WebSphere Adapters

Adapter for Siebel Businesss Applications

Version 6.0

Note

Before using this information, be sure to read the general information in “Notices” on page 55.

7April2006

This edition of this document applies to IBM WebSphere Adapter for Siebel Business Applications (572480), Version 6.0, and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

WebSphere Adapter for Siebel Business Applications Version 6.0 User Guide

The IBM® WebSphere® Adapter for Siebel Business Applications facilitates the exchange of business objects between Siebel Business Applications systems and J2EE-based programming models.

Product overview

The IBM WebSphere Adapter for Siebel Business Applications facilitates the exchange of business objects between Siebel systems and J2EE based application components.

Audience

The information in this topic defines the users of the WebSphere Adapter products and details the skills they require.

The audience for the adapter user guide includes data and application integrators who are responsible for assembling application components into a complete solution and preparing this solution for testing and deployment. These users require the following general skills:

- A good understanding of the business solution and the business environment
- Knowledge of application and solution components, to enable their efficient collaboration at run-time
- A detailed understanding of databases, data access issues, transactional models and connections across heterogeneous relational databases, queues, and web services
- Familiarity with integration tools

The application integrator is also responsible for detailed testing activities and needs these additional skills:

- Creating required scripts, tools, and templates for testing and deployment
- Creating integration workspaces and integrating systems & subsystems
- Resolving interdependencies between entities such as Enterprise Java Beans (EJBs), workflows, and web pages
- Validating the application or solution

The data integrator is also responsible for enabling access to a range of data sources for the application developers. The required skills include:

- Installing and configuring integration capabilities or point-to-point gateways
- Writing procedures to use database access logic efficiently
- Building data models for external data access tools
- Implementing security measures

Task road map: WebSphere Adapter for Siebel Business Applications

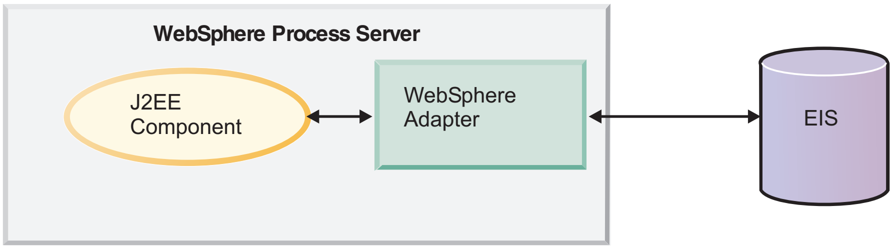
This task road map gives the user the complete perspective from installing to the usage of the adapter.

Task	Description
Installing the adapter	This topic describes how to install WebSphere Adapter for Siebel Business Applications.
Deploying the adapter	This topic describes how to deploy WebSphere Adapter for Siebel Business Applications.
Configuring the adapter	By using WebSphere Process Server administrative console, you can configure WebSphere Adapter for Siebel Business Applications.
Troubleshooting the adapter	This topic describes how to troubleshoot WebSphere Adapter for Siebel Business Applications.
Using the sample applications	This topic describes the sample application for WebSphere Adapter for Siebel Business Applications.

IBM WebSphere Adapters

An IBM WebSphere Adapter implements the Java 2 Enterprise Edition (J2EE) Connector architecture (JCA), version 1.5. Also known as resource adapters or JCA adapters, WebSphere Adapters enable managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components supported by WebSphere Process Server.

A WebSphere Adapter



The IBM(R) WebSphere (R) Adapter portfolio is a new generation of adapters based on the Java 2 Platform, Enterprise Edition (J2EE) standard. JCA is a standard architecture for integrating J2EE applications with enterprise information systems. Each of these systems provides native APIs for identifying a function to call, specifying its input data, and processing its output data. The goal of the JCA is to provide an independent API for coding these functions, to facilitate data sharing, and to integrate J2EE applications with existing and other EISs. The JCA standard accomplishes this by defining a series of contracts that govern interactions between an EIS and J2EE components within an application server.

Fully compliant with the JCA standard, WebSphere Adapters have been developed to run on WebSphere Process Server. A WebSphere Adapter does the following:

- Integrates with WebSphere Process Server.
- Connects an application running on WebSphere Process Server with an EIS.
- Enables data exchange between the application and the EIS.

Each WebSphere Adapter is made up of the following:

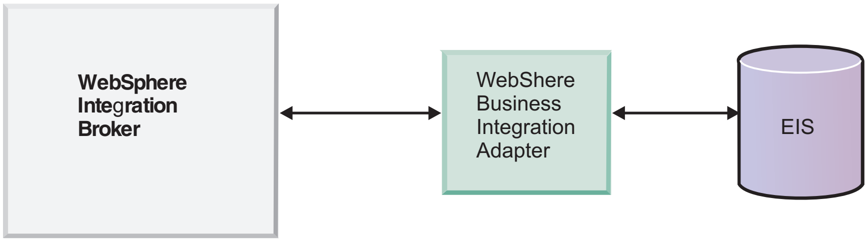
- An implementation of the (J2EE) Connector Architecture (JCA), version 1.5 that supports WebSphere Process Server
- An enterprise metadata discovery component— you use this component with the enterprise service discovery wizard to introspect the EIS— to generate business objects and other service component architecture (SCA) artifacts that are compiled in a standard enterprise application archive (EAR) file.

WebSphere Adapters use service data objects (SDO) for representing data objects.

WebSphere Adapters and WebSphere Business Integration Adapters

Unlike WebSphere Adapters, WebSphere Business Integration Adapters are not JCA-compliant.

A WebSphere Business Integration Adapter



As shown in the figure, WebSphere Business Integration Adapters are distributed. They reside outside of the application server. The server, or integration broker, communicates with this type of adapter through a Java Messaging Service (JMS) transport layer.

Other differences between WebSphere Adapters and WebSphere Business Integration Adapters include the following:

- **Connection management** WebSphere Adapters rely on standard JCA contracts to manage life-cycle tasks such as stopping, starting; WebSphere Business Integration Adapters rely on the WebSphere Adapter Framework to manage connectivity.
- **Event notification** Known as inbound event notification for WebSphere Adapters.
- **Request processing** Known as outbound support in WebSphere Adapters.
- **Object definition** With WebSphere Adapters, you use an enterprise metadata discovery component to probe an EIS and develop business objects and other useful artifacts. This enterprise metadata discovery component is part of the WebSphere Adapter. WebSphere Business Integration Adapters use a separate Object Discovery Agent (ODA) to probe an EIS and generate business object definition schemas.

Enterprise service discovery

The enterprise service discovery wizard allows you to generate business objects for enterprise information system (EIS) or database entities.

The enterprise service discovery wizard provides a blue print for business objects. It allows you to browse the metadata information of an EIS or database, enables selection of the artifacts of interest, and generates deployable service objects and descriptions. By selecting meta-object nodes from the metadata tree structure, you can generate business objects for EIS or database entities. The metadata is transformed into service data objects consisting business graphs and business objects.

The enterprise service discovery wizard allows you to perform the following actions:

- Generate business objects
- Set application-specific information on the business objects
- Set application-specific information on properties
- Provide service descriptions for inbound and outbound events
- Provide connection descriptions for inbound and outbound events

How the WebSphere Adapter for Siebel Business Applications works

WebSphere Adapter for Siebel Business Applications connects to Siebel applications by making calls to the Siebel native interfaces and passing data to and receiving data from the Siebel application. The adapter supports Siebel business services only in this release.

The adapter models Siebel business service method calls as business objects. Outbound event processing consists of the following steps:

1. A business object representing the Siebel method call is passed from the JCA client application to the adapter using the common client interface (CCI) record. The CCI record is an instance of the CCI implementation that enables the interaction between the J2EE application and the adapter, and then to the Siebel back-end system. This record generates and sets the business object.
2. The adapter extracts the elements from the business object and, using the metadata information from the business object, recognizes the Siebel interface to use.
3. The adapter converts the business object data to the appropriate Siebel method call.
4. The adapter executes the method on the designated Siebel business service.

For inbound processing, the adapter models the integration objects as business objects. You select the business service name that processes the required integration object. For example, you select EAI Siebel Adapter if you plan to use an integration object based on a Siebel business object or you select Siebel Account if you plan to use Account Interface as the integration object for your inbound processing.

For outbound processing, Event Method is not used and should be left blank when the service type is outbound. This is used by the adapter for inbound processing alone to retrieve the integration object; this should be the business service method name that does the retrieval.

Inbound processing consists of the following steps:

1. The adapter polls the event component at regular intervals.
2. If an event is found, the integration object represented by the event is retrieved.
3. The appropriate verb is set and dispatched to the registered endpoints.

Outbound event processing

Outbound event processing allows a client to make calls to the adapter to perform a specific operation in a Siebel business application.

The client can request a connection using a connection factory and connection specification that specify the user name and password used for authentication. For service component architecture clients, the functionality of the adapter is exposed through interfaces described by a Web services description language (WSDL). The outbound service description, *EISImportBinding*, is a service component description language artifact that is produced by the enterprise service discovery wizard. It is an import file. The values required at runtime are present in the import file, which specify the function name, user name, and password required.

The client automatically creates an interaction specification that specifies a function name that is a valid method of the business service. The business service name is in the metadata of the incoming business object.

The mechanisms execute a request using the interaction of the adapter. The adapter uses the input business object to determine the business service and the function name to determine the method to invoke.

The adapter creates a copy of the input business object, which is populated with results, as the output business object. The adapter builds the required Siebel property set based on the input business object and invokes the business service. The Siebel property set output is then populated in the output business object. The output business object is returned to the WebSphere Business Integration record implementation, which is returned to the calling client.

Inbound event processing

Inbound event processing is supported by a Siebel event component. The adapter polls the event component at regular intervals. Messages are propagated to endpoints that register for the events.

The event component lists the type of event, the corresponding business graph, and the status of the event. These values are retrieved by the adapter; then, the integration object represented by the event is retrieved. The integration object values are populated in the business graph, which is then dispatched to the registered endpoints.

Asynchronous event delivery

Asynchronous event delivery is supported by the Siebel event component. When events are delivered to endpoints, they can exist only in one of four event states.

An event is represented by a row in the event component that contains the information necessary to instantiate the business service. The event component has fields to represent the information required to process the event. The events in the component within Siebel also have a status that the adapter uses when processing events, as seen in the following event table.

Event states

Event	Description
Event marked as new/ready-to-be-polled in EIS	No record of the event is in the staging table
Event marked as new/ready-to-be-polled in EIS	One or more records exist in the staging table that refer to this event
Event marked as in-progress in EIS	One or more records exist in the staging table that refer to this event
Event marked as in-progress in EIS	No record of the event is in the staging table

Components

Event notification requires the creation of an event store that holds the event records in the Siebel database; this event store is a Siebel business component.

The "IBM Events" Siebel business component, is the default table used to track events that occur in the Siebel enterprise information system.

Event triggers and processes

Creating, updating, or deleting records in a Siebel business application is treated as an event. Siebel supports Visual Basic scripts and Siebel eScripts embedded in the Siebel business component event handlers to populate the event table.

The "IBM Resource Adapter Events" Siebel business component stores information about the event. This information is used by the adapter during event subscription to build the corresponding business object and send it to the registered endpoints.

Inbound business objects that are returned from the adapter are of the same types as business graphs used during outbound interactions. Triggers are placed on the base Siebel business objects.

Event distribution

Field	Description	Example
Description	Any comment associated with the event.	Account interface event
Event ID	ID of the event row.	Automatically generated unique ID in Siebel
Event time stamp	The time stamp for the event. The format is mm/dd/yyy hh:mm:ss	02/24/2005 11:37:56
Event type	The type of event.	Create
Object key	The unique identifier which identifies the business object row for which the event was created. It is a name value pair consisting of the name of the property and the value.	Name=IBM
Object name	The business graph for which the event was detected.	EAIAccountInterfaceBG
Priority	The event priority.	1
Status	The event status. This is initially set to READY_FOR_POLL.	0

The event is retrieved from the event table and the information is used to retrieve the integration object, which is then used to create a business graph to be published to endpoints of interest. There are two ways in which the event can be retrieved by the adapter; the default mechanism and a user defined method using metadata contained in the business object.

The default retrieval mechanism uses the EAI Siebel Adapter business service to execute a query based on the object key. The values of the resulting property set are then populated into the business object.

The retrieval mechanism, defined by you, can be used to perform optimized retrievals. The adapter uses the event method metadata during retrieval as the method on the business service. The business service is also metadata information on the top-level business object. You can specify the event method metadata during business object discovery at design time. The business service is also metadata but this is mandated by the business service in which this business object was generated against, and is automatically populated by the enterprise service discovery wizard.

