWay Business Services Engine (iBSE) and the iWay Connector for JCA to gauge the performance of your run-time environment.

Documentation Conventions

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

Convention	Description
`THIS TYPEFACE` or `this typeface`	Denotes syntax that you must enter exactly as shown.
`this typeface`	Represents a placeholder (or variable) in syntax for a value that you or the system must supply.
`underscore`	Indicates a default setting.
this typeface	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 Application 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 Application 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

    ibserepo.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

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 Application System Adapter for Amdocs ClarifyCRM

The iWay Application 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 Application 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 Application 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 Application 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 Application 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 Application 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:

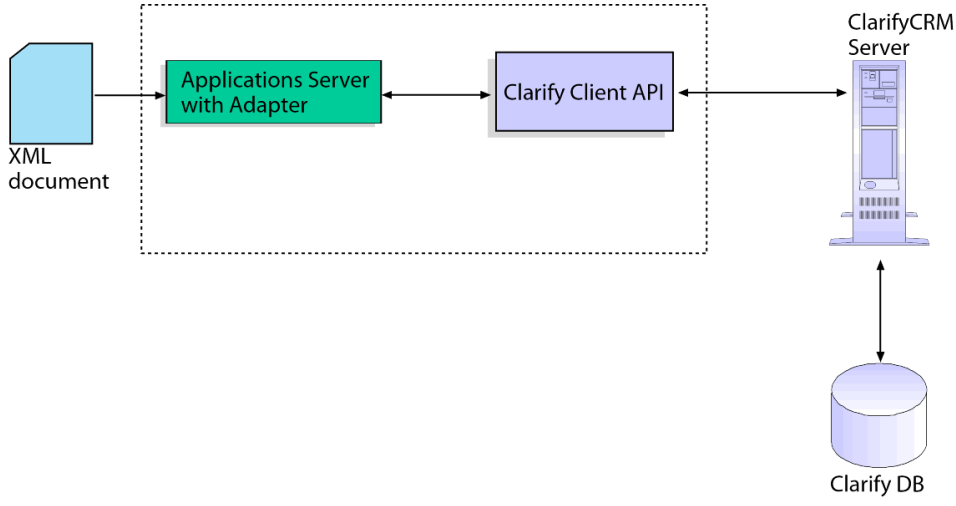
- Applying triggers to specified tables within the ClarifyCRM application.
- Polling the ClarifyCRM events database at user-configured time intervals.
- Translating the event information to XML.

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.

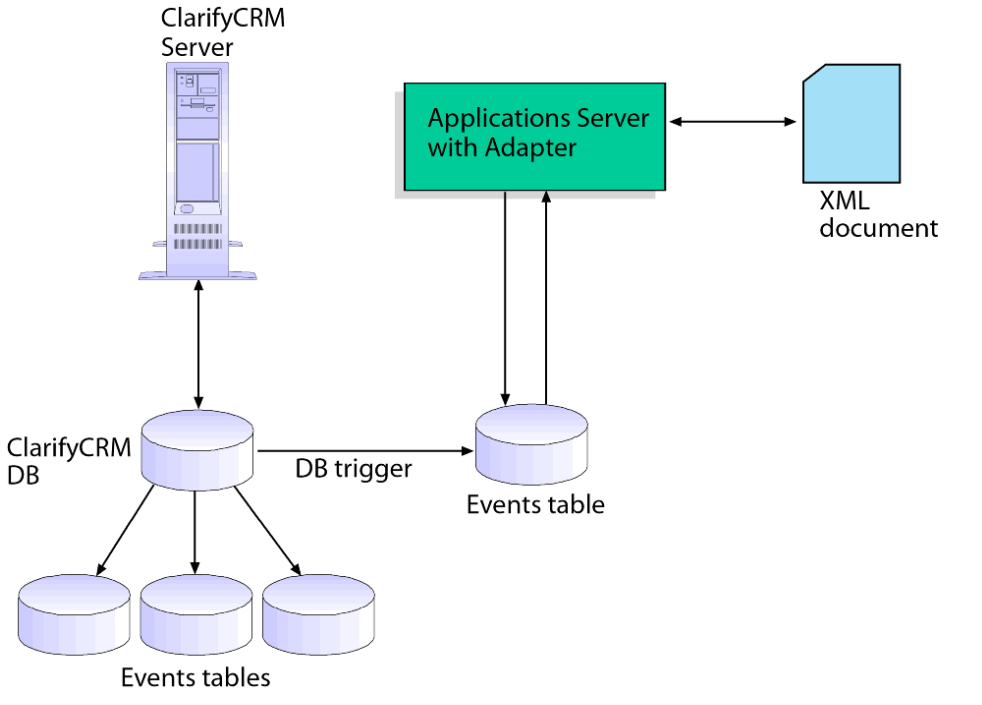


When the iWay Application System Adapter for Amdocs ClarifyCRM processes a service:

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 Application 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 Application System Adapter for Amdocs ClarifyCRM obtains the data and packages it in XML format.

Deployment Options for the iWay Application System Adapter for Amdocs ClarifyCRM

The iWay Application 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 Application 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 of this guide *iWay Installation and Configuration*
iWay Business Services Engine (iBSE)	*iWay Installation and Configuration*
iWay Enterprise Connector for J2EE Connector Architecture (JCA)	*iWay Connector for JCA User's Guide* *iWay Installation and Configuration*

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.

Two connectors are distributed in the iWay installation package. One conforms to the JCA 1.0 specification, with extensions that allow for the consumption of events. The other conforms to the JCA 1.5 specification. The JCA 1.0 connector provides for event functionality through the configuration of ports and channels. When using the adapter in conjunction with a JCA 1.5 connector, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities. For more information on event capabilities of the iWay JCA connectors, see Chapter 3, *Listening for Events Using ClarifyCRM Business Objects*.

For more information on installing and deploying both connectors, see *iWay Installation and Configuration*.

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).

Generating Schemas and Business Services Overview

The iWay Application 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 2-4.
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 2-10.

   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 2-15
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 2-31.

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 the Sun Java System Application Server.

Procedure: How to Start Application Explorer

To start Application Explorer:

1. Ensure the Sun Java System Application Server is running.
2. Enter the following URL in your browser window

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

   where:

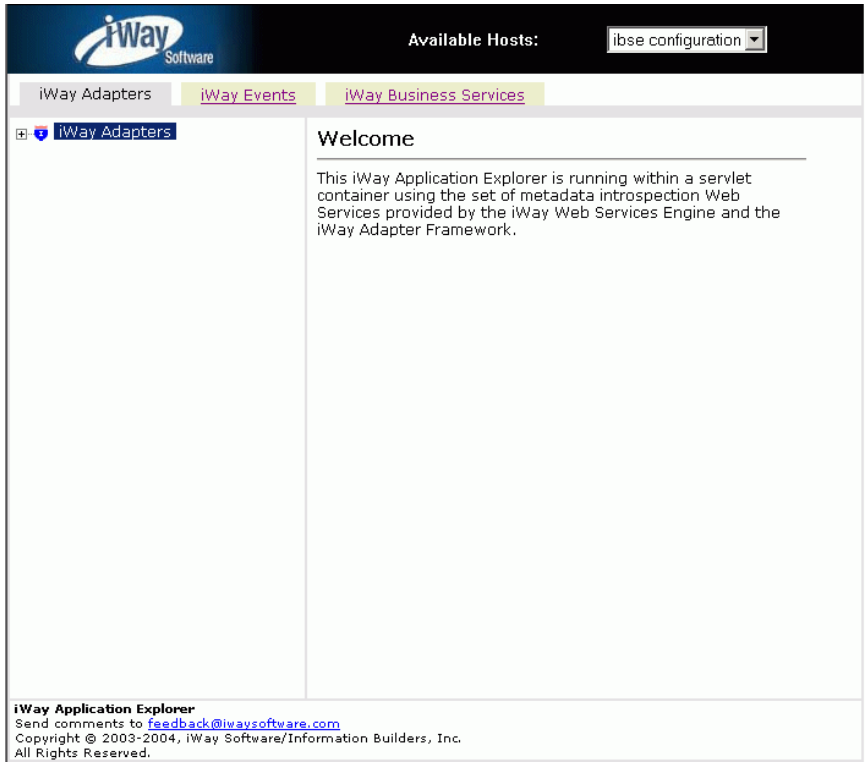
   hostname

   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 iWay Adapters node in the left pane, and a welcome message in the right pane.



The three tabs are:

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

iWay Events, 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 *iWay 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

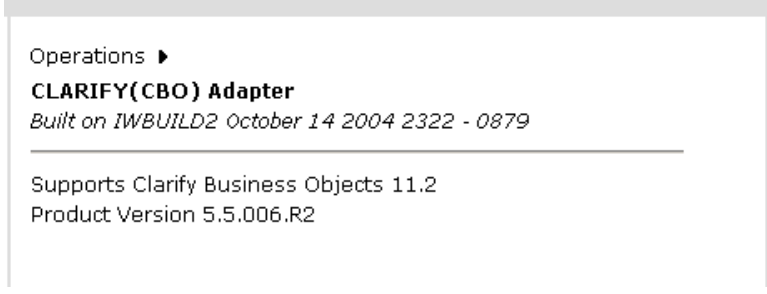
A target serves as the connection point to your Enterprise Information System (EIS) and is automatically saved after you create it. To connect to ClarifyCRM for the first time, you must create a new target from the iWay Adapters tab.

Procedure: How to Create a New CBO Target

To create a new CBO target:

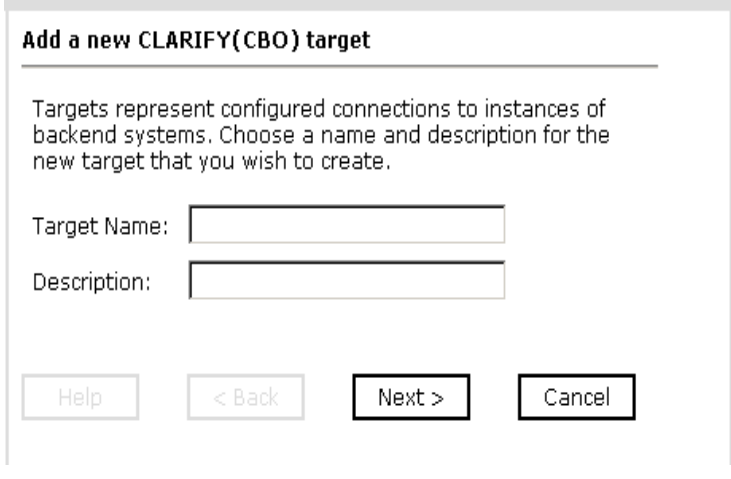
1. In the left pane, expand the *iWay Adapters* node and select the *Clarify(CBO)* node.

   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.

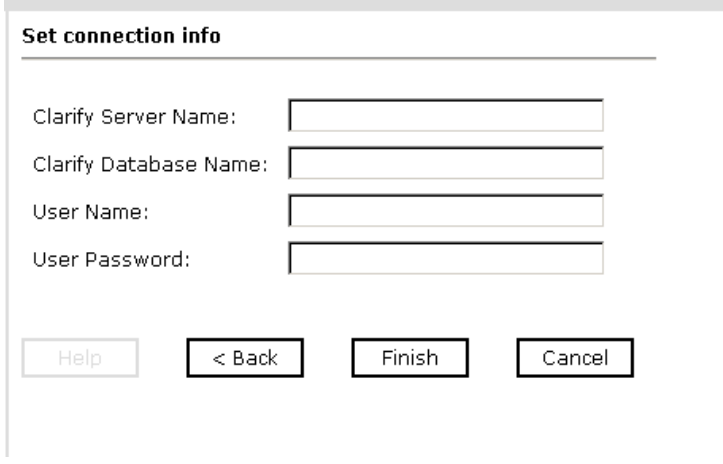


   a. In the Target Name field, type a descriptive name for the target, for example, ClarifyNew.

   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.



   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

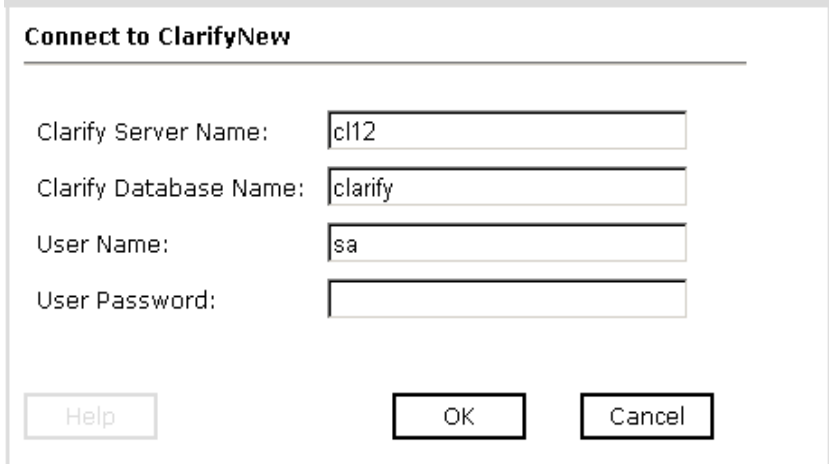
You can use the already defined targets to connect to instances of ClarifyCRM.

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.



   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

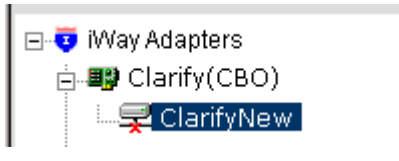
Although you can maintain multiple open connections to different application systems, we recommend that you close connections when they are not in use.

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 2-8.
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.

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 2-31.

You can create XML schemas for Clarify(CBO):

- Services. For instructions, see *How to Create an XML Schema for a Service* on page 2-10.
- Events. For instructions, see *How to Create an XML Schema for an Event* on page 2-13.

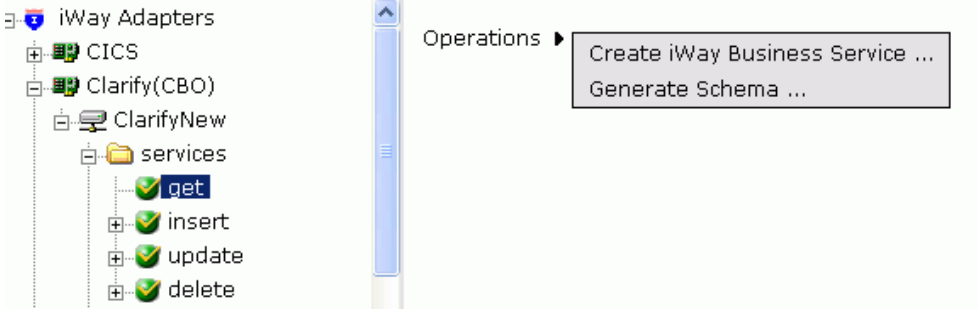
After you create the schema, you can view it or save it and use the schema to create request and response documents. For information about where the schemas are stored, see *Schema Location* on page 2-14.

