

IBM WebSphere Adapters
Version 7 Release 5 Fix Pack 3 (7.5.0.3)

*IBM WebSphere Adapter for Siebel
Business Applications User Guide
Version 7 Release 5 Fix Pack 3
(7.5.0.3)*



IBM WebSphere Adapters
Version 7 Release 5 Fix Pack 3 (7.5.0.3)

*IBM WebSphere Adapter for Siebel
Business Applications User Guide
Version 7 Release 5 Fix Pack 3
(7.5.0.3)*



Note

Before using this information and the product it supports, read the information in "Notices" on page 137.

November 2012

This edition applies to Version 7, Release 5, Fix Pack 3 (7.5.0.3) of IBM WebSphere Adapter for Siebel Business Applications and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email <mailto://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.

© Copyright IBM Corporation 2006, 2012.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Overview of WebSphere Adapter for Siebel Business Applications 1

Hardware and software requirements	2
Technical overview	2
Outbound processing	3
Inbound processing	7
Business objects	13
Service Bean	13
The J2C Bean wizard	13
Standards Compliance	14
Log and Trace Analyzer	15

Chapter 2. Planning for adapter implementation 17

Before you begin	17
Security	17
Support for protecting sensitive user data in log and trace files	17
User authentication	18
Deployment options	19
WebSphere Adapters in clustered environments	22

Chapter 3. Configuring the module for deployment. 25

Configuring the Siebel application to work with the adapter	25
Launching the J2C Bean wizard	27
Configuring the connector dependencies.	28
Setting connection properties for the J2C Bean wizard	28
Configuring the module for outbound processing	31
Selecting business objects	31
Selecting business services	33
Configuring the selected objects	34
Setting deployment properties and generating artifacts.	35
Generating the EJB or JSP project	38
Configuring the module for inbound processing	39
Selecting business objects	39
Selecting business services	41
Configuring the selected objects	42
Setting deployment properties and generating artifacts.	43

Chapter 4. Deploying the module 47

Deployment environments	47
Deploying the module for testing	47
Configuring the connector dependencies.	47
Adding the module to the server	48
Deploying the module for production	49
Configuring the connector dependencies on the server	49

Installing the RAR file (for modules using stand-alone adapters only)	50
Exporting the module as an EAR file	51
Installing the EAR file	52
Deploying the module in a clustered environment	53
Deploying module embedded in the application	53
Deploying module at node level with embedded activation specification	54
Deploying module at node level with JNDI activation specification	55

Chapter 5. Configuring the application on WebSphere Application Server 59

Configuring logging and tracing	59
Configuring logging properties	59
Changing the log and trace file names	61
Changing configuration properties for embedded adapters	62
Setting managed (J2C) connection factory properties for embedded adapters	62
Changing configuration properties for stand-alone adapters	63
Setting resource adapter properties for stand-alone adapters	63
Setting managed (J2C) connection factory properties for stand-alone adapters	64
Setting activation specification properties for stand-alone adapters	65
Adding dependency libraries to the deployed resource adapter.	67
Stand-alone deployment	67
EAR deployment	67

Chapter 6. Troubleshooting and support 69

Techniques for troubleshooting problems	69
First-failure data capture (FFDC) support	71
Solutions to some common problems	71
Support	77
Searching knowledge bases (Web search)	77
Getting Fixes	78
Self-help resources	79

Chapter 7. Reference information 81

Business object information	81
Application-specific information	81
Supported operations	83
Naming conventions	84
Outbound configuration properties	87
Connection properties for the wizard	88
Resource adapter properties	95
Managed connection factory properties	100
Interaction specification properties	108
Inbound configuration properties	108
Connection properties for the wizard	110

Resource adapter properties 117

Activation specification properties 122

Globalization 133

Globalization and bidirectional data transformation 133

Properties enabled for bidirectional data transformation 135

Notices 137

Programming interface information 139

Trademarks and service marks 139

Index 141

Chapter 1. Overview of WebSphere Adapter for Siebel Business Applications

With WebSphere® Adapter for Siebel Business Applications, you can create integrated processes that exchange information with a Siebel Business Applications, without special coding.

With the adapter, an application can send requests to the Siebel Business Applications server or receive notifications of the changes from the server and also can get application data based on the notifications.

The adapter creates a standard interface to the applications and data on the Siebel Business Applications server. The application need not understand the implementation of the Siebel Business Applications or the data structures on the server. An application, for example, can send a request to the server, to query or update an Account record, represented by a Siebel business component instance. It can also receive events from the server, for example, a notification about the updates to a customer record. This provides you with improved business workflow and processes to help manage your customer relations.

The adapter complies with the Java™ Platform, Enterprise Edition (JEE) Connector Architecture (JCA). JCA standardizes the way application components, application servers, and Siebel applications, such as Siebel Business Applications server, interact with each other. The adapter makes it possible for JCA-compliant application servers to connect and interact with the Siebel Business Applications server in a standard way by using Siebel Java Data Bean.

The following example assumes that you are setting up an adapter using Rational® Application Developer for WebSphere Software and deploying the application that includes the adapter to WebSphere Application Server.

Suppose your company uses Siebel Business Applications to coordinate its business operations (for example, customer, and order management). Siebel Business Applications includes a business function that returns a list of customers in response to a range of customer IDs. An application component might be able to use this function as part of an overall business process. For example, when the promotions department within the company sends advertising material to customers, and as part of that process, must first obtain a list of customers.

The Siebel Business Applications function does not have a web service interface. Therefore, the application component used by the promotions department must understand the low-level API and data structures of the Siebel Business Applications function to call the function. Information technology resources must create the links between the component and the Siebel Business Applications function.