An example of an event method is the *QueryByExample* method on a Siebel Application Services Interface business service such as Siebel Account.

As events are retrieved and processed from the event table, the status of the event goes through a cycle, seen in the table below. This is different from the status that is maintained in the staging table.

Status of events during an event notification

Status short name	Description	Event table value
Error processing event	There was an error processing the event.	-1
Processing	The event has been picked up by the adapter but not yet delivered to the event manager or endpoints.	3
Ready for poll	The event has not yet been picked up by the adapter. The event is ready to be picked up.	0
Success	The event has been delivered to the event manager.	1

After an event is successfully sent to the endpoints and marked as *Success*; the adapter deletes the successful event from the table.

The business graph is also populated with the event type from the table which is entered as a top-level verb on the business graph. This is used by the function selector during notification.

Note: WebSphere Adapter for Siebel Business Applications does not support XA transaction or local transaction.

Siebel application architecture

The adapter uses the Java™ application program interfaces provided by the Java Data Bean™ for Siebel to communicate with the Siebel Object Manager for data exchange. The Java Data Bean for Siebel exposes the Siebel business services. The Siebel business services are part of the business objects layer in the Siebel application architecture.

The adapter works with a series of data types or layers. An object definition implements one part of the software: A user interface, an abstract data representation, or a direct data representation construct.

Siebel application architecture is a layered structure containing the following:

- User interface layer
- Business objects layer
- Data objects layer

User interface layer

The user interface defines the visual elements used to lay out applets and views, navigate, and make selections using buttons and check boxes.

Business objects layer

The business objects layer contains entities such as business objects, business components, integration objects, and business services.

A business component is a fundamental binary entity consisting of multiple fields that represent it. A business object is a collection of related business components. A business service is an entity in the Siebel application that encapsulates a set of functionality. The WebSphere Adapter for Siebel Business Applications communicates with this layer using the Siebel Java Data Bean.

Business services:

A Siebel business service is an entity in Siebel that encapsulates and simplifies sets of functionality, such as moving and converting data formats between the Siebel application and external applications.

Siebel business components and business objects are objects that are typically tied to specific data and tables in the Siebel data model. Siebel business services, on the other hand, are not tied to specific objects, but rather operate on objects to achieve a particular goal.

Business services allow you to encapsulate business logic in a central location, abstracting the logic from the data. A business service is much like an object in an object-oriented programming language.

A service has properties and methods and maintains a state. Methods take arguments that can be passed into the object programmatically or, in the case of Siebel EAI, declaratively by way of workflows.

Siebel business services include generic business services, for example EAI Siebel Adapter, applications services interfaces (Siebel Contact), and custom business services.

The adapter supports only business services that are of the class types listed in the table below.

Class types for business services

Business service class type	Description
CSSEAIDataSyncService	This class is used for data synchronization services.

Business service class type	Description
CSSService	This class is used for any business service. It is typically extended for other services. An example is the class, *CSSEAISiebelAdapter*, which inherits from the *CSSService* class and maps to the EAI Siebel Adapter business service.
CSSEAIDTEScriptService	This class is used for data transformation services.
CSSEAISiebelAdapter	This class is used for the EAI Siebel Adapter.

Data objects layer

Object definitions in the data objects layer provide a logical representation of the underlying physical database and are independent of the installed relational database management system. The Java Data Bean for Siebel that is used by the Siebel adapter does not have direct access to this layer.

The adapter supports methods of the generic business services, custom business services, and application services interfaces. The Siebel application services interfaces provide integration-object-specific services. The Siebel Account integrates with the Account Interface integration object. The Siebel Contract integrates with the Contract Interface Integration object.

EAI Siebel Adapter is a general-purpose business service that allows for data-synchronization-based integration by reading and writing integration objects.

Methods for EAI Siebel Adapter

Method	Description
Delete	Deletes the integration object
Execute	Performs a combination of operations on the components of the integration object
Insert	Creates the integration object
Query	Retrieves the integration objects that match the input business object
Query page	Performs a query and returns a limited number of records
Synchronize	Synchronizes the values in the application with those in the business object
Update	Updates the integration object and synchronizes all children
Upsert	Synchronizes, but does not delete

Method examples

Method	Description
InsertOrUpdate	Inserts the integration object and updates it if it exists
QueryByExample	Retrieves the integration object based on non-key values

Method	Description
QueryById	Retrieves the integration object based on the primary row ID

The outbound operation to be performed on the input business object instance is specified as a function name in the interaction specification. The adapter extracts this and the metadata information necessary to access the Siebel object and make the requested changes in Siebel.

Locale and globalization support

This adapter has been globalized so that it can support single- and double-byte character sets and deliver message text in the specified language.

This adapter supports the processing of bidirectional script data for Arabic and Hebrew languages. To use the bidirectional capacity, you must configure the bidirectional properties. In this user guide, the term *bidirectional properties* refers to the properties that control invocation of bidirectional support.

If your enterprise information system (EIS) uses a bidirectional format that differs from the Windows standard format, all properties with bidirectional support are transformed from the Windows standard format to the bidirectional format of the target EIS. The adapter also transforms such data from the EIS into Windows standard format before passing it to WebSphere Process Server.

The Java$^{(TM)}$ runtime environment within the Java Virtual Machine (JVM) represents data in the Unicode character code set. Unicode contains encodings for characters in most known character code sets (both single- and multi-byte). Most components in the WebSphere Business Integration system are written in Java. Therefore, when data is transferred between most WebSphere Business Integration system components, there is no need for character conversion.

To log error and informational messages in the appropriate language and for the appropriate country or territory, the adapter uses the locale of the system on which it is running.

WebSphere Process Server bidirectional language format

WebSphere Process Server uses the bidirectional language format of ILYNN (implicit, left-to-right, on, off, nominal), which is also the Windows bidirectional language format. All other bidirectional language formats must be converted prior to being introduced to WebSphere Process Server.

Five attributes must be set for the proper bidirectional language format. The attributes and settings are listed in the table titled “Bidirectional attributes.”

Bidirectional attributes

Letter position	Purpose	Values	Description	Default setting
1	Order Schema	I or V	Implicit (Logical) or Visual	I
2	Direction	L R C D	Left-to-Right Right-to-Left Contextual Left-to-Right Contextual Right-to-Left	L

Letter position	Purpose	Values	Description	Default setting
3	Symmetric Swapping	Y or N	Symmetric Swapping is on or off	Y
4	Shaping	S N I M F B	Text is shaped Text is not shaped Initial shaping Middle shaping Final shaping Isolated shaping	N
5	Numeric Shaping	H, C, or N	Hindi, Contextual, or Nominal	N

The adapter is responsible for transforming data into a Logical-Left-to-Right format before sending the data into WebSphere Process Server components.

Note: The locale setting of the user interface (browser) defines the bidirectional language display and edit format. WebSphere Process Server user interfaces must convert locale-specific formats to the WebSphere Process Server default format.

Bidirectional property levels

You can set bidirectional properties at several different levels. For more details on these properties and how to set them using the enterprise service discovery wizard, refer to the sections on creating the adapter project and configuring the adapter.

Editing bidirectional properties

You can edit the bidirectional properties for business objects and business object attributes using annotations in the Business Object Editor in WebSphere Integration Developer. The annotations are stored in the business object (the *.xsd file). For more information, refer to the Business Object Editor documentation on the WebSphere Integration Developer website at http://www.ibm.com/software/integration/wid.

You can also edit certain bidirectional properties once they have been defined by using the assembly editor in WebSphere Integration Developer. For more information on using bidirectional properties at run time, refer to the general technical paper and the adapter technical paper regarding bidirectional support. For more information on the assembly editor, refer to the assembly editor documentation on the WebSphere Integration Developer website at http://www.ibm.com/software/integration/wid.

Business objects overview

The topics in this section give you information about working with business objects.

Business object naming conventions

The business object naming convention includes the concatenation of several words for both outbound and inbound events.

The naming convention for the business objects includes the concatenation of several words which include:

- Prefix – An optional prefix used only for outbound artifacts and is placed at the start of the business object name.
- Business Service Name – The business service for the business object.
- Integration Object – The underlying Integration Object for the Siebel Message container business object.
- Integration Component – The underlying Integration component for the Siebel Message container business object.
- Method Name – The method that this business object was generated against.

Outbound

The top-level business object name has the following format:

<Prefix><Business Service Name><Method Name><Names of all the integration objects selected for the Input and InputOutput complex type arguments>

If there are no Input or InputOutput arguments, the names of all the output arguments used in the concatenation have the following format:

<Prefix><Business Service Name><Method Name><Names of all the integration objects selected for the output complex type arguments>

If there are no complex arguments in the method, the name format has the following format:

<Prefix><Business Service Name><Method Name>

The business graph names, for the top-level business objects generated against the business service methods, have the following format:

<Top Level business object Name> + BG.

Note: The Prefix is used only for the business graph (BG) and objects and not for the business objects generated against complex type arguments, for example, integration objects and components.

Examples

When using the prefix, IBM, you generate a business object for the EAI Siebel Adapter and insert method, and then choose the *Account Interface* and *Business Address Interface* integration objects against an *Input* and *InputOutput* method argument. The corresponding business object generated is *IBMEAISiebelAdapterInsertAccountInterfaceBusinessAddressInterface*.

This gets created for the choices mentioned above. The business graph name is *IBMEAISiebelAdapterInsertAccountInterfaceBusinessAddressInterfaceBG*.

For the outbound objects generated against integration components, the following naming convention applies: 'IO' + <Name of Integration Object> + 'IC' + <Name of integration component>.

As an example, the Account Interface integration object has the business object name, IOAccountInterfaceICAccount.

Inbound

Since the objects are generated only for integration components, the naming convention for inbound objects follows the one used for outbound objects generated against integration components. However, there is a business graph generated too.

The business graph has the suffix, BG, added to the business object name, for example, *IOAccountInterfaceICAccountBG*.

Note: The prefix property value is never used for the inbound service type.

Business object attribute properties

Business object attribute properties and their descriptions are listed in the table below.

Business object attribute properties

Attribute	Property
Cardinality	For simple attributes, 1 is used. For container attributes, depending on the method requirements, n is used.
Key and foreign key	These are not used.
Name	Contains the name of the attribute.
Required	This is not used.
Special	None.
Type	The type can be a string or complex type representing an integration object.

How to define metadata

The adapter can retrieve a list of business services from the Siebel enterprise information system repository and generate corresponding business objects for selected business services.

You can generate business objects for selected methods under a business service. You can also provide details for the input and output integration object that correspond with the complex type arguments in the business service method for each selected business service.

Note: Business objects are generated for the input and output integration objects tool along with the one generated for the business service. However, if the integration object details are already present as part of the business service method, you need not provide input.

When you select a business service for generation, the methods under the business service become the functions. For example, if the business service is *EAI Siebel Adapter* and its methods are Insert, Update, and Upsert, the business objects generated on a per-method basis have the same methods as the corresponding functions.

Business object metadata

Business object information can be viewed in the tables and schema that follow.

Business object application-specific text

Parameter	Description
BSN	The name of the business service that is used by the business object.
EventMethod	This identifies the event method to use when you retrieve event data for inbound operations, instead of the default query method.
IC	The name of the Siebel integration component corresponding to the business object.
IO	The name of the Siebel integration object corresponding to the business service name of the business object.

Property metadata

Parameter	Description
FN	The name of the field in the Siebel integration component or business service method that the attribute represents.
ParamType	Identifies whether the property is an input, output or both. Values include *Input*, *Output*, and *InOut*.

Note: There is no verb metadata.