Procedure: How to Create an XML Schema for a Service

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

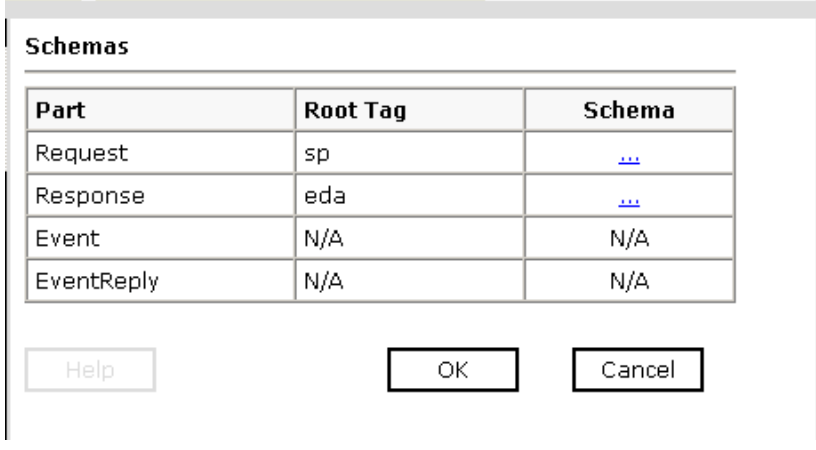
1. Connect to a Clarify(CBO) target, as described in *Opening a Connection to ClarifyCRM* on page 2-4.
2. In the left pane of Application Explorer, expand the target node and then the *services* node.
3. Select the module containing the table for which you want to create schemas.

   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.



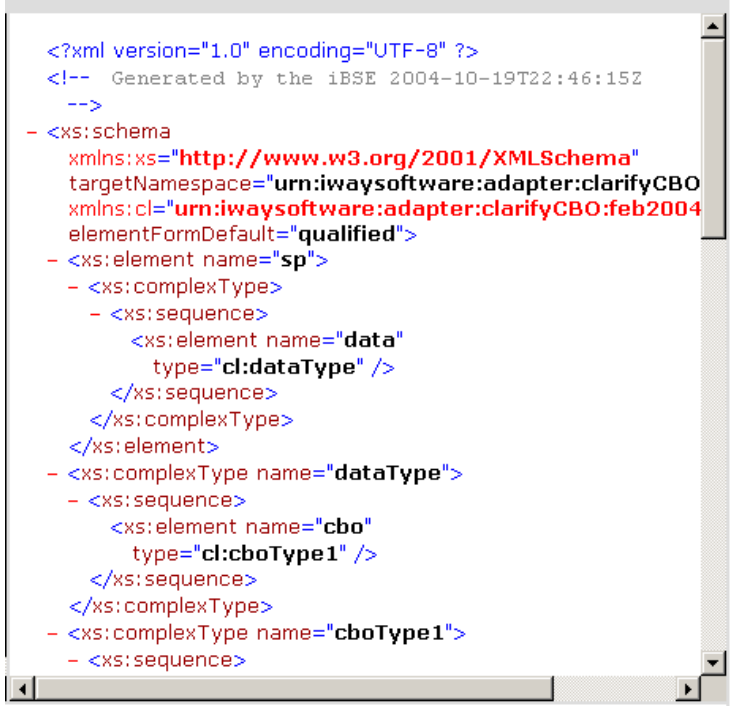
Schemas

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 2-14 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.

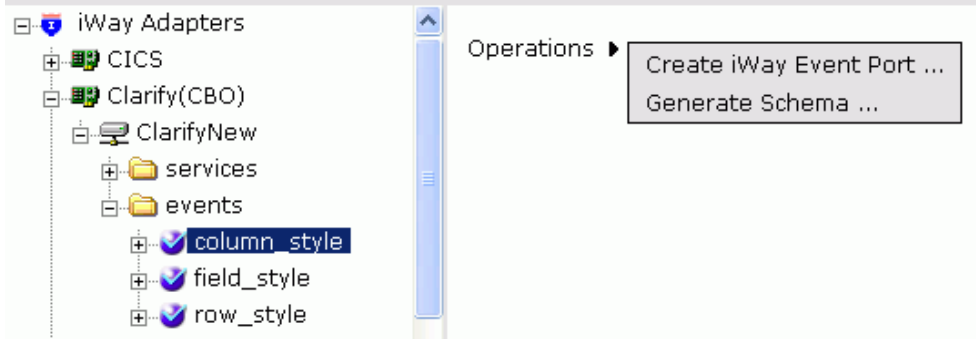
8. To return to the Operations menu from the Schemas table, click *OK*.

Procedure: How to Create an XML Schema for an Event

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

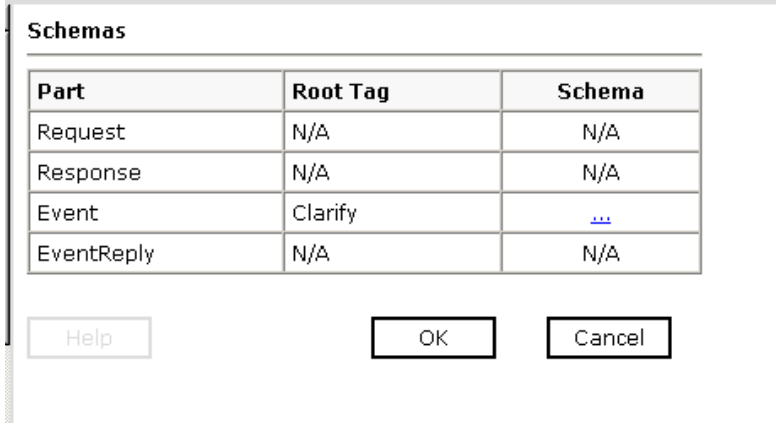
1. Connect to a Clarify(CBO) target, as described in *Opening a Connection to ClarifyCRM* on page 2-4.
2. In the left pane of Application Explorer, expand the target node, and then the *events* node.
3. Select the events module containing the table for which you want to create schemas.

   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 Schema 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 2-14 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

When the adapter is deployed to the Sun Java System Application Server and used with an **iBSE configuration,** Application Explorer stores the schemas in a subdirectory of the Sun Java System Application Server installation directory, for example,

```
D:\Sun\AppServer\domains\domain1\applications\j2ee-apps
\ibse\ibse_war\wsdl\schemas\service\Clarify\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 the adapter is used with a **JCA configuration,** Application Explorer stores the schemas 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: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: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>
```

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: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" minOccurs="0"
         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" />
    </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* 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* manual.

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 2-4.
2. In the left pane of Application Explorer, expand the target node to display its modules.

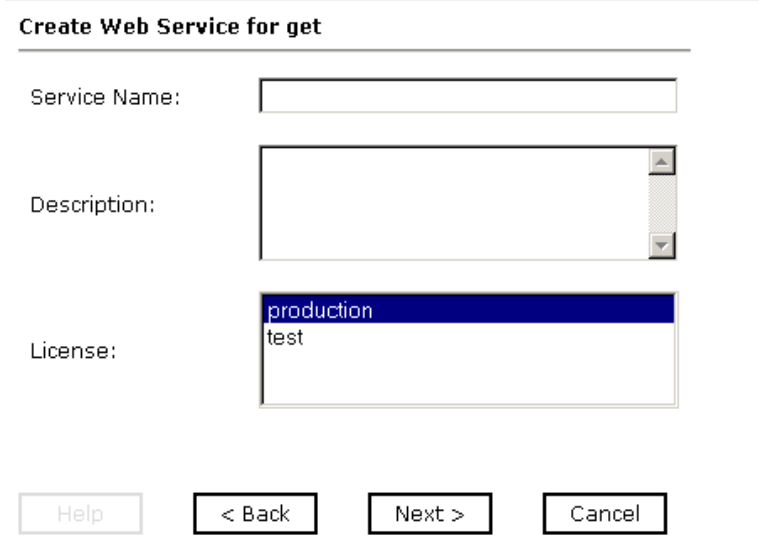
   The list includes all modules currently supported by ClarifyCRM.
3. Open the module containing the table for which you want to create a business service.

4. In the right pane, move the pointer over *Operations* and select *Create iWay Business Service*.

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.

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.



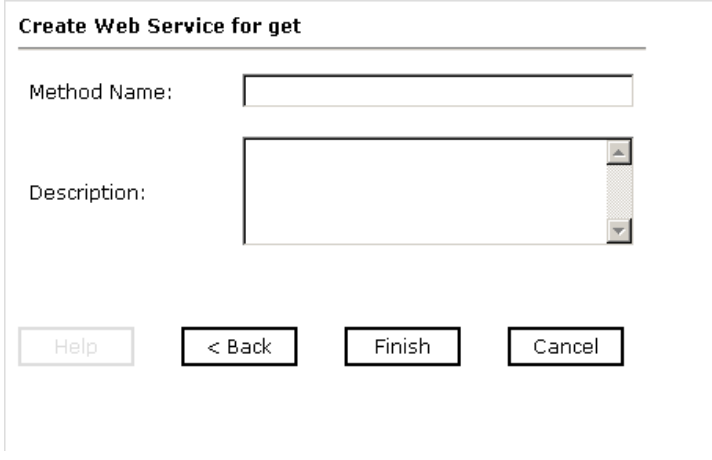
a. In the Service Name field, type a descriptive name for the iWay Business Service.

b. In the Description field, type a brief description of the iWay Business Service.

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.



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 and 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.

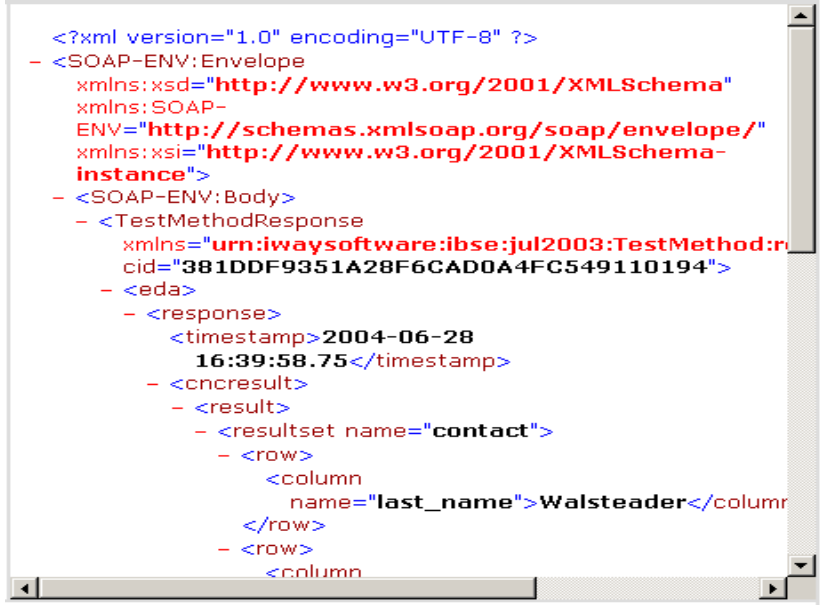
   If you want to test an existing business service, click the *iWay Business Services* tab, expand the list of business services in the left pane, and select the business service you want to test. The test options appear in the right pane.

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 2-36. and *iWay Business Services Input XML for Insert* on page 2-37.

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>
    </cbo>
  </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>
     </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:

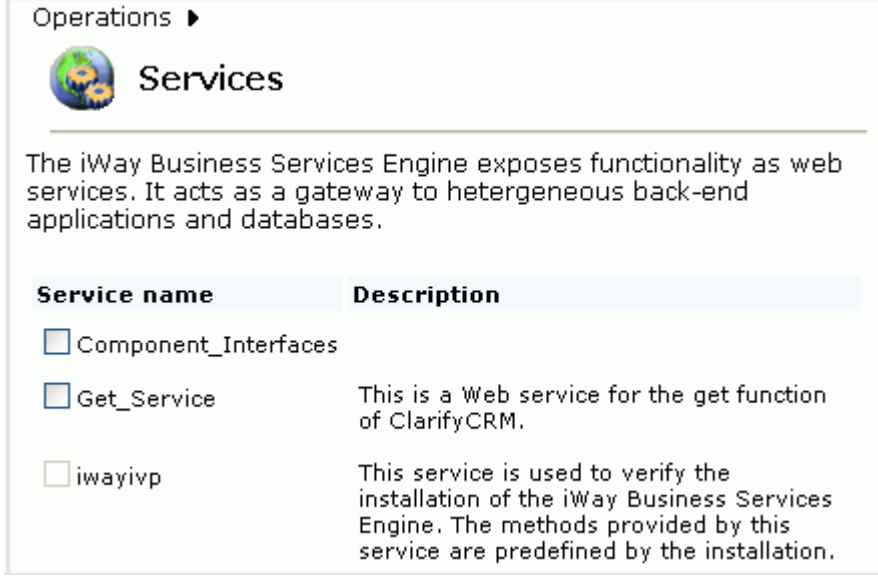
1. Click the *iWay Business Services* tab of Application Explorer.

   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

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 Application System Adapter for Amdocs ClarifyCRM deployed to an application server, such as a Sun Java System Application 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.

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.

Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

The following is a description of how ports and channels work:

- **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 Sun Java System Application Server,a J2EE application server. For more information, see *Creating an Event Port* on page 3-3.

- **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 3-18.

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 iWay Adapters tab or the iWay Events 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 iWay Adapters Tab* on page 3-4
- *Create an Event Port From the iWay Events Tab* on page 3-6

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.

