BEA WebLogic Adapter for ClarifyCRM®

User Guide

Release 7.0
Document Date: January 2003

BEA WebLogic Adapter for ClarifyCRM User Guide

Part Number	Date	Release
N/A	January 2003	7.0

Table of Contents

About This Document

The *BEA WebLogic Adapter for ClarifyCRM User Guide* is organized as follows:

- Chapter 1, "Introducing the BEA WebLogic Adapter for ClarifyCRM," introduces the BEA WebLogic Adapter for ClarifyCRM, describes its features, and provides an overview of how it works.
- Chapter 2, "Creating Application Views," describes how to define and deploy application views, how to test deployed services and events, and how to use services and events in a workflow.
- Chapter 3, "Creating Schema Repositories," addresses the schema repositories, manifests, and schemas that describe the documents entering and exiting a WebLogic Integration system.

What You Need to Know

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 adapter tools to develop online interconnections to ClarifyCRM using BEA WebLogic Integration.

Related Information

The following documents provide additional information for the associated software components:

- *BEA WebLogic Adapter for ClarifyCRM Installation and Configuration Guide*
- *BEA WebLogic Adapter for ClarifyCRM Release Notes*
- BEA WebLogic Server installation and user documentation, which is available at the following URL:

  `http://edocs.bea.com/more_wls.html`
- BEA WebLogic Integration installation and user documentation, which is available at the following URL:

  `http://edocs.bea.com/more_wli.html`

Contact Us!

Your feedback on the BEA Adapter for ClarifyCRM documentation is important to us. Send us e-mail at `docsupport@bea.com` if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the adapter documentation.

In your e-mail message, please indicate which version of the adapter documentation you are using.

If you have any questions about this version of the adapter, or if you have problems using it, contact BEA Customer Support through BEA WebSupport at www.bea.com. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

Documentation Conventions

The following conventions are used throughout this document.

Convention	Item
boldface text	Indicates terms defined in the glossary.
Ctrl+Tab	Indicates that you must press two or more keys simultaneously.
italics	Indicates emphasis or book titles.

`monospace text`	Indicates code samples, commands and their options, data structures and their members, data types, directories, and file names and their extensions. Monospace text also indicates text that you must enter from the keyboard.
	Examples:
	`#include <iostream.h> void main ( ) the pointer psz`
	`chmod u+w *`
	`\tux\data\ap`
	`.doc`
	`tux.doc`
	`BITMAP`
	`float`

Convention	Item
`monospace boldface text`	Identifies significant words in code. *Example*: `void` **`commit`** `( )`
`monospace italic text`	Identifies variables in code. *Example*: `String` *`expr`*
UPPERCASE TEXT	Indicates device names, environment variables, and logical operators. *Examples*: LPT1 SIGNON OR
{ }	Indicates a set of choices in a syntax line. The braces themselves should never be typed.

`[ ]`	Indicates optional items in a syntax line. The brackets themselves should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

|	Separates mutually exclusive choices in a syntax line. The symbol itself should never be typed.

`...`	Indicates one of the following in a command line:
	■ That an argument can be repeated several times in a command line
	■ That the statement omits additional optional arguments
	■ That you can enter additional parameters, values, or other information
	The ellipsis itself should never be typed.
	Example:
	`buildobjclient [-v] [-o name ] [-f` *`file-list`*`]...`
	`[-l` *`file-list`*`]...`

. . .	Indicates the omission of items from a code example or from a syntax line. The vertical ellipsis itself should never be typed.

CHAPTER

1 Introducing the BEA WebLogic Adapter for ClarifyCRM

The BEA WebLogic Adapter for ClarifyCRM provides a means to exchange real-time business data between ClarifyCRM systems and other application, database, or external business partner systems. The adapter allows for inbound and outbound processing with ClarifyCRM.

This section provides information about the adapter that will help you accomplish your integration projects. It includes the following topics:

- Understanding ClarifyCRM
- Resource Adapters
- Executing ClarifyCRM Business Functions
- Accessing Data Stored in ClarifyCRM

Understanding ClarifyCRM

ClarifyCRM eFrontOffice (CeFO) provides an integrated solution that allows you to manage your interactions with your customers. With CeFO, you can track sales, contracts, customers, product development, inventory, and repair operations. You can capture any customer interactions made over the telephone, email, or the Web. You can follow the interactions with a customer from the first request for information through the sales cycle and through subsequent requests for additional product or services. Some of the CeFO client applications include:

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

ClarifyCRM Architecture

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

When you run a CeFO application on the client, the application displays 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 use a form to enter information about an incoming customer call. The information that you enter is stored in the main database. Other employees can use ClearSupport to view and update the information that you have 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 ClarifyCRM Business Object

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

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

How ClarifyCRM Business Object Accesses the Database

A business object has methods and properties that can be used to query and update data in a CeFO database table.

Each business object contains a rowset, which is an in-memory copy of a set of rows from a CeFO database table. When database table is queried, the business object holds the results of the query in its rowset. A row in the rowset can be selected and any of the values of fields in that row can be obtained.

The rowset can be used to make changes to data in the database table. The values of fields in rows can be modified and the changed rows committed back to the database. If a new row must be added to the database table, add the rows to the rowset in the business object and commit the new rows to the database.

Resource Adapters

The BEA WebLogic Adapter for ClarifyCRM is a resource adapter: it connects one application to another when those applications are not originally designed to communicate with each other. The adapter sends requests to ClarifyCRM and returns data retrieved from ClarifyCRM. When the adapter makes requests of ClarifyCRM, it is referred to as a service adapter; when it detects an event in ClarifyCRM, it is referred to as an event adapter.

The ClarifyCRM event adapter monitors a ClarifyCRM database, stored in either MS SQL Server or Oracle, that contains only the most recent changes made in a ClarifyCRM system. That information, when retrieved by the ClarifyCRM event adapter, is formatted into an XML document and sent to the WebLogic Server for further processing, transforming, and/or routing. Once the event is processed successfully, the event data is deleted from the event database.

Event adapters perform the following functions:

- Supplied database triggers move only relevant data about an action that was performed by ClarifyCRM into an events database.
- Poll the ClarifyCRM events database at user-configured time intervals.
- Translate the event information to XML.
- Support transformation of XML documents that contain event information.

The ClarifyCRM service adapter processes requests for ClarifyCRM functions embedded in XML documents and forwards them to a back-end ClarifyCRM system. The resulting response information is then brought back and processed by the service adapter for further routing.

Service adapters receive an XML request document from a client and call a specific function in ClarifyCRM. They are consumers of request messages and, depending on the request, may or may not provide a response. There are two kinds of services:

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

BEA WebLogic Integration supports both of these methods, so you do not have to provide this functionality in your own application code.

Service adapters perform the following functions:

- Receive service requests from an external client.
- Transform the XML request document into the ClarifyCRM-specific format. The request document conforms to the request XML schema for the service. The schema is based on Clarify/CRM metadata.
- Call the underlying function in ClarifyCRM and wait for its response.
- Transform the response from the ClarifyCRM-specific data format to an XML document that conforms to the response XML schema for the service. The schema is based on ClarifyCRM metadata.

Key features of the BEA WebLogic Adapter for ClarifyCRM include support for:

- Bi-directional message interaction between WebLogic Integration and ClarifyCRM.
- Service (ClarifyCRM inbound) and event (ClarifyCRM outbound) adapter integration operations with ClarifyCRM presenting XML schemas to WebLogic Integration Studio.

Application Interaction

The adapter enables non-ClarifyCRM systems to communicate and exchange transactions with ClarifyCRM using WebLogic Integration and XML messages.

Applications that interact with ClarifyCRM to cause a new ClarifyCRM event use WebLogic Integration application views, services, and WebLogic Integration Studio to send request messages to ClarifyCRM via the adapter. If the request retrieves data from ClarifyCRM, the adapter sends the application a response message with the data.

Executing ClarifyCRM Business Functions

You can use the BEA WebLogic Adapter for ClarifyCRM to perform inserts, updates, and deletes against ClarifyCRM objects stored in the ClarifyCRM database. The adapter supports the Contact, PartMaster, ItemPricesQty, ItemPrice, Quote, QuoteSchedule, QuoteLine QuotePONumber, QuoteScheduleBilltoCust, and QuoteScheduleShiptoCust objects. You can also use the adapter to integrate ClarifyCRM with non-ClarifyCRM systems.

Accessing Data Stored in ClarifyCRM

The BEA WebLogic Adapter for ClarifyCRM provides controlled access to the data contained in the ClarifyCRM database, via the ClarifyCRM CBO interface. The GET method is issued via an XML document the same way as Insert, Update, and Delete. The results of the GET are contained in an XML response document. W3C schemas for all the Request and Response documents are supplied, but must be specified in the `manifest.xml` file used when creating the application view.

Compatibility Features

The BEA WebLogic Adapter for ClarifyCRM supports all releases of ClarifyCRM from 8.1 through 11.2. This is because the adapter uses the CBO interface for version 10 and above, and the ClarifyCRM High Level API—a COM-based programming interface—for versions below 10.

Since WebLogic Server is a Java-based server, the adapter must exploit JNI services. The following table lists the different versions of ClarifyCRM and their supported interfaces.

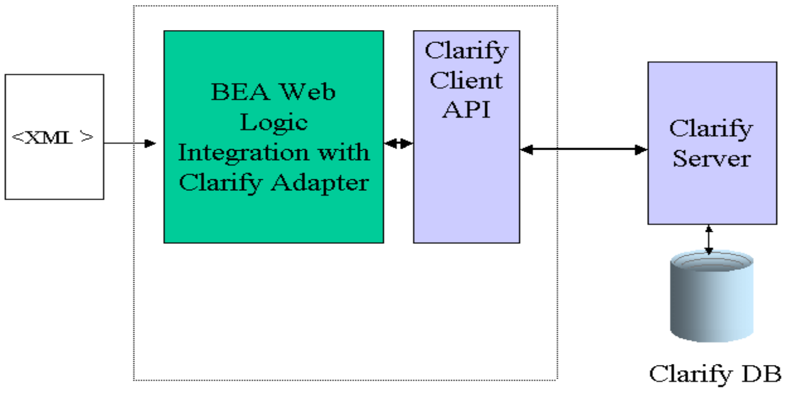
Table 1-1 ClarifyCRM Interfaces Supported

ClarifyCRM Release	API Tool Kits	High Level API	ActiveX (CBO)	Java Beans
8.1	Yes	Yes	No	No
8.5	Yes	Yes	No	No
9	Yes	Yes	Yes	No
10	Yes	Yes	Yes	Yes
10.1	Yes	Yes	Yes	Yes
10.2	Yes	Yes	Yes	Yes
11.1	Yes	Yes	Yes	Yes
11.2	Yes	Yes	Yes	Yes

Processing Services

The following figure illustrates the WebLogic Integration service processing framework.

Figure 1-1 Service Architecture



When the BEA WebLogic Adapter for ClarifyCRM processes a service:

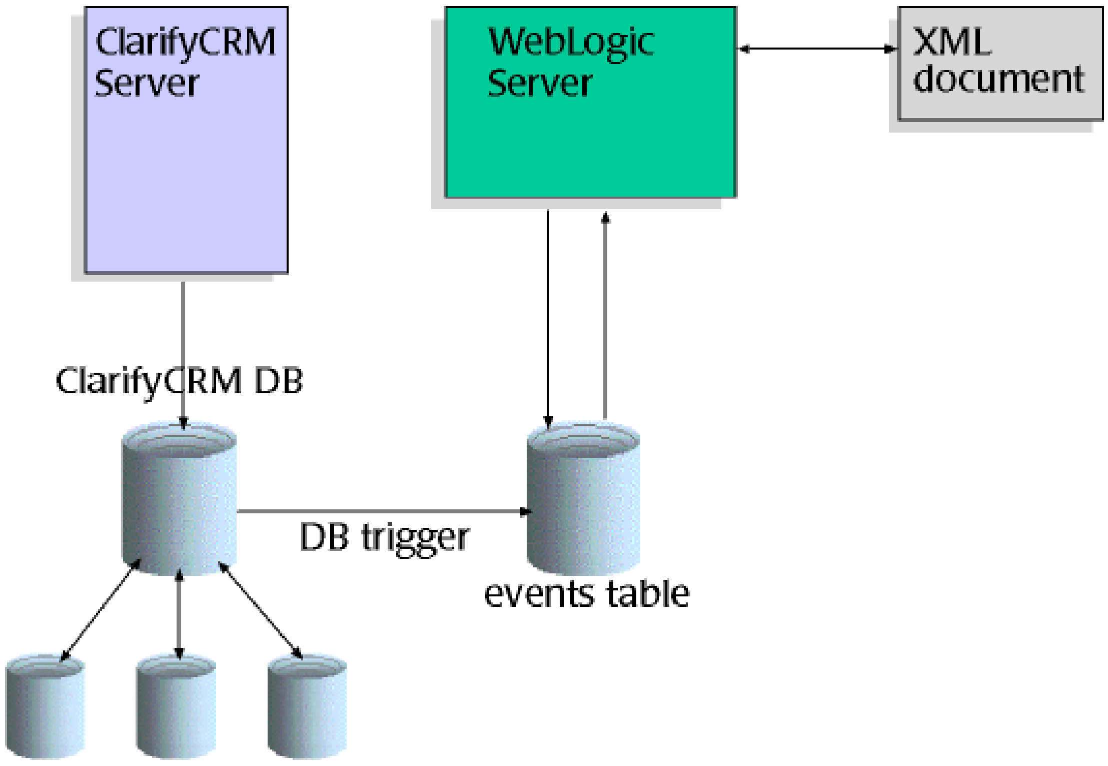
1. The adapter receives an XML service request document from a WebLogic Server workflow and connects it, via the workflow, to a pre-configured WebLogic service.
2. The adapter validates the request to ensure that it conforms to the service's request schema.
3. The adapter interprets the document and 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's response schema.

6. The adapter returns the XML response document to WebLogic Server.
7. WebLogic Server, via a workflow, transforms the document and/or routes it to the service originator.

Processing Events

The following figure illustrates the WebLogic Integration event processing framework.

Figure 1-2 Event Architecture



When the BEA WebLogic Adapter for ClarifyCRM processes an event:

1. An update is run against the ClarifyCRM Server.
2. The update is written to the ClarifyCRM database, which triggers the relevant event data to be copied to the event table.
3. WebLogic Server (WebLogic Integration/Application Integration) periodically polls the events table with a specified SQL query. When the events table satisfies the query, the BEA WebLogic Adapter for ClarifyCRM generates an XML event document describing the event. The document conforms to the event's schema.
4. The adapter emits the event document to WebLogic Server, where it can be routed, or sent back through a workflow. The record in the event table is deleted upon successful generation and emission of the XML document.
5. WebLogic Server, via a workflow, transforms the document and/or routes it. The event's record is then deleted from the events table.

CHAPTER

2 Creating Application Views

When you define an application view, you create an XML-based interface between WebLogic Server and a particular Enterprise Information System (EIS) application within your enterprise. Once you create an application view, a business analyst can use it to create business processes that use the application. With the BEA WebLogic Adapter for ClarifyCRM, you can create any number of application views, each with any number of services and events.

This section provides information on application views and deployed services and events, and includes the following topics:

- Creating an Application View Folder
- Creating an Application View
- Adding a Service to an Application View
- Before Adding an Event to an Application View
- Adding an Event to an Application View
- Deploying an Application View
- Undeploying an Application View
- Testing a Deployed Service
- Testing a Deployed Event
- Using a Service or Event in a Workflow

Creating an Application View Folder

Application views reside within WebLogic Integration. WebLogic Integration provides you with a root folder in which you can store all of your application views; if you wish, you can create additional folders to organize related application views into groups.

To create an application view folder:

1. Log on to the WebLogic Integration Application View Console at `//`*`appserver-host:port`*`/wlai`.

   Here, *`appserver-host`* is the IP address or host name where the WebLogic Integration Server is installed, and *`port`* is the socket on which the server is listening. The port, if not changed during installation, defaults to 7001.

2. If prompted, enter a user name and password.

   Note: If the user name is not `system`, it must be included in the `adapter` group. For more information on adding the administrative server user name to the `adapter` group, see the *BEA WebLogic Adapter for ClarifyCRM Installation and Configuration Guide*.

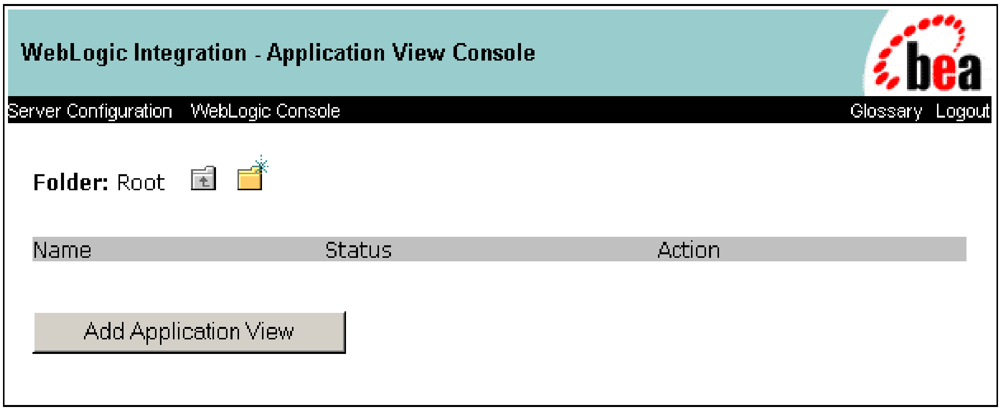
   For more information, see “Logging On to the WebLogic Integration Application View Console” in “Defining an Application View” in *Using Application Integration*:

   - For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/2usrdef.htm`
   - For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/2usrdef.htm`

3. Click Login.

   The WebLogic Integration Application View Console opens.

Figure 2-1 Application View Console Main Window

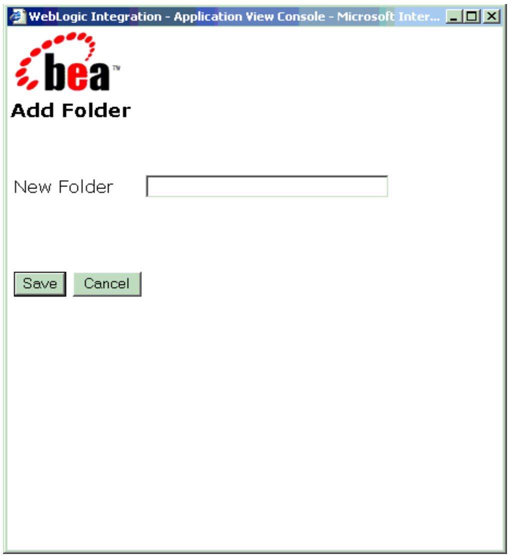


Folder: Root

Name	Status	Action

4. Double-click the new folder icon. The Add Folder window opens.

Figure 2-2 Add Folder Window


WebLogic Integration - Application View Console - Microsoft Inter...
bea
Save
Cancel

Add Folder

New Folder

5. Supply a name for the folder, and then click Save.

You have finished creating the application view folder. To create an application view, see "Creating an Application View" on page 2-4.