With the adapter, however, you can automatically generate an interface to the Siebel Business Applications function to hide the lower-level details of the function. Depending on how you want to use the adapter, you can embed it with the deployed application or install it as a stand-alone component, to be used by more than one application. The embedded adapter is part of an application that is deployed to WebSphere Application Server. The application component interacts with the adapter instead of with the Siebel Business Applications function.

The adapter configuration, which you generate with the J2C Bean wizard within Rational Application Developer for WebSphere Software, uses a standard interface and standard data objects. The adapter takes the standard data object sent by the application component and calls the Siebel Business Applications function. The adapter then returns a standard data object to the application component. The application component need not interact directly with the Siebel Business Applications function; it is the adapter that calls the function and returns the results.

For example, the application component that needs the list of customers sends a standard business object with the range of customer IDs to the adapter. In return, the application component receives the results (the list of customers) in the form of a standard business object. The application component need not know how the function worked or how the data was structured. The adapter performs all the interactions with the actual Siebel Business Applications function.

Similarly, the client application might want to know about a change to the data on the Siebel Business Applications server (for example, a change to a particular customer). You can generate an adapter component that polls for such events on the Siebel Business Applications server and notify client applications of the update. In this case, the interaction begins at the Siebel Business Applications server.

Hardware and software requirements

The hardware and software requirements for WebSphere Adapters are provided on the IBM® Support website.

To view hardware and software requirements for WebSphere Adapters, see <http://www.ibm.com/support/docview.wss?uid=swg27006249>.

Additional information

The following links provide additional information you might need to configure and deploy your adapter:

- The compatibility matrix for WebSphere Business Integration Adapters and WebSphere Adapters identifies the supported versions of required software for your adapter. To view this document, go to the WebSphere Adapters support page: <http://www.ibm.com/support/docview.wss?uid=swg27006249>.
- Technotes for WebSphere Adapters provide workaround and additional information that are not included in the product documentation. To view the technotes for your adapter, go to the following Web page, select the name of your adapter from the **Product category** list, and click the search icon: <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>.

Technical overview

WebSphere Adapter for Siebel Business Applications supports the exchange of information between your applications and Siebel Business Applications. The adapter supports Siebel entities, including business objects, business components, and business services. This enables you to create business processes that exchange data.

The adapter supports outbound processing (requests for data or services from an application to the Siebel Business Applications), and inbound processing (event notification from a Siebel Business Applications server to an application).

With the adapter, you can use the existing or newly created applications that run in a supported run time environment to send requests for data, and services to the Siebel Business Applications.

You can also add event-generation triggers to Siebel business objects to have notifications of events, such as the creation, update, and deletion of a record, sent to one or more of your applications.

The adapter is metadata driven. Even after you import the adapter into Rational Application Developer for WebSphere Software, you can continue to add new business objects to the existing module. The adapter can process the new business objects without additional coding or customization.

Outbound processing

WebSphere Adapter for Siebel Business Applications supports synchronous outbound processing. Assume, the component sends a request in the form of a WebSphere business object hierarchy to the adapter. The adapter processes the request and returns a WebSphere business object hierarchy that represents the result of the operation.

When the adapter receives a WebSphere business object hierarchy, the adapter processes it as follows:

1. The adapter extracts the metadata from the WebSphere business object hierarchy.
2. It identifies the appropriate Siebel objects to access (for example, Siebel business objects and business components, or Siebel business services, integration objects, and integration components) depending on the objects against which the artifacts were generated.
3. The adapter extracts the outbound operation to be performed from the WebSphere business object hierarchy.
4. After accessing the required Siebel objects, the adapter retrieves, updates, deletes, or creates a Siebel business component hierarchy or performs the corresponding business service method on the integration component hierarchy.
5. If there are updates (Create, Update, and/or Delete), the adapter populates that Siebel object (business or integration component hierarchy) with data from the hierarchy of WebSphere business objects.

Supported outbound operations

The adapter supports the following outbound operations: Create, Delete, Exists, Retrieve, Retrieve All, and Update.

The following table lists and describes each of these operations.

Table 1. Supported outbound operations

Operation	Description
Create	Creates the business component.
Delete	Deletes the business component and its child components.

Table 1. Supported outbound operations (continued)

Operation	Description
Exists	Checks for the existence of incoming business objects. The output business object, "ExistsResult" is returned with the Boolean value populated.
Retrieve	Specifies the value of the business component.
RetrieveAll	Retrieves multiple instances of the same business component.
Update	Updates the Siebel application with the incoming business object.

Outbound operations

Application components must invoke operations such as data retrieval from a database. An operation is the action that an adapter can perform on the database during the outbound processing. The name of the operation indicates the type of action that the adapter performs, such as Create or Delete.

WebSphere Adapter for Siebel Business Applications supports the following operations during outbound processing.

Create operation:

The Create operation creates the corresponding Siebel business component hierarchy in the Siebel application.

During the Create operation, the following occurs:

- Each of the child object is created first, and then the parent object is created. After all the child objects are created, the parent object is saved. By saving the parent object after creating all the child object, you can roll back all the child objects that were created, if the top-level Create operation fails.
- Field values are set for the simple attributes.
- Each of the business component records is saved.
- In case of a failure, the adapter generates a `SiebelCreateFailedException` error.
- If the Siebel application generates a duplicate record error, the adapter generates a `DuplicateRecordException` error.
- The output business object is populated with values for the key attributes.

Each of the child container attributes is processed as follows:

- The key attributes are not set on the parent business component. The Siebel application generates the key attributes when a record is created. If the key attributes are set, Siebel returns an exception and sets the read-only fields populated with values.
- If the child container is a business component of a `PickList`, and you want to select an existing child instance, do not create a child business object. Instead, set the value on the parent container. For example, if you pick an existing `Business Address` instance under the `Account` object, set the `PrimaryAddressId` instead of creating a `Business Address` instance. However, to create a child instance, you must populate the non-key attributes in the child business component so that the adapter tries to create the corresponding child instance in the Siebel application.