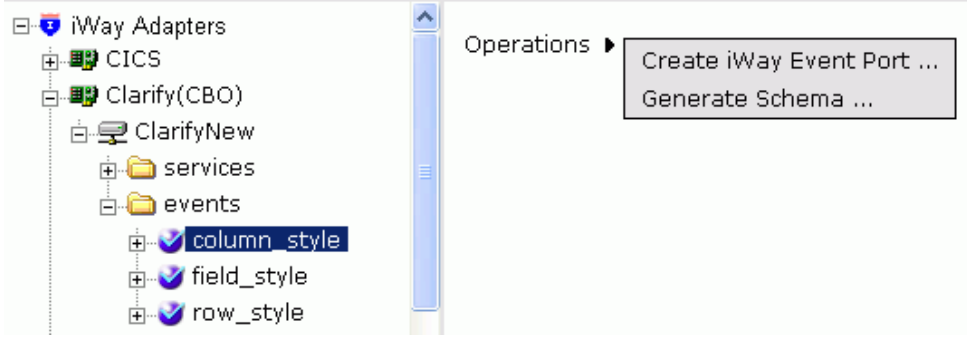
Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

Procedure: How to Create an Event Port From the iWay Adapters Tab

To create an event port from the iWay Adapters tab:

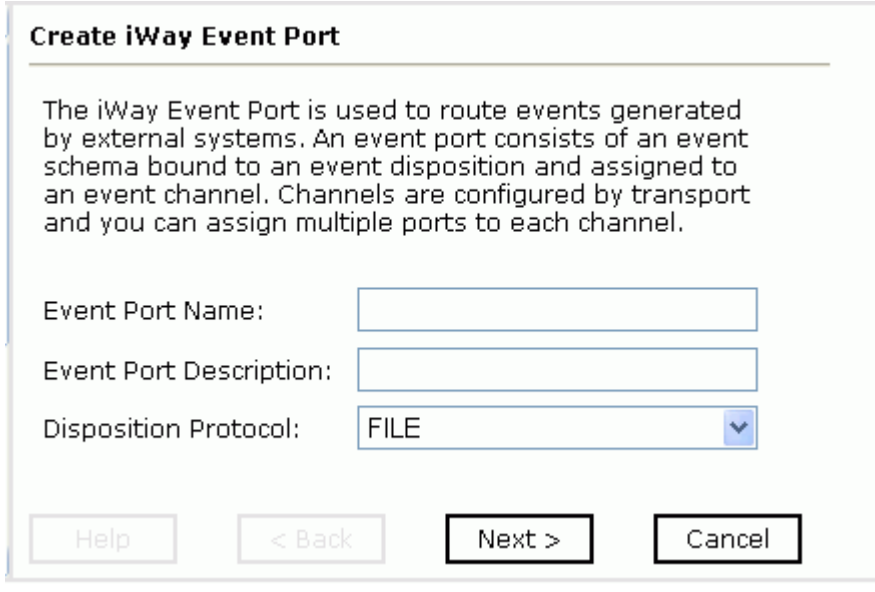
1. Click the *iWay Adapters* tab.
2. In the left pane, expand the *iWay Adapters* node, the *Clarify(CBO)* node, and the target node for the event.
3. Expand the *events* node and select an event component.

   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 iWay Event Port*.

The Create iWay 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.



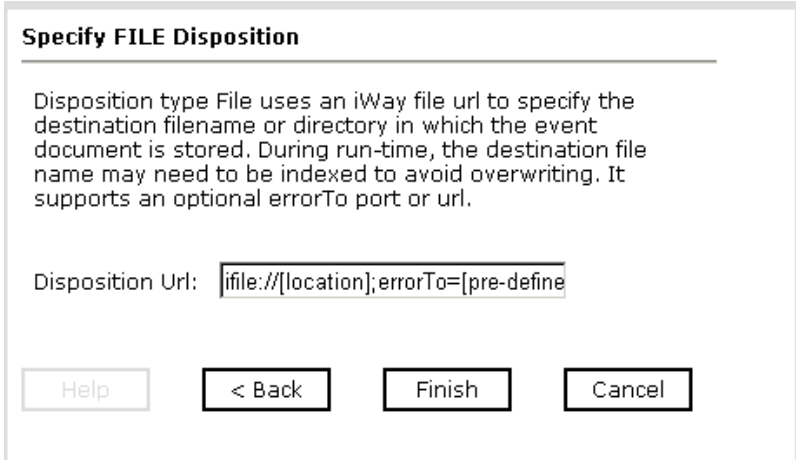
a. In the Event Port Name field, type a name for the event port.

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.



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 3-8.

7. Click *Finish*.

   The iWay Events 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 3-8.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-18.

Procedure: How to Create an Event Port From the iWay Events Tab

To create an event port from the iWay Events tab:

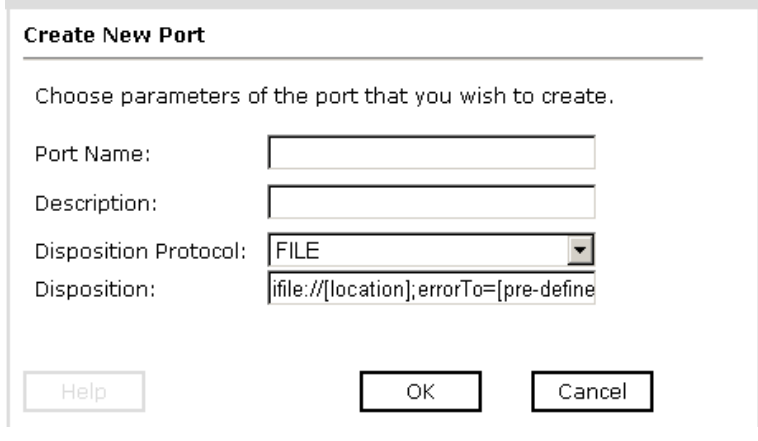
1. Click the iWay Events tab.
2. In the left pane, expand the iWay Event Adapters node and the *Clarify(CBO)* node.

3. Select the *ports* node.

   The following image shows an expanded iWay Event Adapters 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 *Add a new port*.

   The Create New 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 3-8.

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 3-8.

You are ready to associate the event port with a channel. For more information, see *Creating a Channel* on page 3-18.

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
  ```

  Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

The following table lists and 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 on the right. The following image shows a listing and description for a FILE event port.

iBSE Disposition URL Format

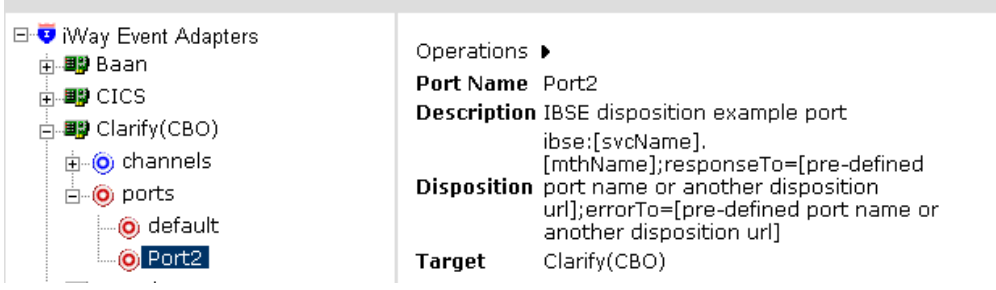
The format of the URL to the iBSE disposition event document is:

`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

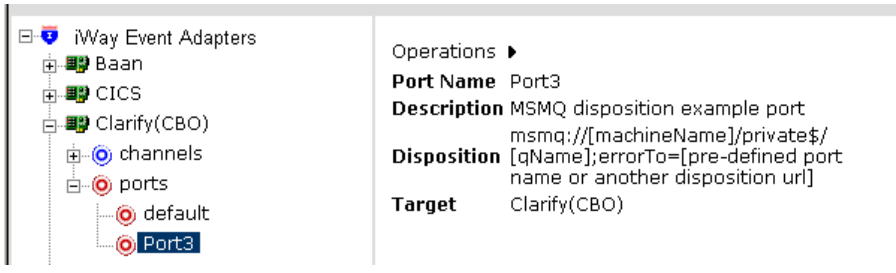
The format of the URL to the MSMQ disposition event document is:

`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 *Private$*. 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
  ```

  Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

The following table lists and describes the JMS disposition parameters.

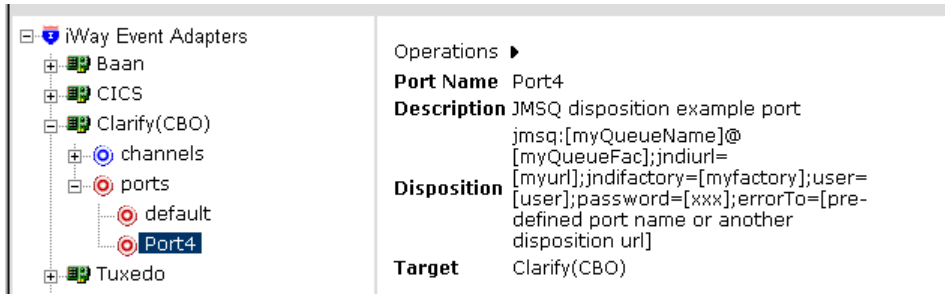
Parameter	Description
queue	Name of a queue to which events are emitted.
conn_factory	The connection factory, a resource that contains information about the JMS Server. You must create the connection factory, for example: `sampleQCF`

Parameter	Description

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:
	`java.naming.provider.url`
	The URL of the Sun Java System Application Server is:
	`iiop://localhost:3700 (3700 is a default port)`

jndi_factory	Is JNDI context.INITIAL_CONTEXT_FACTORY and is provided by the JNDI service provider. For Sun Java System Application Server, this is: `com.sun.jndi.cosnaming.CNCtxFactory.`
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 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]
```

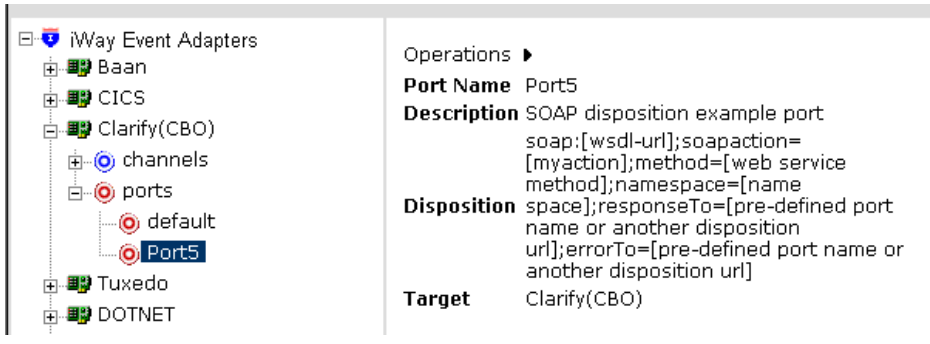
The following table lists and describes the SOAP disposition parameters.

Parameter	Description

wsdl-url	The URL to the WSDL file that is required to create the SOAP message. For example,
	`http://localhost:7001/ibse/IBSEServlet/test/webservice.ibs?wsdl`
	where:
	`webservice`
	Is the name of the Web service you created using Application Explorer.
	To find this value, navigate to the iWay Business Services tab, expand the *Services* node, select the service you created, and click the *Service Description* option on the right. The WSDL URL appears in the Address field of the window that opens.
	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.

soapaction	The method that will be called by the SOAP disposition.
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 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*

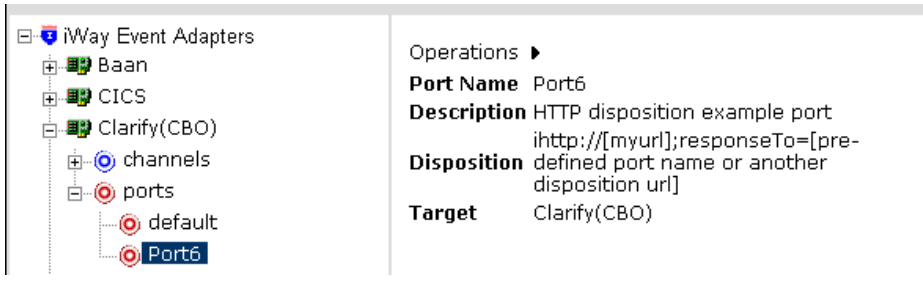
  Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

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.
port	Port number on which the Web server is listening.

Parameter	Description
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=
  ```

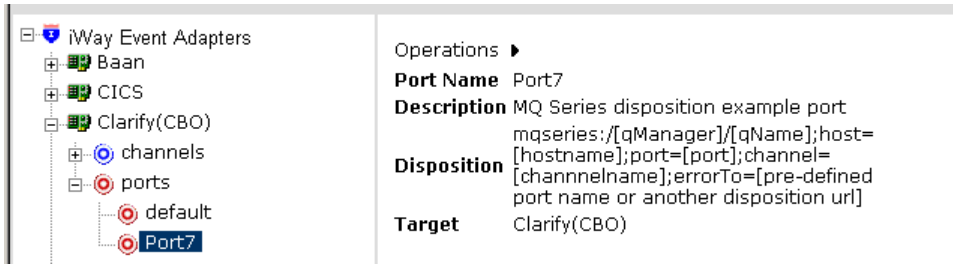
  Important: When using the adapter in conjunction with the iWay Connector for JCA 1.5, there is no need to create event ports to dispose of event data. However, you must create a channel to enable event listening capabilities.

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).

Parameter	Description
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 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.

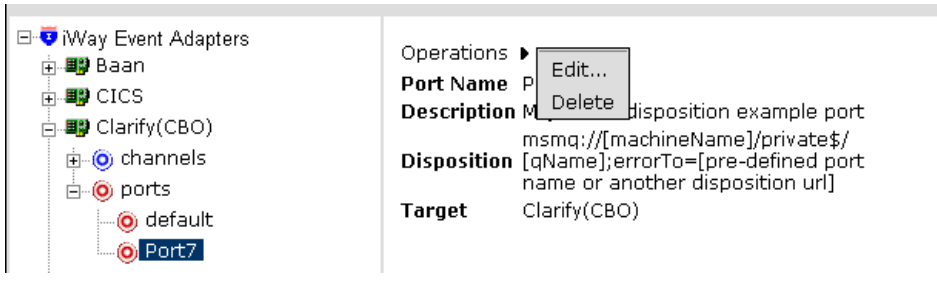
Procedure: How to Edit an Event Port

To edit an existing event port using Application Explorer:

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

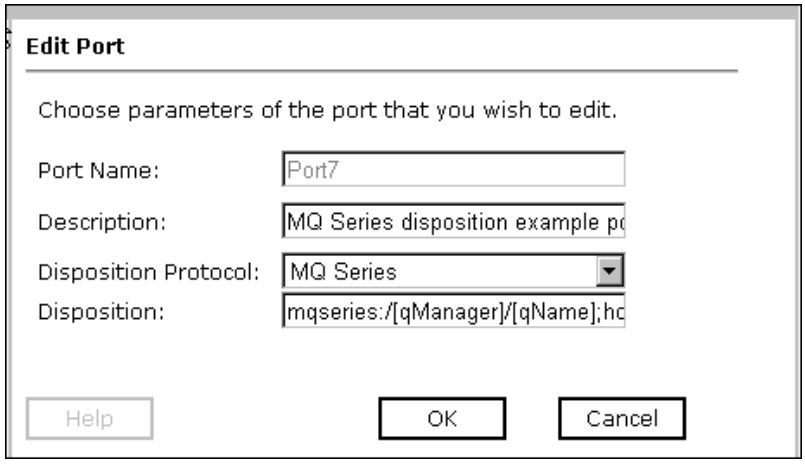
3. Select the event port you want to edit.