Siebel business object metadata schema

```
<schema targetNamespace="urn:app:sieb:asi" xmlns:sasi="urn:app:sieb:asi
"xmlns:xs="http://www.w3.org/2001/XMLSchema
"xmlns:botm="http://www.ibm.com/wbi/BusinessObjectMetadata"
elementFormDefault="qualified
"xmlns="http://www.w3.org/2001/XMLSchema">
<complexType name="SiebelBusinessObjectTypeMetadata">
<sequence minOccurs="1" maxOccurs="1">
<!--  Identifies the Business Service Name -->
<element name="BSN" type="string" />
<!--  Identifies the Integration Object for the Business Object. -->
<element name="IO" type="string" />
   <!--  Identifies the Integration Component for this Business Object. -->
<element name="IC" type="string" />
<!-- Identifies the Method to invoke during event notification
If none specified there is a default method used -->
<element name="EventMethod" type="string" minOccurs="0" />
</sequence>
</complexType>
<complexType name="SiebelAttributeTypeMetadata">
<sequence>
<!--  Identifies the field name to match to a property set. -->
<element name="FN" type="string" />
<!--  Identifies whether the property is a Input,Output or InOut.
This value is set only on the top level object. -->
<element name="ParamType" type="sasi:ParamTypes" />
</sequence>
</complexType>
<simpleType name="ParamTypes">
<restriction base="string">
```

```
<enumeration value="Input" />
<enumeration value="Output" />
<enumeration value="InOut" />
</restriction>
</simpleType>
</schema>
```

Supported verbs

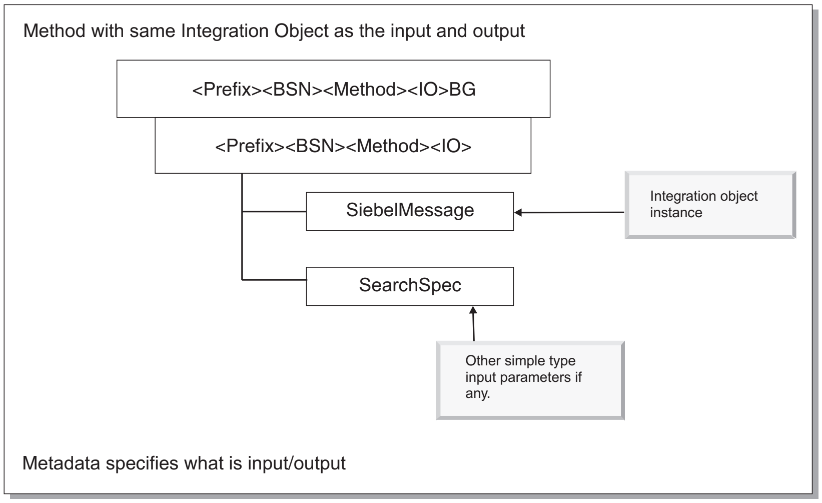
The create, update, and delete verbs are supported on the integration-object based business objects for inbound interactions.

Business object structure: Example 1

Metadata defines what can be used as input and output for a method.

The following diagram represents a method with the same integration object being used as input and output.

- <Prefix> - Prefix as stated in EMD
- <BSN> - The business service name for the business object
- <Method> - The method the business object was generated against
- <IO> - The underlying IO chosen to be used in the method


Method with same Integration Object as the input and output
<Prefix><BSN><Method><IO>BG
<Prefix><BSN><Method><IO>
SiebelMessage
Integration object instance
SearchSpec
Other simple type input parameters if any.
Metadata specifies what is input/output

A method with the same integration object as the input and output

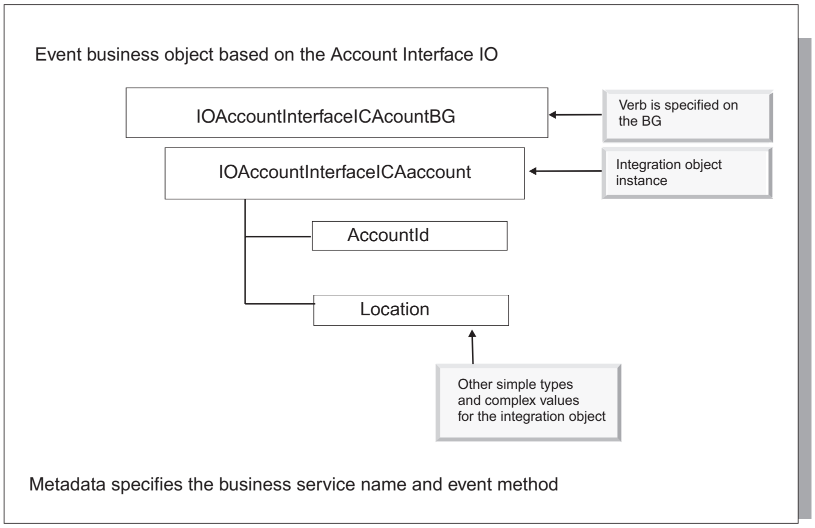
Business object structure: Example 2

The following business object structure follows the template business graph pattern of the business object framework. The data parameters for the methods on the business service are seen where there is a business graph, template, or top-level object that is a single element to the graph.

The top-level object contains metadata information that states the business service for which the business object corresponds. The business graph contains top-level verbs which are used during event notification as an emit event which is based on the verb. The verbs currently supported are create, update, and delete.

The top-level data is a combination of the representation of both input and output arguments with the Siebel Message as a container. This same business object can be used for both request and response to and from the adapter to interact with the underlying Siebel EIS. This means that the same business object type that you send in as a request is returned as the result of the execution.

The Siebel message is a wrapper similar to the wrapper that the Siebel EIS uses to wrap integration objects and their respective fields and components within business services.


Event business object based on the Account Interface IO
IOAccountInterfaceICAcountBG
Verb is specified on the BG
IOAccountInterfaceICAaccount
Integration object instance
AccountId
Location
Other simple types and complex values for the integration object
Metadata specifies the business service name and event method

An event business object based on the Account Interface IO

Installing the adapter

Topics in this section provide information about the hardware and software requirements for installing the adapter, as well as a listing of files that are installed with the adapter.

For information on installing, see the *WebSphere Adapters Installation Guide*.

For information on installing, see Installing the adapter.

Adapter environment

The information in this topic provides a link for hardware, software, and other requirements for WebSphere Adapter for Siebel Business Applications.

Hardware and software requirements

For hardware and software requirements for this adapter, see IBM WebSphere Adapters and IBM WebSphere Business Integration Adapters: Hardware and Software Requirements. Select your adapter from the list of WebSphere adapters.

Adapter-specific information

You must create an event table in the Siebel application.

Adapter-specific information

Specific to WebSphere Adapter for Siebel Business Applications, you must create an event table in the Siebel application.

You may already have an event table created in the Siebel application. If not, you must create an event table in the Siebel application.

Related tasks

"Creating an event table in the Siebel application" on page 39
To create an event table, you must be familiar with Siebel tools functionality. For detailed information on any of the steps, refer to the documentation for Siebel tools.

Installed file structure

After installation, you can view a list of installed directories and files.

The following table lists the UNIX and Linux directories and files for the WebSphere Adapter for Siebel Business Applications. The directories and files are grouped into categories.

Directory and file structure for UNIX and Linux

File and directory category	Directories and files
RAR files	/adapter/Siebel/deploy/ CWYEB_SiebelAdapter.rar
Sample files	/adapter/Siebel/samples/Apps/acctoutApp.ear
	/adapter/Siebel/samples/Apps/acctinApp.ear
	/adapter/Siebel/samples/Event_pkg/Account.js
	/adapter/Siebel/samples/Event_pkg/ BusinessAddress.js
	/adapter/Siebel/samples/Event_pkg/Contact.js
	/adapter/Siebel/samples/Event_pkg/ IBMAudit.sif
Notices file	/adapter/Siebel/notices.txt
ISA plugin zip file	/adapter/Siebel/ISAPlugin/ com.ibm.com.esupport.client. SS6FE6_RASiebel.zip
IBM Tivoli® License Manager (ITLM) file	/adapter/Siebel/5724L80E060000.sys
Log Message zip file	/adapter/Siebel/messages/ CWYBS_AdapterFoundation_messages.zip
	/adapter/Siebel/messages/ CWYBS_AdapterFoundation_messages.tar
	/adapter/Siebel/messages/ CWYEB_SiebelAdapter_messages.zip
	/adapter/Siebel/messages/ CWYEB_SiebelAdapter_messages.tar

The following table lists the Windows® directories and files for the WebSphere Adapter for Siebel Business Applications. Directories and files are grouped into categories.

Directory and file structure for Windows

File and directory category	Directories and files
RAR files	\adapter\Siebel\deploy\ CWYEB_SiebelAdapter.rar
Sample files	\adapter\Siebel\samples\Apps\acctoutApp.ear
	\adapter\Siebel\samples\Apps\acctinApp.ear

File and directory category	Directories and files
	\adapter\Siebel\samples\Event_pkg\Account.js
	\adapter\Siebel\samples\Event_pkg\BusinessAddress.js
	\adapter\Siebel\samples\Event_pkg\Contact.js
	\adapter\Siebel\samples\Event_pkg\IBMAudit.sif
Notices file	\adapter\Siebel\notices.txt
ISA plugin zip file	\adapter\Siebel\ISAPlugin\com.ibm.com.esupport.client.SS6FE6_RASiebel.zip
IBM Tivoli License Manager (ITLM) file	\adapter\Siebel\5724L80E060000.sys
Log Message zip file	\adapter\Siebel\messages\CWYBS_AdapterFoundation_messages.zip
	\adapter\Siebel\messages\CWYBS_AdapterFoundation_messages.tar
	\adapter\Siebel\messages\CWYEB_SiebelAdapter_messages.zip
	\adapter\Siebel\messages\CWYEB_SiebelAdapter_messages.tar

Note: On Windows, the CWYBS_AdapterFoundation_messages.zip and CWYEB_SiebelAdapter_messages.zip files should be extracted to the <WAS-ROOT>/properties directory. On UNIX and Linux, the CWYBS_AdapterFoundation_messages.tar and CWYEB_SiebelAdapter_messages.tar files should be extracted to the <WAS-ROOT>/properties directory.

Deploying the adapter

The topics in this section explain how to deploy WebSphere Adapter for Siebel Business Applications using the adapter project that you create.

You must install these products before you can install and deploy the adapter:

- WebSphere Integration Developer Version 6.0 (WebSphere Integration Developer)

  For more information, see http://www.ibm.com/software/integration/wid
- WebSphere Adapter for Siebel Business Applications EIS, installed on the same machine as WebSphere Integration Developer
- IBM WebSphere Process Server for Multiplatforms, Version 6.0

  For more information, see http://www.ibm.com/software/integration/wps

To deploy the adapter, you must create a project for the adapter and export the project.

Creating the project

Before you deploy the adapter, you must create an adapter project.

The following steps are required when you create an adapter project:

- Create a project for the adapter
- Add vendor libraries or external dependencies
- Configure the service
- Generate reference bindings for the service

Creating a project for the adapter

The first task in deploying the adapter is to create a J2EE connector project for the adapter.

The resource adapter archive (RAR) file for the Adapter for Siebel Business Applications needs to be imported into WebSphere Integration Developer. This sets up the project in your workspace in WebSphere Integration Developer.

1. Start WebSphere Integration Developer.
2. From the WebSphere Integration Developer main window, switch to the J2EE perspective and select **File** → **Import**.
3. From the Import window, select **RAR file** from the list of import sources and click **Next**.
4. From the Connector Import window, browse to the RAR file location and select the RAR file. The default location for the RAR file is usually the Install directory Deployment folder. Once the RAR file is selected, information is automatically populated in the dialog box.
5. Deselect the check box, **Add Module to an EAR project**.
6. Click **Finish**.
7. From the Confirm Perspective dialog box, click **No**.

The J2EE connector project for the adapter is now created in your workspace.

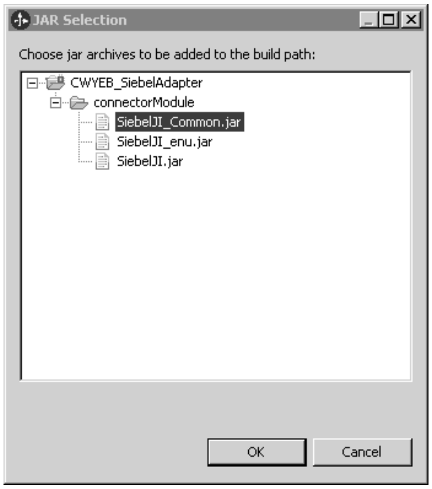
Adding vendor libraries

Third-party libraries and other external software dependencies can be added to the project by using WebSphere Integration Developer.

A connector project for the adapter must be created on your workspace.

Once an adapter project is created on your workspace, you can add external dependencies.

1. Copy external dependencies into the connectorModule folder under the connector project generated in the WebSphere Integration Developer. The following is an example of a destination folder: WID\workspace\CWYEB_SiebelAdapter\connectorModule.
2. Right-click and select **Refresh**.
3. Add the external dependencies to the project as internal libraries.
   a. Under the Connector Projects folder, right click and select **Properties**.
   b. From the Properties window, select Java build path in the left pane.
   c. Click on the **Libraries** tab.
   d. Select **Add JARs**.
   e. From the JAR selection window, expand the nodes and select the JAR files, as seen below.



Choose jar archives to be added to the build path:

JAR Selection window

Based on the Siebel version, the dependency JARs may vary.

f. Click **OK** in the following two windows to conclude the selection.