- If there are static Picklists, then the corresponding simple attribute in the business object is required. If a value is not listed in the Picklists list of values, the adapter tries to populate a value. If the list is unbounded and Insert is enabled, the operation is successful. If the adapter is unable to populate the value by itself, the operation fails.
- With multi value links, the child record is created. If the child Create operation fails, the child record is rolled back. If the parent Create operation fails, all the child and the parent records are rolled back.
- The adapter attempts to select an existing child business component instance (for example, Association) based on the value of all the attributes in the corresponding child business object. If a matching child business component instance is found, it is associated with the parent. If a matching child business component instance is not found, a new child business component is created and associated with the parent business component.
- If the child Create operation fails, the child component is rolled back. If the parent Create operation fails, only the parent component is rolled back. Because association components can also be individual components, as well as child component of other components, a child component, or parent component cannot be rolled back after an associated child component has been created.

Delete operation:

The Delete operation deletes the top-level business object and its child objects.

The Delete operation results in a physical delete, and the record is removed from the Siebel application. If the key attributes (for example, ID) are not set correctly, the Delete operation fails.

Exists operation:

The Exists operation looks for a specified business component instance in the Siebel application.

If the specified business component instance exists, the component receives a successful response (ExistsResults) in the form of a business object. A Boolean property set in the business object confirms that the business component exists in the database. For example, if the business component instance exists in the Siebel application, the Boolean property is set to True. If the specified record does not exist, the Boolean property is set to False. If there are additional errors during processing, then a `BusinessProcessingFailedException` error is returned.

Retrieve operation:

The Retrieve operation retrieves the Siebel business component that corresponds to the specified key in the incoming business object.

During the Retrieve operation, the following occurs:

- The keys are set on the top-level business component to search the records. If the record is not found, an error message is logged, and the Retrieve operation fails with an exception. If the record is found, the simple attribute values are retrieved from the top-level business component and set on the business object.

Note: A simple PickList attribute is retrieved differently from other simple attributes. A simple PickList attribute is retrieved by obtaining the value of the PickListKey from the child picklist component. To retrieve a child picklist

component, a `FieldName` value must be used. Then, the adapter uses the `Restrict` application-specific information to set more conditions on the child picklist component before retrieving the `PickListKey` value.

- The child components are retrieved based on the type of relationship they have to the parent business object in the business object hierarchy.
- The child `PickList` instances are retrieved. The adapter uses the `From` and `To` application-specific information to obtain the child component. Before the adapter retrieves the child component, the `From` attribute value from the parent component must be set against the `To` attribute in the child component. If the `From` attribute is not set, the adapter retrieves the child component without setting any values on the `To` attribute. For example, only `executeQuery` is called.
- With multi-value link and association, if the child component record is found, the attribute value is set on the parent component.

RetrieveAll operation:

The adapter uses the `RetrieveAll` operation to retrieve multiple instances of the same business object.

The adapter supports the `RetrieveAll` request for business objects that contains the key and non-key attributes. All the key and non-key attributes populated in the incoming business object determine the selection criteria for the retrieval. For business objects on which non-key attributes are set, the selection criteria is `select "*"`. If the values of the attributes are part of the search criteria containing `"*"`, the `RetrieveAll` operation is processed for that attribute. Any additional search criteria is applied to the operation if it has already been set.

The steps involved in processing the `RetrieveAll` operation are similar to the `Retrieve` operation. However, the adapter does not check if all the keys are set on the incoming business object.

Note: The adapter supports attributes values that contain these special characters: `'(, ')', "''", "''", "''", "''", "<", ">"`. The number of records that the adapter can return cannot exceed the `MaxRecords` property value of the `WBIInteractionSpec` instance. If the number of records retrieved from the Siebel application exceeds the `MaxRecords` value, the adapter generates the `MatchesExceededLimitException` error. If the records are not retrieved, the adapter generates the `RecordNotFoundException` error.

Update operation:

The adapter uses the `Update` operation to compare the business component hierarchy retrieved from Siebel with the incoming top-level business object hierarchy.

The adapter compares the object retrieved from the Siebel application with the incoming top-level business object. Depending on the results of the comparison, the creation, update, or deletion of any child object will be processed and updated. If the child `Create` operation fails because the record exists, the adapter returns a `RecordAlreadyExistsException` error.

If the `Update` operation fails, the adapter returns a `SiebelUpdateFailedException` error.

Inbound processing

WebSphere Adapter for Siebel Business Applications supports asynchronous inbound processing. The adapter polls the Siebel applications at specified intervals for events. When the adapter detects an event, it converts the event data into a business object and sends it to the component.

When the adapter detects an event for Siebel event business components or integration components, it processes it by retrieving the updated data and converting it into a business object. Then, the adapter sends the business object to the event business component. For example, if an event business component (an account) is updated, an event trigger adds an event record to the event business component. The adapter then polls the event business component, retrieves the event record, and processes it.

When the adapter finds an event for a Siebel event business component, it is processed as follows:

1. The adapter retrieves the event information from the Siebel event business component.
2. It retrieves the corresponding event business component instance hierarchy.
3. It populates the associated WebSphere business object with the values it retrieves from the event business component.
4. It sends a notification to each registered application.

Note: Before inbound processing can occur, a Siebel event business component must be created in the Siebel application (IBM2 for Siebel version 7.x and IBM_EVENT for Siebel version 8) and its name specified against the corresponding property in the adapter activation specification.

Event store

The event store is a persistent cache where event records are saved until the polling adapter can process them. The adapter uses the event store to track the inbound events.