   The following image shows an expanded iWay 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.

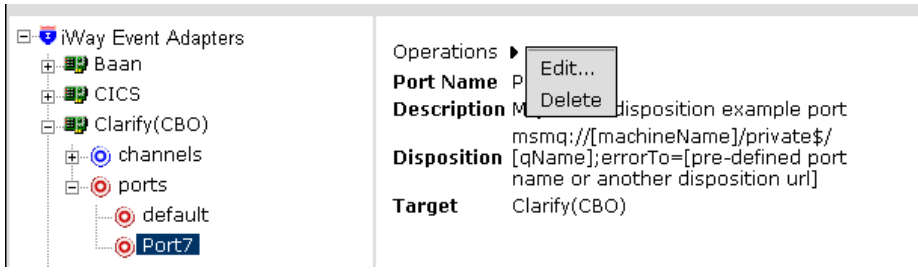
5. Make the required changes to the Description, Disposition Protocol, or Disposition fields, and click *OK*.

Procedure: How to Delete an Event Port

To delete an existing event port using Application Explorer:

1. Click the *iWay Events* tab.
2. In the left pane, expand the *iWay 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 iWay 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

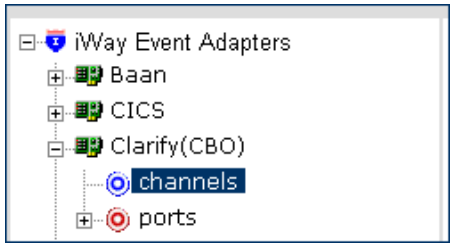
The following procedure describes how to create a channel for a ClarifyCRM event. All defined event ports must be associated with a channel.

Procedure: How to Create a Channel

To create a channel using Application Explorer:

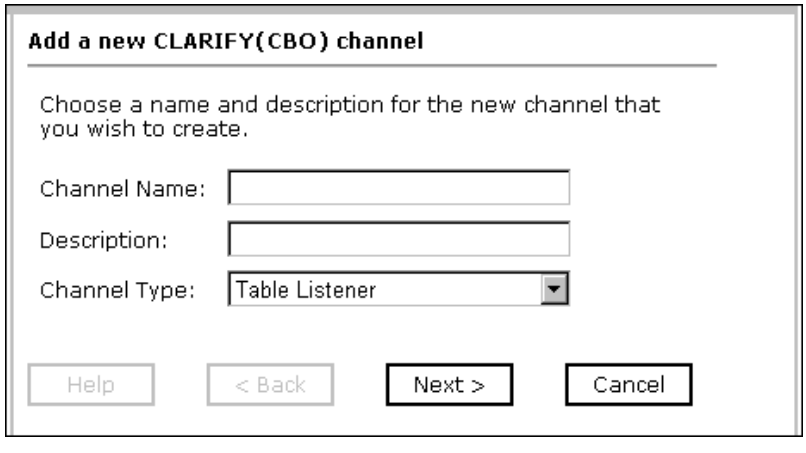
1. Click the *iWay Events* tab.
2. In the left pane, expand the *iWay Event Adapters* node and the *Clarify(CBO)* node.

   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.

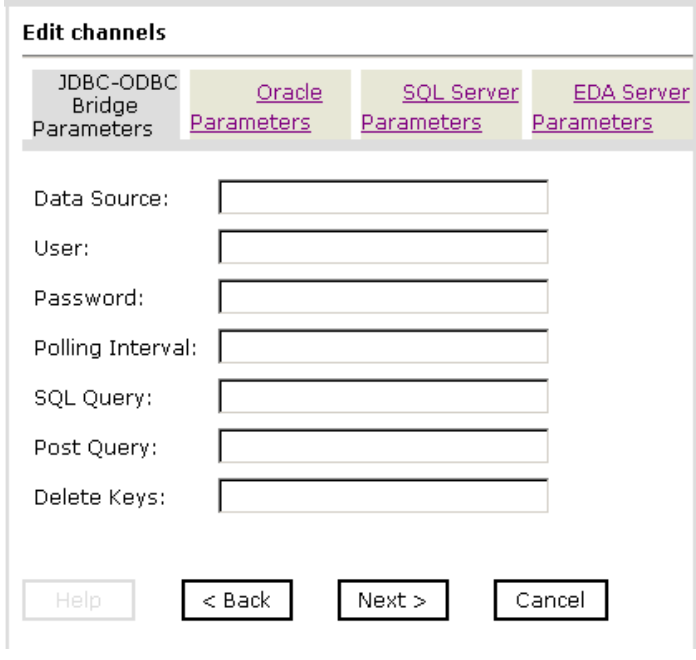


   a. In the Channel Name field, enter a name for the channel.
   b. In the Description field, enter a brief description of the channel.
   c. From the Channel Type drop-down list, accept the default channel type, *Table Listener*.

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).



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.

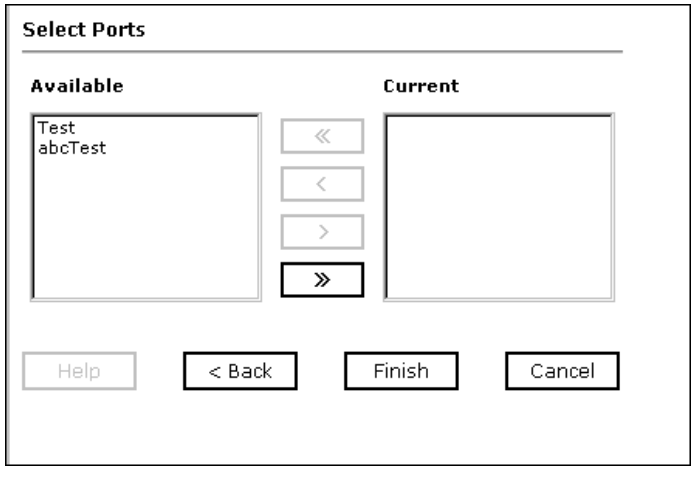
Field	Description
SID	A unique name of the database service, chosen by the database administrator or the person who installed ClarifyCRM.
Database Name	Name of the database.
User	The ClarifyCRM database user ID to access the ClarifyCRM database underlying the ClarifyCRM system. The user ID must have 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: `select * from iw_events`

Field	Description
Post Query*	One or more SQL statements that are executed after each new record is read from the table. Separate multiple statements with a semicolon (;). 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. When you specify a value for this parameter, the table data is retained after it is read. 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: • **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. * 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. There are two field operators, ? and ^, that you can use in a post-query SQL statement; for more information, see *Post Query Parameter Operators* on page 3-24.

Field	Description
Delete Keys*	Comma-separated list of key columns used in the default DELETE statement. DELETE operates on keys, therefore specify the table key columns. 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. * 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.

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.



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 *iWay Events* tab.
2. In the left pane, expand the *iWay 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 *iWay Events* tab.
2. In the left pane, expand the *iWay 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 *iWay Events* tab.
2. In the left pane, expand the *iWay 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 4

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.

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 *How to Create a User to Associate With a Policy* on page 4-3 or *How to Create a Group to Associate With a Policy* on page 4-5.

An execution policy governs who can execute the business service to which the policy is applied. For more information, see *How to Create an Execution Policy* on page 4-8.

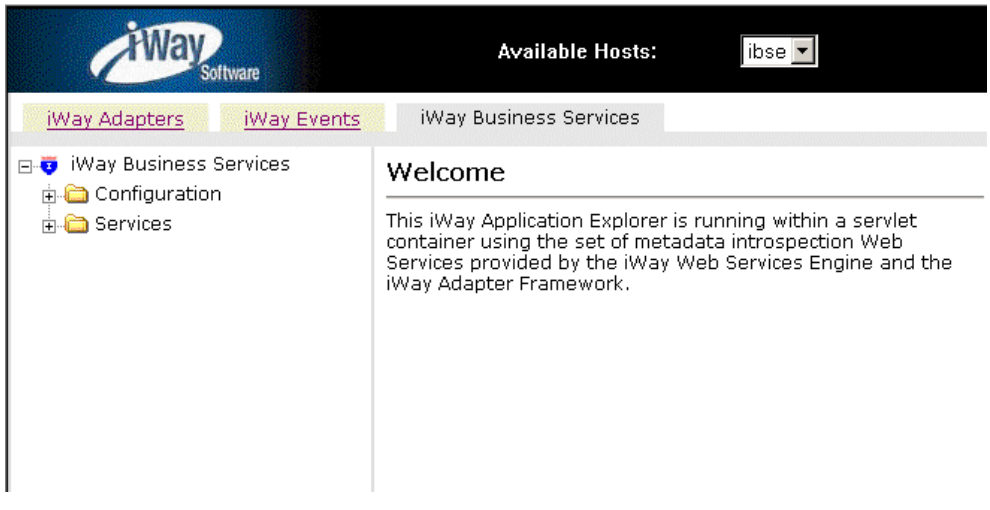
You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to iBSE and therefore, need not be applied to an individual business service. You need not create a policy, however, you must enable the Security Policy option in Servlet Application Explorer. For more information, see *How to Configure IP and Domain Restrictions* on page 4-12.

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 iWay Adapters, iWay Events, 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.

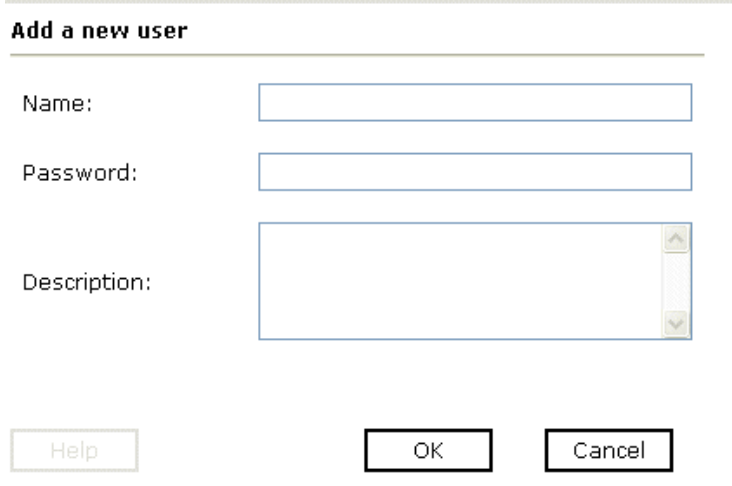
c. Expand the *Security* node.

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.



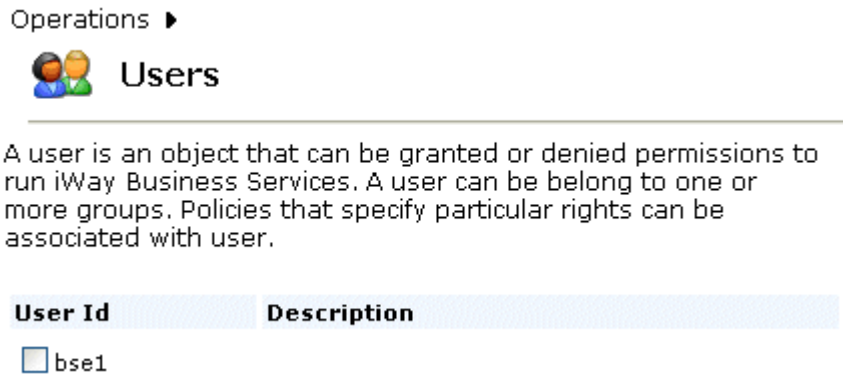
a. In the Name field, type a user ID.

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.



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*.

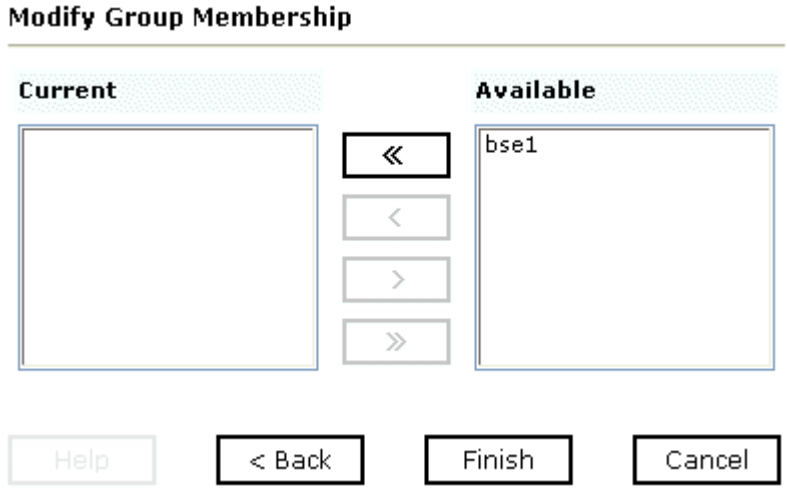
The following image shows the Add new group pane that opens with fields where you enter a name and a description for the group. To continue after typing inputs, click the *Next* button. The pane also includes a Help button, a Back button to return to the previous screen, and a Cancel button to escape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

a. In the Name field, type a 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.

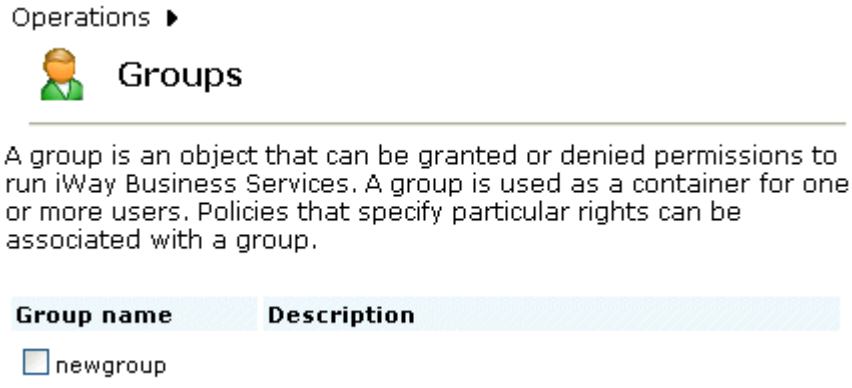


You can either highlight a single user in the list of available users and add it to the current list by clicking the left arrow, or you can click the double left arrow to add all users in the list of available users to the group.

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.