Creating an Application View

To create an application view:

1. Log on to the WebLogic Application View Console at `//appserver-host:port/wlai`.

   Here, *`appserver-host`* is the IP address or host name where the WebLogic Integration Server is installed, and *`port`* is the socket on which the server is listening. The port, if not changed during installation, defaults to 7001.

2. If prompted, enter a user name and password.

   Note: If the user name is not `system`, it must be included in the `adapter` group. For more information on adding the administrative server user name to the `adapter` group, see the *BEA WebLogic Adapter for ClarifyCRM Installation and Configuration Guide*.

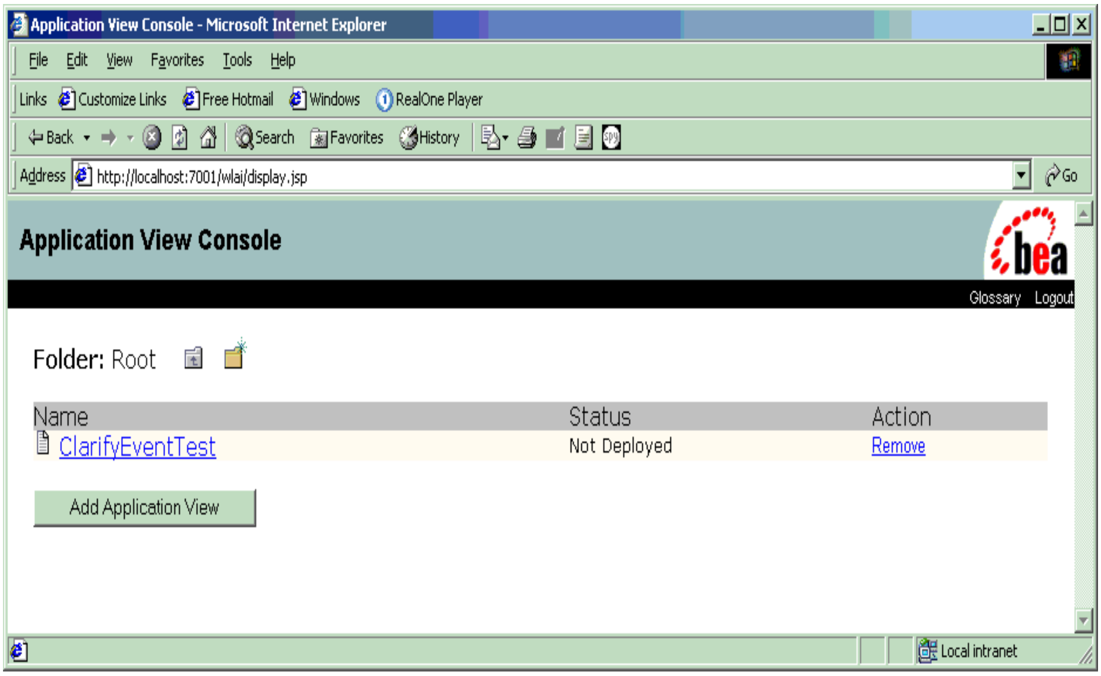
   For more information, see "Logging On to the WebLogic Integration Application View Console" in "Defining an Application View" in *Using Application Integration*:

   - For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/2usrdef.htm`
   - For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/2usrdef.htm`

3. Click Login.

   The WebLogic Integration Application View Console opens.

Figure 2-3 Selecting a Folder in the Main Window


Application View Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address http://localhost:7001/wlai/display.jsp
Go
Application View Console
bea
Glossary Logout
Add Application View
Local intranet

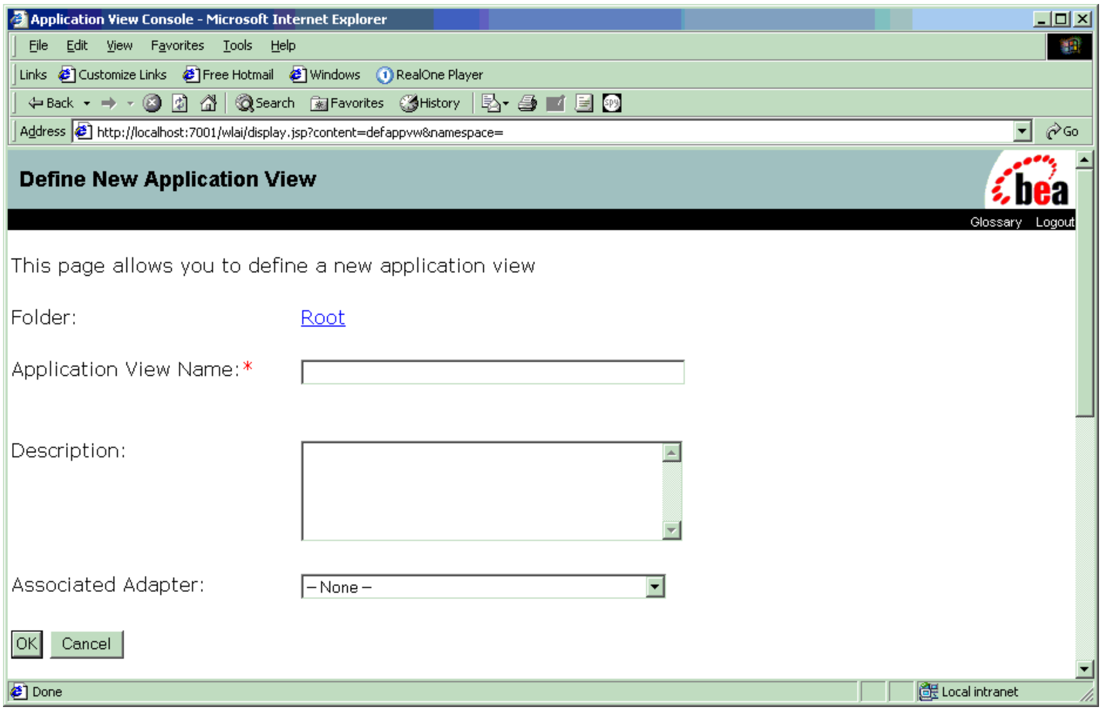
Folder: Root

Name	Status	Action
ClarifyEventTest	Not Deployed	Remove

4. Select the desired application view folder.
5. Click Add Application View.

The Define New Application View window opens.

Figure 2-4 Define New Application View Window


Application View Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address http://localhost:7001/wlai/display.jsp?content=defappvw&namespace=
Go
bea
Glossary Logout
Done
Local intranet

Define New Application View

This page allows you to define a new application view

Folder: Root

Application View Name:*

Description:

Associated Adapter: – None –

OK Cancel

An application view enables a set of functions for the adapter's target EIS application. For detailed information, see “Defining an Application View” in *Using Application Integration*:

- For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/2usrdef.htm`
- For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/2usrdef.htm`

6. Enter a name and description for the application view.

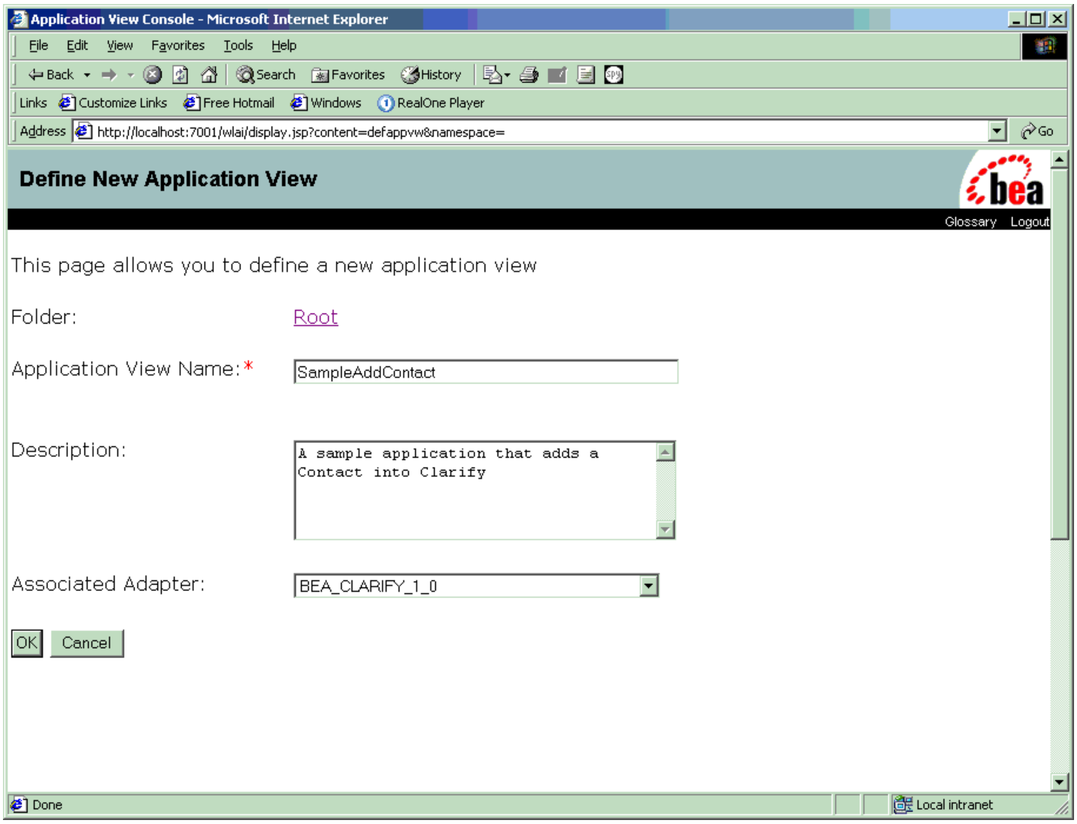
   The name should describe the set of functions performed by this application. Each application view name must be unique to its adapter. Valid characters are a-z, A-Z, 0-9, and _ (underscore).

The description is seen by users when they use this application view in workflows.

7. Select `BEA_CLARIFY_1_0` from the Associated Adapter drop-down list.

   The following is an example of a completed window.

Figure 2-5 Completed Define New Application View Window


Application View Console - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address
Go
bea
Glossary Logout
OK Cancel
Done
Local intranet

http://localhost:7001/wlai/display.jsp?content=defappvw&namespace=

Define New Application View

This page allows you to define a new application view

Folder: Root

Application View Name:* SampleAddContact

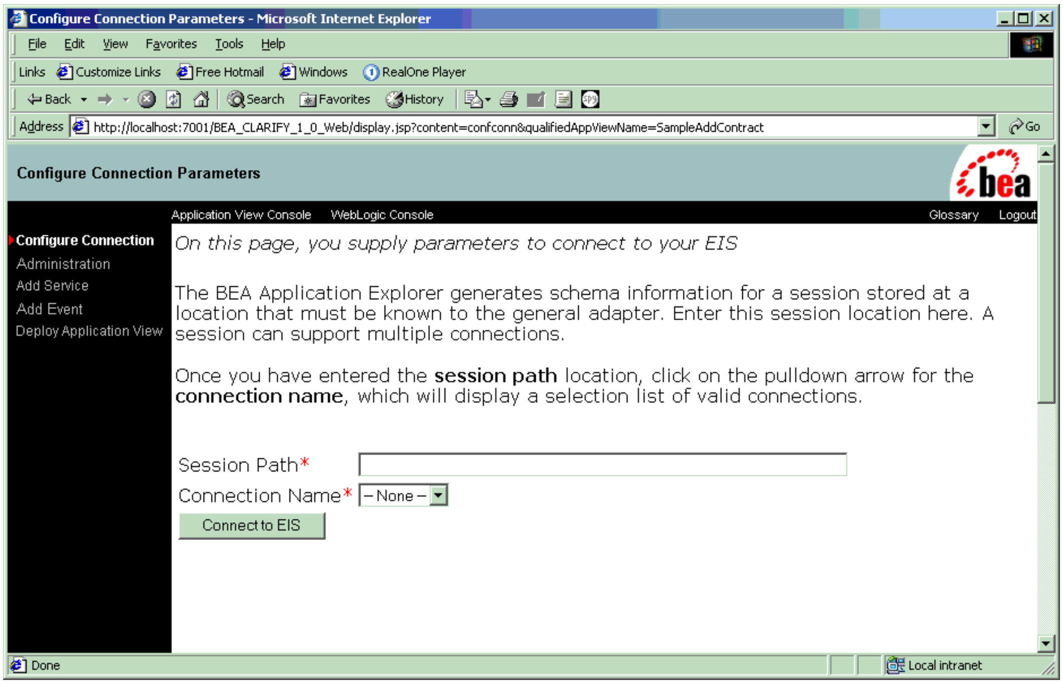
Description: A sample application that adds a Contact into Clarify

Associated Adapter: BEA_CLARIFY_1_0

8. Click OK.

   The Configure Connection Parameters window opens.

Figure 2-6 Configure Connection Parameters Window


Configure Connection Parameters - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address http://localhost:7001/BEA_CLARIFY_1_0_Web/display.jsp?content=confconn&qualifiedAppViewName=SampleAddContract
Go
Configure Connection Parameters
bea
Application View Console WebLogic Console Glossary Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Connect to EIS
Done
Local intranet

On this page, you supply parameters to connect to your EIS

The BEA Application Explorer generates schema information for a session stored at a location that must be known to the general adapter. Enter this session location here. A session can support multiple connections.

Once you have entered the **session path** location, click on the pulldown arrow for the **connection name**, which will display a selection list of valid connections.

Session Path*	
Connection Name*	– None –

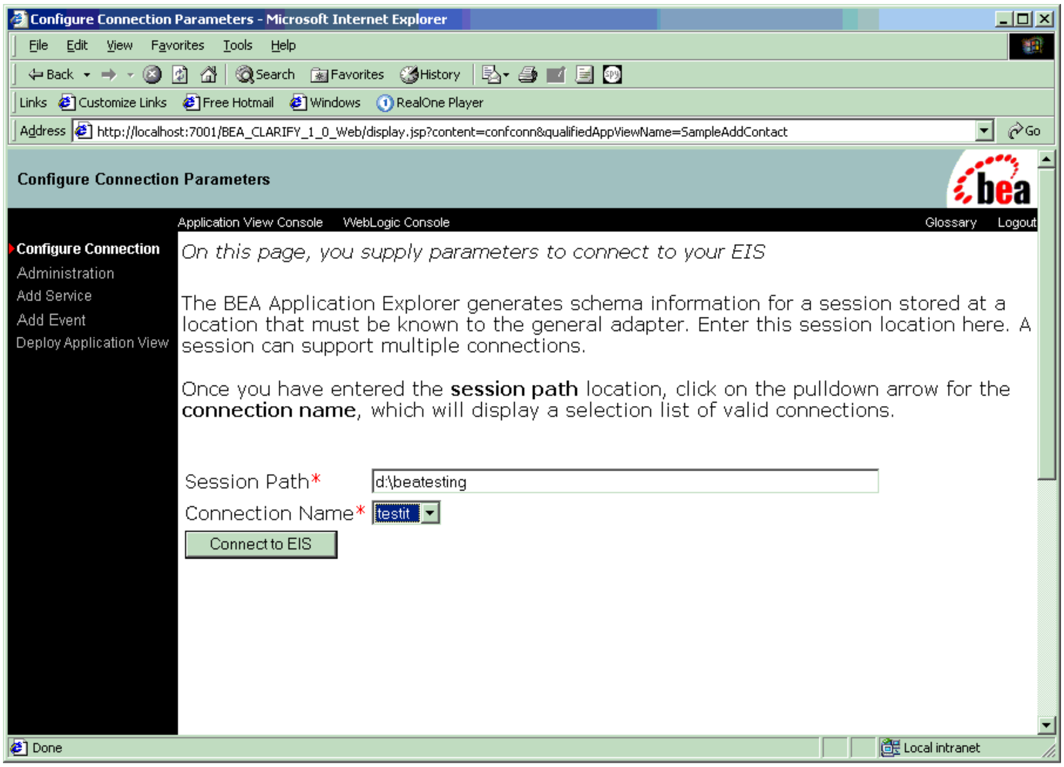
9. Enter the name of the BEA WebLogic Adapter for ClarifyCRM session path (sometimes known as the session base directory).

   This path holds your ClarifyCRM schema and connection information.

   For more information about the session path and schemas, see Chapter 3, “Creating Schema Repositories.”

10. Select the connection name (sometimes known as the session name) from the Connection Name drop-down list.

Figure 2-7 Completed Configure Connection Parameters Window


Configure Connection Parameters - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address http://localhost:7001/BEA_CLARIFY_1_0_Web/display.jsp?content=confconn&qualifiedAppViewName=SampleAddContact
Go
Configure Connection Parameters
bea
Application View Console WebLogic Console Glossary Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Connect to EIS
Done
Local intranet

On this page, you supply parameters to connect to your EIS

The BEA Application Explorer generates schema information for a session stored at a location that must be known to the general adapter. Enter this session location here. A session can support multiple connections.

Once you have entered the **session path** location, click on the pulldown arrow for the **connection name**, which will display a selection list of valid connections.

Session Path* d:\beatesting
Connection Name* testit

11. Click Connect to EIS. The Application View Administration window opens.

12. Click Save. You have finished creating the application view.

    You can now add a service or event to the application view, as described in "Adding a Service to an Application View" on page 2-10 and "Before Adding an Event to an Application View" on page 2-15. (You must add a service or event before you can deploy the application view.)

    Note that you can access the Configure Connection Parameters window (displayed in a previous step) when the application view is not deployed, simply by clicking the Reconfigure connection parameters link. If the application view is deployed, you can access the window by first undeploying the application view.

Adding a Service to an Application View

You can add a service to an application view to support the application's functions. (For information about creating an application view, see "Creating an Application View" on page 2-4.)