The creation, update, or deletion of an event record in Siebel Business Applications is an 'event'. Each time a business object is created, updated, or deleted, the adapter updates the status of the event in an event store. For recovery purposes, the status of each event is continuously updated by the adapter until the events are delivered to a configured export in WebSphere Application Server.

For example, if you have a customer component and a new customer has been added to it, this signals an update. If the adapter is configured to receive the events about the new update, there are triggers attached to the Siebel end and connected to the customer component. The triggers add a record to the event business component. The record contains information about the new customer, such as the customer ID. This information is stored in the object key. The object key is the unique identifier that provides the key name and value of the event business component that is updated (for example, Id=1-20RT). The object name is the WebSphere business object name that represents the customer component (for example, AccountBG or Account). The adapter retrieves this event and the new customer information that relates to it. It then processes the event and delivers it to the export.

Each event store created by Siebel Business Applications is associated with a particular module; however the same event store can be used by multiple modules. If you want multiple modules to use the same event store, you must set the

EventTypeFilter property to a different event type for each module using the event store. Set the value for the property in the ActivationSpecification instance for each module.

During inbound processing, the adapter polls the event business components from the event store at regular intervals. Each time it polls, a number of events are processed by the adapter. Events are processed in ascending order of priority and ascending order of the event time stamp. In each poll cycle, new events are picked up. The adapter retrieves the value set in the object key field for the event and loads the business object that corresponds to it. The business object, is created from the retrieved information and is delivered to the exports.

If you set the activation specification property AssuredOnceDelivery to true, a transaction ID (XID) value is set for each event in the event store. After the event is retrieved for processing, the XID value for it is updated in the event store and displayed in the XID column in the event business component. The event is then delivered to its corresponding export, and the status is updated to show that the event has been successfully delivered. If the application is stopped or the event is not processed completely, the XID column is filled with a value. This ensures that the event is reprocessed and sent to the export. After the connection is reestablished or the adapter starts again, the adapter checks for events in the event store that have a value in the XID column. The adapter processes these events first, and then polls the other events during the poll cycles.

The adapter can either process all events or process events filtered by business object type. You set the filter through the activation specification property, EventTypeFilter. This property contains a comma-delimited list of business object types. Only the types specified in the property are processed. If the EventTypeFilter property is not set, then all the events are processed. If the FilterFutureEvents property is set to true, the adapter filters events based on the time stamp. The adapter compares the system time in each poll cycle to the time stamp on each event. If an event is set to occur in the future, it is not processed until that time.

After an event is successfully posted and delivered to the export, the entry is deleted from the event store. Failed events (posting and delivery to the export is unsuccessful), remain in the event store and are marked -1. This prevents duplicate processing.

Event store structure for Siebel business objects and business components

The event business component stores information about the event. The information stored is used by the resource adapter during event subscription to build the corresponding business object and send it to the registered exports. The information that is stored, as well as the structure of the event store used by the adapter, is shown in the following table.

Table 2. Event store structure for Siebel event business objects and business components

Field	Description	Example
Object key	A unique identifier that provides the key name and value of the actual component instance that was updated (for example: An Account component instance was updated and it triggers to insert an event into the event component). The event field holds the key name and value of the Account component instance as a name-value pair for the adapter to do the retrieval later on when polling.	ID=1-20RT
Object name	The name of the business object, corresponding to the Siebel business component for which the event was detected.	B0AccountBCAccount
Event type	The type of event such as Create, Update, or Delete.	Create
Priority	The event priority.	1
Status	The event status. This is initially set to the value for a new event and updated by the adapter as it processes the event. The status can have one of the following values: <ul style="list-style-type: none"> • 0: Identifies a new event. • 1: Identifies an event that has been delivered to an export. • -1: An error occurred while processing the event. The status cannot be null.	0
Description	Any comment associated with the event.	Account create Event
Event ID	The key field ID of the event component. The value is auto-generated by Siebel.	
Event timestamp	The time stamp for the event. The format is MM/dd/yyyy HH:mm:ss.	02/24/2007 11:37:56
XID	The transaction ID. This is to ensure 'assured once delivery'. This removes the dependency on the EDT. For more information about XID, check the AFC FS.	
CONNECTOR ID	The ID set for the adapter.	Siebel001

Event store structure for Siebel business services

The event is retrieved from the event business component and the information is used to retrieve the event business component.

Table 3. Event store structure for Siebel business services

Field	Description	Example
Object key	A unique identifier that provides the key name and value of the actual component instance that was updated (for example: An Account component instance was updated and it triggers to insert an event into the event component). The event field holds the key name and value of the Account component instance as a name-value pair for the adapter to do the retrieval later on when polling.	Name=TestName; Location=BGM, where 'Name' and 'Location' are the keys in the integration component 'Account' under Integration object 'Account (PRM ANI)' and 'TestName' and 'BGM' are the values specified, and; is the event key delimiter.
Object name	The name of the business object, corresponding to the Siebel business component for which the event was detected.	IOAccountPRMANIICAccount
Event type	The type of event such as Create, Update, or Delete.	Create
Priority	The event priority.	1
Status	The event status. Initially the value is set for a new event and updated by the adapter as it processes the event. The status can be set to one of the following values: <ul style="list-style-type: none"> • 0: Identifies a new event. • 1: Identifies an event that has been delivered to an export. • -1: An error occurred while processing the event. The status cannot be null.	0
Description	Any comment associated with the event.	Account PRM ANI Event
Event ID	The key field ID of the event component. The value is auto-generated by Siebel.	
Event timestamp	The time stamp for the event. The format is in <i>MM/dd/yyyy HH:mm:ss</i> .	02/24/2007 11:37:56
XID	The transaction ID. This is to ensure 'assured once delivery'. This removes the dependency on the EDT. For more information about XID, check the AFC FS.	
CONNECTOR ID	The ID set for the adapter.	Siebel001

Monitoring inbound events

The adapter supports monitoring inbound events from the Siebel application, in addition to the other events you are monitoring using Business Monitor or WebSphere Business Events.