Procedure: How to Create an Execution Policy

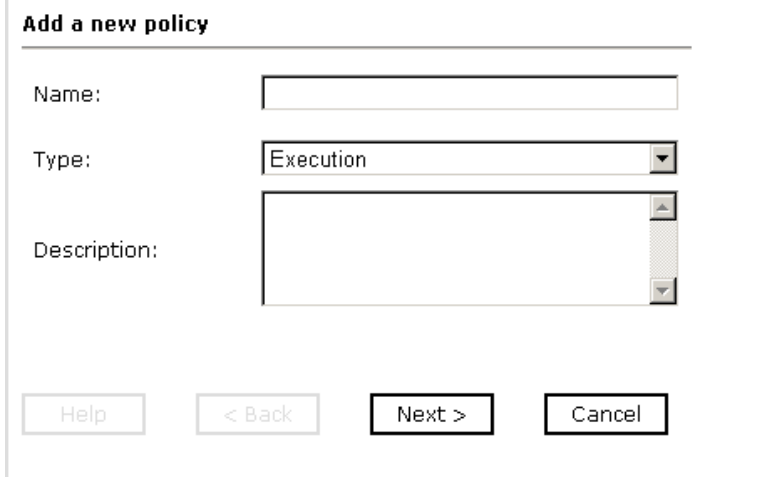
To create an execution policy:

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.



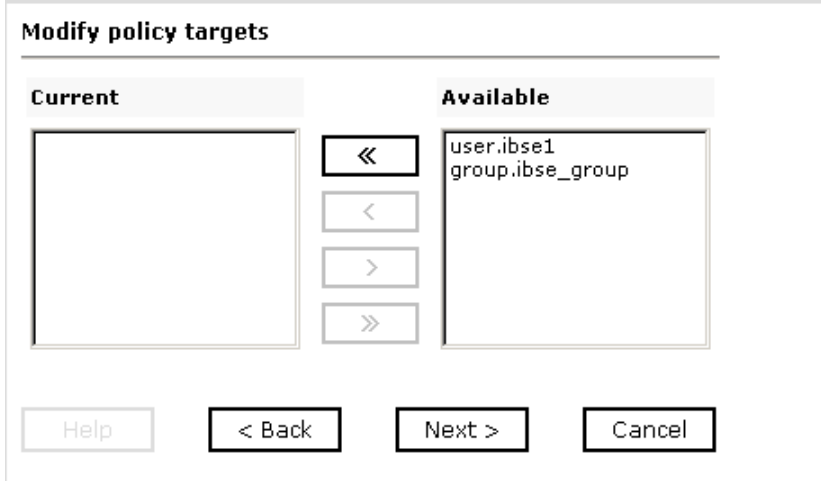
a. In the Name field, type a a name for the policy.

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.

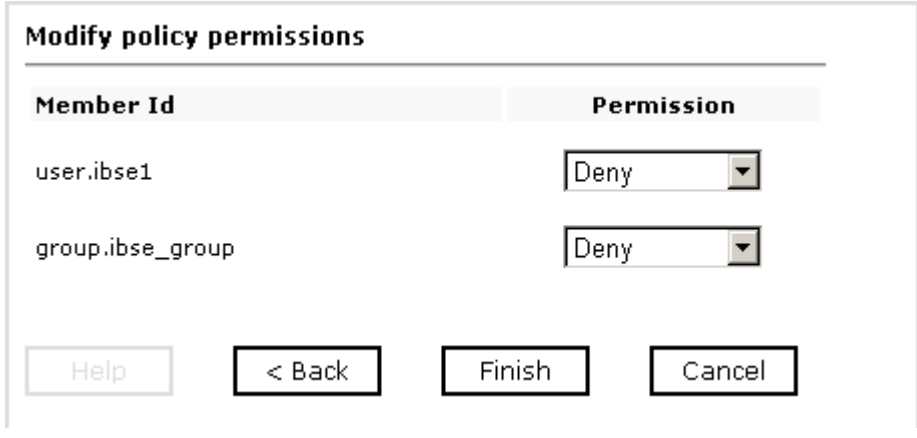


4. Select a minimum of one user or group from the Available pane.

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*.

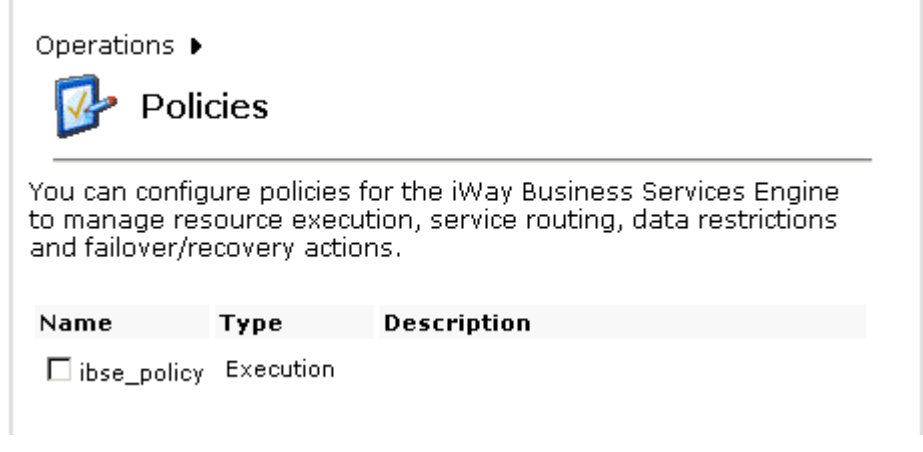
   The following image shows the Modify policy permissions pane that opens and includes drop-down lists where you can select to grant or deny permission to members and then click a button to finish. The pane also includes a Help button, a Back button to return to the previous screen, and a Cancel button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



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.

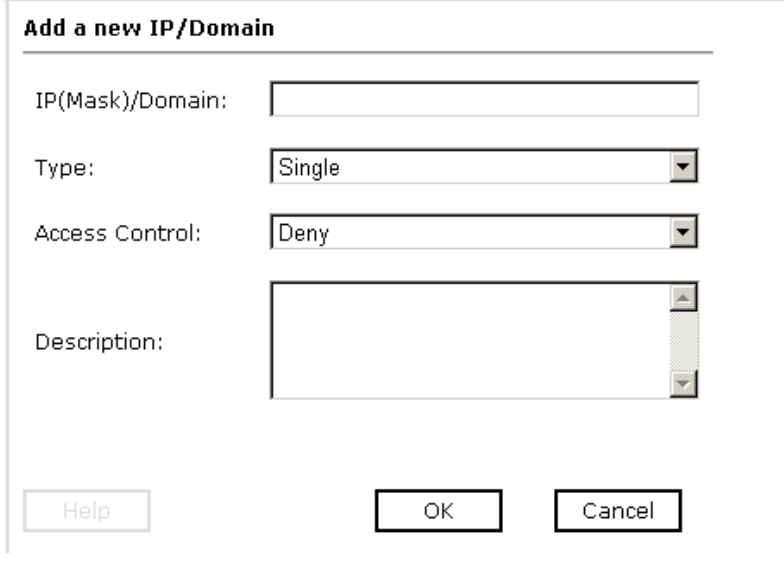


Procedure: How to Configure IP and Domain Restrictions

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.



   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).

CHAPTER 5

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 iWay 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

iWay 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 application 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.

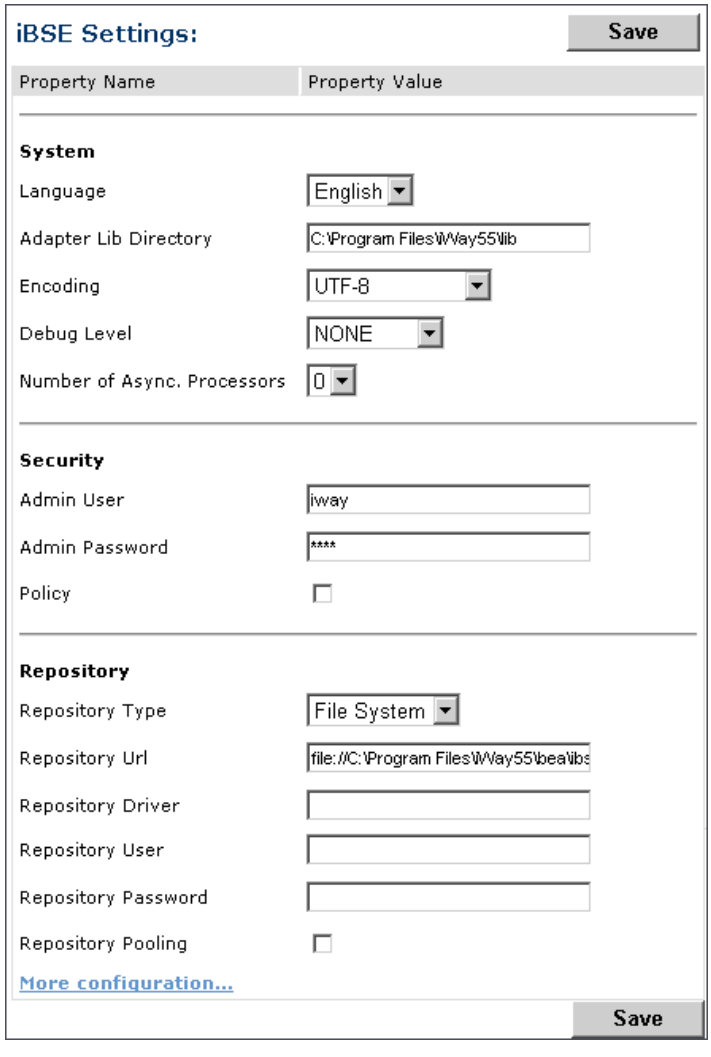
   port

   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.



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:

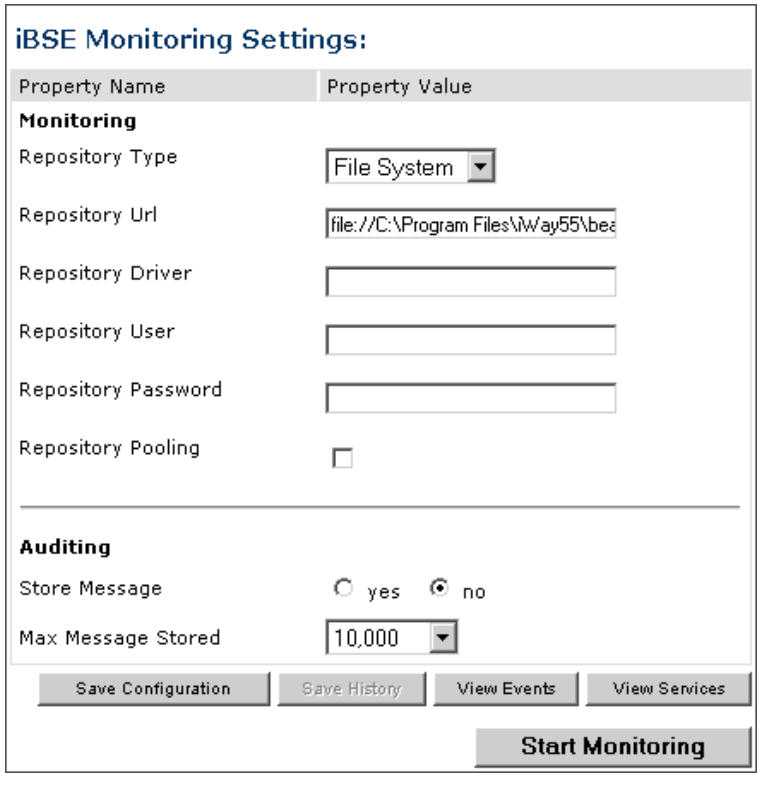
localhost

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.



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

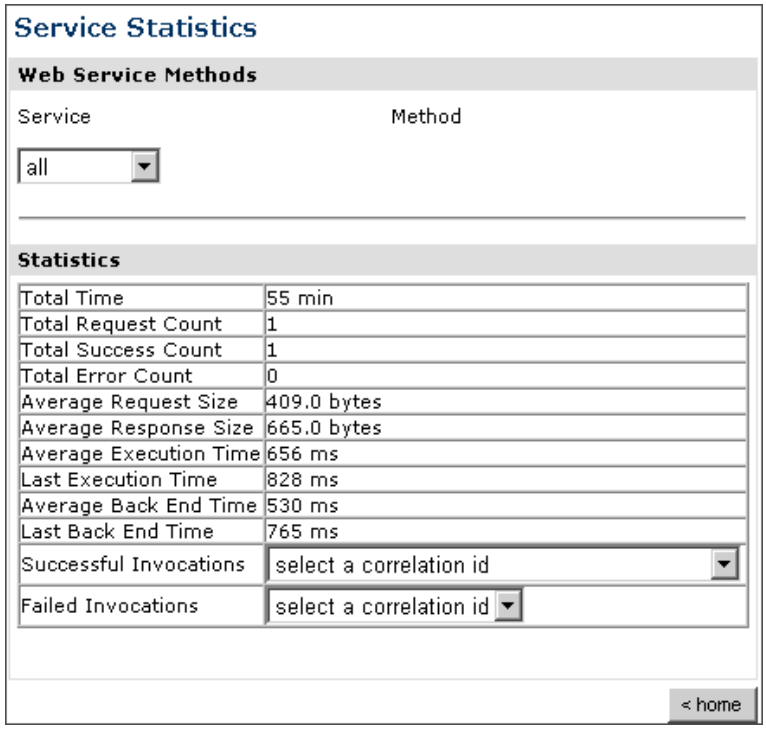
To monitor services:

1. Ensure that your application 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.



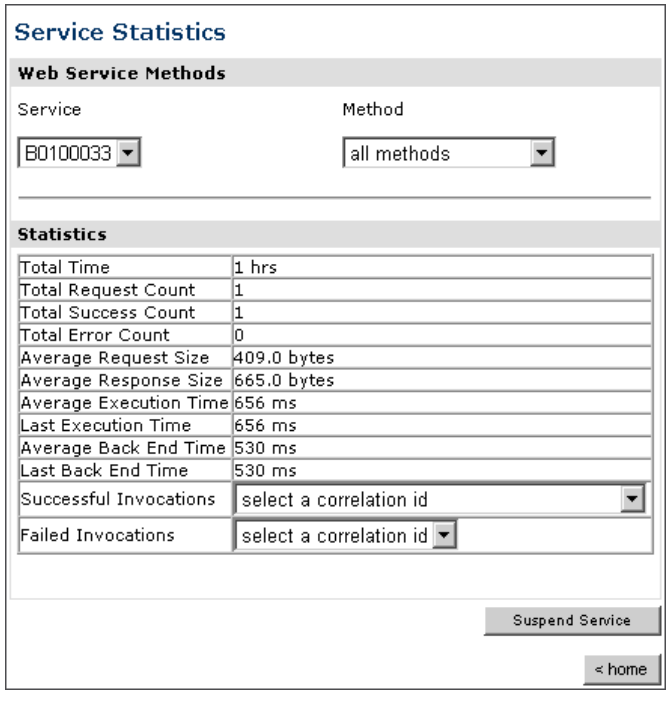
The system level summary provides services statistics at a system level.

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).