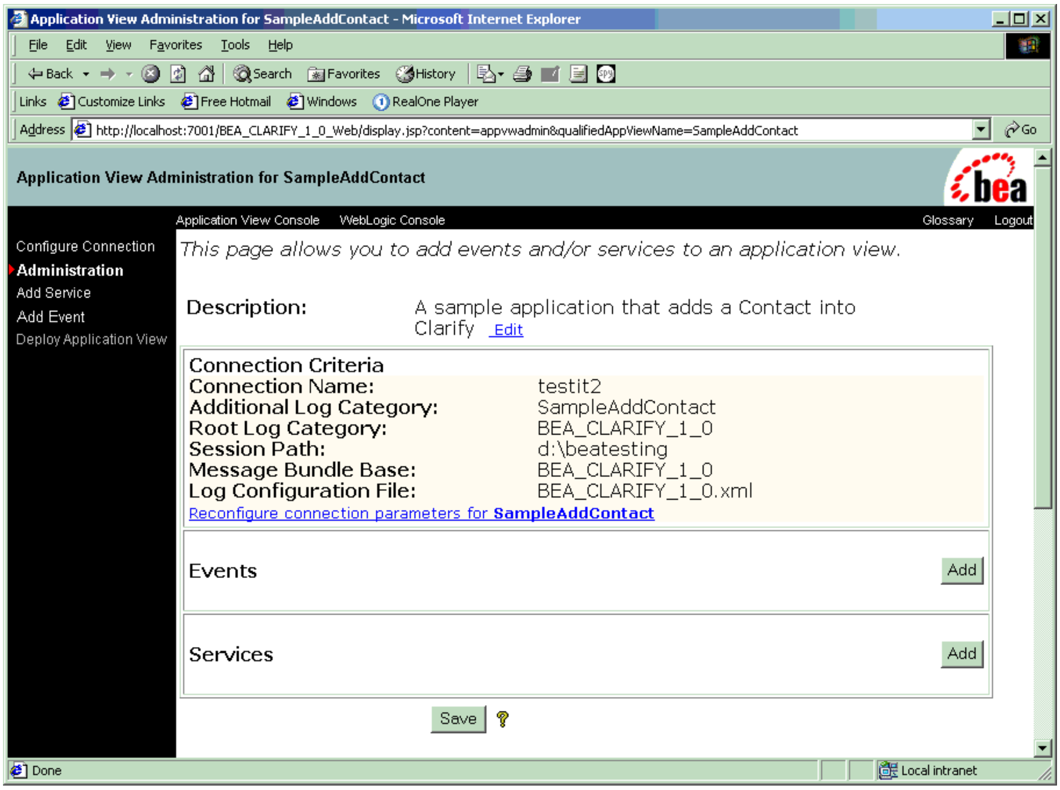
Note: If the application view is deployed, you must undeploy it before adding the service. For information about undeploying, see "Undeploying an Application View" on page 2-32.

To add a service:

1. Log on to the WebLogic Application View Console as described in "Creating an Application View" on page 2-4.
2. Select the folder in which this application view resides, and then select the application view.

   The Application View Administration window opens.

Figure 2-8 Application View Administration Window (Active Section)


Application View Administration for SampleAddContact - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address http://localhost:7001/BEA_CLARIFY_1_0_Web/display.jsp?content=appvwadmin&qualifiedAppViewName=SampleAddContact
Go
Application View Administration for SampleAddContact
bea
Application View Console
WebLogic Console
Glossary
Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Events
Add
Services
Add
Save
Done
Local intranet

This page allows you to add events and/or services to an application view.

Description: A sample application that adds a Contact into Clarify Edit

Connection Criteria

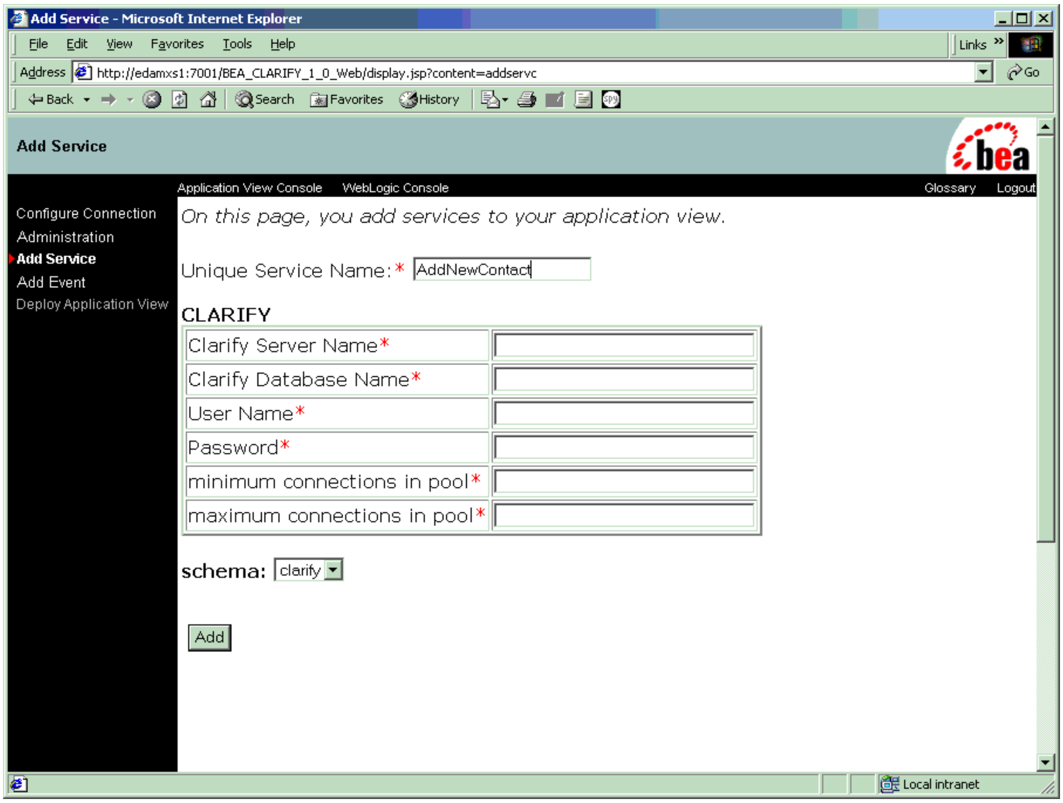
Connection Name:	testit2
Additional Log Category:	SampleAddContact
Root Log Category:	BEA_CLARIFY_1_0
Session Path:	d:\beatesting
Message Bundle Base:	BEA_CLARIFY_1_0
Log Configuration File:	BEA_CLARIFY_1_0.xml

Reconfigure connection parameters for **SampleAddContact**

3. Select Add Service in the left pane, or click Add in the Services section.

   The Add Service window opens.

Figure 2-9 Add Service Window


Add Service - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://edamxs1:7001/BEA_CLARIFY_1_0_Web/display.jsp?content=addservc
Back Search Favorites History
Add Service
Application View Console WebLogic Console Glossary Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Add
Local intranet

On this page, you add services to your application view.

Unique Service Name:* AddNewContact

CLARIFY

Clarify Server Name*	
Clarify Database Name*	
User Name*	
Password*	
minimum connections in pool*	
maximum connections in pool*	

schema: clarify

4. Specify the service's properties, which are described in the following table. Required values are indicated on the screen with an asterisk (*).

Table 2-1 Service Properties

Property	Description
Unique Service Name	The name of the service that you are adding. The name must be unique within the application view. Valid characters include a-z, A-Z, 0-9, and underscore (_).
Clarify Server Name	Network id name for computer hosting ClarifyCRM Server

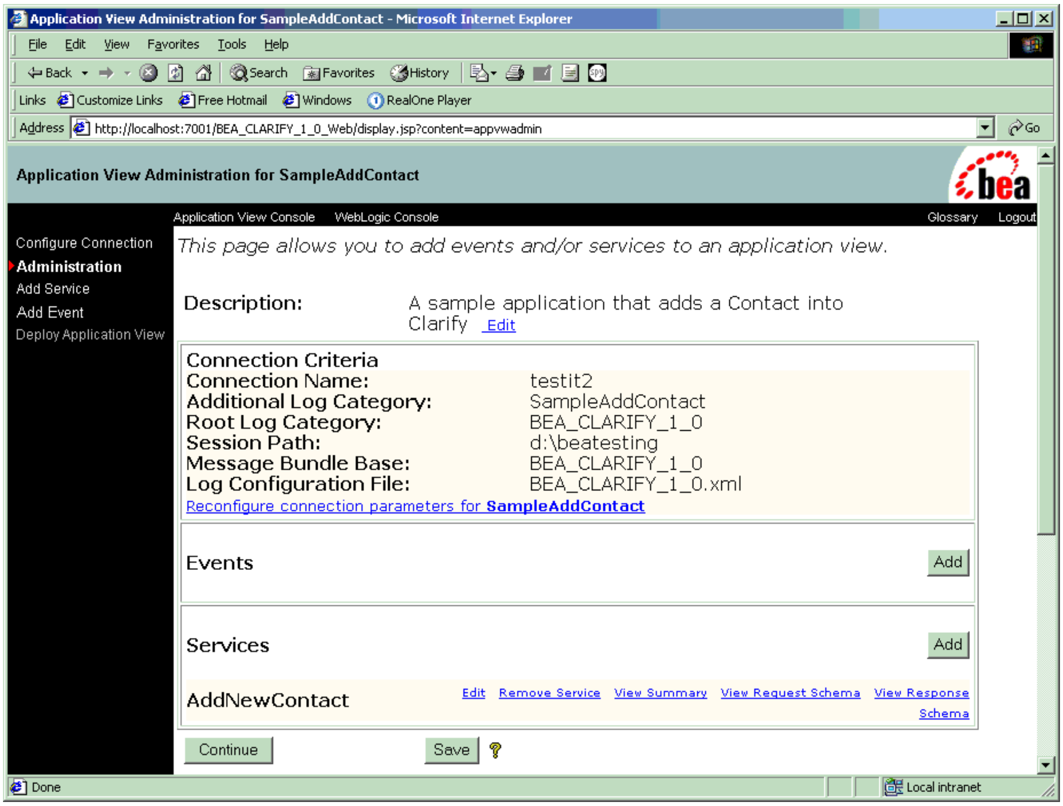
Table 2-1 Service Properties (Continued)

Property	Description
Clarify Database Name	Name of the database used by the ClarifyCRM Server
User Name	The user ID required to access the ClarifyCRM service.
Password	The password associated with the specified user ID.
minimum connections in pool	The minimum number of connections to the ClarifyCRM Server the adapter should open.
maximum connections in pool	The maximum number of connections to the ClarifyCRM Server the adapter should open.
schema	The name of the schema that describes the service.

5. Click Add.

   The Application View Administration window opens.

Figure 2-10 Service Added to Application View


Application View Administration for SampleAddContact - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address http://localhost:7001/BEA_CLARIFY_1_0_Web/display.jsp?content=appvwadmin
Go
Application View Administration for SampleAddContact
bea
Application View Console WebLogic Console Glossary Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View

This page allows you to add events and/or services to an application view.

Description:	A sample application that adds a Contact into Clarify Edit

Connection Criteria

Connection Name:	testit2
Additional Log Category:	SampleAddContact
Root Log Category:	BEA_CLARIFY_1_0
Session Path:	d:\beatesting
Message Bundle Base:	BEA_CLARIFY_1_0
Log Configuration File:	BEA_CLARIFY_1_0.xml

Reconfigure connection parameters for **SampleAddContact**

Events Add
Services Add
AddNewContact Edit Remove Service View Summary View Request Schema View Response Schema
Continue Save
Done
Local intranet

6. Click Continue.

   The Deploy Application View window opens.

   To deploy the application view, see “Deploying an Application View” on page 2-23.

Before Adding an Event to an Application View

Before using the BEA ClarifyCRM Adapter for event processing, some preliminary database maintenance must be performed, a process that must be carried out only once. Once done, you can add events to Application Views. The steps that follow describe how to use the provided programs and files to set up your ClarifyCRM database to work with the adapter for event processing.

Creating the Event Tables

The adapter polls an event table inside the ClarifyCRM database, which is populated with information about external updates, through database triggers applied to the tables used by the CBO objects supported by the adapter. This database must be created using the SQL script files provided with the BEA WebLogic Adapter for ClarifyCRM.

For ClarifyCRM Systems with a database stored in Oracle, the `WLI_CLARIFY_EVENTS.sql` will create the events database inside Oracle. Also, the `WLI_CLARIFY_EVENTS_SEQ.sql` will create a second table that keeps track of the sequences, to prevent the same event being processed twice. You can run these scripts inside the Oracle SQL. For more information on running Oracle SQL scripts, refer to your Oracle documentation).

For ClarifyCRM Systems with a database stored in Microsoft SQL Server, the `wli_clarify_events.sql` will create the events database inside Microsoft SQL Server. Also, the `wli_clarify_events_seq.sql` will create a second table that keeps track of the sequences to prevent the same event being processed twice. You can run these scripts inside the SQL Query Analyzer application. For more information on running MSSQL SQL scripts, refer to your MS SQL Server documentation).

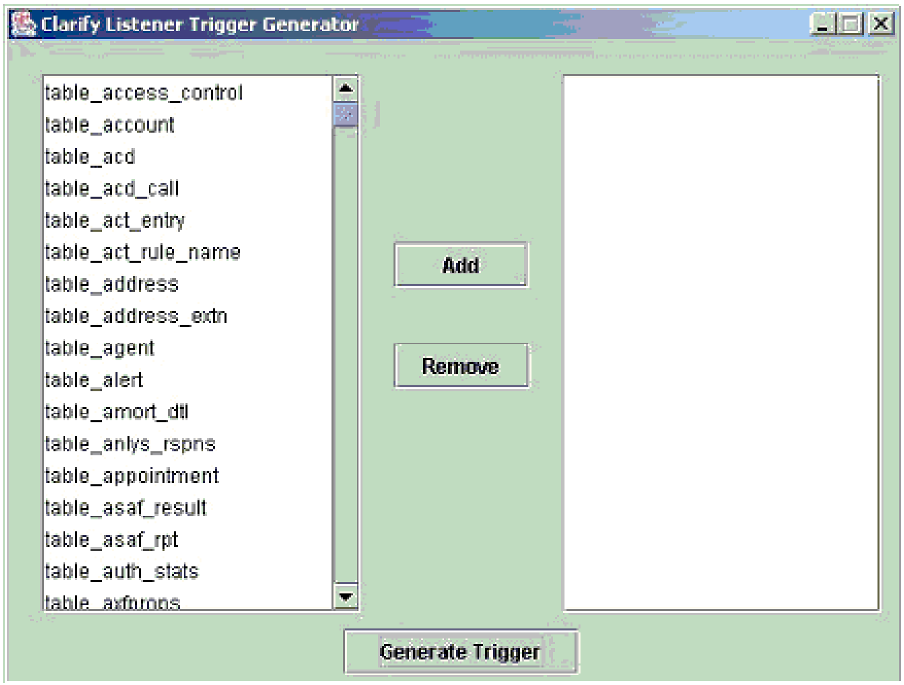
Applying DB Triggers

The ClarifyCRM trigger generator presents a list of all the ClarifyCRM tables that can be listened to. When used, the table will place Create, Delete, and Update triggers onto the tables you have selected. The trigger generator supports only Oracle 8i and Microsoft SQL Server 2000, both though the respective JDBC drivers, which must be downloaded from the RDBMS vendor's download site. For details on where to obtain these drivers, see the *BEA WebLogic Adapter for ClarifyCRM Installation and Configuration Guide.*

The ClarifyCRM Trigger Creation tool should be started using the provided batch files. The batch file you choose will depend on your ClarifyCRM installation. If you have ClarifyCRM installed on Oracle, run "Oracle_TriggerGen.bat"; if you use ClarifyCRM installed on MS SQL, run "MSSQL_TriggerGen.bat". You must modify the batch files to change the user name and password used for the logon and provide a DB Connection URL and database name for the ClarifyCRM database.

After launching the tool, a window similar to that shown in the following figure opens.

Figure 2-11 Trigger Creation Window



The left pane displays a list of tables that the SQL Listener will be able to listen to for events. To add triggers to a table or a set of tables:

1. Select the table(s) from the left pane and click Add.
2. Click Generate Triggers.

   You can remove a table by selecting it in the right pane and clicking Remove.

   Note: Because selecting Remove does not remove a table from the database, you must remove tables manually.

Adding an Event to an Application View

You can add an event to an application view to support the application's functions. (For information about creating an application view, see “Creating an Application View” on page 2-4.)

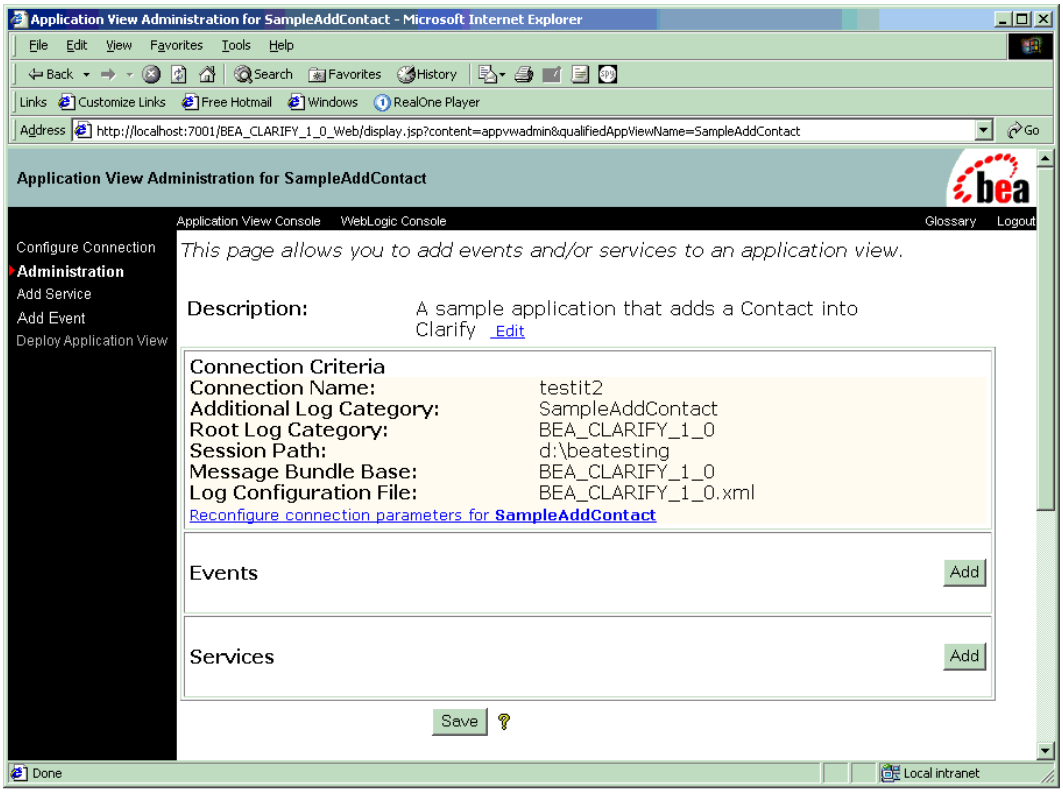
Note: If the application view is deployed, you must undeploy it before adding the event. For information about undeploying, see “Undeploying an Application View” on page 2-32.

To add an event:

1. Log on to the WebLogic Application View Console as described in “Creating an Application View” on page 2-4.
2. Select the folder in which this application view resides, and then select the application view.

   The Application View Administration window opens.