Monitoring inbound events using IBM Business Monitor:

You can use Rational Application Developer for WebSphere Software and the adapter to send inbound events to WebSphere Application Server Common Event Infrastructure (CEI), where they are accessible to Business Monitor.

When you select the option to monitor inbound events in the Rational Application Developer for WebSphere Software J2C Bean wizard, the required artifacts are generated to monitor inbound events. These artifacts include the message-driven J2C bean, as well as the interface, Service Bean, interceptor class, helper class, and the event schemas that are required to create a monitor model. You can then deploy the resulting adapter inbound event monitoring application containing the message-driven bean (the adapter application) to a server, either a Business Monitor server or a remote server. The message-driven bean invokes the stateless session bean that makes the events accessible to the client. More importantly, it also listens for events coming in from the Siebel application (inbound events) and uses the interceptor to set up the intercepted inbound events as Common Base Events (CBE). Then, it posts these Common Base Events into a designated Java Message Service (JMS) destination - Common Event Infrastructure queue, where they are accessible to Business Monitor for further processing.

Important: Inbound event monitoring is available to your application only if you have Business Monitor installed in your environment. For more information about installing Business Monitor, see <http://pic.dhe.ibm.com/infocenter/dmndhelp/v7r5m1/index.jsp?topic=%2Fcom.ibm.wbpm.mon.imuc.doc%2Finst%2Fintro.html>. For more information about the software requirements and configuration, see <http://www-304.ibm.com/support/docview.wss?uid=swg27008414>.

For enabling inbound event monitoring function, perform the following tasks:

1. Configure the Siebel application to work with the adapter (see “Configuring the Siebel application to work with the adapter” on page 25)
2. Launch the J2C Bean wizard (see “Launching the J2C Bean wizard” on page 27)
3. Configure the connector dependencies (see “Configuring the connector dependencies” on page 28)
4. Set the connection properties in J2C Bean wizard (see “Setting connection properties for the J2C Bean wizard” on page 28)
5. Configuring the module for inbound processing (see “Configuring the module for inbound processing” on page 39)

Related References

For a sample on enabling inbound event monitoring for Business Monitor, see <http://publib.boulder.ibm.com/infocenter/radhelp/v8r5/topic/com.ibm.j2c.doc/topics/tcreatinginboundapps.html>.

For information about how to disable the event monitor function, see <http://publib.boulder.ibm.com/infocenter/radhelp/v8r5/topic/com.ibm.j2c.doc/topics/t disablingwbe.html>.

For a sample end-to-end scenario on publishing events to the Business Monitor, see <http://pic.dhe.ibm.com/infocenter/dmndhelp/v7r5m1/topic/com.ibm.wbpm.mon.doc/scen/eis.html>.

Monitoring inbound events using WebSphere Business Events:

You can use Rational Application Developer for WebSphere Software and the adapter to send inbound events to WebSphere Application Server JMS topic, where they are accessible to WebSphere Business Events.

Note: You cannot create a JMS connection to the remote server when the same connection factory name is duplicated. For more information, see http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.pmc.doc/tasks/tjn0033_.html

When you select the option to monitor inbound events in the Rational Application Developer for WebSphere Software J2C Bean wizard, the required artifacts are generated to monitor inbound events. These artifacts include the message-driven J2C bean, as well as the interface, Service Bean, interceptor class, helper class, and the event schemas that are required to create a monitor model. You can then deploy the resulting adapter inbound event monitoring application containing the message-driven bean (the adapter application) to a server. The message-driven bean invokes the stateless session bean that makes the events accessible to the client. More importantly, it also listens for events coming in from the Siebel application (inbound events) and uses the interceptor to set up the intercepted inbound events as Common Base Events (CBE). Then, it posts these Common Base Events into a designated Java Message Service (JMS) destination - JMS topic, where they are accessible to WebSphere Business Events for further processing.

Important: Inbound event monitoring is available to your application only if you have WebSphere Business Events installed in your environment. For information about installing WebSphere Business Events, see <http://publib.boulder.ibm.com/infocenter/wbevents/v6r2m1/index.jsp?topic=/com.ibm.wbe.install.doc/doc/install.html>. The WebSphere Business Events works with WebSphere Application Server version 6.1; it is not supported in WebSphere Application Server version 7.0. For more information about the software requirements and configuration, see <http://www.ibm.com/software/integration/wbe/requirements/>.

For enabling inbound event monitoring function, perform the following tasks:

1. Configure the Siebel application to work with the adapter (see “Configuring the Siebel application to work with the adapter” on page 25)
2. Launch the J2C Bean wizard (see “Launching the J2C Bean wizard” on page 27)
3. Configure the connector dependencies (see “Configuring the connector dependencies” on page 28)
4. Set the connection properties in J2C Bean wizard (see “Setting connection properties for the J2C Bean wizard” on page 28)
5. Configuring the module for inbound processing (see “Configuring the module for inbound processing” on page 39)
6. Generate the eventBOTypeMapping.xml and eventBOTypeMapping.xsd files from the generated inbound session bean. The eventMapping file provides the mapping between the event and the business object schema that the WebSphere Business Event requires for monitoring the event. To generate the eventBOTypeMapping.xml and eventBOTypeMapping.xsd files:
 - a. Right-click your session bean.
 - b. Select **Source > Generate Event Mapping**.

The EventMapping files get generated in the same folder as your business object schema files.