External dependencies must be added to the EAR file, which is later exported. For more information, refer to the documentation on the WebSphere Integration Developer Web site at http://www.ibm.com/software/integration/wid.

Configuring the service

This topic contains important information about J2C configuration properties used to configure the service and accompanying business objects by using the enterprise service discovery wizard.

Important: During deployment, if you specify J2C activation specification properties when you initially configure the service, those property settings will remain in place. In other words, you cannot update the properties later. The WebSphere Process Server administrative console allows you to update, but the changes to the properties will not take effect (for example, you can see the updated values once you save through WebSphere Process Server administrative console, but the adapter continues to pick up the information from the export file, if the property is set there). If, for any reason, you want to set the J2C activation specification properties after installing the application using the administrative console; you should refrain from setting them during deployment. Note that the J2C connection factory properties can be set during deployment and updated using the administrative console after you install the application.

Related reference

"Enterprise service discovery properties" on page 43
The topics in this section describe the properties that can be configured when you create a project by using the enterprise service discovery wizard.

"Configuration properties" on page 43
This section contains information on properties that can be configured.

Setting connection properties:

By using WebSphere Integration Developer, you can set the values of the metadata discovery connection properties.

After you have created the adapter project, initiate the enterprise service discovery wizard for WebSphere Adapter for Siebel Business Applications and change the values of the metadata discovery connection properties for your database instance.

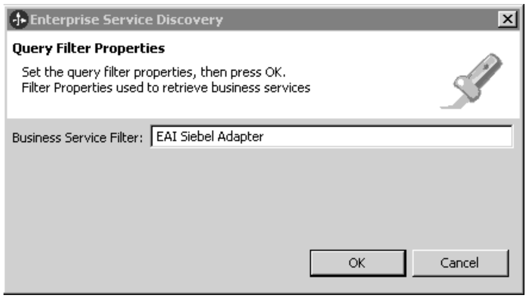
1. From the WebSphere Integration Developer window, switch to the business integration perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed.
2. From the **File** menu, select **New** → **Enterprise Service Discovery**.
3. From the Import Configurations window, select **IBM WebSphere Adapter for Siebel Business Applications (version 6.0.0)** from the CWYEB_SiebelAdapter connector project, and click **Next**.
4. From the Configuration setting for discovery agent window, enter Connection Configuration values in the fields provided and click **Next**.

Finding and discovering enterprise services:

After the EIS connection has been established, you can retrieve the Siebel business service objects that you need from the EIS. The adapter fetches the business services based on the value filled in against the filter property. For example, if you enter a value, such as *EAI*, all the business services that start with *EAI* are retrieved by the adapter. If there is no value provided and you run the query, all the business services are retrieved and listed in a tree structure, where each tree node represents a business service.

1. From the Find and Discover Enterprise Services window, click **Edit Query**.
2. Enter a Business Service Filter value and click **OK**.

   In the following example, the value, *EAI Siebel Adapter* was entered against the *Business Service Filter* property, as seen below. If you do not enter any search parameters, a list of all EIS business services are listed in the tree structure. To narrow the search parameters, you can enter the first few characters in the field click OK.

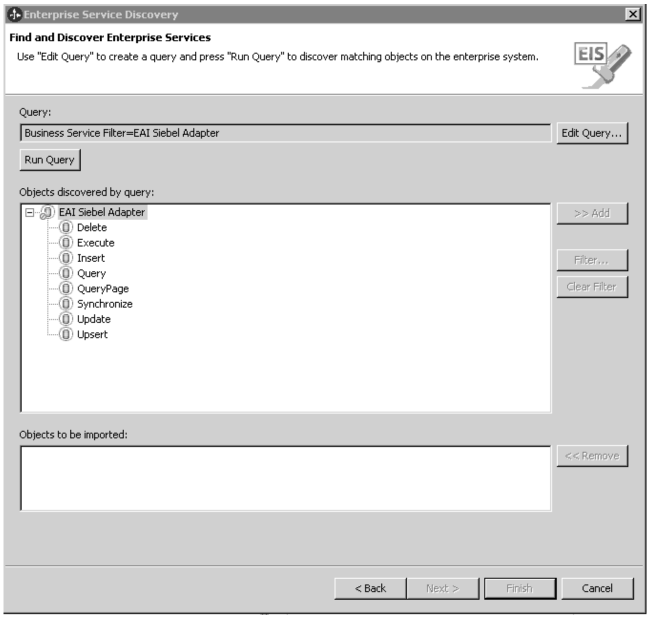
   Query Filter Properties

   Set the query filter properties, then press OK.
   Filter Properties used to retrieve business services

   Business Service Filter: EAI Siebel Adapter

   Query Filter Properties window
3. From the Find and Discover Enterprise Services window, click **Run Query**.

   The query results show the *EAI Siebel Adapter* business service filter displayed in the tree structure as seen below.



Enterprise Service Discovery

Find and Discover Enterprise Services

Use "Edit Query" to create a query and press "Run Query" to discover matching objects on the enterprise system.

Query: Business Service Filter=EAI Siebel Adapter

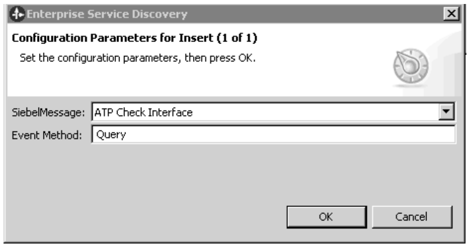
Objects discovered by query:

- EAI Siebel Adapter
  - Delete
  - Execute
  - Insert
  - Query
  - QueryPage
  - Synchronize
  - Update
  - Upsert

Objects to be imported:

Find and Discover Enterprise Services window

4. From the search results, expand the business service node, select the business service method that you want, and click **Add**.



Enterprise Service Discovery

Configuration Parameters for Insert (1 of 1)

Set the configuration parameters, then press OK.

SiebelMessage: ATP Check Interface

Event Method: Query

Configuration Parameters for Internet window

5. For complex arguments, under the business service, select the appropriate integration object.
6. If the service type is inbound, select an **Event Method**. The default value is query.
7. Optionally, you can remove an object by selecting the object from the lower pane of the **Find and Discover Enterprise Service**s window and clicking **Remove**.
8. When you finish making your selections, click **Finish**.

Related reference

"Enterprise service discovery properties" on page 43
The topics in this section describe the properties that can be configured when you create a project by using the enterprise service discovery wizard.

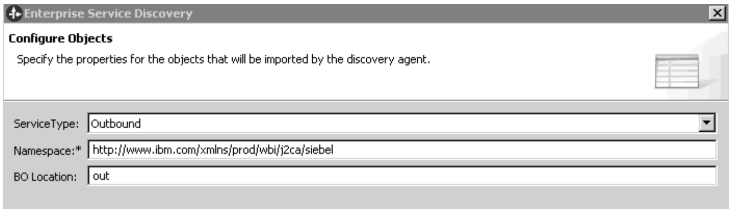
Configuring business objects:

You can configure business objects based on the configuration of Siebel objects.

Once the Siebel objects are selected, you can configure the business objects.

1. Select the **Service Type**, either inbound or outbound.
2. From the Selection Properties window, specify **Namespace**. The name space is initially set to the default for all business objects.

3. Enter a **BO Location** where the business objects can be stored, for example, under the top-level module folder.



Enterprise Service Discovery

Configure Objects

Specify the properties for the objects that will be imported by the discovery agent.

ServiceType: Outbound

Namespace:* http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel

BO Location: out

Configure Objects window

4. For the inbound service type only: From the Generate Artifacts window, do the following:
   a. Select the check box, **Deploy connector with module**.
   b. Select, **Use discovered connection properties**
   c. Enter **Properties for Activation Spec** information in the fields provided.

Related reference

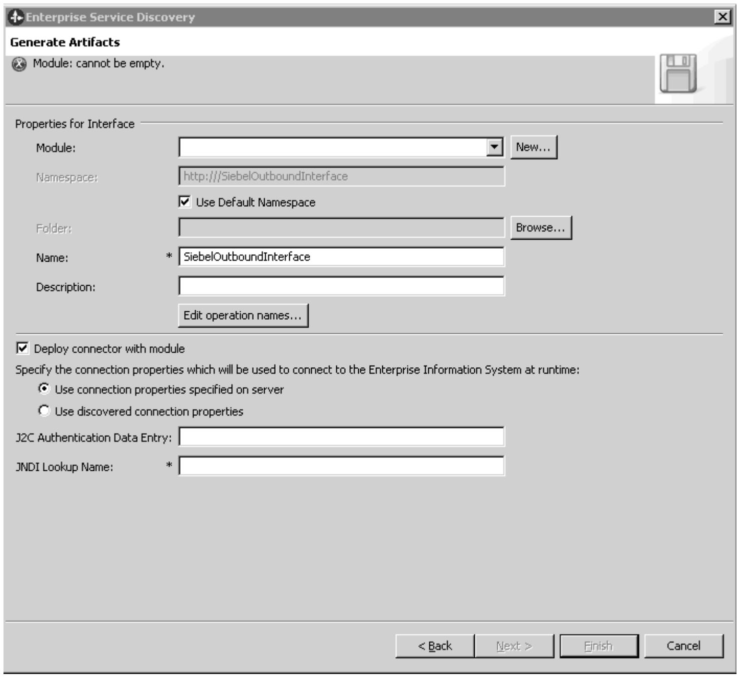
"J2C activation specification properties" on page 47
You can configure the inbound J2C activation specification properties, including bidirectional properties, by using WebSphere Process Server administrative console.

Saving the adapter project:

Create and save a new business integration module.

After you specify selection properties, you need to create the new business integration module where all the artifacts and property values can be saved.

1. From the Generate Artifacts window, create a new module by clicking **New**.



Enterprise Service Discovery

Generate Artifacts

Module: cannot be empty.

Properties for Interface

Module:

Namespace: http:///SiebelOutboundInterface

☑ Use Default Namespace

Folder:

Name: * SiebelOutboundInterface

Description:

☑ Deploy connector with module

Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

● Use connection properties specified on server

○ Use discovered connection properties

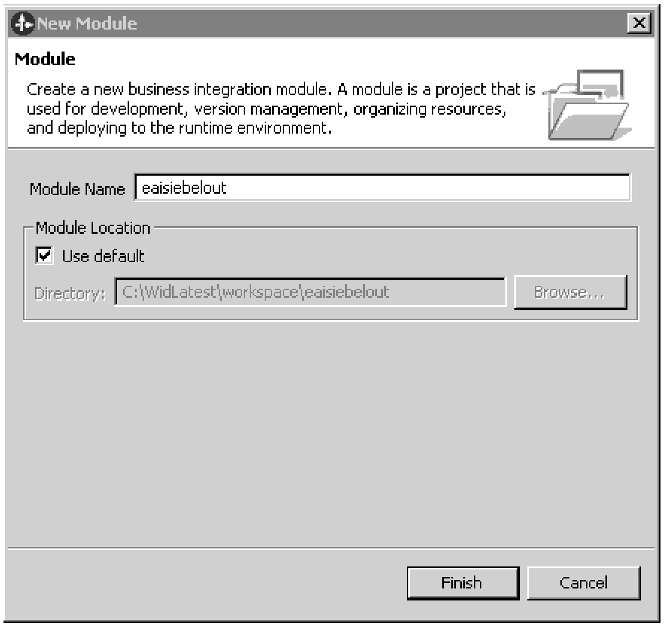
J2C Authentication Data Entry:

JNDI Lookup Name: *

Generate Artifacts window

2. Enter a module name and location, and click **Finish**.

   A module is created at the default location, as seen below, or at the location that you specify.



Module

Create a new business integration module. A module is a project that is used for development, version management, organizing resources, and deploying to the runtime environment.