Figure 2-12 Application View Administration Window (Active Section)


Application View Administration for SampleAddContact - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address http://localhost:7001/BEA_CLARIFY_1_0_Web/display.jsp?content=appvwadmin&qualifiedAppViewName=SampleAddContact
Go
Application View Administration for SampleAddContact
Application View Console
WebLogic Console
Glossary
Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Add
Add
Save
Done
Local intranet

This page allows you to add events and/or services to an application view.

Description:	A sample application that adds a Contact into Clarify Edit

Connection Criteria

Connection Name:	testit2
Additional Log Category:	SampleAddContact
Root Log Category:	BEA_CLARIFY_1_0
Session Path:	d:\beatesting
Message Bundle Base:	BEA_CLARIFY_1_0
Log Configuration File:	BEA_CLARIFY_1_0.xml

Reconfigure connection parameters for **SampleAddContact**

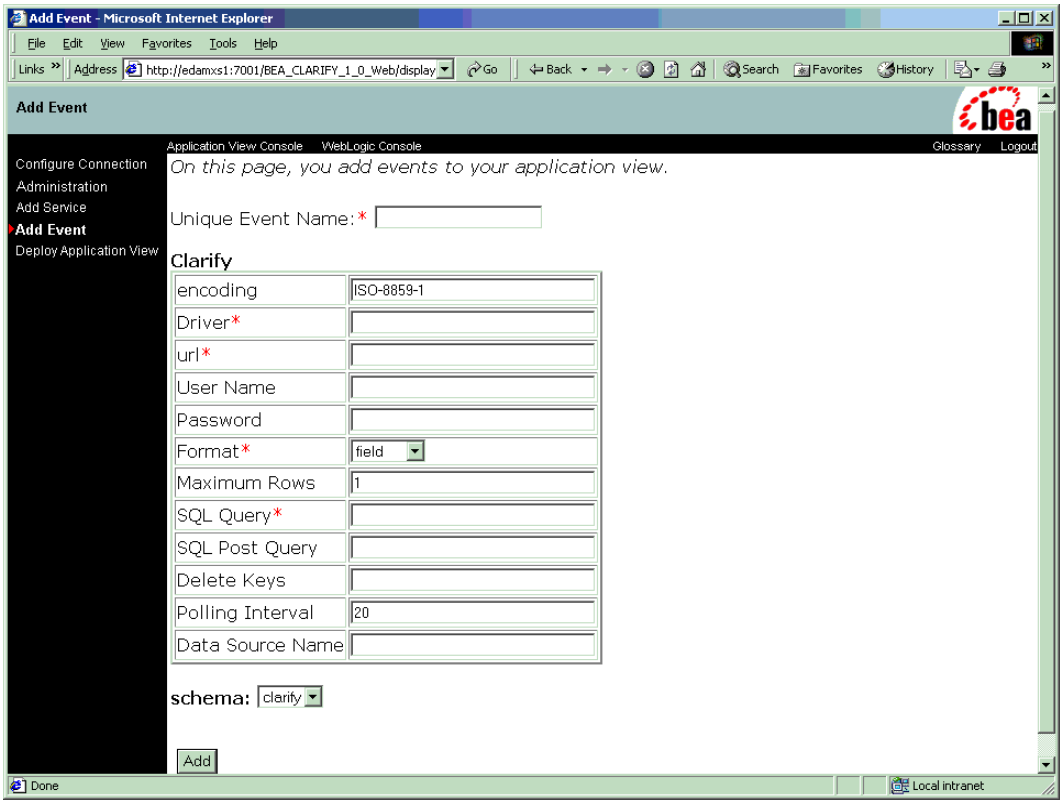
Events

Services

3. Select Add Event in the left pane, or click Add in the Events section.

   The Add Event window opens.

Figure 2-13 Add Event Window


Add Event - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Address http://edamxs1:7001/BEA_CLARIFY_1_0_Web/display Go Back Search Favorites History
Add Event
bea
Application View Console WebLogic Console Glossary Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Add
Done
Local intranet

On this page, you add events to your application view.

Unique Event Name:*

Clarify

encoding	ISO-8859-1
Driver*	
url*	
User Name	
Password	
Format*	field
Maximum Rows	1
SQL Query*	
SQL Post Query	
Delete Keys	
Polling Interval	20
Data Source Name	

schema: clarify

4. Specify the event's properties, which are described in the following table. Required values are indicated on the screen with an asterisk (*).

Table 2-2 Event Properties

Property	Description
Unique Event Name	The name of the event that you are adding. The name must be unique within the application view. Valid characters include a-z, A-Z, 0-9, and underscore (_).
encoding	Sets the character set encoding to be used (default value ISO-8859-1 -US and Western Europe).

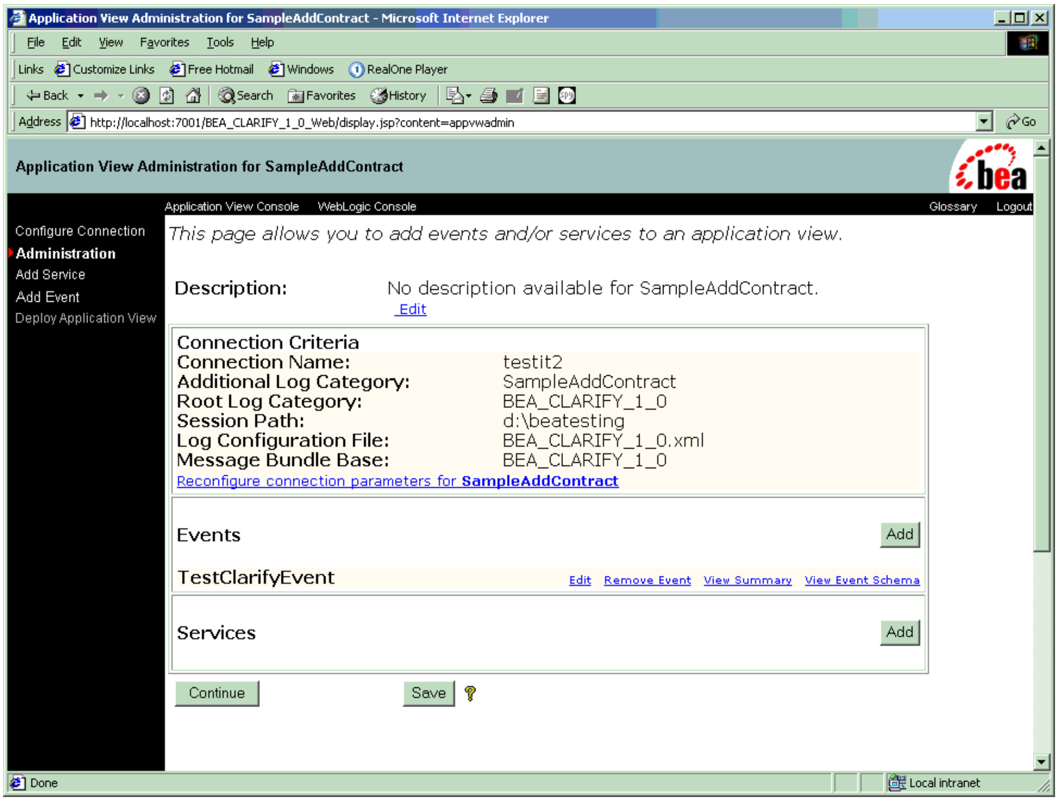
Table 2-2 Event Properties (Continued)

Property	Description
Driver	Is the name of the JDBC driver that the adapter should use when querying the ClarifyCRM event table. Note that ClarifyCRM supports storing data in both Microsoft SQL Server and Oracle databases, so be sure to specify the JDBC driver for the DBMS your ClarifyCRM environment uses.
url	URL for the JDBC driver to access the database.
User Name	User name for the JDBC driver to access the database.
Password	Database password for user.
Format	Style of XML document containing results of the SQL query. The value is Column or Field.
Maximium Rows	Maximum number of rows to include in each document.
SQL_Query	The query that the BEA WebLogic Adapter for ClarifyCRM will issue against the event table. By default, the value is `SELECT * FROM` *`event_tablename`*, but you can specify a different query so that the event captures the data you desire.
SQL Post Query	SQL statement issued after the query. If omitted, `delete <fields> from table where <field values> or <keys>` is used.
Delete Keys	Comma-separated list of keys to be used to build delete statement. If omitted, all fields are used. Do not use if a listener exit is provided. Case sensitive.
Polling Interval	Indicates how often, in seconds, the adapter should issue the SQL query. The higher this value, the longer the interval, and so the fewer system resources are used. The default value is 20 seconds.
Data Source Name	The name of the data source.
schema	The name of the schema that describes this event.

5. Click Add.

The Application View Administration window opens.

Figure 2-14 Event Added to Application View


Application View Administration for SampleAddContract - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address http://localhost:7001/BEA_CLARIFY_1_0_Web/display.jsp?content=appvwadmin
Go
bea
Application View Console WebLogic Console Glossary Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Edit
Reconfigure connection parameters for SampleAddContract
Add
Edit Remove Event View Summary View Event Schema
Add
Continue
Save
Done
Local intranet

Application View Administration for SampleAddContract

This page allows you to add events and/or services to an application view.

Description: No description available for SampleAddContract.

Connection Criteria	
Connection Name:	testit2
Additional Log Category:	SampleAddContract
Root Log Category:	BEA_CLARIFY_1_0
Session Path:	d:\beatesting
Log Configuration File:	BEA_CLARIFY_1_0.xml
Message Bundle Base:	BEA_CLARIFY_1_0

Events

TestClarifyEvent

Services

To deploy the application view, see “Deploying an Application View” on page 2-23.

Deploying an Application View

You can deploy an application view when you have added at least one service or event to it. You must deploy an application view before you can test its services and events and before you can use it with WebLogic Server.

Deploying an application view places relevant metadata about its services and events in a run-time metadata repository. Deployment makes the application view available to other WebLogic Server clients, enabling it to interact with business processes.

Deploying an Application View With a Service

To deploy an application view with a service:

1. If you are not already in the Application View Console's Administration window:

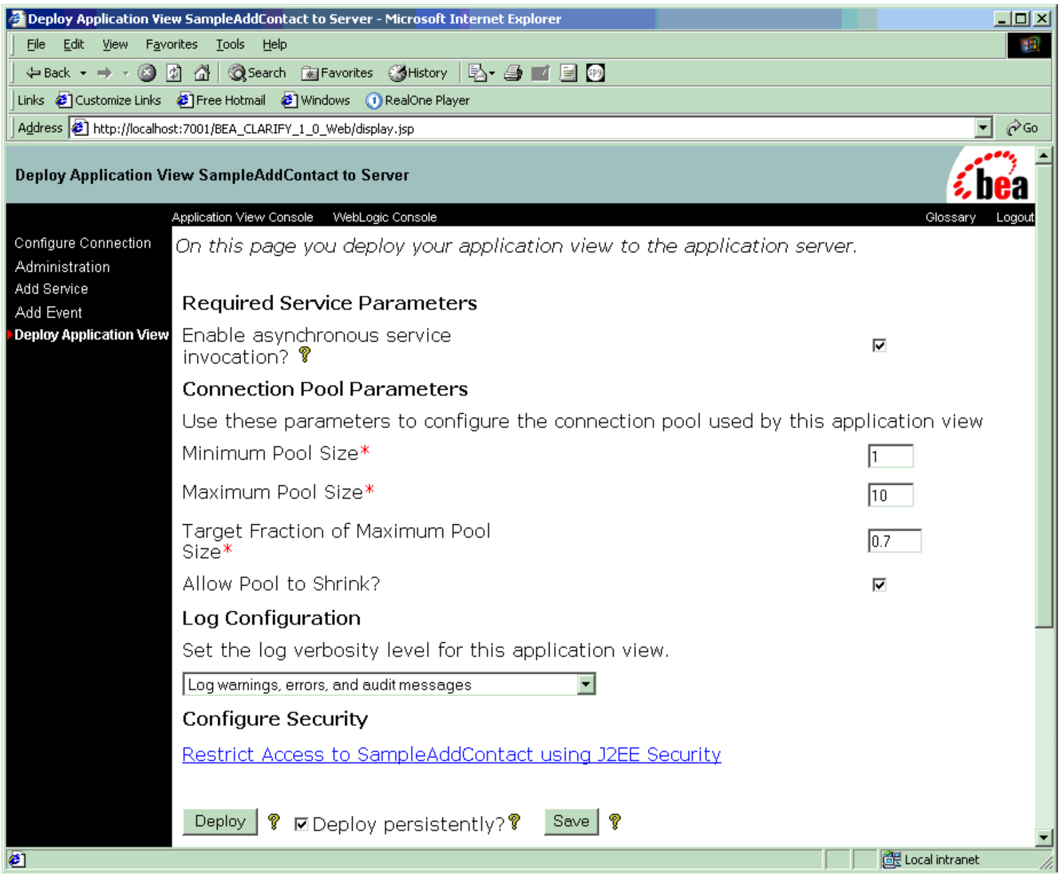
   a. Log on to the WebLogic Application View Console as described in "Creating an Application View" on page 2-4.

   b. Select the folder in which this application view resides, and then select the application view. The Administration window opens.

2. Click Continue in the lower left corner of the window.

   The Deploy Application View window opens.

Figure 2-15 Deploy Application View Window


Deploy Application View SampleAddContact to Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address http://localhost:7001/BEA_CLARIFY_1_0_Web/display.jsp
Go
bea
Application View Console WebLogic Console Glossary Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Deploy
Save
Local intranet

Deploy Application View SampleAddContact to Server

On this page you deploy your application view to the application server.

Required Service Parameters

Enable asynchronous service invocation? ☑

Connection Pool Parameters

Use these parameters to configure the connection pool used by this application view

Minimum Pool Size* 1

Maximum Pool Size* 10

Target Fraction of Maximum Pool Size* 0.7

Allow Pool to Shrink? ☑

Log Configuration

Set the log verbosity level for this application view.

Log warnings, errors, and audit messages

Configure Security

Restrict Access to SampleAddContact using J2EE Security

☑ Deploy persistently?

3. Update service parameters, connection pool parameters, log configuration, and security as necessary. Parameters that appear on the screen with an asterisk (*) are required.

Table 2-3 Service Deployment Parameters

Parameter	Description
Enable asynchronous service invocation	Enables you to run this service asynchronously.

Table 2-3 Service Deployment Parameters (Continued)

Parameter	Description
Minimum Pool Size	The minimum number of threads to ClarifyCRM.
Maximum Pool Size	The maximum number of threads to ClarifyCRM.
Target Fraction of Maximum Pool Size	The optimal thread level.
Allow Pool to Shrink	Enables the removal of threads that are no longer used.
Log verbosity level	The level of messages sent to the log.
Restrict Access using J2EE Security	Security parameter.
Deploy persistently	Persistence parameter.

For more information about these parameters, see "Defining an Application View" in "Using Application Integration":

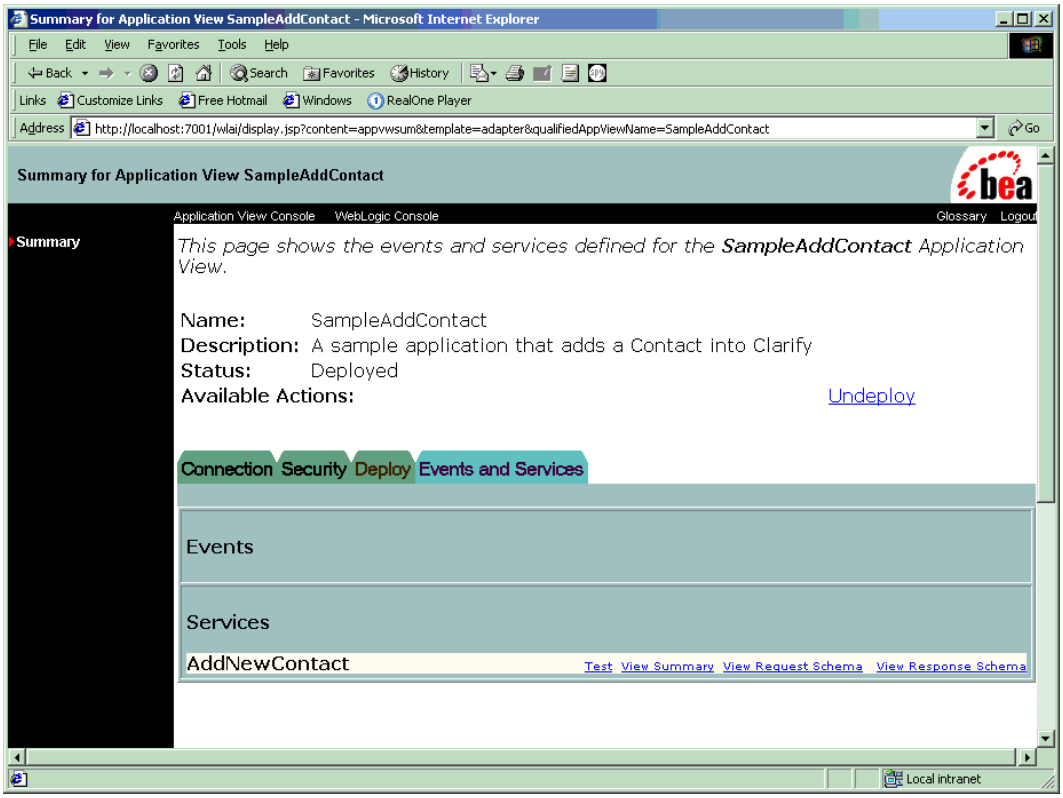
- For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/2usrdef.htm`
- For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/2usrdef.htm`

4. Click Deploy to deploy the application view.

   Note: To save the parameters without first deploying the application view, click Save.

   The Summary window for the application view opens.

Figure 2-16 A Deployed Application View With a Service


Summary for Application View SampleAddContact - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address http://localhost:7001/wlai/display.jsp?content=appvwsum&template=adapter&qualifiedAppViewName=SampleAddContact
Go
Application View Console WebLogic Console Glossary Logout
Summary
Connection Security Deploy Events and Services
Test View Summary View Request Schema View Response Schema
Local intranet

Summary for Application View SampleAddContact

This page shows the events and services defined for the ***SampleAddContact*** *Application View.*

Name:	SampleAddContact
Description:	A sample application that adds a Contact into Clarify
Status:	Deployed
Available Actions:	Undeploy

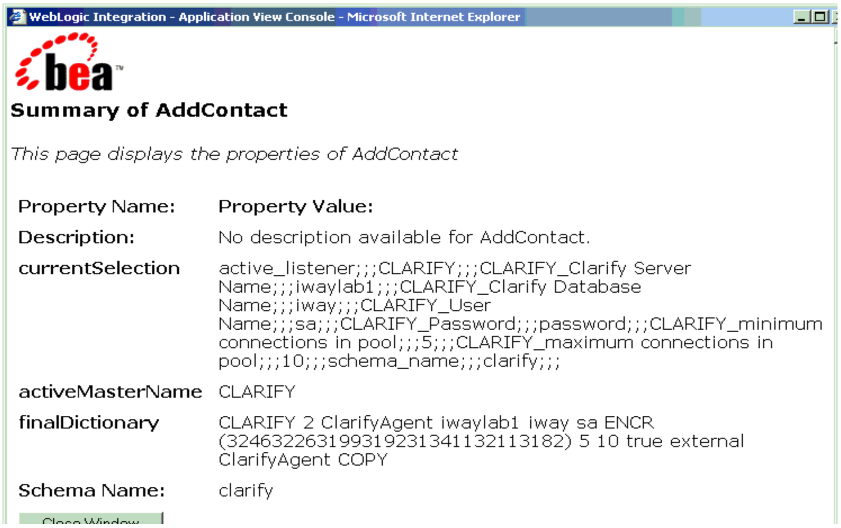
Events

Services

AddNewContact

5. To view a summary of the service as deployed, select the service and click View Summary.

   A summary of the service is displayed in a new window.