Related Reference

For integrating WebSphere Business Events with WebSphere Application Adapters, see <http://publib.boulder.ibm.com/infocenter/wbevents/v6r2m1/topic/com.ibm.wbe.integrating.doc/doc/integratingusingwebsphereadapters.html>.

Business objects

To send data or obtain data from Siebel Business Applications, the adapter uses business objects. A business object is a structure that consists of data, the action to be performed on the data, and additional instructions, if any, for processing the data. The data can represent either a business entity, such as an invoice, an employee record, or an unstructured text.

How business objects are created

You create business objects by using the J2C Bean wizard, launched from Rational Application Developer for WebSphere Software. The J2C Bean wizard connects to the application, discovers data structures in the application, and generates business objects to represent them. It also generates other artifacts needed by the adapter.

Business object structure

The adapter supports business objects that are hierarchically structured. The top-level business object must have a one-to-one correspondence with the Siebel business component, and collections that occur within the top-level object are child objects of it. Information about the processed object is stored in the application-specific information for the object and each of its attributes.

Service Bean

The business data exchanged between the client application and the resource adapter is represented as Service Bean. The metadata describing the business data is defined as business objects and represented as the XSD schemas. The Service Bean is generated from these XSDs and is the realization of the business objects.

A Service Bean is a structure that consists of data and, in some cases, metadata with additional instructions, for processing the data. It is a generated, hierarchical, Java objects implementing Record interface. The data can represent a business entity, such as an invoice or an employee record.

You create Service Bean by using the J2C Bean wizard, started from Rational Application Developer for WebSphere Software connector tools. The wizard connects to the Siebel Business Applications, discovers data structures in the EIS, and generates Service Bean to represent them. The adapter supports records that are hierarchically structured. Information about the processed object is stored in the application-specific information for the object and each of its attributes.

The J2C Bean wizard

The J2C Bean wizard is a tool you use to configure your adapter before deploying it to WebSphere Application Server. The J2C Bean wizard establishes a connection to the Siebel application server, discovers business objects and services (based on search criteria you provide), and generates business objects based on the services discovered.

Using Rational Application Developer for WebSphere Software, you establish a connection to the Siebel application server to browse the metadata repository on

the server. In the wizard, you select the business objects you need. The wizard automatically generates the XSD schemas that represent the business objects and the corresponding Service Bean.

The result of running the J2C Bean wizard is a module that contains the Siebel business objects and services along with the adapter. You deploy this module on WebSphere Application Server.

For example, if you run the J2C Bean wizard to create a module called Outbound, you can see under Data Types, a list of generated business objects.

Standards Compliance

This product is compliant with several government and industry standards, including accessibility standards and Internet Protocol standards.

Accessibility

Administration

The run time administrative console is the primary interface for deployment and administration of enterprise applications. These consoles are displayed within a standard web browser. By using an accessible Web browser, such as Microsoft Internet Explorer or Netscape Browser, you are able to:

- Use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen
- Use voice recognition software, such as IBM via Voice, to enter data and to navigate the user interface
- Operate features by using the keyboard instead of the mouse

You can configure and use product features by using standard text editors and scripts or command line interfaces instead of the graphical interfaces that are provided. When appropriate, the documentation for specific product features contains additional information about the accessibility of the features.

J2C Bean wizard

The J2C Bean wizard is the primary component used to create application accessing EIS systems. This wizard is implemented as an Eclipse plug-in that is available through Rational Application Developer for WebSphere Software is fully accessible.

Keyboard navigation

This product uses standard Microsoft Windows® navigation keys.

IBM and accessibility

See the IBM Accessibility Center website <http://www.ibm.com/able/> for more information about the commitment that IBM has to accessibility.

Internet Protocol, Version 6 (IPv6)

WebSphere Application Server, version 6.1.0 and later and its JavaMail component support dual-stack Internet Protocol Version 6.0 (IPv6). For more information about this compatibility in WebSphere Application Server, see IPv6 support in the WebSphere Application Server information center. For more information about IPv6, see <http://www.ipv6.org>.

Log and Trace Analyzer

The adapter creates log and trace files that can be viewed with the Log and Trace Analyzer.

The Log and Trace Analyzer can filter log and trace files to isolate the messages and trace information for the adapter. It can also highlight the messages of an adapter and trace information in the log viewer.

The adapter's component ID for filtering and highlighting is a string composed of the characters SEBLRA plus the value of the adapter ID property. For example, if the adapter ID property is set to 001, the component ID is SEBLRA001.

If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter. For example, when you set the adapter ID property of two instances of WebSphere Adapter for Siebel Business Applications to 001 and 002. The component IDs for those instances, SEBLRA001 and SEBLRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SEBLRAInstanc.

For outbound processing, the adapter ID property is located in both the resource adapter and managed connection factory property groups. If you update the adapter ID property after using the J2C Bean wizard to configure the adapter for outbound processing, be sure to set the resource adapter and managed connection factory properties consistently. It prevents inconsistent marking of the log and trace entries. For inbound processing, the adapter ID property is located only in the resource adapter properties, so this consideration does not apply.

For more information, see the “Adapter ID (AdapterID)” on page 96 property.

Chapter 2. Planning for adapter implementation

Before you configure WebSphere Adapter for Siebel Business Applications Software, consider whether you must set up the adapters in a clustered environment, in which the workload of the server is distributed across multiple machines.

Before you begin

Before you begin to set up and use the adapter, you must possess a thorough understanding of business integration concepts, the capabilities, and requirements of the integration development tools and runtime environment you use, and the Siebel Business Applications server environment where you build and use the solution.