Module Name: eaisiebelout

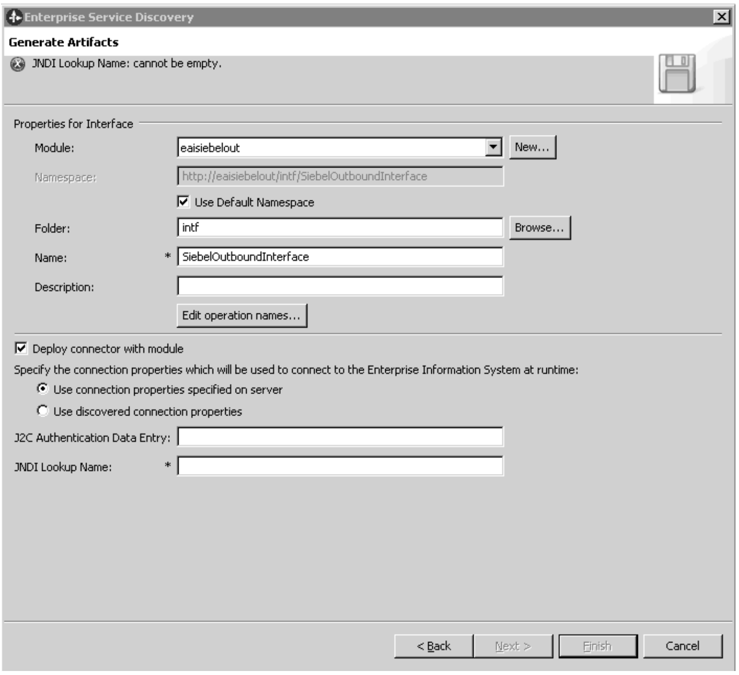
Module Location

☑ Use default

Directory: C:\WidLatest\workspace\eaisiebelout

Module window

3. From the Generate Artifacts window, follow these directions:
   a. Deselect the check box, **Deploy connector with module**.
   b. Select **Use discovered connection properties**, as seen below.



Generate Artifacts

JNDI Lookup Name: cannot be empty.

Properties for Interface

Module: eaisiebelout

Namespace: http://eaisiebelout/intf/SiebelOutboundInterface

☑ Use Default Namespace

Folder: intf

Name: * SiebelOutboundInterface

Description:

☑ Deploy connector with module

Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

● Use connection properties specified on server

○ Use discovered connection properties

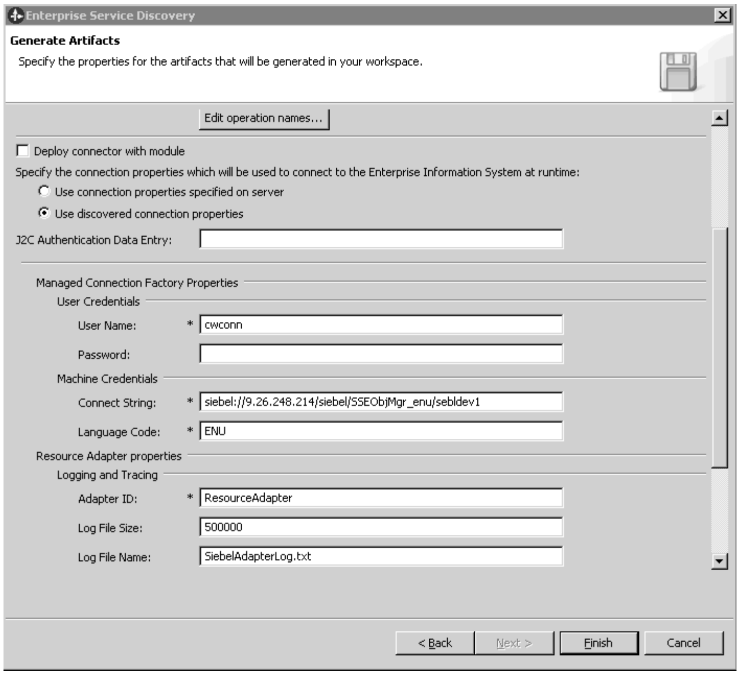
J2C Authentication Data Entry:

JNDI Lookup Name: *

Generate Artifacts window

4. In the Generate Artifacts window, select the following:
   a. User name and password.

      A user name and password are not needed for the outbound service type because an authentication alias is provided. You must create the corresponding authentication alias in WebSphere Process Server for that particular user name and password.
   b. Logging and tracing information for **Adapter ID**, **Log File Size**, and **Log File Name**, as seen below.



Enterprise Service Discovery

Generate Artifacts

Specify the properties for the artifacts that will be generated in your workspace.

□ Deploy connector with module

Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

○ Use connection properties specified on server

● Use discovered connection properties

J2C Authentication Data Entry:

Managed Connection Factory Properties

User Credentials

User Name: * cwconn

Password:

Machine Credentials

Connect String: * siebel://9.26.248.214/siebel/SSEObjMgr_enu/sebldev1

Language Code: * ENU

Resource Adapter properties

Logging and Tracing

Adapter ID: * ResourceAdapter

Log File Size: 500000

Log File Name: SiebelAdapterLog.txt

Generate Artifacts window

c. If supported by the Siebel application, select **Resonate Support**.

5. To create the service component artifacts and business object definitions in the project, click **Finish**.

Note: If you want to create a test environment, you need to generate reference bindings for the service.

For more information on WebSphere Process Server, see IBM WebSphere Process Server for Multiplatforms, Version 6.0

Exporting the project

There are two deployment options for deploying the business integration module project for the adapter. Option A allows you to deploy the adapter by using the WebSphere Process Server administrative console, and option B allows you to deploy the adapter by using the WebSphere Integration Developer.

An adapter project must be created on your workspace.

To complete the deployment process, you must deploy the business integration module project for the adapter. You must choose between two deployments options, Option A or Option B.

- Option A: You can export the project as an enterprise application archive (EAR) file by using the WebSphere Process Server administrative console.
- Option B: You can select the project in WebSphere Integration Developer and deploy it on a server that has been configured to start within the WebSphere Integration Developer. Start the WebSphere Integration Developer, start the server from the WebSphere Integration Developer, and simply step through the process.

Option A: Deploying the adapter from WebSphere Process Server

You can use the WebSphere Process Server administrative console to deploy the adapter.

Create a project for the adapter.

Once you have created a project for the adapter, you must deploy the project. This deployment option shows you how to export the project to the WebSphere Process Server administrative console to export the business integration module project as an EAR file.

1. To export the EAR file, start WebSphere Integration Developer.
2. Select **File** → **Export**.
3. From the list of export resources seen in the Export window, select **EAR file** and click **Next**.
4. Select the EAR project and destination location for the newly created EAR file.
5. To create the EAR file, click **Finish**.

Installing the application from the WebSphere Process Server administrative console:

Use the WebSphere Process Server administrative console to export the business integration module project as an EAR file.

1. Configure the process server that you want to use.
2. Start the WebSphere Process Server administrative console. You can start the WebSphere Process Server administrative console by clicking on the configured server entry and selecting **Run administrative console**.
3. Login to the process server and follow the instructions.
4. To install the new application, from the left pane of the WebSphere Process Server administrative console, select **Install New Application**.
5. From the dialog box that appears, specify the path to the new application EAR file and click **Next**.
6. Select the default mappings by clicking **Next**.
7. Select the default installation options by clicking **Next**.
8. Scroll to the bottom of the screen and select the authentication alias from the drop-down menu.
9. Select the record below and click **Apply**.
10. To complete the deployment, click **Next**.

Once you deploy the adapter, edit the ManagedConnectionFactory properties to match the import file. You can edit the hostname, username, password, and more.

Option B: Deploying the application from the WebSphere Integration Developer

Use the WebSphere Integration Developer to deploy the application on a server.

A project must be created for the adapter.

Once you create the project application, you can deploy the application on a server.

1. Right click on the server and add the project that you have created.
2. Select **Publish**.

Setting up global security and the authentication alias

If you have not already set up global security and the authentication alias, you can do so now by using the WebSphere Process Server administrative console.

The application must be installed.

Once you deploy the WebSphere Adapter for Siebel Business Applications, you can set up the global security and authentication alias.

1. From the WebSphere Administrative Console main window, expand the **Application** node in the left pane.
2. For the next two windows, click **Next**.
3. Select the authentication alias from the drop down menu at the bottom of the screen. The authentication alias should have been created in WebSphere Application Server. If it is not created and ready to use, create it now.
   a. From the left pane, select **Security** → **Global Security**
   b. In the right pane, select and expand the selection for **JAAS Configuration**.
   c. Select **J2C authentication data**.
4. Select the **New** button.
5. Enter required information in the fields provided for user name, password, and description. This is the same user name and password that you will be using for establishing connection to the EIS for the outbound operations.
6. To complete the deployment, click **Next**.

Starting the application

Once the adapter is deployed on your workstation, you can start the adapter.

Select the application on your workstation and click **Start**.

Configuring the adapter

Use the WebSphere Process Server administrative console to configure the adapter properties.

To configure properties using the WebSphere Process Server administrative console, follow these steps:

1. Start the WebSphere Process Server administrative console.
2. Click on the installed application.
3. Go to **Connector Modules**.
4. From the components listed, select the adapter RAR file.
5. Under Additional Properties, select **Resource Adapter**.
6. Under Resource Adapter, select one of the following:
   - **J2C connection factories** To configure the managed connection specification properties, which are used to configure a target EIS instance. Next, select Connection pool properties, Advanced connection factory properties, or Custom properties, depending on which J2C connection factory properties you want to configure. Custom properties are those J2C connection factory properties that are unique to WebSphere Adapter for Siebel Business Applications. The connection pool and advanced connection factory properties are properties that you configure if you are developing your own adapter.

- **J2C activation specifications** Select the name of the J2C activation specification that you want to configure. Next, select the name of the message endpoint property you want to configure and set the value as desired.
- **Custom properties** From the Custom properties page, select the name of the default configuration property you want to configure and set the value.

Related reference

"Configuration properties" on page 43
This section contains information on properties that can be configured.

Troubleshooting the adapter

The topics in this section explain how to access troubleshooting information for the adapter.

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2® and WebSphere products that run on Windows or UNIX® operating systems), enroll in Passport Advantage® in one of the following ways:
  - **Online:** Go to the Passport Advantage Web page and click How to Enroll.
  - **By phone:** For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries®, pSeries®, and iSeries™ environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web page.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

- Determine the business impact of your problem.
- Describe your problem and gather background information.
- Submit your problem to IBM Software Support.

1. Determine the business impact of your problem. When you report a problem to IBM, you will be asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting. Use the following criteria:

Severity	Description
Severity 1	Critical business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
Severity 2	Significant business impact: The program is usable but is severely limited.
Severity 3	Some business impact: The program is usable with less significant features (not critical to operations) unavailable.
Severity 4	Minimal business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.

2. Describe your problem and gather background information. When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:
   - What software versions were you running when the problem occurred?
   - Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
   - Can the problem be recreated? If so, what steps led to the failure?
   - Have any changes been made to the system? (For example, hardware, operating system, networking software, and so on.)
   - Are you currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.
3. Submit your problem to IBM Software Support. You can submit your problem in one of two ways:
   - **Online:** Go to the Submit and track problems page on the IBM Software Support site. Enter your information into the appropriate problem submission tool.
   - **By phone:** For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support will create an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail.

Whenever possible, IBM Software Support will provide a workaround for you to implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM product support Web pages daily, so that other users who experience the same problem can benefit from the same resolutions.

Enabling logging

WebSphere Adapter for Siebel Business Applications maintains a log file that you can view to determine the status of event processing. All events and errors that relate to the adapter are tracked by the log file, along with the date, time, and event for each log entry. Since the adapter logs an error message when it encounters an error or warning condition, the log file is a good source to start troubleshooting problems.

For the Siebel Adapter, logging is enabled through the WebSphere Process Server Administrative Console. Follow the steps below to enable the logging feature.

1. Start WebSphere Application Server.
2. Start the WebSphere Process Server administrative console.

3. Log into the WebSphere Process Server administrative console.
4. From the administrative console, select **Troubleshooting** → **Logs and Trace**.
5. Click **Component** to specify a log detail level for individual components or click **Groups** to specify a log detail for a predefined group of components.
6. Select the logging level that you need. The Logging levels table below describes different logging levels that you can set through the WebSphere Process Server administrative console.

   Note: To view detail-level log events, you must enable the Diagnostic Trace Service. Log events that are at Detail Level or above can be viewed in the *SystemOut* log, the IBM Service log (when enabled), or the Diagnostic Trace Service (when enabled).

   Logging levels

Level	Indicator	Description
Audit	A	Significant event affecting server state or resources
Config	C	Configuration change or status.
Detail	D	General information detailing subtask progress.
Fatal	F	Task cannot continue. Component cannot function.
Info	I	General information outlining overall task progress
Severe	E	Task cannot continue. Component can still function. This also includes conditions that indicate an impending fatal error - i.e. reporting on situations that strongly suggest that resources are on the verge of being depleted.
Warning	W	Potential error or impending error. This also includes conditions that indicate a progressive failure - for example, the potential leaking of resources.

7. Click **Apply** to save your changes.

Enabling tracing

Tracing determines what level of errors or warnings are captured in the adapter log file. You can trace messages regarding adapter processing by defining a tracing level.