WebLogic Integration - Application View Console - Microsoft Internet Explorer
bea
Close Window

Summary of AddContact

This page displays the properties of AddContact

Property Name:	Property Value:
Description:	No description available for AddContact.
currentSelection	active_listener;;;CLARIFY;;;CLARIFY_Clarify Server Name;;;iwaylab1;;;CLARIFY_Clarify Database Name;;;iway;;;CLARIFY_User Name;;;sa;;;CLARIFY_Password;;;password;;;CLARIFY_minimum connections in pool;;;5;;;CLARIFY_maximum connections in pool;;;10;;;schema_name;;;clarify;;;
activeMasterName	CLARIFY
finalDictionary	CLARIFY 2 ClarifyAgent iwaylab1 iway sa ENCR (324632263199319231341132113182) 5 10 true external ClarifyAgent COPY
Schema Name:	clarify

You are now ready to test your service, as described in "Testing a Deployed Service" on page 2-36.

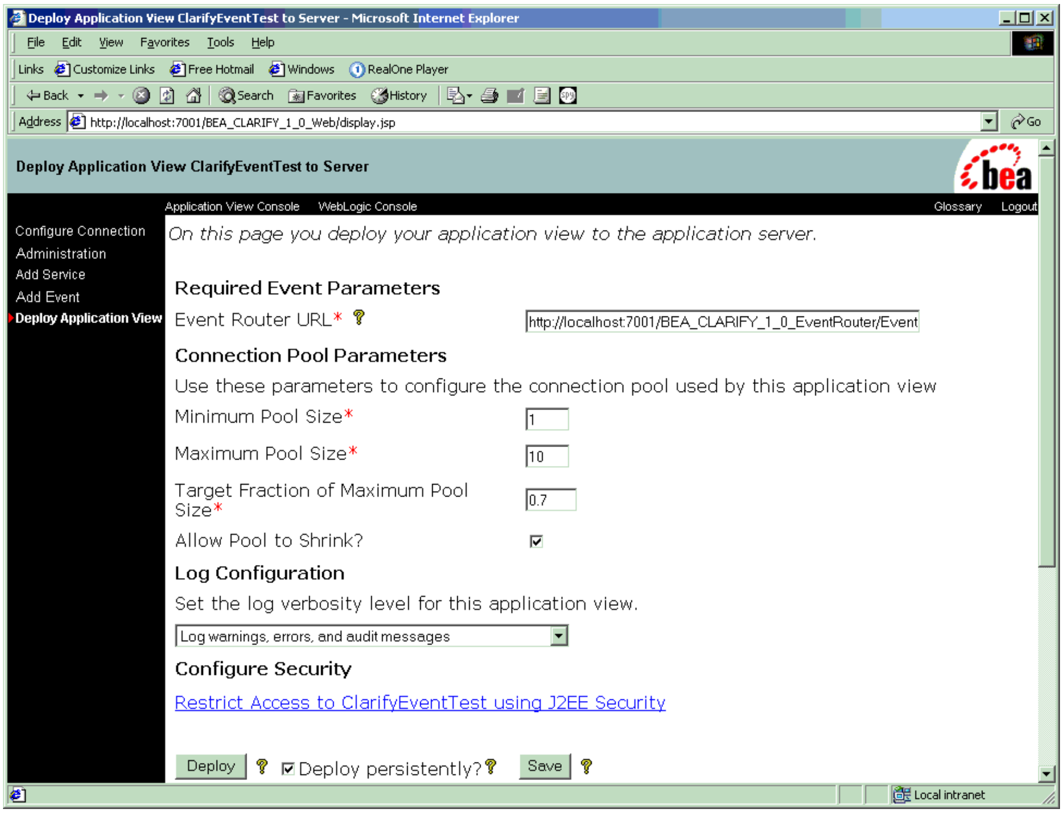
Deploying an Application View With an Event

To deploy an application view with an event:

1. If you are not already in the Application View Console's Administration window:
   a. Log on to the WebLogic Application View Console as described in "Creating an Application View" on page 2-4.
   b. Select the folder in which this application view resides, and then select the application view. The Administration window opens.
2. Click Continue in the lower left corner of the window.

   The Deploy Application View window opens.

Figure 2-17 Deploy Application View Window


Deploy Application View ClarifyEventTest to Server - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address http://localhost:7001/BEA_CLARIFY_1_0_Web/display.jsp
Go
bea
Application View Console WebLogic Console Glossary Logout
Configure Connection
Administration
Add Service
Add Event
Deploy Application View
Deploy
Save
Local intranet

Deploy Application View ClarifyEventTest to Server

On this page you deploy your application view to the application server.

Required Event Parameters

Event Router URL*	http://localhost:7001/BEA_CLARIFY_1_0_EventRouter/Event

Connection Pool Parameters

Use these parameters to configure the connection pool used by this application view

Minimum Pool Size*	1
Maximum Pool Size*	10
Target Fraction of Maximum Pool Size*	0.7
Allow Pool to Shrink?	✓

Log Configuration

Set the log verbosity level for this application view.

Log warnings, errors, and audit messages

Configure Security

Restrict Access to ClarifyEventTest using J2EE Security

✓ Deploy persistently?

3. Update event parameters, connection pool parameters, log configuration, and security as necessary. Parameters that appear on the screen with an asterisk (*) are required.

Table 2-4 Event Deployment Parameters

Parameter	Description
Event Router URL	The location of the router for this event.
Minimum Pool Size	The minimum number of threads to ClarifyCRM.
Maximum Pool Size	The maximum number of threads to ClarifyCRM.

Table 2-4 Event Deployment Parameters (Continued)

Parameter	Description
Target Fraction of Maximum Pool Size	The optimal thread level.
Allow Pool to Shrink	Enables the removal of threads that are no longer used.
Log verbosity level	The level of messages sent to the log.
Restrict Access using J2EE Security	Security parameter.
Deploy persistently	Persistence parameter.

For more information about these parameters, see "Defining an Application View" in "Using Application Integration":

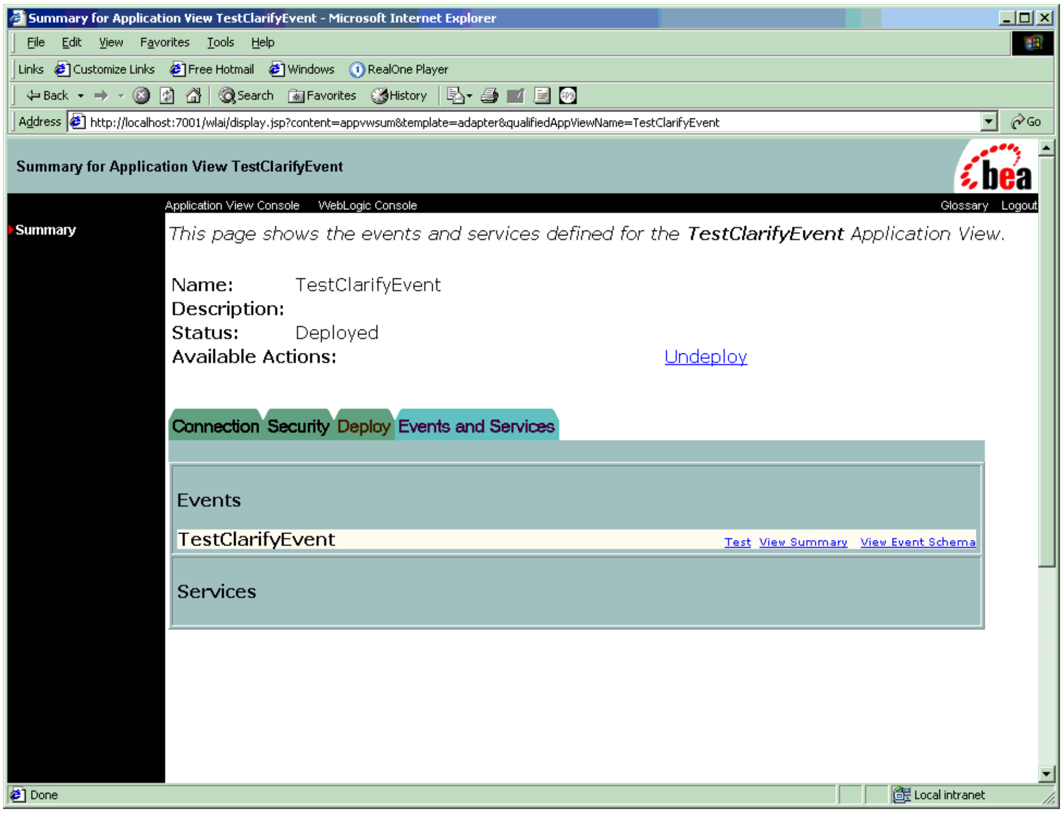
- For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/2usrdef.htm`
- For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/2usrdef.htm`

4. Click Deploy to deploy the application view.

   Note: To save the parameters without first deploying the application view, click Save.

   The Summary window for the application view opens.

Figure 2-18 A Deployed Application View With an Event


Summary for Application View TestClarifyEvent - Microsoft Internet Explorer

Address: http://localhost:7001/wlai/display.jsp?content=appvwsum&template=adapter&qualifiedAppViewName=TestClarifyEvent

Summary for Application View TestClarifyEvent

This page shows the events and services defined for the **TestClarifyEvent** Application View.

Name: TestClarifyEvent
Description:
Status: Deployed
Available Actions: Undeploy

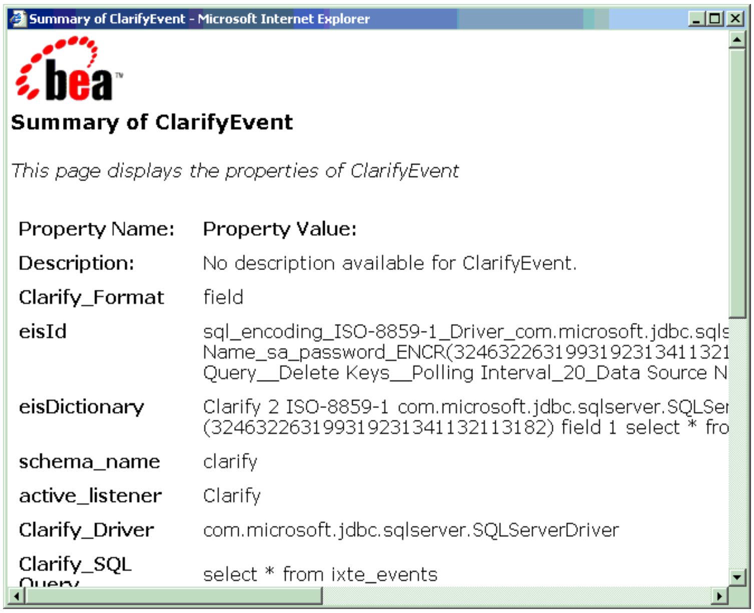
Events
TestClarifyEvent — Test | View Summary | View Event Schema

Services

File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History Go
bea
Application View Console WebLogic Console Glossary Logout
Summary
Connection Security Deploy Events and Services
Done Local intranet

5. To view a summary of the event as deployed, select the event and click View Summary.

   A summary of the event is displayed in a new window.



Summary of ClarifyEvent

This page displays the properties of ClarifyEvent

Property Name:	Property Value:
Description:	No description available for ClarifyEvent.
Clarify_Format	field
eisId	sql_encoding_ISO-8859-1_Driver_com.microsoft.jdbc.sqls Name_sa_password_ENCR(3246322631993192313411321 Query__Delete Keys__Polling Interval_20_Data Source N
eisDictionary	Clarify 2 ISO-8859-1 com.microsoft.jdbc.sqlserver.SQLSe (324632263199319231341132113182) field 1 select * fro
schema_name	clarify
active_listener	Clarify
Clarify_Driver	com.microsoft.jdbc.sqlserver.SQLServerDriver
Clarify_SQL Query	select * from ixte_events

You are now ready to test your event, as described in "Testing a Deployed Event" on page 2-41.

Undeploying an Application View

Once an application view is deployed, it cannot be modified. In order to modify it (for example, to add another service or event) you must first undeploy it.

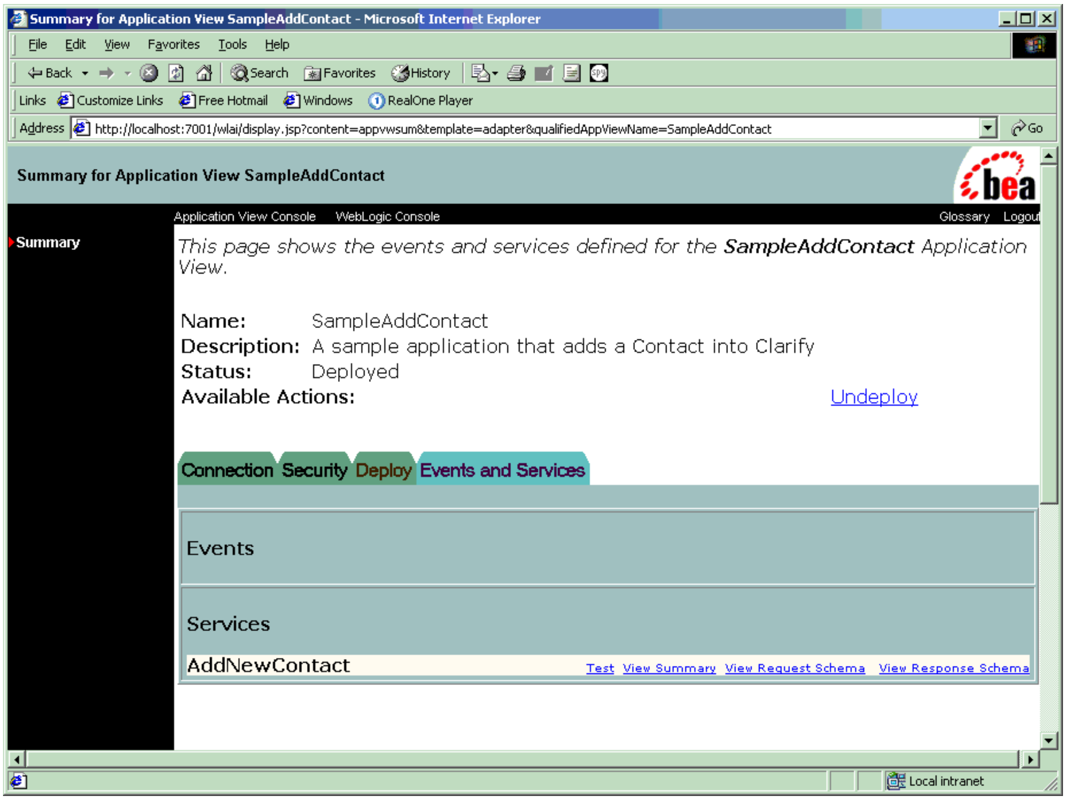
To undeploy an application view:

1. If you are not already in the Application View Console's Summary window for the application view you want to undeploy:

   a. Log on to the WebLogic Application View Console as described in "Creating an Application View" on page 2-4.

   b. Select the folder in which this application view resides, and then select the application view. The Summary window opens.

Figure 2-19 Undeploying an Application View


Summary for Application View SampleAddContact - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address http://localhost:7001/wlai/display.jsp?content=appvwsum&template=adapter&qualifiedAppViewName=SampleAddContact
Go
bea
Application View Console WebLogic Console
Glossary Logout
Summary
Connection Security Deploy Events and Services
Test View Summary View Request Schema View Response Schema
Local intranet

Summary for Application View SampleAddContact

This page shows the events and services defined for the ***SampleAddContact*** *Application View.*

Name: SampleAddContact
Description: A sample application that adds a Contact into Clarify
Status: Deployed
Available Actions: Undeploy

Events

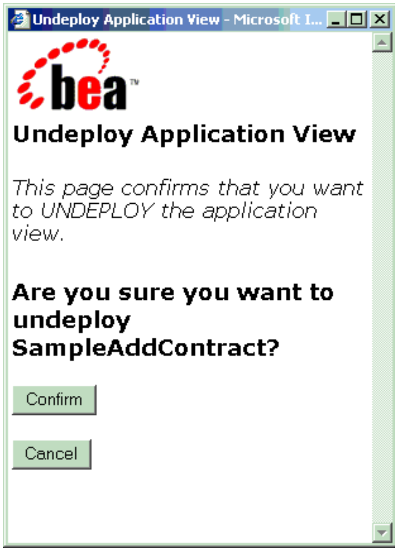
Services

AddNewContact

2. Click Undeploy.

A confirmation window asks if you are sure you want to undeploy the application view.

Figure 2-20 Undeployment Confirmation



Undeploy Application View - Microsoft I...

Undeploy Application View

This page confirms that you want to UNDEPLOY the application view.