To configure and use WebSphere Adapter for Siebel Business Applications, you must understand and have experience with the following concepts, tools, and tasks:

- The business requirements of the solution you are building.
- The Siebel Business Applications environment in which you are working. This includes a detailed understanding of the Siebel version that supports your applications.
- Business integration concepts and models.
- The capabilities provided by the integration development tools you use to build the solution. You must know how to use these tools to create modules, test components, and complete other integration tasks.
- The capabilities and requirements of the runtime environment you use for the integration solution. You must know how to configure and administer the host server and how to use the administrative console to set and modify property definitions, configure connections, and manage events.
- The capabilities provided by the Siebel Business Applications version that supports your applications. You must be familiar with Siebel business objects, business components, business services, integration objects, and integration components.

Security

The adapter uses the J2C authentication data entry, or the authentication alias feature of Java EE security to provide secure user name and password authentication. For more information about security features, see the documentation for WebSphere Application Server.

Support for protecting sensitive user data in log and trace files

You can configure the adapter to prevent sensitive or confidential data, in the log and trace files, from being viewed by users without authorization.

Log and trace files for the adapter can contain data from your Siebel application, which might contain sensitive or confidential information. Sometimes these files

might be seen by individuals without authorization to view sensitive data. For example, a support specialist must use the log and trace files to troubleshoot a problem.

To protect the data in such situations, the adapter lets you specify whether you want to prevent confidential user data from displaying in the adapter log and trace files. You can select this option in the J2C Bean wizard or change the `HideConfidentialTrace` property. When this property is enabled, the adapter replaces the sensitive data with XXX's.

See “Managed connection factory properties” on page 100 for information about this optional property.

The following types of information are considered potentially sensitive data and are disguised:

- The contents of a business object
- The contents of the object key of the event record
- User name and password
- The URL used to connect to the Siebel application

The following types of information are not considered user data and are not hidden:

- The contents of the event record that are not part of the event record object key, for example, the `XID`, event ID, business object name, and event status
- Business object schemas
- Transaction IDs
- Call sequences

User authentication

The adapter supports several methods for supplying the user name and password that are needed to connect to the Siebel application. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

To integrate an adapter into your application, a user name and password are needed at the following times:

- When the J2C Bean wizard connects to Siebel application to extract, or *discover*, information about the objects and services that you can access with the adapter.
- At run time on WebSphere Application Server, when the adapter connects to Siebel application to process outbound requests and inbound events.

Authentication in the wizard

The J2C Bean wizard prompts for the connection information for the discovery process, and then reuses it as the default values of the adapter properties that specify the connection information used at run time. You can use a different user name and password while running the wizard than you use when the application is deployed to the server. You can even connect to a different Siebel application, although the schema name must be the same in both databases. For example, while developing and integrating an application that uses WebSphere Adapter for Siebel Business Applications, you might not use the production database; using a test database with the same data format but fewer, simulated records allows you to

develop and integrate the application without impacting the performance of a production database and without encountering restrictions caused by the privacy requirements for customer data.

The wizard uses the user name and password that you specify for the discovery process only during the discovery process; they are not accessible after the wizard is completed.

Authentication at run time

At run time, the adapter needs to provide the user name and password to connect to the Siebel application. To connect without user intervention, the adapter must access a saved copy of the user information. In a server environment, there are several methods for saving user information. You can configure the adapter to get your user information, through any of the following methods:

- Adapter properties
- J2C authentication alias

Saving the user name and password in adapter properties is a direct way to provide this information at run time. You provide this user name and password when you use the J2C Bean wizard to configure your module. Although directly specifying the user name and password seems the most straightforward method, it has important limitations. Adapter properties are not encrypted; the password is stored as clear text in fields that are accessible to others on the server. Also, when the password changes, you must update the password in all instances of the adapter that access that Siebel application. This includes the adapters embedded in application EAR files and adapters that are separately installed on the server.

Using a J2C authentication data entry, or authentication alias, created with the Java Authentication and Authorization Service (JAAS) feature of Java EE security is a robust, secure way to deploy applications. An administrator creates the authentication alias that is used by one or more applications that need to access a system. The user name and password must be known only to that administrator, who can change the password in a single place, when a change is required.

Deployment options

There are two ways to deploy the adapter. You can either embed it as part of the deployed application, or you can deploy it as a stand-alone RAR file. The requirements of your environment affect the type of deployment option you choose.

The following are the deployment options:

- When you deploy the adapter as an embedded component, the adapter is included within an enterprise application archive (EAR) file and is available only to the application in the EAR file.
- When you deploy the adapter as a stand-alone component, the adapter is represented by a stand-alone resource adapter archive (RAR) file. When it is deployed, it is available to all applications deployed in the server instance.
- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an

embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

While creating the project for your application using Rational Application Developer for WebSphere Software, you can choose how to package the adapter [either bundled with the (EAR) file or as a stand-alone (RAR) file]. Your choice affects how the adapter is used in the run time environment and how the properties for the adapter are displayed on the administrative console.

Choosing either to embed an adapter with your application or to deploy the adapter as a stand-alone module depends on how you want to administer the adapter. If you want a single copy of the adapter and do not care about disruption to multiple applications when you upgrade the adapter, then you would be more likely to deploy the adapter as a stand-alone module.

If you plan to run multiple versions, and if you care more about potential disruption when you upgrade the adapter, you would be more likely to embed the adapter with the application. Embedding the adapter with the application allows you to associate an adapter version with an application version and administer it as a single module.

To deploy the RAR file to the application server, you must obtain and install Adapter for Siebel Business Applications. This provides the RAR file that you install following instructions supplied with WebSphere Application Server.

Considerations for embedding an adapter in the application

Consider the following items if you plan to embed the adapter with your application:

- An embedded adapter has class loader isolation.
A class loader affects the packaging of applications and the behavior of packaged applications deployed on run time environments. *Class loader isolation* means that the adapter cannot load classes from another application or module. Class loader isolation prevents two similarly named classes in different applications from interfering with each other.
- Each application in which the adapter is embedded must be administered separately.