The trace levels can be configured in the WebSphere Process Server administrative console. Follow the steps below to enable and set the tracing feature.

1. Start WebSphere Application Server.
2. Start the WebSphere Process Server administrative console.
3. From the administrative console, select **Troubleshooting** → **Logs and Trace**.

4. Select the tracing level that you need. The table below describes different tracing levels that can be set using the WebSphere Process Server administrative console.

   Tracing levels

Level	Indicator	Description
Fine	1	General trace. Includes broad actions being taken by adapter such as establishing a connection to the EIS, converting an event in the EIS to a business object (only key values), processing a business object (only key values).
Finer	2	Detailed trace that provides more granular information on the logic being performed by the adapter including the various API calls being made to the EIS and any parameters or return values.
Finest	3	This is the most detailed level and should include method entry / exit / return values. Complete business object dumps should be included. At this level, all detail needed to debug problems should be provided.

5. Click **Apply** to save your changes.

Enabling the Common Event Infrastructure (CEI)

This topic describes how to enable the Common Event Infrastructure (CEI) for the adapter.

You must publish the IBM WebSphere Adapters Event Definitions file to the CEI catalog before you can set these event definitions. For instruction on how to do this, refer to the CEI documentation found on the WebSphere Process Server web site at http://www.ibm.com/software/integration/wps.

1. Start the WebSphere administrative console.
2. Go to **Troubleshooting** → **Log and Trace** and select <your server name>.
3. There are many options for the General Properties. Select **Change Log Detail Level**, and then select **com.ibm.j2ca.*** for JCA components. Under this section there is a subcomponent for each adapter type:
   - com.ibm.j2ca.flatfile.* (WebSphere Adapter for Flat Files)
   - com.ibm.j2ca.jdbc.* (WebSphere Adapter for JDBC)
   - com.ibm.j2ca.peoplesoft.* (WebSphere Adapter for PeopleSoft)
   - com.ibm.j2ca.sap.* (WebSphere Adapter for SAP)
   - com.ibm.j2ca.siebel.* (WebSphere Adapter for Siebel)
4. Select the component that matches your adapter. Each adapter component has two subcomponents, one for logging and one for CEI. They are:

- *subcomponent name*.log.*adapter id*
- *subcomponent name*.cei.*adapter id*

For example, com.ibm.j2ca.siebel.cei.<AdapterID1>. For each instance of a deployed adapter, the system will show a separate ID.

5. Select the CEI adapter ID that you want to enable.
6. From the drop-down menu, you can choose from the following:
   - off - turn CEI off
   - fine- turn CEI on with Event Content set to Empty
   - finer- turn CEI on with Event Content set to Digest
   - finest- turn CEI on with Event Content set to Full
   - all - same as finest

For information on what each Event Content level means (Empty, Digest and Full), and for more information on using the Common Base Event model and the Common Event Infrastructure, refer to the documentation on the WebSphere Process Server web site at http://www.ibm.com/software/integration/wps

Using the sample applications

This sample application illustrates how to deploy an application package and how the adapter processes business objects. The application presents a scenario for each audience; the application integrator and the data integrator.

For each of the samples provided, the two scenarios presented are as follows:

Sample application scenarios

Scenario	Description	Audience
Scenario 1	• Provides the already-generated artifacts and illustrates how the adapter processes business objects. Since this scenario provides an instance of the adapter that is already configured, you simply import the EAR file in your project using WebSphere Integration Developer. • This scenario is targeted at an audience, such as an application integrator, that is responsible for assembling application components into a solution and preparing this solution for testing and deployment.	Application integrator

Scenario	Description	Audience
Scenario 2	• Illustrates how you use the enterprise service discovery wizard to discover Siebel application components and develop the business objects that the adapter processes. • This scenario is targeted at an audience, such as an data integrator, that has the same responsibilities as the application integrator, but is further responsible for enabling access to a range of data sources for the application developers. You use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the service component architecture artifacts.	Data integrator

Structure of the sample application package for scenario 1

The sample application files are installed when you install the adapter. A comprehensive set of files, for a non-enterprise service discovery sample, are packaged in an archive file that is installed in the Sample folder. This sample illustrates the inbound and outbound operations.

Scenario 1: All-inclusive set of sample files

By using this sample, you can do the following:

- Delete
- Insert or update
- QueryByExample
- QueryByID
- Synchronize an account interface using Siebel Account Business Service
- Update
- Upsert

For Scenario 1, targeted at the Application Integrator, the sample application package includes all the required artifacts, so you do not need to use the enterprise metadata discovery wizard to obtain these. In a real-world deployment of the adapter, you would need to generate all these artifacts and configure the adapter using the enterprise metadata discovery wizard, as illustrated in Scenario 2.

Outbound A comprehensive set of files (non-enterprise service discovery sample: all artifacts are included, there is no need to run the enterprise service discovery wizard to generate files) for outbound operation.

In the samples folder, you save the **accoutApp.ear** file to the location of your choice.

The following files are extracted from the **accoutApp.ear** file:

- A configured instance of the adapter that is deployed by default to the host: **CWYEB_SiebelAdapter.rar**
- Service component architecture (SCA) module with various SCA artifacts:
  - Sca.module
  - Sca.modulex
  - Sca.references
- WSDL: **SiebelOutboundInterface.wsdl**
- Business objects:
  - out/IOAccountInterfaceICAAccount.xsd
  - out/IOAccountInterfaceICBusinessAddress.xsd
  - out/IOAccountInterfaceICCreditPRofile.xsd
  - out/IOAccountInterfaceICRelatedContact.xsd
  - out/IOAccountInterfaceICRelatedIndustry.xsd
  - out/IOAccountInterfaceICRelatedOrganization.xsd
  - out/IOAccountInterfaceICRelatedSalesRep.xsd
  - out/SiebelAccountDeleteAccountInterface.xsd
  - out/SiebelAccountDeleteAccountInterfaceBG.xsd
  - out/SiebelAccountInsertAccountInterface.xsd
  - out/SiebelAccountInsertAccountInterfaceBG.xsd
  - out/SiebelAccountInsertOrUpdateAccountInterface.xsd
  - out/SiebelAccountInsertOrUpdateAccountInterfaceBG.xsd
  - out/SiebelAccountQueryByExampleAccountInterface.xsd
  - out/SiebelAccountQueryByExampleAccountInterfaceBG.xsd
  - out/SiebelAccountQueryByIdAccountInterface.xsd
  - out/SiebelAccountQueryByIdAccountInterfaceBG.xsd
  - out/SiebelAccountSynchronizeAccountInterface.xsd
  - out/SiebelAccountSynchronizeAccountInterfaceBG.xsd
  - out/SiebelAccountUpdateAccountInterface.xsd
  - out/SiebelAccountUpdateAccountInterfaceBG.xsd

The enterprise service discovery wizard generates a BG and a data object for each method selected. While generating the EAR all the methods present on **Siebel Account** were selected, hence, business objects corresponding to *Delete*, *Insert*, *InsertOrUpdate*, *QueryByExample*, *QueryById*, *Synchronize*, and *Update* can be seen. In addition, all the methods have the same integration object. You can see business objects for each integration component present under **Account Interface**.

Inbound: A comprehensive set of files (non-enterprise service discovery sample: all artifacts are included, there is no need to run the enterprise service discovery wizard to generate files) for the inbound operation.

In the Samples folder, you save the **accinApp.ear** file to the location of your choice.

The following files are extracted from the **accinApp.ear** file:

- A configured instance of the adapter that is deployed by default to the host: **CWYEB_SiebelAdapter.rar**
- SCA module with various SCA artifacts:

  - Sca.module
  - Sca.modulex
- WSDL: **SiebelInboundInterface.wsdl**
- Business objects:
  - in/IOAccountInterfaceICAccount.xsd
  - in/IOAccountInterfaceICAccounBG.xsd
  - in/IOAccountInterfaceICBusinessAddress.xsd
  - in/IOAccountInterfaceICCreditProfile.xsd
  - in/IOAccountInterfaceICRelatedContact.xsd
  - in/IOAccountInterfaceICRelatedIndustry.xsd
  - in/IOAccountInterfaceICRelatedOrganization.xsd
  - in/IOAccountInterfaceICRelatedSalesRep.xsd

For inbound, the adapter processes the integration objects directly, hence, the enterprise service discovery wizard generates a BG for the top-level integration component present in the integration object **Account Interfacer**. You can see business objects for each integration component present under the **Account Interface** integration object.

Structure of the sample application package for scenario 2

For scenario 2, targeted at the Data Integrator, the sample application package provides the capability for you to create the service component architecture (SCA) artifacts and configure the adapter using the enterprise service discovery wizard in the WebSphere Integration Developer. You can access copies of all the files that you eventually generate using the enterprise service discovery wizard, as a way of verifying that what you create with the enterprise service discovery wizard is indeed correct and what the application expects in order to run properly.

Scenario 2: EMD requirement

A minimal set of files (requires using the enterprise service discovery wizard to generate comprehensive set of artifacts) are required for inbound and outbound event processing. You can use any zip utility to look at the contents of the files:

- For inbound, the files are archived in \Sample\acctinApp.ear
- For outbound, the files are archived in \Sample\acctoutApp.ear

Unzip **acctout.jar** located in **accountApp.ear**.

The enterprise service discovery wizard is used to generate artifacts, and configure the adapter for deployment and use.

You can view examples of the artifacts that are created when you use the enterprise service discovery wizard.

- Sample Import: **Import1.import** found under **\samples\accountoutApp.ear\accout.jar**
- Sample Export: **SiebelInboundInterface.export** found under **\samples\accountinApp.ear\acctin.jar**
- Other SCA Artifacts can be found under the **module.jars**. For outbound, they are located in **acctout.jar**, and for inbound, they are located in **acctin.jar**.
- WSDL files can be found under the **module.jars**. For outbound they are located in **acctout.jar** and for inbound, they are located in **acctin.jar**.

- Business objects can be found under the **module.jars**. For outbound they are located in **acctout.jar**, and for inbound, they are located in **acctin.jar**.

Deploying and configuring the sample for scenario 1

Scenario 1, targeted at the Application Integrator, provides a configured instance of the adapter and all the necessary SCA artifacts. Once you deploy and configure the sample for scenario 1, you must create the listener MDB class for inbound operations.

Related reference

"Configuration properties" on page 43
This section contains information on properties that can be configured.

Deploying and configuring the sample for scenario 1: The Application Integrator

Scenario 1, targeted at the Application Integrator, provides a configured instance of the adapter and all the necessary SCA artifacts, therefore, you are not required to deploy the package and configure the adapter.

Since this scenario of the sample provides an instance of the adapter that is already configured, you simply follow the steps using WebSphere Integration Developer to deploy and configure the adapter.

The adapter configuration properties are already set to the appropriate values, however, you can change the configuration property values, including the local host of the adapter, to suit your configuration by using the WebSphere administrative console.

1. Find the EAR, **acctoutApp.ear**.
2. Use WinZip to extract the contents to a directory. You should see the files listed.
3. In WebSphere Integration Developer, import the adapter RAR file. Importing the RAR creates a connector project for the Siebel adapter.
4. Update the Java build path so that it includes the Siebel dependency jars.
5. In WebSphere Integration Developer, create a module. Give it the same name as the EAR file without the "App". For example, if the EAR is "acctoutApp.ear", call the module "acctout."
6. Use WinZip to extract the contents of the module JAR file into the module. For example, extract "acctout.jar" into the module called "acctout". In WebSphere Integration Developer, refresh the module.
7. Update the Java build path; go to the Projects tab and check the box next to the connector project.
8. Start the WebSphere Process Server.
9. Once the WebSphere Process Server is started, add the project to the WebSphere Process Server.
10. Start the WebSphere administrative console and ensure that the application has successfully started.
11. If anything has not started successfully, stop the server and restart.
12. Also, ensure that the *Siebel_Auth_Alias* has been properly configured with the user ID and password required to log on to the Siebel instance.
    a. To create the authentication alias on WebSphere Process Server, select **Global Security** → **JAAS Authentication data** → **J2C Authentication alias**
    b. Click **New** and specify the name as **Siebel_Auth_Alias**.

c. Use the assembly editor to edit the authentication alias specified **nodeName/Siebel_Auth_Alias** with the appropriate node name after you create it in WebSphere Process Server.

13. For importing the inbound application *acctinApp.ear* file, repeat steps 1 through 12. In addition, you must create the listener MDB class and bind it in the *acctin* assembly diagram.

The adapter properties are set to default values. You can choose to change these, including the local host of the adapter. For the inbound EAR file, modify the values in the *ActivationSpec*. For the outbound EAR file, modify the values in *ManagedConnectionFactory*.