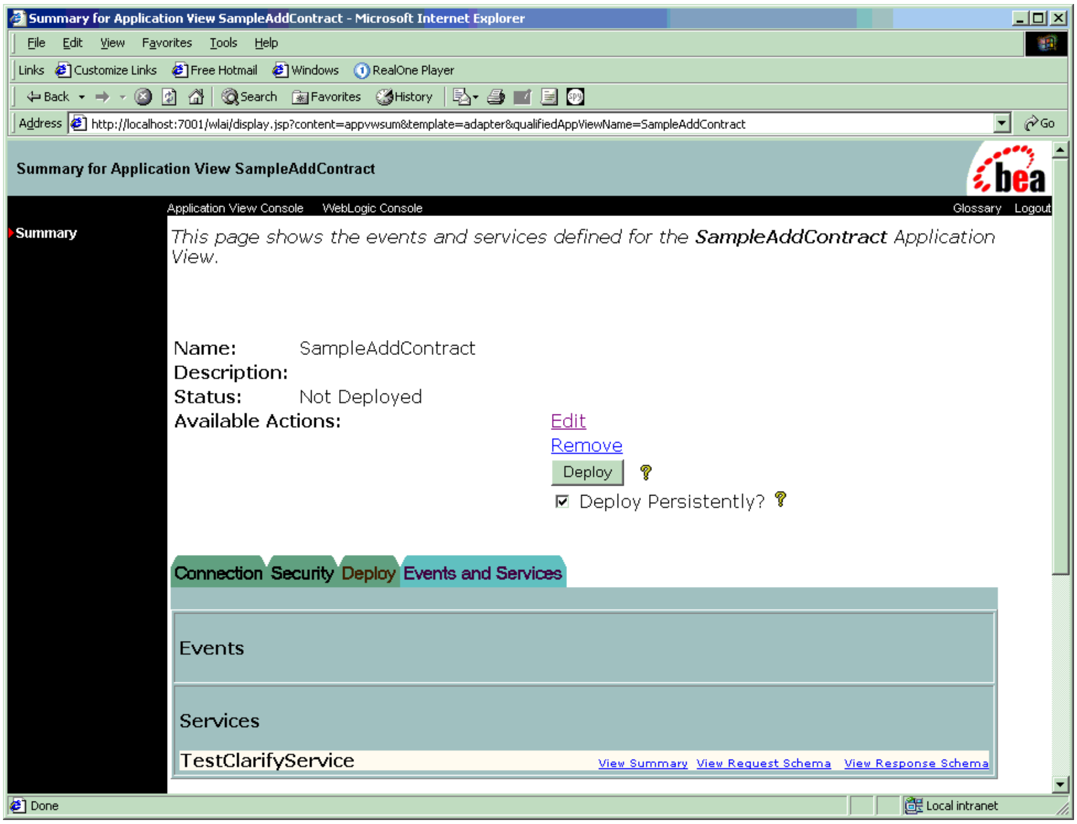
Are you sure you want to undeploy SampleAddContract?

3. Click Confirm to undeploy the application view.

For more information see "Optional Step: Undeploying an Application View" in "Defining an Application View" in *Using Application Integration*:

- For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/2usrdef.htm`
- For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/2usrdef.htm`

Figure 2-21 Undeployed Application View



Summary for Application View SampleAddContract - Microsoft Internet Explorer

Address: http://localhost:7001/wlai/display.jsp?content=appvwsum&template=adapter&qualifiedAppViewName=SampleAddContract

Summary for Application View SampleAddContract

*This page shows the events and services defined for the **SampleAddContract** Application View.*

Name:	SampleAddContract
Description:	
Status:	Not Deployed
Available Actions:	Edit Remove Deploy ☑ Deploy Persistently?

Events

Services

TestClarifyService — View Summary | View Request Schema | View Response Schema

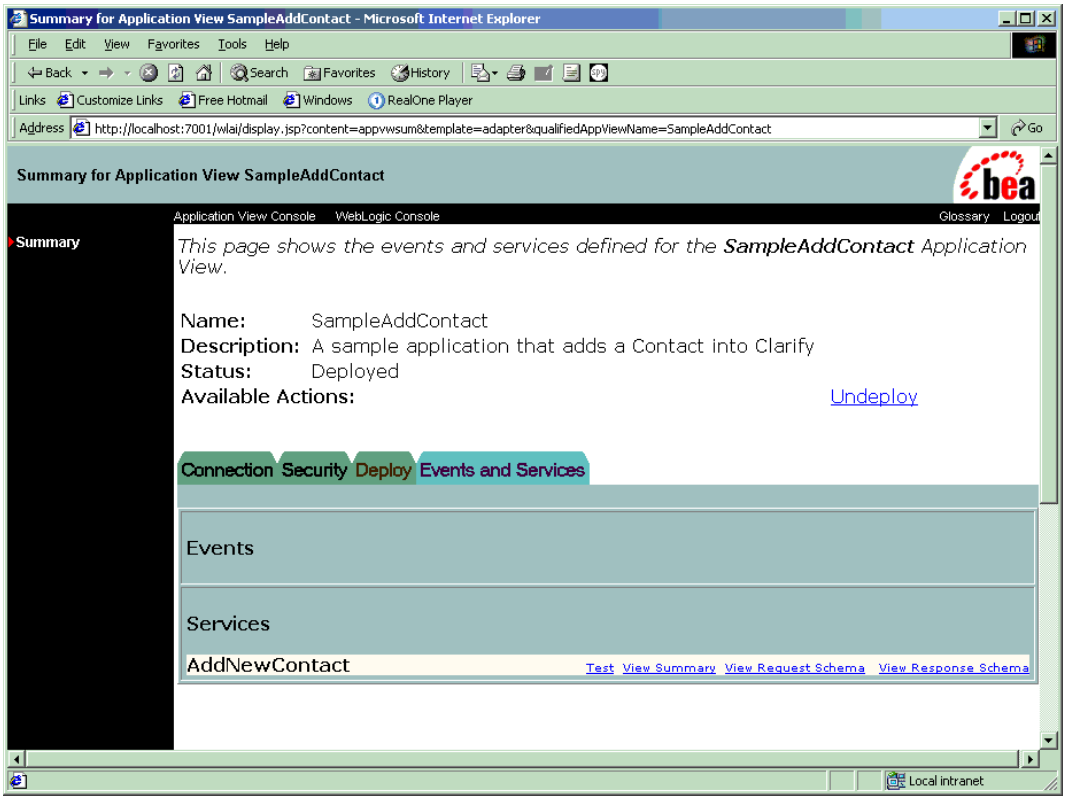
Testing a Deployed Service

After you create and deploy an application view that contains a service, you can test the application view service to evaluate whether it interacts properly with the BEA WebLogic Adapter for ClarifyCRM.

To test a deployed service:

1. If you are not already in the Application View Console's Summary window for the application view service you want to test:
   a. Log on to the WebLogic Application View Console as described in "Creating an Application View" on page 2-4.
   b. Select the folder in which this application view resides, and then select the application view. The Summary window opens.

Figure 2-22 Summary Window for a Service to Be Tested


Summary for Application View SampleAddContact - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Links Customize Links Free Hotmail Windows RealOne Player
Address http://localhost:7001/wlai/display.jsp?content=appvwsum&template=adapter&qualifiedAppViewName=SampleAddContact
Go
Application View Console WebLogic Console Glossary Logout
Summary
Connection Security Deploy Events and Services
Test View Summary View Request Schema View Response Schema
Local intranet

Summary for Application View SampleAddContact

This page shows the events and services defined for the ***SampleAddContact*** *Application View.*

Name:	SampleAddContact
Description:	A sample application that adds a Contact into Clarify
Status:	Deployed
Available Actions:	Undeploy

Events

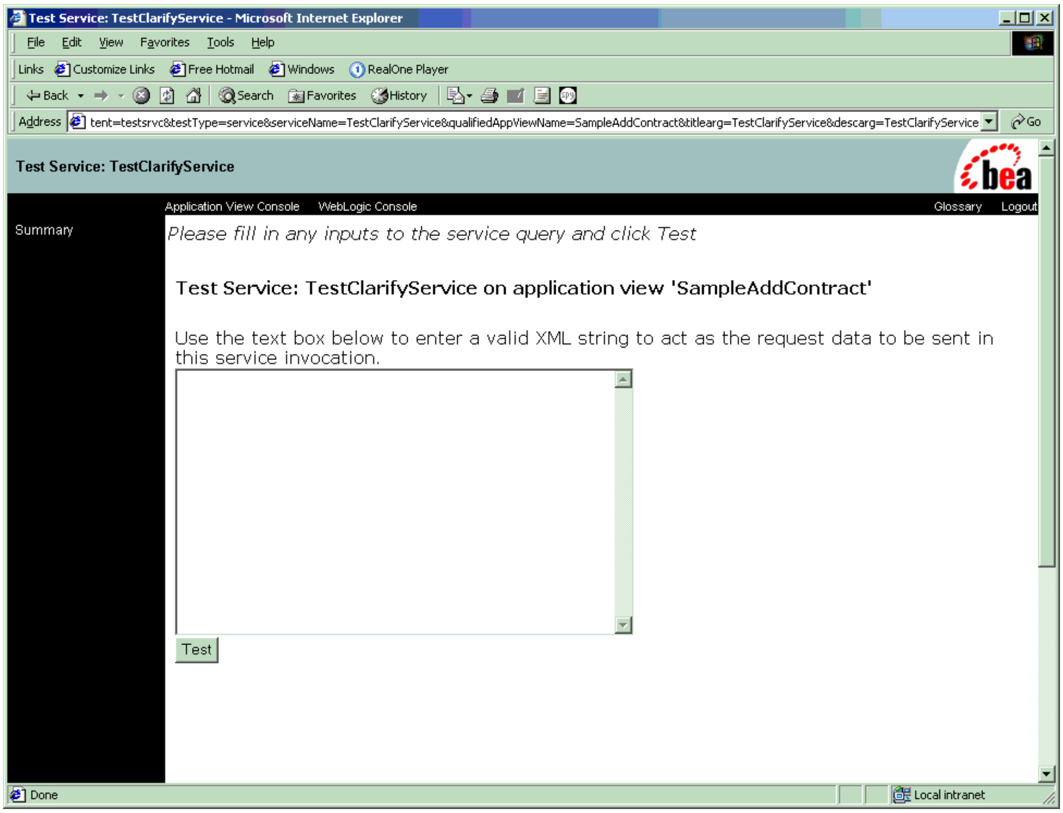
Services

AddNewContact

2. For the service you want to test, click Test.

   The Test Service window opens.

Figure 2-23 Test Service Window


Test Service: TestClarifyService - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address tent=testsrvc&testType=service&serviceName=TestClarifyService&qualifiedAppViewName=SampleAddContract&titlearg=TestClarifyService&descarg=TestClarifyService
Go
Application View Console WebLogic Console Glossary Logout
Summary
Test
Done
Local intranet

Test Service: TestClarifyService

Please fill in any inputs to the service query and click Test

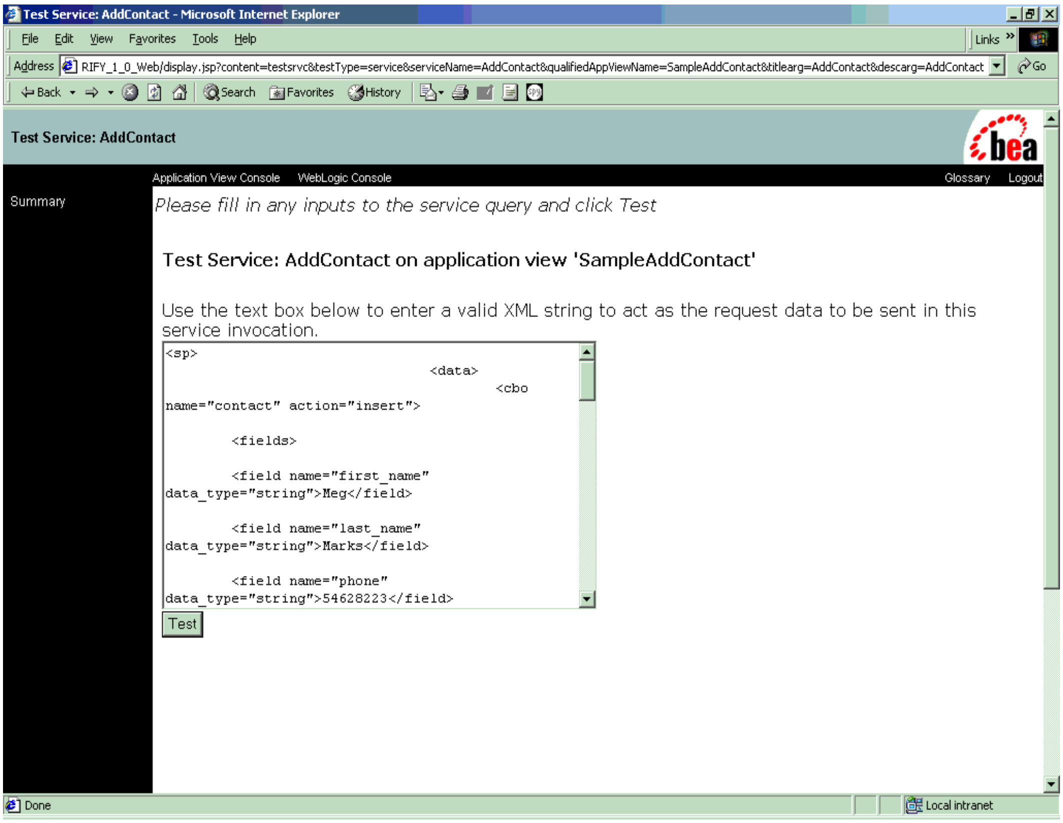
Test Service: TestClarifyService on application view 'SampleAddContract'

Use the text box below to enter a valid XML string to act as the request data to be sent in this service invocation.

3. Enter the XML that invokes the service request.

   In the following figure, the request that has been entered adds a new contact to the ClarifyCRM database.

Figure 2-24 Entering Request XML


Test Service: AddContact - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address RIFY_1_0_Web/display.jsp?content=testsrvc&testType=service&serviceName=AddContact&qualifiedAppViewName=SampleAddContact&titlearg=AddContact&descarg=AddContact
Back Search Favorites History
Test Service: AddContact
Application View Console WebLogic Console Glossary Logout
Summary
Test
Done
Local intranet

Please fill in any inputs to the service query and click Test

Test Service: AddContact on application view 'SampleAddContact'

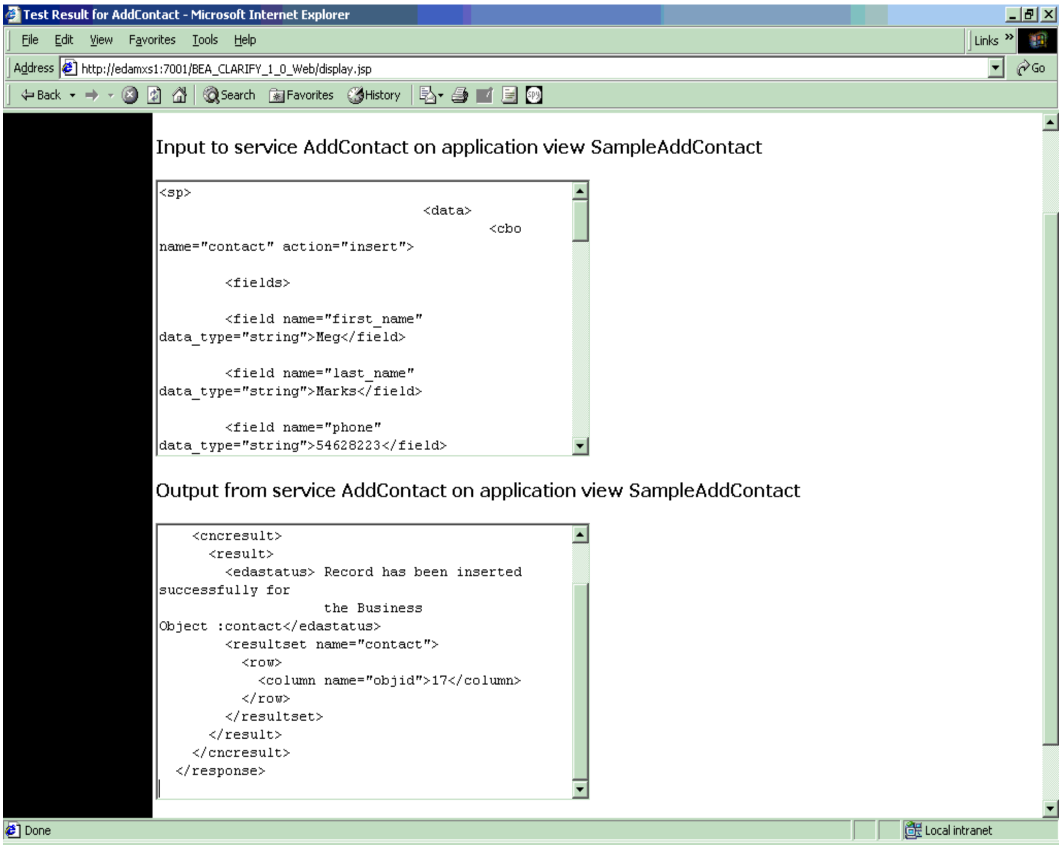
Use the text box below to enter a valid XML string to act as the request data to be sent in this service invocation.

```
<sp>
                                <data>
                                        <cbo
name="contact" action="insert">

        <fields>

        <field name="first_name"
data_type="string">Meg</field>

        <field name="last_name"
data_type="string">Marks</field>

        <field name="phone"
data_type="string">54628223</field>
```

4. Click Test to test the service.

The request and response documents are then displayed.

Figure 2-25 Test Service Results


Test Result for AddContact - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links
Address http://edamxs1:7001/BEA_CLARIFY_1_0_Web/display.jsp
Go
Back
Search
Favorites
History
Done
Local intranet

Input to service AddContact on application view SampleAddContact

```
<sp>
                                <data>
                                        <cbo
name="contact" action="insert">

        <fields>

        <field name="first_name"
data_type="string">Meg</field>

        <field name="last_name"
data_type="string">Marks</field>

        <field name="phone"
data_type="string">54628223</field>
```

Output from service AddContact on application view SampleAddContact

```
    <cncresult>
      <result>
        <edastatus> Record has been inserted
successfully for
                    the Business
Object :contact</edastatus>
        <resultset name="contact">
          <row>
            <column name="objid">17</column>
          </row>
        </resultset>
      </result>
    </cncresult>
  </response>
```

If the test fails, a response document displays the appropriate error messages. You can correct the error and resubmit the request.

If the test succeeds, the response document that corresponds to the request is displayed in the output field.

You have successfully deployed and tested the application view service.