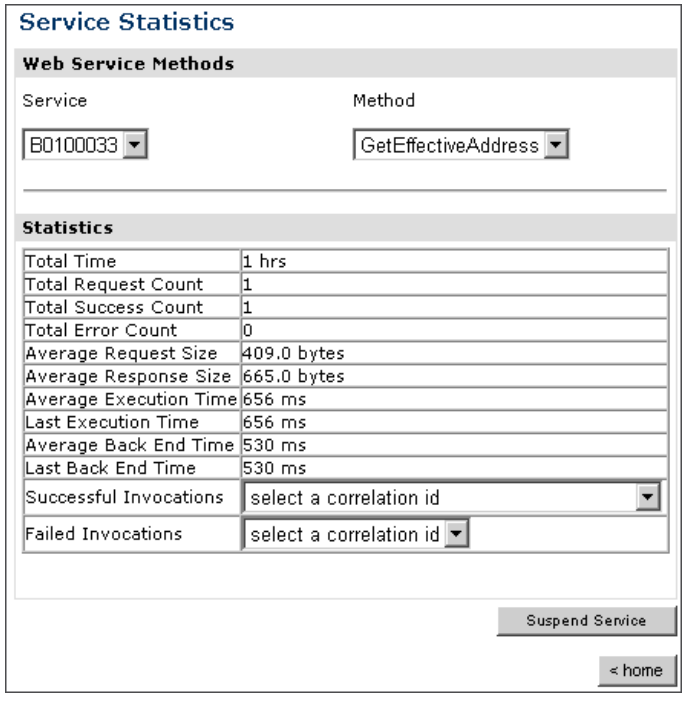


a. To stop a service at any time, click *Suspend Service*.

b. To restart the service, click *Resume Service*.

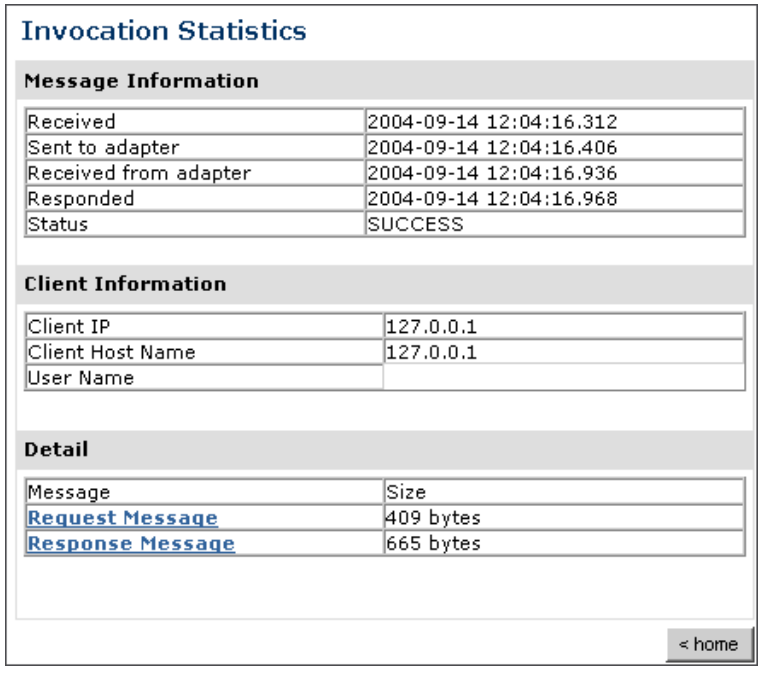
5. Select a method for the service from the Method drop-down list.

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).



6. 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.



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*.

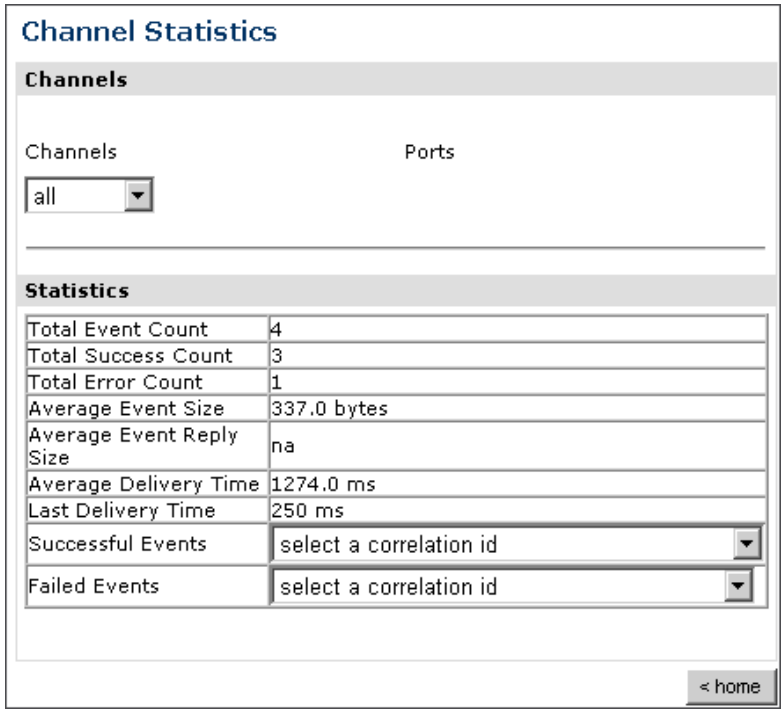
Procedure: How to Monitor Events

To monitor events:

1. Ensure that your application 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.



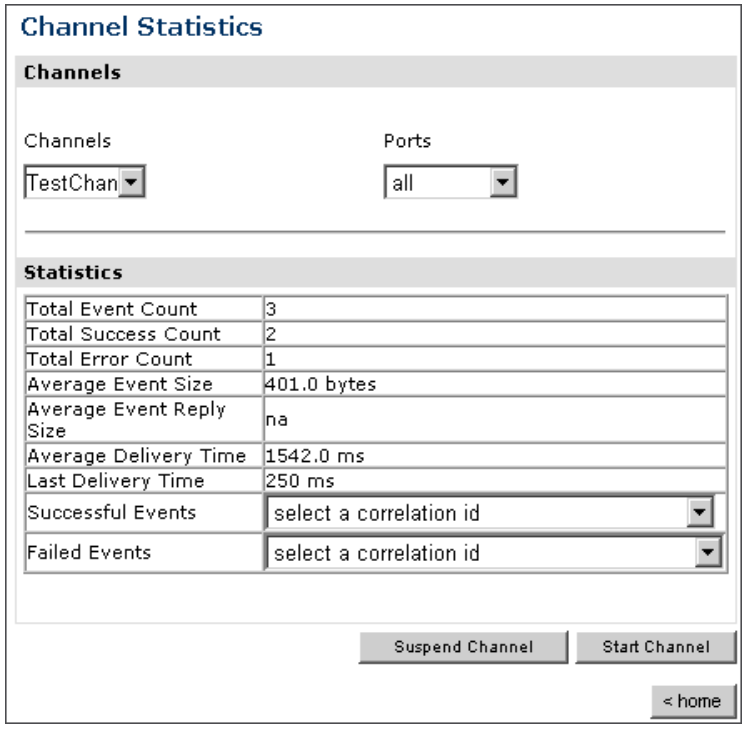
The system level summary provides event statistics at a system level.

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.

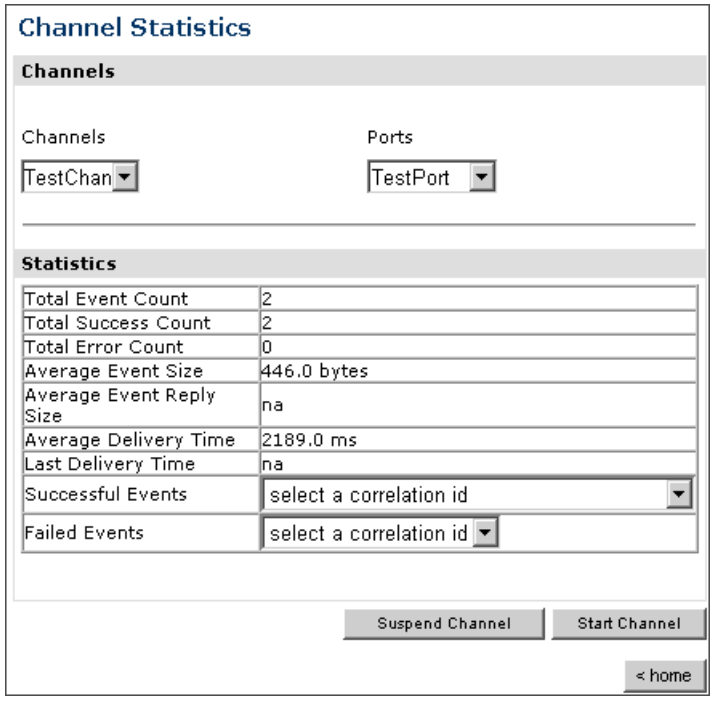


a. To stop a channel at any time, click *Suspend Channel*.

b. To start the channel, click *Start Channel*.

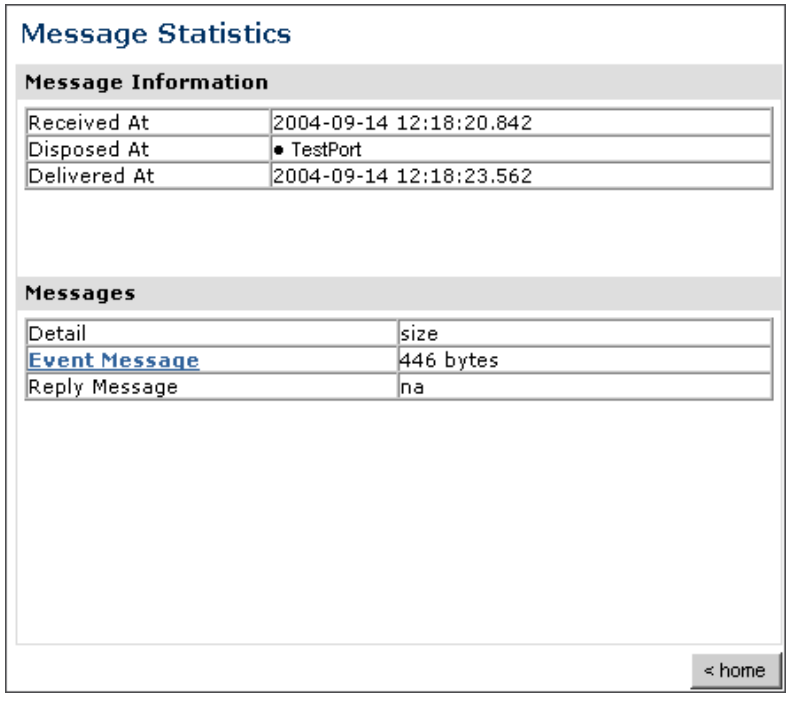
5. From the Ports drop-down list, select a port for the channel.

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.



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.



   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

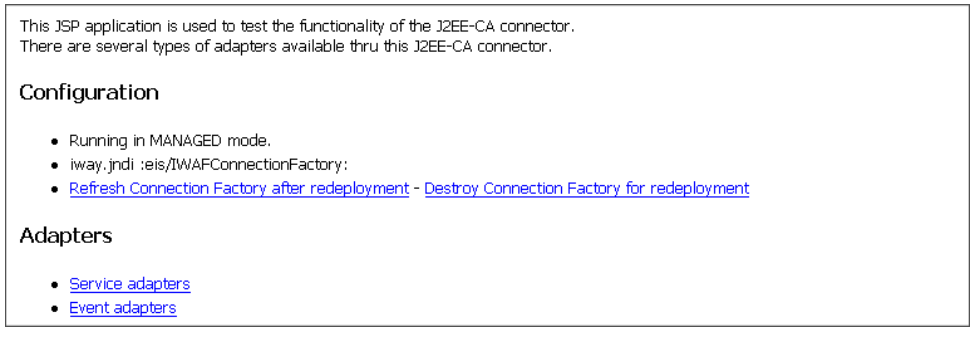
   Is the name of the machine where your application server is running.

   port

   Is the HTTP port for the application server, 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.

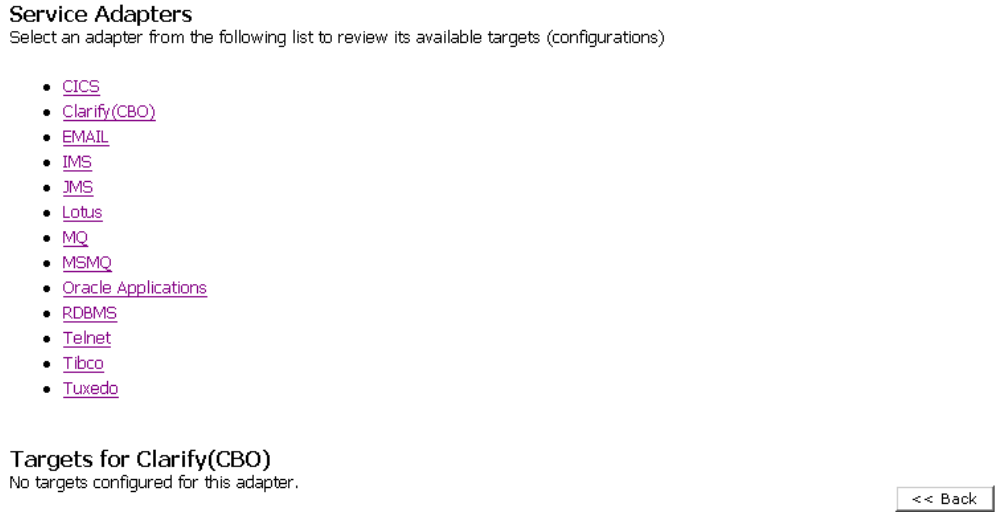
a. Click *Destroy Connection Factory for redeployment*.

b. Redeploy the JCA connector.

c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.