Creating the listener MDB class for inbound operations

Once you have completed the steps for deploying and configuring the sample for scenario 1, you must create the listener MDB class and bind it in the *acctin* assembly diagram.

You must complete the steps for deploying and configuring the sample for scenario 1, outlined in the previous section, for inbound operations.

The following steps describe how to create the listener MDB class for inbound operations and bind it to the *acctin* assembly diagram.

1. Open the *acctin* assembly diagram under the *acctin* project.
2. Select the component without the implementation link.
3. Click the assembly diagram, **Component1**.
4. Link the *SiebelInboundInterface* to *Component1*, just created.
5. Right click Component1 and select **Generate Implementation>Java**.
6. Add Java code as needed to the method on MDB that support operations. For example, you can see the following methods: *emitCreateAfterImageIOAccountInterfaceICAccount*, *emitUpdateAfterImageIOAccountInterface1cAccount*, and *emitDeleteAfter1mageIOAccountInterfacer1cAccount*
7. Go to the *Projects* tab and select the check-box next to the connector project that you have just created.
8. Start WebSphere Process Server.
9. After WebSphere Process Server has started, export an EAR out of the project and install it through the administrative console.
10. Check through the administrative console to ensure that the application has successfully started.

If the application has not successfully started, stop and restart the server.

Deploying and configuring the sample for scenario 2

Scenario 2 targets the Data Integrator and requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the service component architecture artifacts.

Before you begin to deploy and configure, import the **CWYEB_SiebelAdapter.rar** file into the project.

For this task, you must run the enterprise service discovery wizard within the WebSphere

Integration Developer to set adapter configuration properties.

1. In the J2EE perspective, select **File** → **Import** → **RAR file**.
2. Browse to the folder where the **CWYEB_SiebelAdapter.rar** file exists and select the file.
3. Click **Finish**.
4. Switch to the business integration perspective and select **File** → **New** → **Enterprise Service Discovery**.
5. From the Service Discovery window, select **Siebel ESD Adapter** and click **Next**.
6. Provide connection configuration properties for the following:
   - Prefix
   - Siebel repository
   - User name
   - Password
   - Connect string
   - Language code
   - If needed, check **Bi-di transformation**
7. Click **Next**.
8. From the Metadata Query window, click **Run Query**.
9. Specify the appropriate value for the **Business Service Filter** field, for example, EAI Siebel Adapter.
10. Click **OK**.
11. Expand the node (+) next to the business service to display the methods available.
12. Select the method that you want to generate definitions against and click **Add selected** to the right of the pane.
13. From the Configuration Parameters window, select the integration object whose definitions will be generated against the complex argument. There will be no complex arguments in this window if the Integration Object type is already set. Provide the event method value, for example, **QueryByExample**. This is needed only when the service type operation is inbound.
14. Individually select additional metadata objects needed and click **Add selected**.
15. Click **Next**.
16. From the Selection Properties window, select the required service type, either inbound or outbound. A default namespace provided. You can provide a different one if needed. Provide the folder name, against business object location, where the .xsds generated are stored and click **Next**.
17. From the Saving Properties window, specify the module name where the artifacts are to be saved. In this example, since there was no module created earlier, click on **New** button to create a new Business Integration module. Specify a folder name within the module where Service Description will be saved and click Finish. Ensure this is not the same one that you provided against business object Location property earlier.
18. Optionally, you can edit the generated method names by selecting the check box for **Edit generated method names**.
19. To generate the artifacts, click **Finish**. After the adapter has been deployed, you can change the property values using the WebSphere Process Server administrative console.

Related reference

"Enterprise service discovery properties" on page 43
The topics in this section describe the properties that can be configured when you create a project by using the enterprise service discovery wizard.

Running the sample application

After you deploy and configure the sample application package and adapter, you can run the application to illustrate how the adapter supports outbound processing of business objects.

Ensure that you are in the Business Integration Perspective. Right-click on the **acctout project → Test → Test module** This brings up the test client.

Follow the instructions below to run the sample application.

1. Select the operation as needed, there are seven from which to choose. As the names of the various operations would suggest, each one performs the corresponding operation. For example, insertSiebelSiebelAccountInsertAccountInterface is used to insert a new Account, updateSiebelSiebelAccountUpdateAccountInterface is used to update, and so on.
2. Set the values as needed. For example, you can set insertSiebelSiebelAccountInsertAccountInterface, updateSiebelSiebelAccountUpdateAccountInterface, and deleteSiebelSiebelAccountDeleteAccountInterface for operations.
3. Click **Continue**.
4. Confirm that you were successful by checking the returned object.
5. For inbound operations only. Once the application starts, check the following to ensure success: For Inbound operations, testing of the sample applications ends at this point.
   a. Navigate to the WebSphere Process Server folder: **\bi-v6\profiles\default\logs**
   b. Examine the adapter trace file, **Siebelinboundtrce.txt**, and make sure it contains line, *emit<operation>SiebelAccountAccountInterfaceSuccessful*, when <operation> can be create, update, or delete.
   c. Examine the *IBM-Events* in Siebel to ensure that it contains events.
   d. Examine the data in the Siebel EIS by checking the event status. If the transaction was successful, the status will be set to 1. These steps conclude running the sample application for inbound operations.

References

The topics in this section allow you to access reference information for the adapter.

Creating an event table in the Siebel application

To create an event table, you must be familiar with Siebel tools functionality. For detailed information on any of the steps, refer to the documentation for Siebel tools.

This procedure uses the Siebel Sales Enterprise application as an example. Substitute all references to Siebel Sales Enterprise with the name of the Siebel application in use. For this example **IBM Events** has been used for the event table name. Based on your needs, this name can be changed, but it is easier to use the same names for both.

Related concepts

"Adapter-specific information" on page 16
Specific to WebSphere Adapter for Siebel Business Applications, you must create an event table in the Siebel application.

Step 1: Setting up columns

There are five steps required to create an event table in the Siebel Sales Enterprise application. The first step involves setting up columns in the event table.

To create an event table and an object in the Siebel application, perform the following procedure using the Siebel tools:

1. Create an IBM project and lock your project. Except for event triggers, you must complete all Siebel customization under the new project.
2. If you are installing multiple connectors, create multiple tables with different names.
3. Using New Object Wizard, create a stand alone table named CX_IBM_Events.
4. Create the following columns in your new table:
   - Column Name/User Name
   - Type
   - Length
   - Physical type
   - Required
   - Nullable
   - Status

Column example

Column Name/ User Name	Type	Length	Physical Type	Required	Nullable	Status
DESCRIP TION	Data (public)	255	Varchar		Yes	Active
EVENT_ID	Data (public)	30	Varchar	Yes		Active
EVENT_ TIME STAMP	Data (public)	7	Date Time	Yes		Active
EVENT_ TYPE	Data (public)	20	Varchar	Yes		Active
OBJECT _KEY	Data (public)	255	Varchar	Yes		Active
OBJECT_ NAME	Data (public)	255	Varchar	Yes		Active
PRIORITY	Data (public)	10	Number		Yes	Active
STATUS	Data (public)	20	Number	Yes		Active

Step 2: Creating a business component

Follow these steps to create a business component.

1. Create a new Business Component (BC), **IBM Events**, based on your new table. All fields are single value fields.
2. Create a new business object (BO) named **IBM Events**.
3. Associate **IBM Event BC** to the **IBM Event BO**.
4. Create **IBM Event List View Applet** based on **IBM Event BC**.
5. Create **IBM Event List View** based on **IBM Event BO**.
6. Create **IBM Event Screen** and associate it to the **IBM Event List View** in screen view.
7. Create a page tab as follows:
   a. Access the **Application** → **Siebel Sales** → **Page tab**.
   b. Right-click and select **New Record** from the menu.
   c. Enter **IBM Event** as the screen name and **IBM Event** as the text name.
   d. For the sequence, enter a number greater than the rest of the sequence numbers. This selection determines where the tab is displayed in the application.
   e. Leave the inactive field unchecked.
8. Create a screen menu item as follows:
   a. From the menu, select **Application** → **Siebel Sales Enterprise** → **Screen menu**.
   b. Right-click and select **New Record**.
   c. Enter **IBM Event** as the screen name and **IBM Event** as the text name.
   d. For the sequence, enter a number greater than the rest of the sequence numbers. This selection determines where the tab is displayed in the application.
   e. Leave the inactive field unchecked.
   f. Go to the screen menu item, **Locale** and create a new record for **IBMEvent**.

Step 3: Applying a schema to the event table

Follow these steps required to apply a schema to the event table.

In this set of steps, you apply the physical schema for the new tables to your local database.

1. Query for the new table, **CX_IBM_EVENT**.
2. Select the current query to create a physical schema. Leave the table space and index space blank.
3. To activate the new schema, click **Activate**.
4. Add or modify the Siebel VB or e-scripts for the BC that corresponds to the business objects used at your site. The Siebel script triggers the event notification for business objects. (Samples are in the Sample folder) If you are planning to use multiple connectors, make sure the correct name is specified in the Siebel script. In the Siebel script, make sure the event is not created for the adapter user name, it will create a repeat effect.
5. Compile the updated and locked projects on your local database to create a new Siebel repository (.srf) file.

Step 4: Creating a new account in the IBM event list view

Follow these steps to create a new account.

To perform the following steps, you must have administrative privileges to your local database.

1. Open Siebel Sales Enterprise on your local database.
   a. Create a new view called IBM Event List View. Tip: You can copy the view name from tools and paste it into the **View Name** field.
   b. Create a new responsibility called IBM Responsibility for IBM Event List View.
   c. Add the employees or teams who are responsible for reviewing events to the newly created IBM Responsibility.
   d. Create the IBMCONN (or your adapter user name) user and add it to IBM Responsibility and Administrative Responsibility.
2. Test the application in your local environment.
3. Ensure that you have visibility to IBM Event List View and that an event is generated in the view after you create a record in Supported object. For example, create a new account in Siebel and check that a new account event appears in the IBM Event List View.
4. Check in the new and updated projects to your development server.
5. Activate the new tables in the development database.
6. Compile a new Siebel.srf file in the server.
7. Enable Enterprise Application Integration by going to **Sitemap > Server Administration > Component Group** and selecting **Enable**.

Now that the event table is created in Siebel, you can set the timeout value for the Siebel Java Beans ^{TM}.

Step 5: Setting a timeout value

Set a timeout value for the Siebel JavaBeans™.

You must have previously created a Siebel event table.

To set the Siebel JavaBeans:

1. Select, **Site Map** → **Server Admin** → **Components** (Sales Object Manager).
2. In the lower applet, go to **Component Parameter** and enter a timeout value.

You have now set the timeout value.

Generating reference bindings for the service (test environment only)

Reference bindings are used by other WebSphere Business Integration SCA components to access the adapter. You create a reference to the adapter from the project module by linking the adapter to other server processes. This is required in a testing environment only. It is not necessary when deploying the adapter in a production environment.

An adapter project must be created and configured on your workspace.

Once an adapter project is created, you must generate reference bindings to bind to the service component.

1. From the WebSphere Integration Developer main window under **All Resources**, select the module name for your project.
2. Double-click on your project folder. The project module appears in the upper-right pane.

3. Select the **Import** icon. Hold your cursor over the icons on the left side of the pane until you find the icon that you want. When you select the Import icon, several more icons appear.
4. Double-click on the **Standalone References** icon. The Standalone Reference box appears in the right pane.
5. Drag and drop the yellow bulb around the Standalone Reference box onto the Import Module to bind them.
6. From the Add Wire dialog box, click **OK**.
7. Save the outbound service type file.

Configuration properties

This section contains information on properties that can be configured.

Related concepts

"Configuring the service" on page 20
This topic contains important information about J2C configuration properties used to configure the service and accompanying business objects by using the enterprise service discovery wizard.

Related tasks

"Configuring the adapter" on page 27
Use the WebSphere Process Server administrative console to configure the adapter properties.

"Deploying and configuring the sample for scenario 1" on page 36
Scenario 1, targeted at the Application Integrator, provides a configured instance of the adapter and all the necessary SCA artifacts. Once you deploy and configure the sample for scenario 1, you must create the listener MDB class for inbound operations.

Enterprise service discovery properties

The topics in this section describe the properties that can be configured when you create a project by using the enterprise service discovery wizard.

Related concepts

"Configuring the service" on page 20
This topic contains important information about J2C configuration properties used to configure the service and accompanying business objects by using the enterprise service discovery wizard.

Related tasks

"Deploying and configuring the sample for scenario 2" on page 37
Scenario 2 targets the Data Integrator and requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the service component architecture artifacts.