If you wish, you can write custom code to create a workflow. For more information, see “Using Application Views in the Studio” in *Using Application Integration*:

- For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/3usruse.htm`
- For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/3usruse.htm`

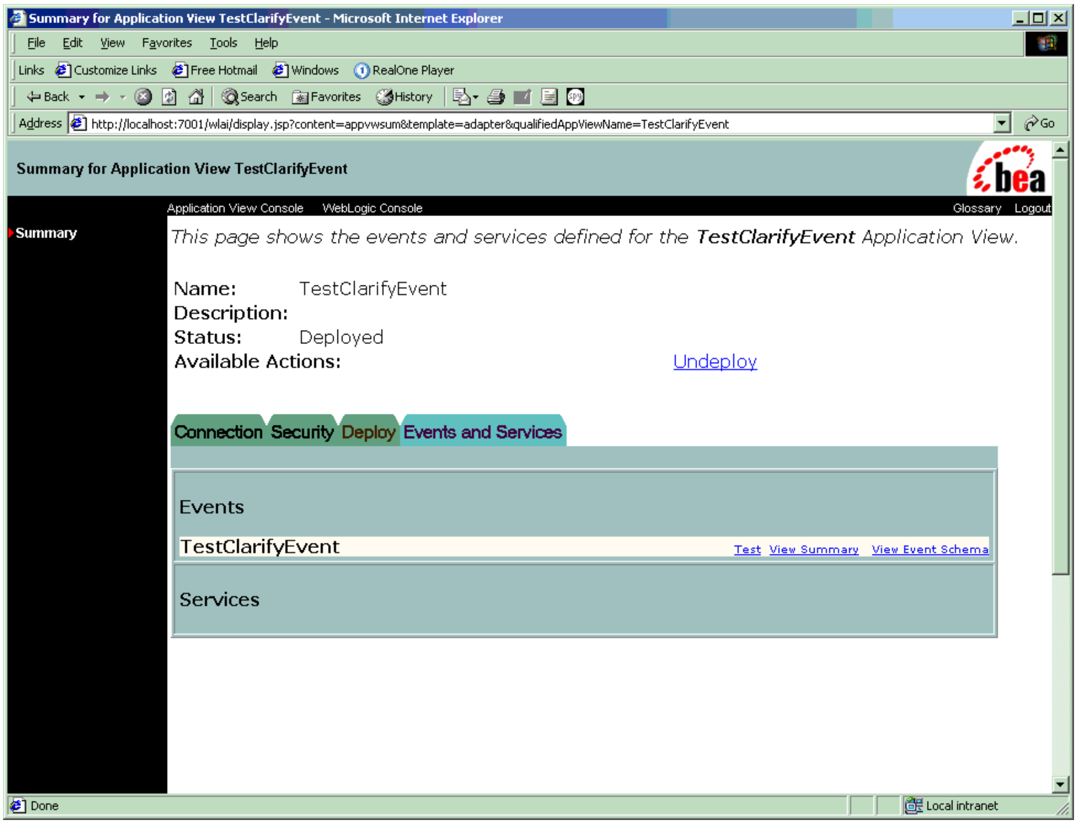
Testing a Deployed Event

After you create and deploy an application view that contains an event, you can test the application view event to evaluate whether it interacts properly with the BEA WebLogic Adapter for ClarifyCRM.

To test a deployed event:

1. If you are not already in the Application View Console’s Summary window for the application view event you want to test:
   a. Log on to the WebLogic Application View Console as described in “Creating an Application View” on page 2-4.
   b. Select the folder in which this application view resides, and then select the application view. The Summary window opens.

Figure 2-26 Summary Window for an Event to Be Tested


Summary for Application View TestClarifyEvent - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address http://localhost:7001/wlai/display.jsp?content=appvwsum&template=adapter&qualifiedAppViewName=TestClarifyEvent
Go
Application View Console WebLogic Console
Glossary Logout
Summary
Connection Security Deploy Events and Services
Test View Summary View Event Schema
Done
Local intranet

Summary for Application View TestClarifyEvent

This page shows the events and services defined for the ***TestClarifyEvent*** *Application View.*

Name: TestClarifyEvent
Description:
Status: Deployed
Available Actions: Undeploy

Events

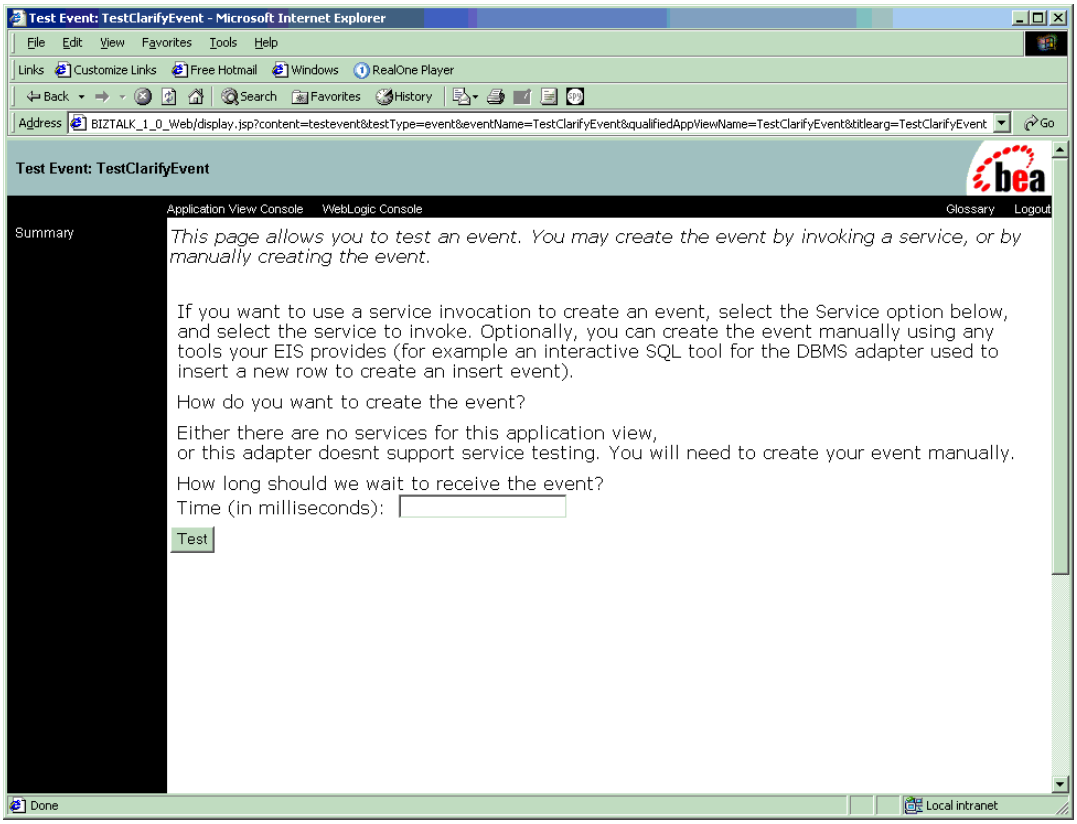
TestClarifyEvent

Services

2. For the event you want to test, click Test.

   The Test Event window opens.

Figure 2-27 Test Event Window


Test Event: TestClarifyEvent - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address
Go
Application View Console WebLogic Console
Glossary Logout
Summary
Done
Local intranet

BIZTALK_1_0_Web/display.jsp?content=testevent&testType=event&eventName=TestClarifyEvent&qualifiedAppViewName=TestClarifyEvent&titlearg=TestClarifyEvent

Test Event: TestClarifyEvent

This page allows you to test an event. You may create the event by invoking a service, or by manually creating the event.

If you want to use a service invocation to create an event, select the Service option below, and select the service to invoke. Optionally, you can create the event manually using any tools your EIS provides (for example an interactive SQL tool for the DBMS adapter used to insert a new row to create an insert event).

How do you want to create the event?

Either there are no services for this application view, or this adapter doesnt support service testing. You will need to create your event manually.

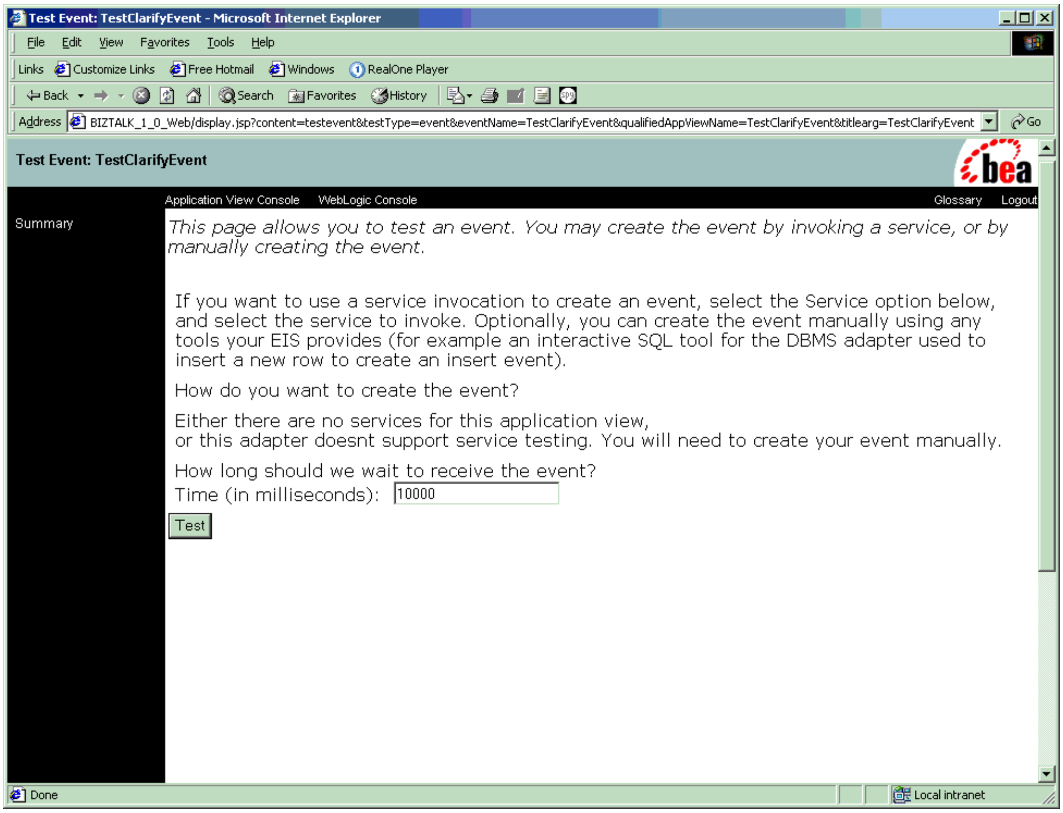
How long should we wait to receive the event?

Time (in milliseconds):

Test

3. Enter a time, in milliseconds, that indicates how long to wait to receive the event. Make sure to specify sufficient time to initiate the event.

Figure 2-28 Testing an Event


Test Event: TestClarifyEvent - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links Customize Links Free Hotmail Windows RealOne Player
Back Search Favorites History
Address BIZTALK_1_0_Web/display.jsp?content=testevent&testType=event&eventName=TestClarifyEvent&qualifiedAppViewName=TestClarifyEvent&titlearg=TestClarifyEvent
Go
Application View Console
WebLogic Console
Glossary
Logout
Summary
Test
Done
Local intranet

Test Event: TestClarifyEvent

This page allows you to test an event. You may create the event by invoking a service, or by manually creating the event.

If you want to use a service invocation to create an event, select the Service option below, and select the service to invoke. Optionally, you can create the event manually using any tools your EIS provides (for example an interactive SQL tool for the DBMS adapter used to insert a new row to create an insert event).

How do you want to create the event?

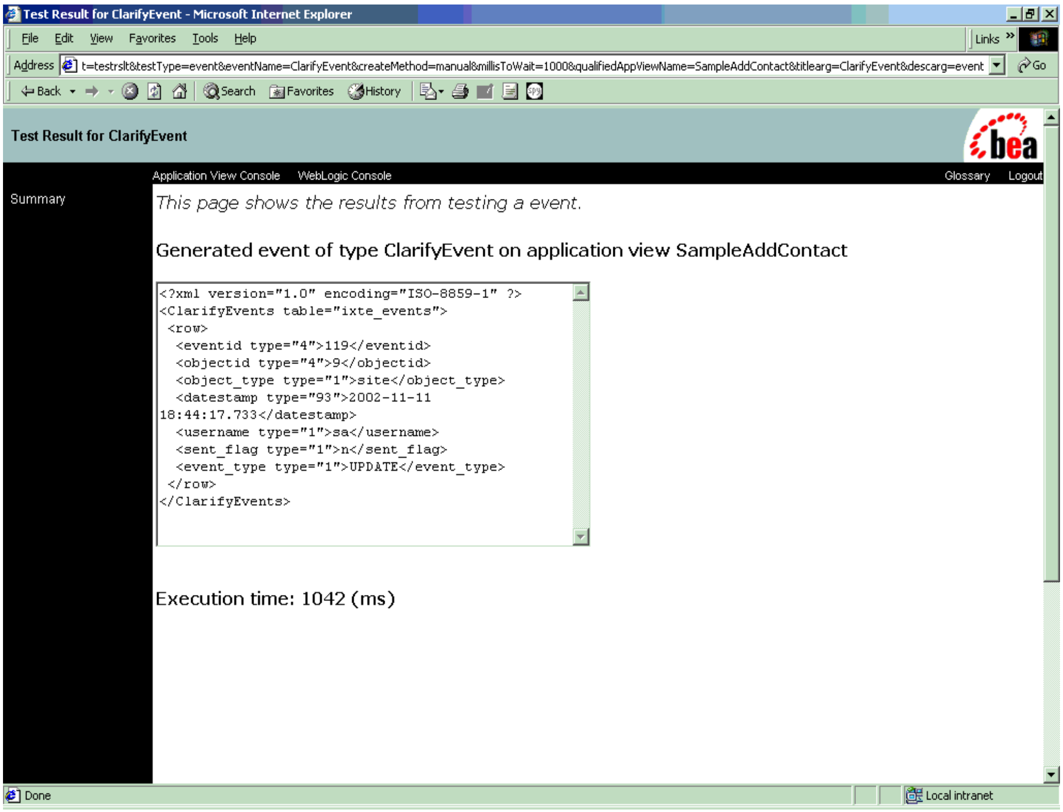
Either there are no services for this application view,
or this adapter doesnt support service testing. You will need to create your event manually.

How long should we wait to receive the event?
Time (in milliseconds): 10000

4. Click Test to test the event.

   When the event occurs, the event test result document is displayed.

Figure 2-29 Event Test Results


Test Result for ClarifyEvent - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links
Address t=testrslt&testType=event&eventName=ClarifyEvent&createMethod=manual&millisToWait=1000&qualifiedAppViewName=SampleAddContact&titlearg=ClarifyEvent&descarg=event
Go
Back Search Favorites History
Test Result for ClarifyEvent
bea
Application View Console
WebLogic Console
Glossary
Logout
Summary
Done
Local intranet

This page shows the results from testing a event.