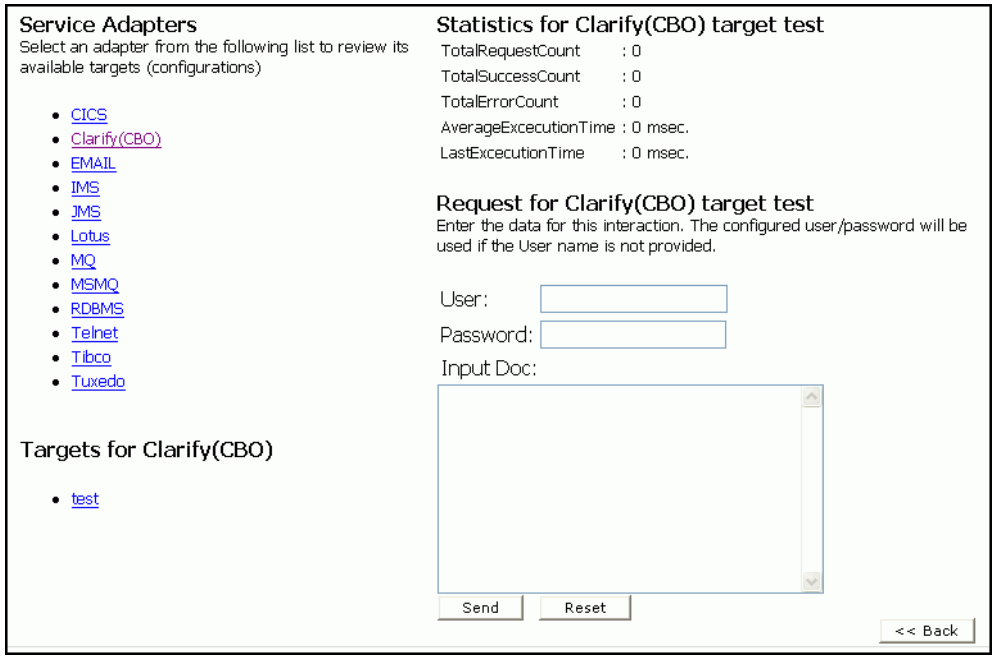
3. Click *Service adapters*.

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.



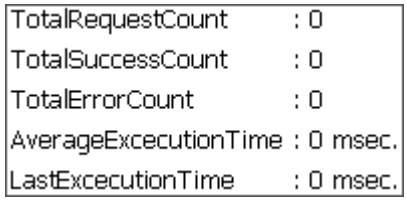
a. Click the desired target for your service adapter.

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.



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

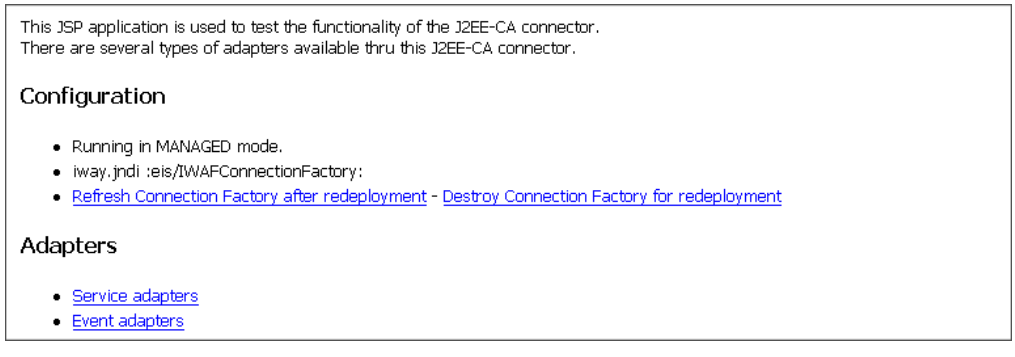
   Is the name of the machine where your application server is running.

   port

   Is the HTTP port for the application server, 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 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.

   c. In the JCA Test Tool, click *Refresh Connection Factory after redeployment*.

3. Click *Event adapters*.

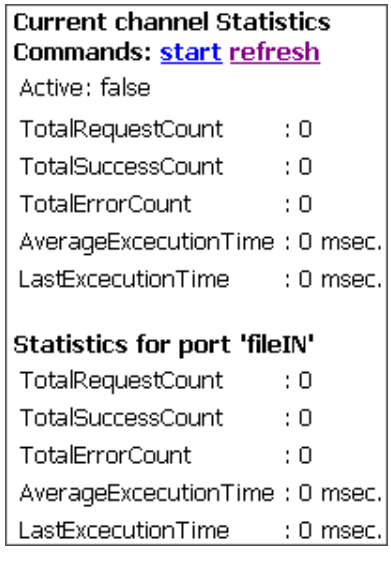
   The Event Adapters page opens.

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.



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* 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 HTTP port for the application server, 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*.

Tracing information is written to the ibselogs directory where your application server accesses or has expanded Servlet iBSE.

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* 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.

File Repositories

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

```
drive:\Program Files\iWay55\ibse\ibserepo.xml
```

where:

drive

Is the location of your iWay 5.5 installation.

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

```
drive:\ProductionConfig\ibse\ibserepo.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* 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 5-28.

Procedure: How to Migrate an iBSE Repository Configured for Oracle

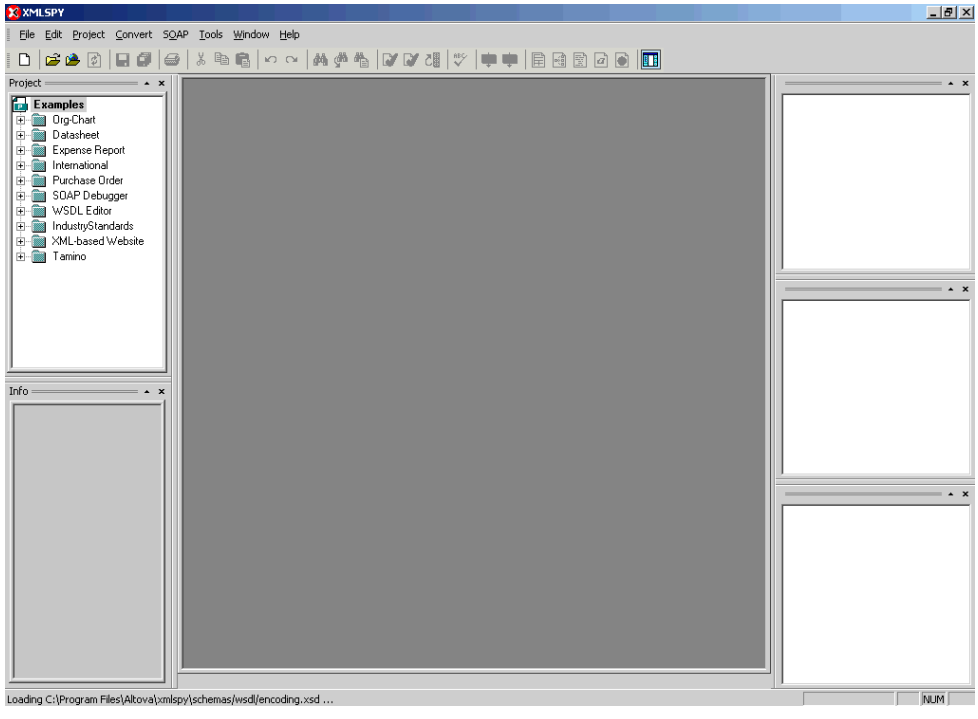
To migrate an iBSE repository that is configured for Oracle:

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

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

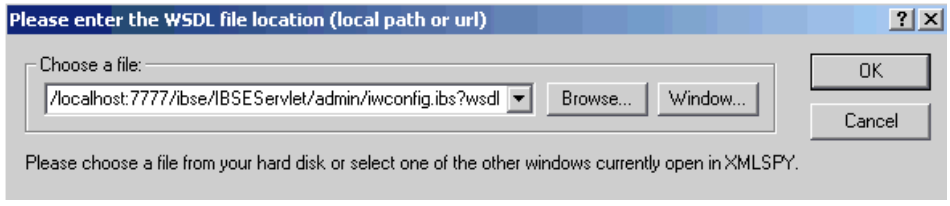
2. Open a third-party XML editor, for example, XMLSPY.

   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*.

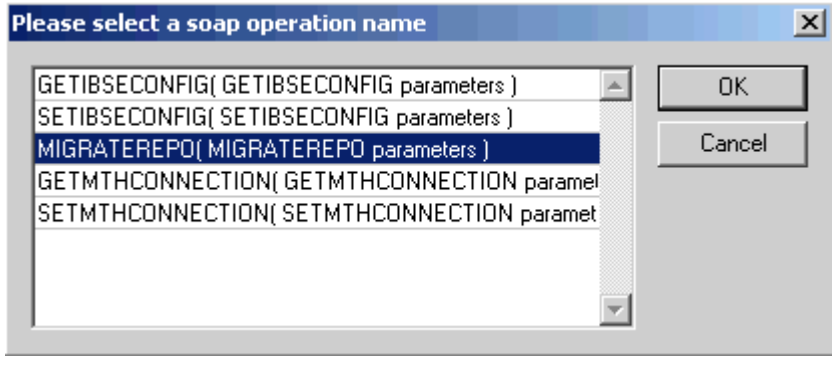
   The following image shows the WSDL file location dialog box that opens, where you enter a local path or URL. The dialog includes Browse, Window, OK, and Cancel buttons.



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

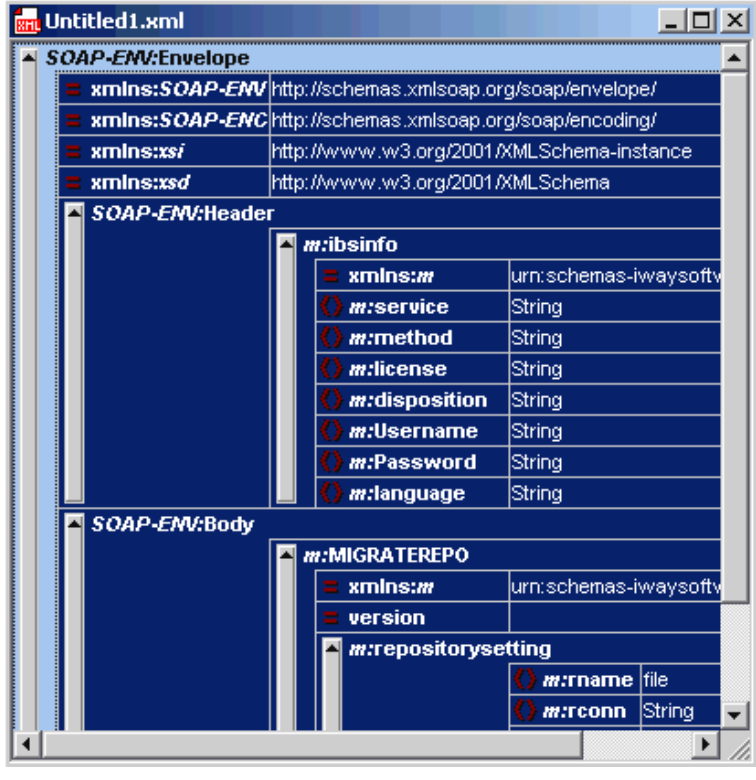
5. Click *OK*.

   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* 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:

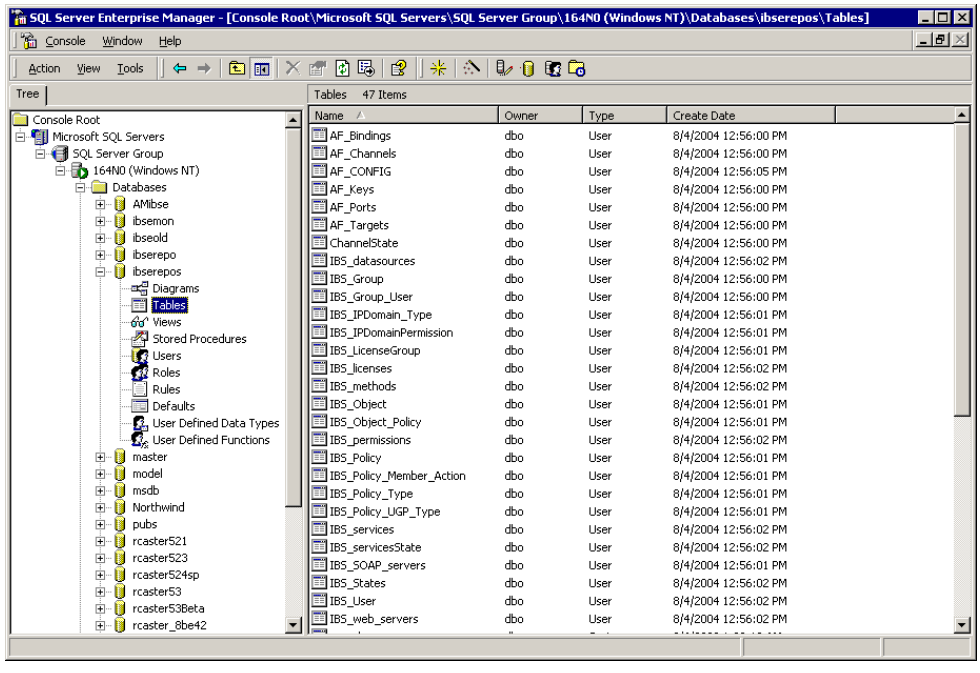
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   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* 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:

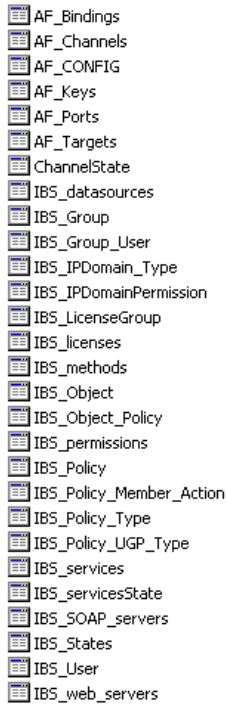
   For Oracle 8

   ```
   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.



   For more information on configuring the Oracle repository, see the *iWay Installation and Configuration* 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:

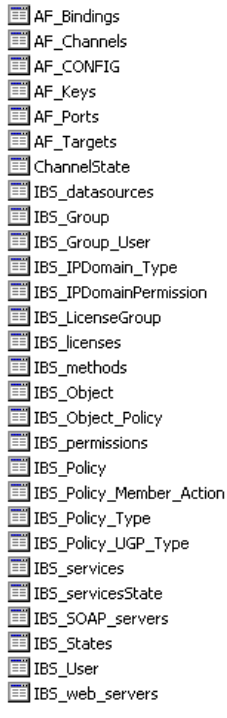
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   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.