"Finding and discovering enterprise services" on page 21
After the EIS connection has been established, you can retrieve the Siebel business service objects that you need from the EIS. The adapter fetches the business services based on the value filled in against the filter property. For example, if you enter a value, such as *EAI*, all the business services that start with *EAI* are retrieved by the adapter. If there is no value provided and you run the query, all the business services are retrieved and listed in a tree structure, where each tree node represents a business service.

Bidirectional properties:

You can configure bidirectional properties using the enterprise metadata discovery (EMD) tool for the adapter.

Bidirectional (BiDi) properties for EMD

BiDi property name	Possible values	Default value	Meaning
BiDiEISDirection	LTR, RTL, contextualLTR, contextualRTL	LTR	The orientation component of the BiDi format specification
BiDiEISNumeric Shaping	Nominal, national, contextual	Nominal	The numeric shaping component of the BiDi format specification
BiDiEISOrdering Schema	Implicit, visual	Implicit	The ordering scheme of the BiDi format specification
BiDiEISShaping	Nominal , shaped, initial, middle, final, isolated	Nominal	The shaping component of the BiDi format specification
BiDiEISSymmetric Swapping	True or false	True	The symmetric swapping component of the BiDi format specification
BiDiTransformation	True or false	True	If true, the BiDi support is turned on. If false, BiDi support is turned off

Custom properties:

You can configure custom properties for the adapter.

Custom Properties

Property	Description
Connect string	Determines the Siebel instance information. For Siebel 7.5, Siebel://GatewayServer/enterprisename/ObjectManager/SiebelServer
Event component name	Specifies the name of the Siebel Component for the event table.
Language code	The Siebel instance language.
Password	The password for the corresponding user name.
Resonate support	Specifies whether the adapter should use Attach and Detach calls on the SiebelDataBean.
User name	The user name to login to the Siebel.

Data descriptions:

Data descriptions are generated from the Siebel business methods service and integration objects by the adapter. They are represented as XML schemas.

The following apply to data descriptions:

- Business objects map to complex type definitions
- Properties of the business object map to element type definitions
- Application-specific information for the business object is contained in annotations for the complex type
- Application-specific information for each property in the business object is contained in annotations for the element types

A business graph is generated for every method selected under a business service when the service type is outbound. A business service contains a complex type, the business object, and the properties for the method arguments. The data type of the arguments can be a string or a complex type. The arguments with the string data type become properties of type string in the business object.

A complex type argument is represented by an input or output integration object. The input or output integration is based on the details entered in the process or retrieved from the EIS repository. The adapter automatically generates the business objects for the integration components under these integration objects. The business object hierarchy is created based on the relationship between the integration components, for example, parent and child.

When a business service does not have any arguments for a method, no properties are generated under the business object. The adapter automatically sets the application-specific information at the business object and properties levels.

Note: The template for the application-specific properties for the business object and the property levels is defined in the metadata schema for the Siebel adapter. The name of the schema file is **SiebelASI.xsd**. The generated XML schemas refer to this template.

Metadata discovery adapter type:

The only adapter type supported by the enterprise service discovery wizard is the Siebel adapter type.

Metadata discovery adapter type properties

Property	**Description**
Description	This is the description for the adapter. Value: “IBM WebSphere Adapter for Siebel Business”
Display name	This is the display name of the adapter. Value: “IBM WebSphere Adapter for Siebel Business”
ID	This is the ID for the adapter type. Value: “Siebel”
Vendor	This is the vendor providing the adapter. Value: “IBM”
Version	This is the version of the adapter. Value: “6.0.0”

Metadata discovery connection type:

WebSphere Adapter for Siebel Business Applications enterprise service discovery wizard supports only one connection type; the Siebel connection type.

Metadata discovery connection-type properties

Name	Property	Globalized	Possible Values/Is Required
Connect string	The connection string needed to connect to the Siebel EIS. Required property.	Yes	Protocol:// machinename / enterprisename/ object manager/ servername For Siebel 7.7.x: Protocol:// machinename: portnumber/ enterprisename/ objectmanager Required
Language code	The Siebel EIS instance language.	No.	ENU/Required
Password	The user name password.	Yes.	Siebelpassword/ Required
Prefix	The prefix for the object generated for business service methods when the service type is Outbound. A prefix is not required.	No.	IBM/Optional
Siebel repository	The name of the Siebel repository. This is a required property.	No.	MyRepository/ Required
User name	The user account for the Siebel EIS. This is a required property.	Yes.	User 1/Required

Method descriptions:

The following table contains inbound and outbound method description properties.

Properties for inbound methods

Name	Description
EIS function name	The name of the method that is returned by the function selector.
Input data description	The business object that is input to the method.
Name	The name of the method.
Output data description	The business object that is output from the method.

Properties for outbound methods

Property	Description
Input data description	The business object that is input to the method.
InteractionSpec	The instance of class representing the Siebel adapter interaction spec.
Name	The name of the method.
Output data description	The business object that is output from the method.

J2C activation specification properties

You can configure the inbound J2C activation specification properties, including bidirectional properties, by using WebSphere Process Server administrative console.

J2C activation specification properties

Property	Type	Description
AutoCreateEDT	Boolean	Flag that indicates whether the adapter should create the EDT table automatically if it does not already exist. The default is true
Connect string	String	Determines the Siebel instance information. For Siebel 7.5, Siebel: //GatewayServer/ enterprisename/ ObjectManager/ SiebelServer For Siebel 7.7, Siebel: //GatewayServer: portnumber/ enterprisename/ ObjectManager
DeliveryType	String	Ordered or unordered. This determines the order in which the events are published. Ordered means one at a time, while unordered means all at once. The default is ordered
EDTDatabaseName	String	The name of event recovery database
EDTDriverName	String	The name of the XA database driver to use to connect to the event distribution table for inbound events. For example: com.ibm.db2j.DB2jXAData Source. If no value is present, the event manager cannot perform recovery

Property	Type	Description
EDTSchemaName	String	The schema used for automatically creating the event recovery database
EDTTableName	String	The name of event recovery table
EDTURL	String	The URL to the EDT database
EDTUserName	String	The user name for connecting to the database
EDTUserPassword	String	The user password for connecting to the database
Event component name	String	Specifies the name of the Siebel component for the event table
Language code	String	The Siebel instance language
Password	String	The password for the corresponding user name
PollPeriod	Integer that is equal to or greater than 0	The rate in milliseconds at which to poll the EIS event store for new inbound events. If 0, the adapter will not wait between cycles. The poll cycle is established at a fixed rate, meaning that is an execution of the poll cycle is delayed (for example, the prior poll cycle takes longer than expected to complete the next cycle will occur immediately to catch up.) This is a required property. The default is 500
PollQuantity	Integer that is greater than 0.	This property is used to determine the number of events to deliver to each endpoint per poll cycle. It is a required property
RetryInterval	Integer	Used to enable connection reestablishment for inbound delivery.
User name	String	The user name to login to the Siebel.

Bidirectional (BiDi) J2C activation specification properties

BiDi property	Type	Possible values	Default value	Description
Connection string BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	Specified BiDI format for connection string
EDT BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	Specified BiDI format for EDT properties
EDT URL BiDi special format	String	WIN_DIR, WIN_NET_DIR, UNIX_DIR, MVS_DIR, URL_WIN_DIR, URL_UNIX_DIR, URL_FTP_HTTP, EMAIL_FOLDER, JDBC_URL_SQL, SIEBEL _ CONNSTR _77, NORMAL _STRING	NORMAL _STRING	Specifies the name of category for values subject to special treatment during BiDi transformation invocation.
ESI BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	Specifies BiDI format used by EIS (Siebel) to store content business data
EIS BiDi special format	String	WIN_DIR, WIN_NET_DIR, UNIX_DIR, MVS_DIR, URL_WIN_DIR, URL_UNIX_DIR, URL_FTP_HTTP, EMAIL_FOLDER, JDBC_URL_SQL, SIEBEL _CONNSTR_77, NORMAL _STRING	NORMAL _STRING	Specifies the name of the category for values subject to special treatment during BiDi transformation invocation.

BiDi property	Type	Possible values	Default value	Description
Event component name BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	
Metadata BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	
Password BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	
Skip BiDi transformation	String	True, false, <empty string>	<empty string>	
Skip BiDi transformation for connection string	String	True, false, <empty string>	<empty string>	
Skip BiDi transformation for EDT	String	True, false, <empty string>	<empty string>	
Skip BiDi transformation for EDT URL	String	True, false, <empty string>	<empty string>	
Skip BiDi transformation for event component name	String	True, false, <empty string>	<empty string>	
Skip BiDi transformation for password	String	True, false, <empty string>	<empty string>	

BiDi property	Type	Possible values	Default value	Description
Skip BiDi transformation for user name	String	True, false, <empty string>	<empty string>	Used to control invocation of BiDi transformation on the user name. If true, the transformation is invoked. If false the transformation is not invoked. An empty string is used for the invocation of the lookup mechanism
User name BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	Specified BiDI format for the user name

Related tasks

"Configuring business objects" on page 22
You can configure business objects based on the configuration of Siebel objects.

J2C connection factory properties

You can configure the J2C connection factory (ManagedConnectionFactory) properties, including bidirectional properties, by using the WebSphere Process Server administrative console.

J2C connection factory properties

Property	Description
User name	The user name for logging into the Siebel EIS.
Password	The password for the user.
Connect string	This determines the Siebel instance information. Siebel://GatewayServer/ enterprisename/ ObjectManager/ SiebelServer
Language code	The Siebel instance language. For example, enu for English, or jpn for Japanese.

Bidirectional (BiDi) J2C connection factory properties

BiDi property	Type	Possible values	Default values	Description
EIS BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	Specifies the BiDI format used by the Siebel EIS to store content business data
Metadata BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	Specifies the BiDi format used by the Siebel EIS to store metadata (for example, business service names)
Skip BiDI transformation	String	True, false, <empty string>	<empty string>	Used to control the invocation of BiDi transformation on the user name. If true, the transformation is invoked. If false, the transformation is not invoked. An empty string is used to invoke the lookup mechanism
Password BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format	ILYNN	Spedified BiDi format for the password

J2C adapter properties

You can configure the J2C adapter properties, including bidirectional properties, by using WebSphere Process Server administrative console.

J2C adapter properties

Property	Type	Description
BONameSpace	String	Namespace for the business object definitions to be used by this adapter. This value should be taken from the value you provided during the enterprise service discovery process. This property is required.

Property	Type	Description
LogFileName	String	The full path of the log file. This property is required. For example, *LogFileName* can be c:\logs\log.txt.
LogNumberOfFiles	Integer	The number of log files to use. When a log file reaches its maximum size, another log file is started. If no value is specified, the value is set to 1.
LogMaxFileSize	Integer	Size of the log files in kilobytes. If no value is specified, the files have no maximum size.
ResonateSupport	Boolean	Specifies whether the adapter should use Attach and Detach calls on the SiebelDataBean.

Bidirectional (BiDi) J2C adapter properties

BiDi property	Type	Possible value	Default value	Description
EIS BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format		Specifies the BiDI format used by the Siebel EIS to store content business data
Metadata BiDi format	String	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format		Specifies the BiDi format used by the Siebel EIS to store metadata (for example, business service names)
Skip BiDi transformation	String	True, false, <empty string>		Used to control invocation of BiDi transformation on content data. If true, the transformation is invoked. If false, the transformation is not invoked. An empty string is used to invoke the lookup mechanism

BiDi property	Type	Possible value	Default value	Description
EIS BiDi special format	String	WIN_DIR, WIN_NET_DIR, UNIX_DIR, MVS_DIR, URL_WIN_DIR, URL_UNIX_DIR, URL_FTP_HTTP, EMAIL_ FOLDER, JDBC_URL_SQL, SIEBEL _CONNSTR _77, NORMAL _STRING	NORMAL _STRING	Specifies the name of the category for values subject to special treatment during BiDi transformation invocation
Turn BiDi off	Boolean	True, false	True	If true, BiDi transformation support is turned off. If false, BiDi transformation support is turned on

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, a nd represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

i5/OS
IBM
the IBM logo
AIX
AIX 5L
CICS
CrossWorlds
DB2
DB2 Universal Database
Domino
HelpNow
IMS
Informix
iSeries
Lotus
Lotus Notes
MQIntegrator
MQSeries
MVS
Notes
OS/400
Passport Advantage
pSeries
Redbooks
SupportPac
WebSphere
z/OS

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

This product includes software developed by the Eclipse Project (http://www.eclipse.org/).

WebSphere Adapters, Version 6.0

Printed in USA