Generated event of type ClarifyEvent on application view SampleAddContact

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<ClarifyEvents table="ixte_events">
 <row>
  <eventid type="4">119</eventid>
  <objectid type="4">9</objectid>
  <object_type type="1">site</object_type>
  <datestamp type="93">2002-11-11
18:44:17.733</datestamp>
  <username type="1">sa</username>
  <sent_flag type="1">n</sent_flag>
  <event_type type="1">UPDATE</event_type>
 </row>
</ClarifyEvents>
```

Execution time: 1042 (ms)

If the test fails, a message is displayed indicating that the event timed out. You can correct the error and retest the event.

If the test succeeds, the event result document is displayed.

You have successfully deployed and tested the application view event.

If you wish, you can write custom code to create a workflow. For more information, see “Using Application Views in the Studio” in *Using Application Integration*:

- For WebLogic Integration 7.0, see `http://edocs.bea.com/wli/docs70/aiuser/3usruse.htm`
- For WebLogic Integration 2.1, see `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/3usruse.htm`

Using a Service or Event in a Workflow

This section contains specific examples of the use of a service and an event in a workflow. A workflow is the process that links an event in one Enterprise Integration System (EIS), such as a ClarifyCRM system, to the services in another EIS. You can build complex workflows that involve multiple events and services, transformations, and other business logic.

This section does not provide a comprehensive description of workflows. For more information, see “Using Application Views in the Studio” in *Using Application Integration*:

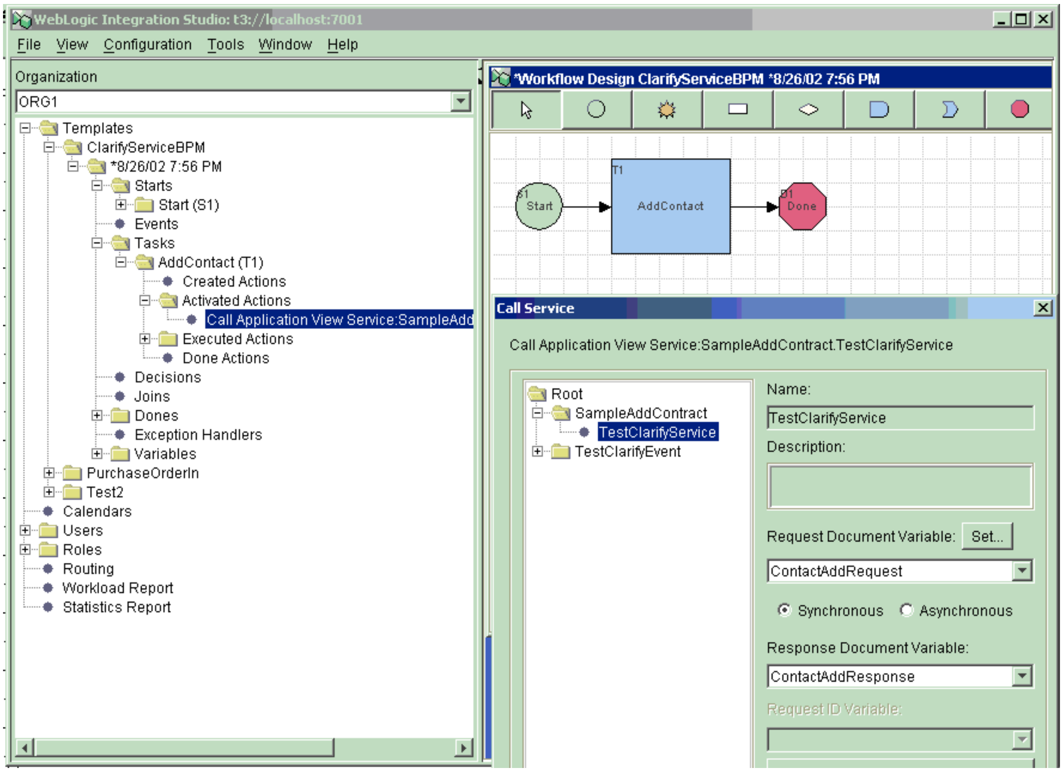
- For WebLogic Integration 7.0, see
  `http://edocs.bea.com/wli/docs70/aiuser/3usrruse.htm`
- For WebLogic Integration 2.1, see
  `http://edocs.bea.com/wlintegration/v2_1sp/aiuser/3usrruse.htm`

Example: Using a Service in a Workflow

In this example, a workflow is built using a static document called ContactAddRequest. This document is then used as input to the ClarifyCRM service defined in “Adding a Service to an Application View” on page 2-10. The resulting document is placed in the ContactResponse.

The following figures illustrate the service request in a workflow. In this workflow, the input document is defined as an action of the Start node. The AddContact action defines the use of the XML document passed to it from the Start node and sends it to the ClarifyCRM service.

Figure 2-30 Application View Service in a Workflow



Call Application View Service:SampleAddContract.TestClarifyService

Name: TestClarifyService

Description:

Request Document Variable: ContactAddRequest

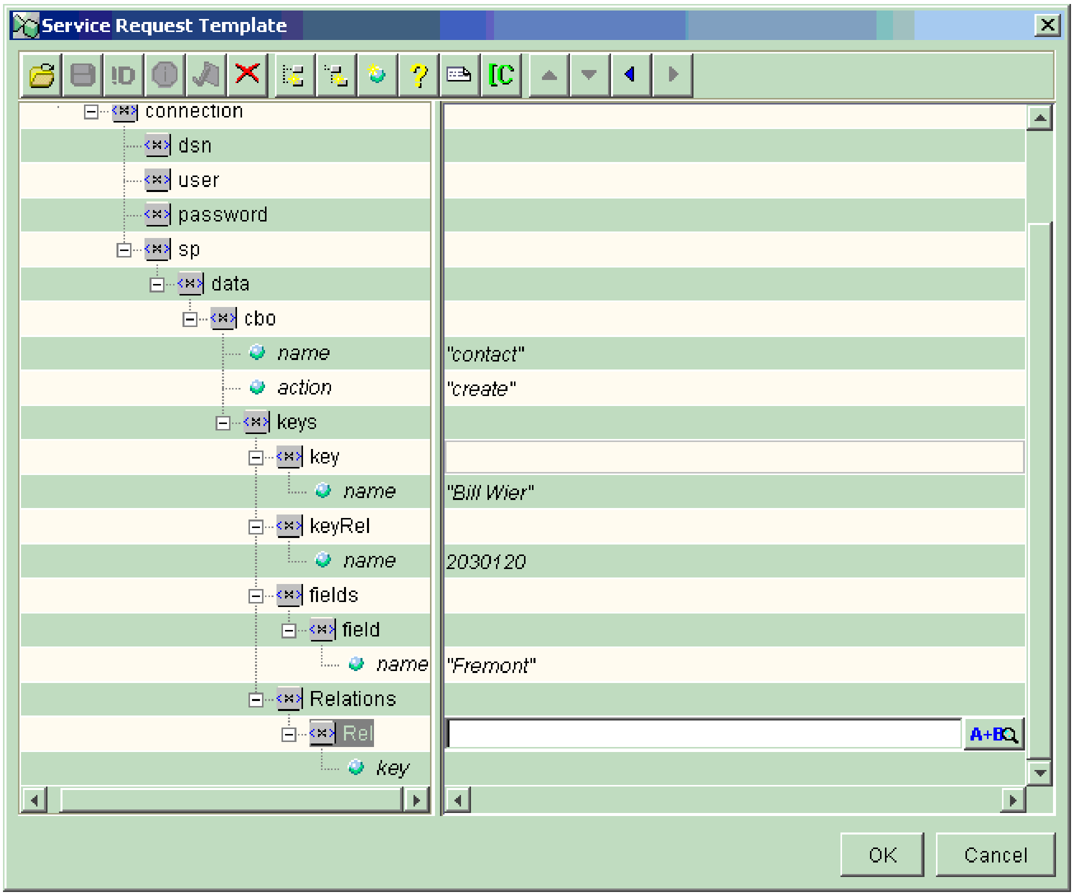
[✓ Synchronous] Asynchronous

Response Document Variable: ContactAddResponse

The ClarifyCRM Request

The following figure shows the static document created for the workflow. In this case, the document is a ContactRequest intended to be added into the ClarifyCRM application. In a real-world workflow, the document would typically be the result of an event in another EIS, with transformations and enrichment steps performed. For testing, however, this Service Request Template allows you to fill in test values.

Figure 2-31 Static Document

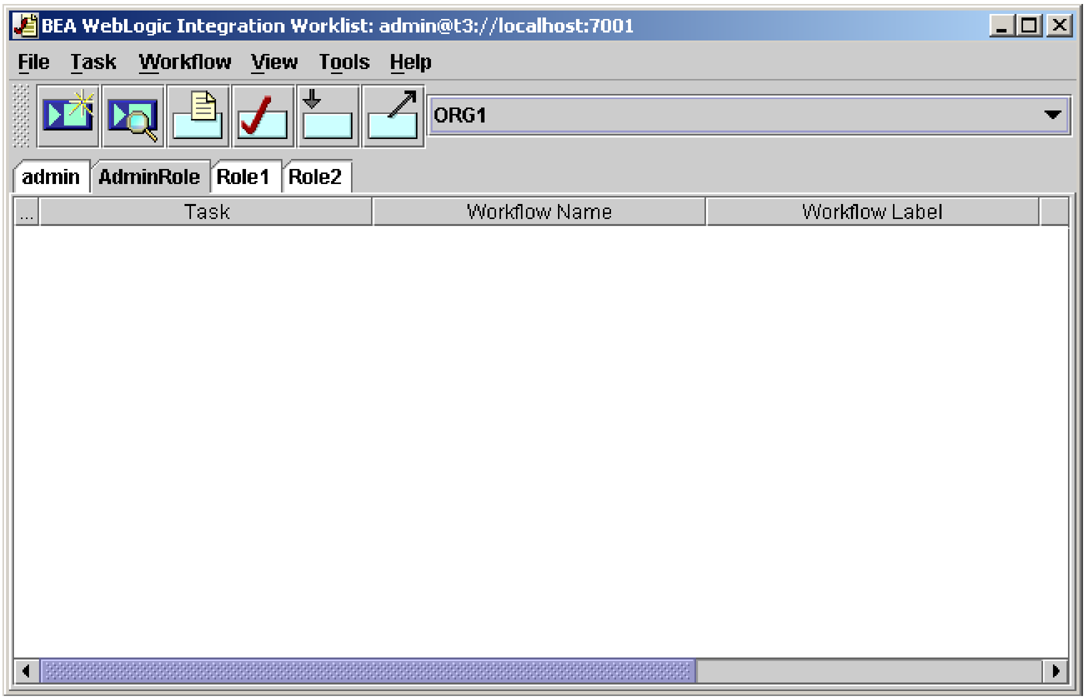

Service Request Template

connection	
dsn	
user	
password	
sp	
data	
cbo	
name	*"contact"*
action	*"create"*
keys	
key	
name	*"Bill Wier"*
keyRel	
name	*2030120*
fields	
field	
name	*"Fremont"*
Relations	
Rel	
key	

Start the Workflow

Once the workflow is built, Worklist is used to start the process manually. Worklist is required since the input document is static and not the result of an event from another EIS.

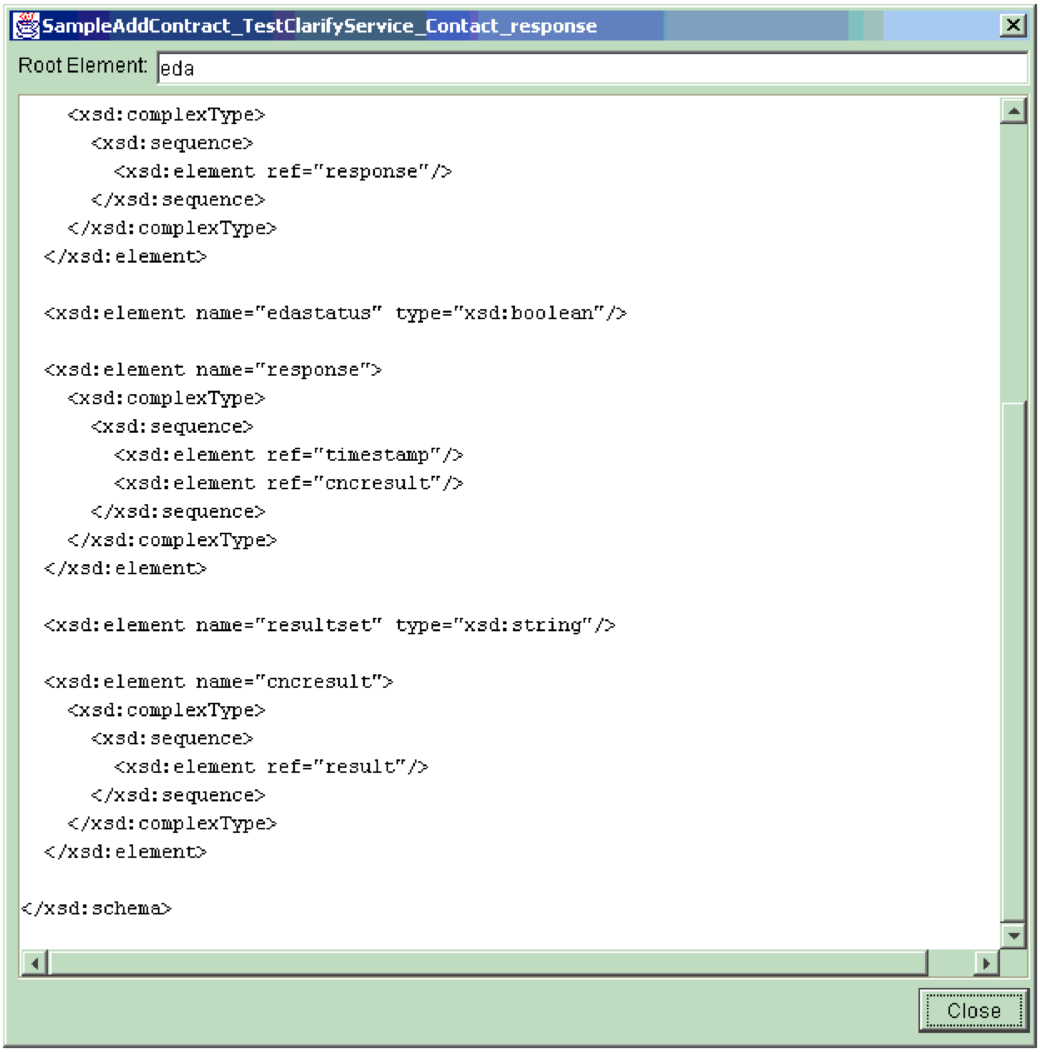
Figure 2-32 Starting the Workflow



The ClarifyCRM Response

When the workflow is started, the Task1 step takes the document and sends it to the ClarifyCRM application. Upon completion of the task, the workflow places the response XML document in the ContactAddResponse variable. By pressing the View Response Definition from the Call Service window, the XML Schema is displayed.

Figure 2-33 Response Document



Root Element: eda

```
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="response"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="edastatus" type="xsd:boolean"/>

  <xsd:element name="response">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="timestamp"/>
        <xsd:element ref="cncresult"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="resultset" type="xsd:string"/>

  <xsd:element name="cncresult">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="result"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

</xsd:schema>
```

CHAPTER

3 Creating Schema Repositories

This section explains how to describe metadata for ClarifyCRM to WebLogic Integration. After you have described the metadata, you can create and deploy application views using the WebLogic Application View Console.

This section includes the following topics:

- Understanding Metadata
- Introducing Schemas and Repositories
- Naming a Schema Repository
- The Repository Manifest

Understanding Metadata

When you define an application view, you are creating an XML-based interface between WebLogic Integration and ClarifyCRM or another enterprise information system (EIS) or application within your enterprise. The BEA WebLogic Adapter for ClarifyCRM provides schemas for services and events that will handle any of the supported ClarifyCRM operations. These schemas must be specified in the `manifest.xml` file for the application view which uses the adapter.

Introducing Schemas and Repositories

You describe all the documents entering and exiting your WebLogic Integration system using W3C XML schemas. These schemas describe each event arriving to and propagating out of an event, and each request sent to and each response received from a service. There is one schema for each event and two for each service (one for the request, one for the response). The schemas are usually stored in files with an `.xsd` extension.

Use the WebLogic Integration Application View Console to access events and services, and to assign a schema to each event, request, and response. Assign each application view to a schema repository; you can assign several application views to the same repository.

BEA WebLogic adapters all make use of a schema repository to store their schema information and present it to the WebLogic Application View Console. The schema repository is a directory containing:

- A manifest file that describes the event and service schemas.
- The corresponding schema descriptions.

To work with schemas, you must know how to:

- Name a schema repository.
- Create a manifest.
- Create a schema.

Naming a Schema Repository

The schema repository has a three-part naming convention:

```
session_path\clarify\connection_name
```

Here,

- *session_path* is the schema's session base path (also known as the session base directory), beneath which you can store multiple sessions of schemas.
- `clarify` is the type of adapter.
- *connection_name* is a name representing a particular instance of the adapter.

For example, if the session path is `/usr/opt/bea/bse`, the adapter type is ClarifyCRM, and the connection name is TestClarify, then the schema repository is the directory:

```
/usr/opt/bea/bse/ClarifyCRM/TestClarify
```

The Repository Manifest

Each schema repository has a manifest that describes the repository and its schemas. This repository manifest is stored as an XML file named `manifest.xml`.

The manifest file relates documents (through their schemas) to services and events. The manifest exposes schema references to the event, relating the required document (via the root tag) to the corresponding schema. Schemas and manifests are stored in the same schema repository.

The following is an example of a manifest file. It illustrates the relationships between events and services and their related schemas:

Listing 3-1 Sample Manifest File

```
<manifest>
<connection/>
    <schemaref name="clarify">
                  <request root="sp" file="bea_clarify_request.xsd" />
                  <response root="response" file="bea_clarify_response.xsd" />
                <event root="ClarifyEvents" file="bea_clarify_events_row.xsd" />
          </schemaref>
</manifest>
```

The manifest has a schema reference section, named `schemaref`. The schema reference name is displayed in the schema drop-down list on the Add Service and Add Event windows in the WebLogic Application View Console.

This sample manifest has two schema references:

- One for events.

  Events require only one schema, defined by the `event` tag. This relates the root tag of an XML document produced when a ClarifyCRM event occurs.

- One for services.

  Services require two schemas: one for the document being passed to the service, represented by the `request` tag, and one for the expected response document received from the service operation, represented by the `response` tag.

You may also create a schema reference that comprises both services and events.

Creating a Repository Manifest

The repository manifest is an XML file with the root element `manifest` and two sub-elements:

- `connection`, which appears once, and which you can ignore because it is not used by the BEA WebLogic Adapter for ClarifyCRM at runtime, but only by the BEA Application Explorer when generating schemas from ClarifyCRM metadata.

Note: At the time this document was created, BEA Application Explorer did not support the dynamic creation of service and event schemas for ClarifyCRM. Check with BEA Systems for the latest BEA Application Explorer adapter support.

- `schemaref`, which appears multiple times, once for each schema name, and which contains all three schemas—request, response, and event.

To create a manifest:

1. Create an XML file with the following structure:

```
<manifest>
   <connection>
   </connection>
</manifest>
```

2. For each new event or service schema you define, create a `schemaref` section using this model:

```
<schemaref name="Product">
   <request root="sp" file="bea_clarify_request.xsd"/>
   <response root="response" file="bea_clarify_response.xsd"/>
   <event root="clarify" file="bea_clarify_event_row.xsd"/>
</schemaref>
```

   Here, the value you assign to:

   - `file` is the name of the file in the schema repository.
   - `root` is the name of the root element in the actual instance documents that will arrive at, or be sent to, the event or service.

The following is an example of an instance document for the Product event referred to in "Creating a Repository Manifest" on page 3-4.

Listing 3-2 Request Document for ClarifyCRM Contact_Insert Service

```
<?xml version="1.0"?>
      <sp>
       <data>
        <cbo name="contact" action="create">
         <keys>
          <key  name="first_name">Bill</key>
          <key  name="last_name">Wier</key>
          <key  name="phone">408-777-9000</key>
          <keyRel name="contact2contact_role">2030120</keyRel>
          <fields>
           <field name="address_1">235 darwin drive</field>
           <field name="address_2">APT 235</field>
           <field name="City">fremont</field>
           <field name="e_mail">bwier@bww.com</field>
           <field name="fax_number">408-999-9898</field>
           <field name="mail_stop">M123</field>
           <field name="salutation">Mr.</field>
           <field name="status">0</field>
           <field name="title">CEO</field>
          </fields>
          <Relations>
           <Rel name="contact2contact_role">26589412</Rel>
          </Relations>
         </keys>
        </cbo>
       </data>
      </sp>
```

The following is a schema matching this instance document; it may be manually coded or generated from any XML editor:

Listing 3-3 Schema Matching Contact_Add Service Request Document

```
<?xml version=”1.0” encoding=”UTF-8”?>
<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”
elementFormDefault=”qualified”>
       <xsd:complexType name=”cboType”>
              <xsd:sequence>
                     <xsd:element name=”keys” type=”keysType”
minOccurs=”0”/>
                     <xsd:element name=”fields” type=”fieldsType”
minOccurs=”0”/>
              </xsd:sequence>
              <xsd:attribute name=”name” type=”xsd:string”
use=”required”/>
              <xsd:attribute name=”action” type=”xsd:string”/>
              <xsd:attribute name=”relation” type=”xsd:string”/>
       </xsd:complexType>
       <xsd:complexType name=”dataType”>
              <xsd:sequence>
                     <xsd:element name=”cbo” type=”cboType”/>
              </xsd:sequence>
       </xsd:complexType>
       <xsd:element name="sp" type="spType"/>
       <xsd:complexType name="fieldType">
              <xsd:simpleContent>
                     <xsd:extension base="xsd:string">
                            <xsd:attribute name="name"
type="xsd:string" use="required"/>
                            <xsd:attribute name="data_type"
type="xsd:string"/>
                     </xsd:extension>
              </xsd:simpleContent>
       </xsd:complexType>
       <xsd:complexType name="fieldsType">
              <xsd:sequence>
                     <xsd:element name="cbo" type="cboType"
minOccurs="0" maxOccurs="unbounded"/>
                     <xsd:element name="field" type="fieldType"
minOccurs="0" maxOccurs="unbounded"/>
                     <xsd:element name="cbo" type="cboType"
minOccurs="0" maxOccurs="unbounded"/>
              </xsd:sequence>
       </xsd:complexType>
       <xsd:complexType name="keysType">
              <xsd:sequence>
                     <xsd:element name="field" type="fieldType"/>
              </xsd:sequence>
       </xsd:complexType>
       <xsd:complexType name="spType">
              <xsd:sequence>
                     <xsd:element name="data" type="dataType"/>
              </xsd:sequence>
       </xsd:complexType>
</xsd:schema>
```

Sample Files

Supplied with the BEA WebLogic Adapter for ClarifyCRM are XSD schema files for service requests and responses and for events. With these schemas, all of the ClarifyCRM methods and objects that the adapter supports can be used in application views. These files are supplied in a zip file called `samples.zip`. These schemas can be used in the `manifest.xml` files you build.

For services, use the schemas as defined in the table that follows.

Table 3-1 Service Schemas

For . . .	Use the following schema . . .
Sevice requests	`bea_clarify_request.xsd`
Service responses	`bea_clarify_response.xsd`

For events, use one of three files in the `manifest.xml` file, depending on the format you select when you build the event, as defined in the table that follows.

Table 3-2 Event Schemas

Format	Use the following schema . . .
Row	`bea_clarify_event_row.xsd`
Column	`bea_clarify_event_col.xsd`
Field	`bea_clarify_event_field.xsd`

The `samples.zip` file also includes sample request files that work with the above-mentioned schemas. These instances contain the object and operation inside ClarifyCRM to be executed, in addition to sample data.