   For more information on configuring the Sybase repository, see the *iWay Installation and Configuration* 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:

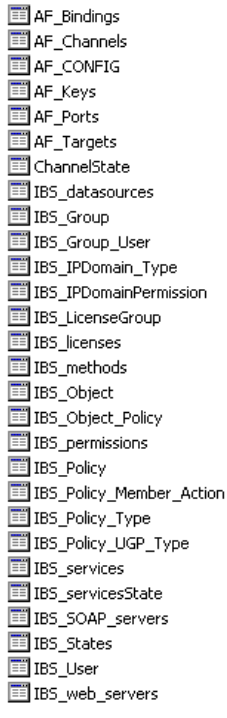
1. Open a command prompt and navigate to the iWay setup directory. The default location on Windows is:

   ```
   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.



   For more information on configuring the DB2 repository, see the *iWay Installation and Configuration* 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

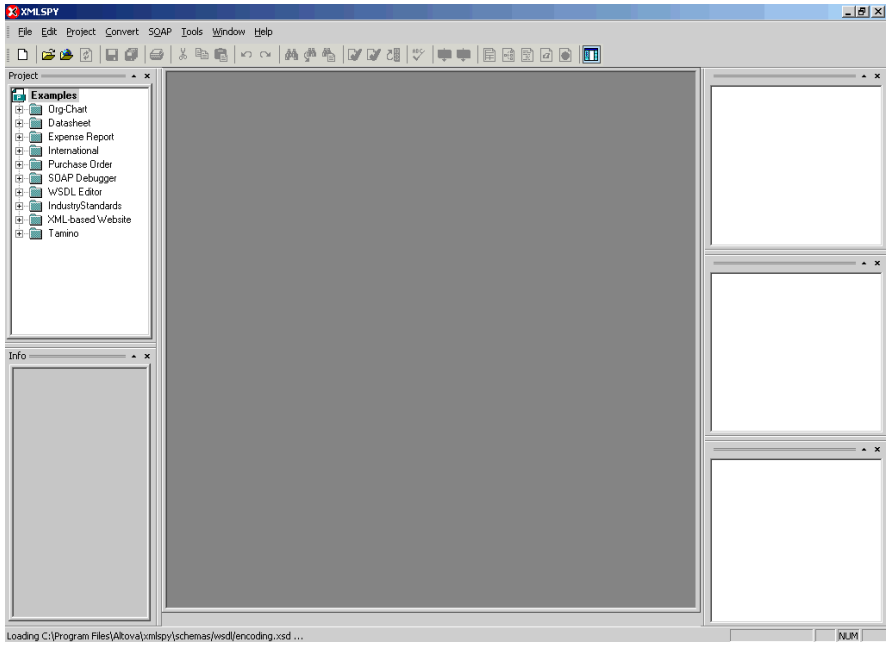
To export a target:

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

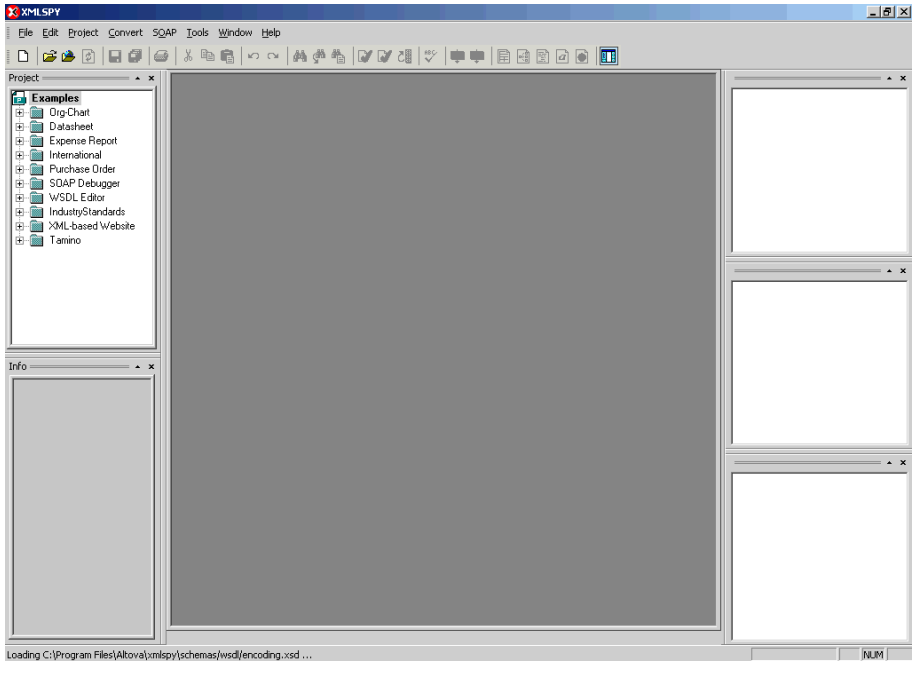
To import a target:

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>R0lGODlhcgGSALMAAAQCAEMmCZtuMFQxDS8b</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

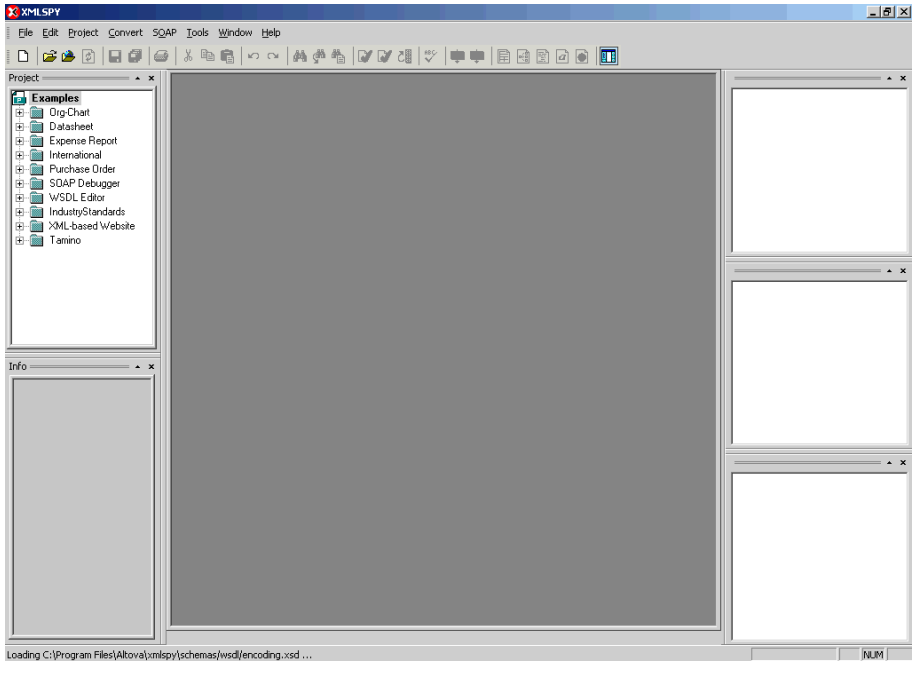
To retrieve Web service method connection information:

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

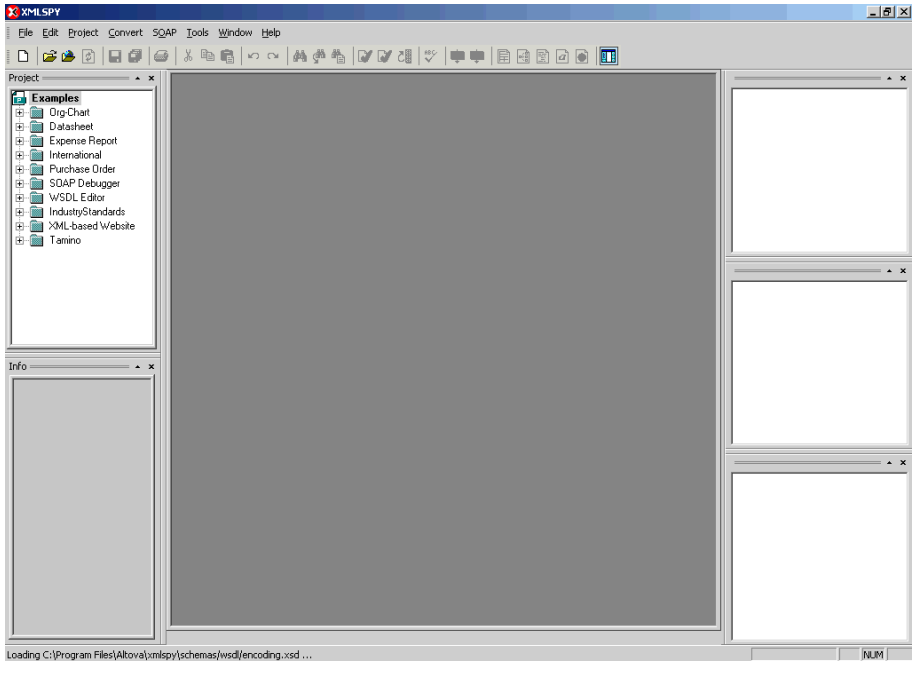
To update Web service method connection information:

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

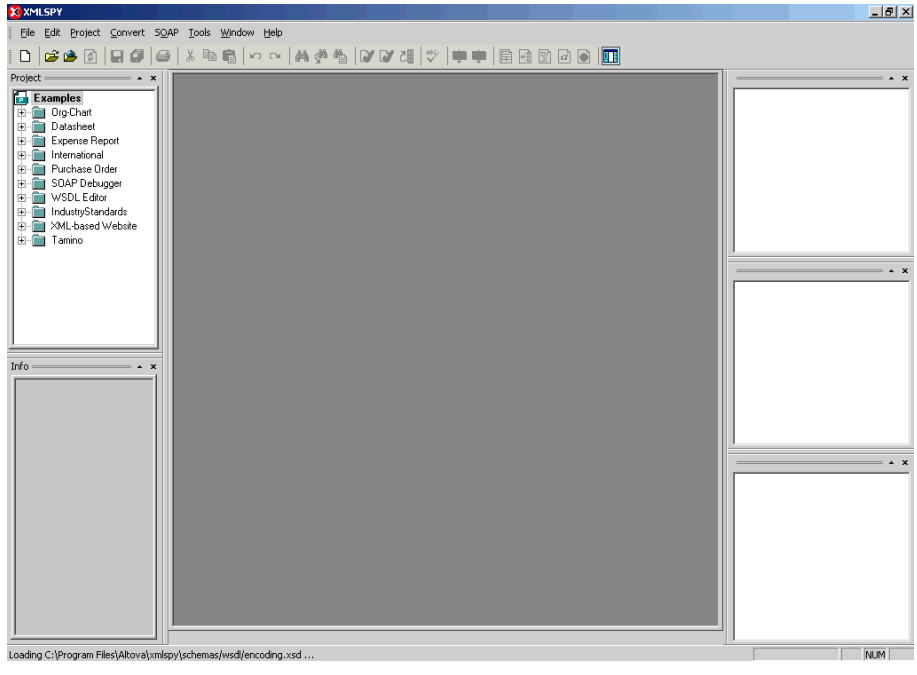
To start a channel programmatically:

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.

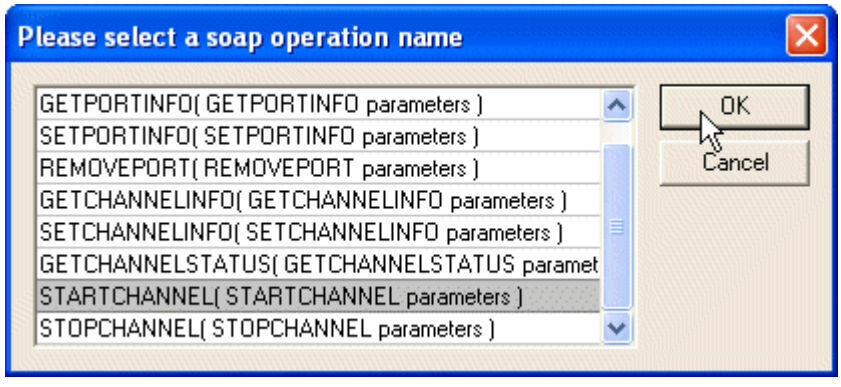


3. From the SOAP menu, select *Create new SOAP request*.

   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

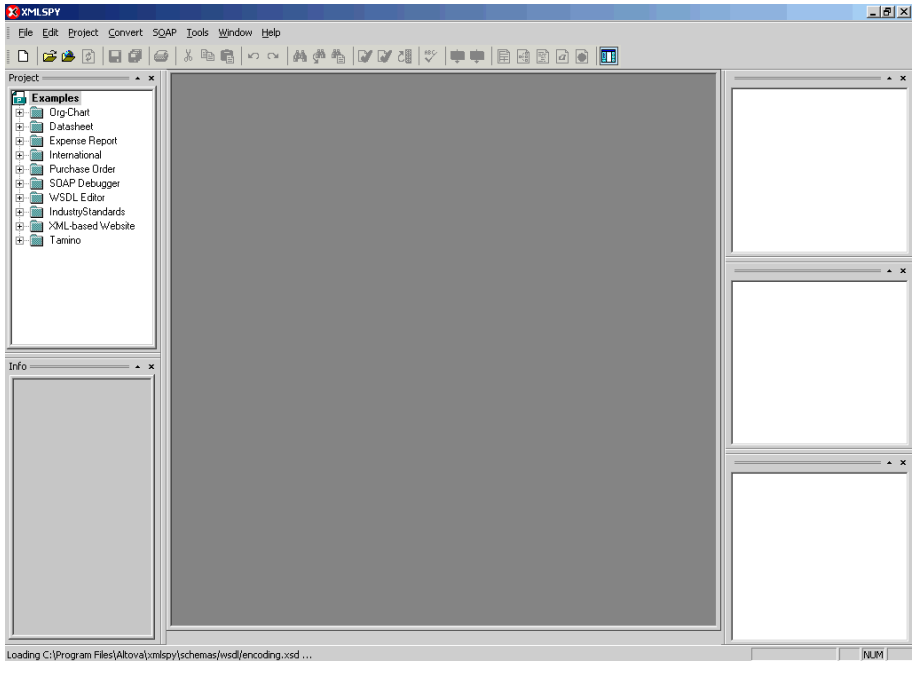
To stop a channel programmatically:

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.

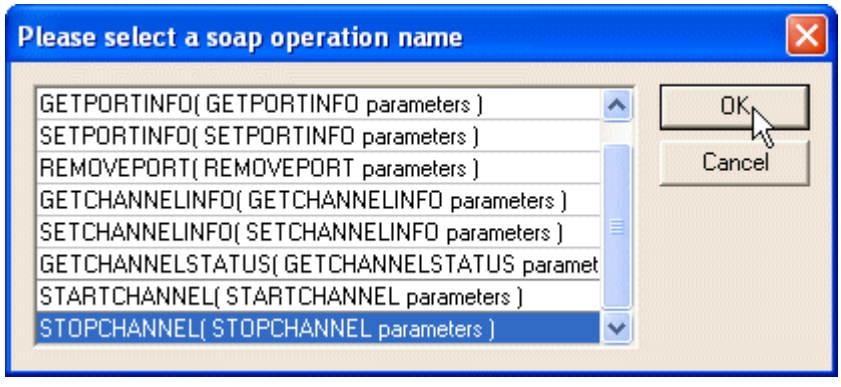


3. From the SOAP menu, select *Create new SOAP request*.

   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.*

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

Information Builders, Two Penn Plaza, New York, NY 10121-2898 **(212) 736-4433**

iWay Application System Adapter for Amdocs ClarifyCRM User's Guide DN3501482.0305
Version 5 Release 5