Considerations for using a stand-alone adapter

Consider the following items if you plan to use a stand-alone adapter:

- Stand-alone adapters have no class loader isolation.
Because stand-alone adapters have no class loader isolation, only one version of any defined Java artifact is run and the version and sequence of that artifact is undetermined. For example, when you use a stand-alone adapter there is only *one* resource adapter version, *one* adapter foundation class (AFC) version, or *one*

third-party JAR version. All adapters deployed as stand-alone adapters share a single AFC version, and all instances of a defined adapter share the code version. All adapter instances using a given third-party library must share that library.

- If you update any of these shared artifacts, all applications using the artifacts are affected.

For instance, if you have an adapter that is working with server version X, and you update the version of the client application to version Y, your original application might stop working.

- Adapter Foundation Classes (AFC) is compatible with previous versions, but the latest AFC version must be in every RAR file that is deployed in a stand-alone manner.

If more than one copy of any JAR file is in the class path in a stand-alone adapter, the one that is used is random; therefore, they all must be the latest version.

Considerations while deploying adapters with different versions

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the CWYBS_AdapterFoundation.jar is loaded during run time, the adapter returns the ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for Siebel Business Applications version 7.5.0.3, the following error message is displayed "The version of CWYBS_AdapterFoundation.jar is not compatible with IBM WebSphere Adapter for Siebel Business Applications" as IBM WebSphere Adapter for Siebel Business Applications loads file:/C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3. However, the base level of this jar required is version 7.5.0.3. To overcome this conflict, you must ensure that all adapters are at same version level. For further assistance, contact WebSphere Adapters Support for help.

There are occasions when you work with embedded adapters that do not need a client-server communication, stand-alone adapters that need a server connection, or a hybrid mix of adapter connections.

The following scenarios cover the different behaviors of AFC version conflict detection, when you are deploying two or more adapters and at least one of the adapter version is 7.5 or higher.

Deploying a stand-alone Adapter

1. Install WebSphere Adapter for Siebel Business Applications version 7.0.1.0 through the WebSphere Application Server administrative console.
2. Install WebSphere Adapter for SAP Software version 7.5.0.0 through the administrative console.
3. Create ActivationSpec for an ALE pass-through inbound operation.
4. Create an application in Rational Application Developer for WebSphere Software for a stand-alone ALE pass-through inbound operation.
5. Install and start the application through the administrative console.
6. Verify the error.

Note: An error message is generated in the log/trace area of WebSphere Application Server, to indicate an AFC version conflict.

Deploying an embedded Adapter

1. Import a build of WebSphere Adapter for PeopleSoft Enterprise version 7.0.1.0, using a RAR file.
2. Create a Peoplesoft Inbound EMD operation.
3. Import a build of WebSphere Adapter for Oracle E-Business Suite version 7.5.0.0, using a RAR file.
4. Create an Oracle inbound EMD operation, in the same module where you have created the Peoplesoft Inbound EMD operation.
5. Deploy the module to WebSphere Application Server.
6. Check the trace.

At step 5, the deployment fails. At step 6, you get an internal error message due to the AFC version conflict.

Note: To avoid a name conflict between the business object generated by the two adapters, generate the artifacts into different folders.

Deploying a combination of stand-alone and embedded Adapters

1. Install WebSphere Adapter for Oracle E-Business Suite version 7.0.1.0 through the WebSphere Application Server administrative console.
2. Create an ActivationSpec for a Oracle inbound operation.
3. Create an application in Rational Application Developer for WebSphere Software for a Oracle inbound operation, for the stand-alone Adapter deployment.
4. Deploy the Oracle inbound application and trigger your inbound events.
5. Create an application in Rational Application Developer for WebSphere Software for a WebSphere Adapter for SAP Software version 7.5.0.0 inbound embedded Adapter deployment.
6. Deploy an SAP inbound application, and trigger your inbound events.

Note: You can resolve the AFC version conflict by using different class loaders for the stand-alone and embedded deployments. You can start both Oracle and SAP inbound applications successfully, and process Inbound events without exception.

For further assistance, visit <http://www.ibm.com/support/docview.wss?uid=swg27006249>.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying a module on a clustered server environment. Clusters are groups of servers that are managed together to balance workloads and to provide high availability and scalability.

The module you deployed is replicated across all servers in a cluster, regardless of whether you deploy the module using a stand-alone or an embedded adapter. The following IBM products support WebSphere Adapters in a clustered environment:

- WebSphere Application Server
- WebSphere Application Server Network Deployment
- WebSphere Extended Deployment

To deploy and configure WebSphere Adapter for Siebel Business Applications in a clustered environment, see: “Deploying the module in a clustered environment” on page 53. When you set up a server cluster, you create a Deployment Manager profile. The HAManager, a subcomponent of the Deployment Manager, notifies the Java Platform, Enterprise Edition (JEE) Connector Architecture (JCA) container to activate an adapter instance.

Using WebSphere Extended Deployment, you can optionally enhance the performance of adapter instances in your clustered environment. WebSphere Extended Deployment extends the WebSphere Application Server Network Deployment capabilities by using a dynamic Workload Manager instance instead of a static Workload Manager. The dynamic Workload Manager instance can optimize the performance of adapter instances in the cluster by dynamically balancing the load of the requests. It means that application server instances can be automatically stopped and started based on the load variations, allowing systems with different capacities and configurations to handle load variations evenly.

In clustered environments, adapter instances can handle both inbound and outbound processes.

High availability for inbound processes

Inbound processes are based on events triggered as a result of updates to data in the Siebel application. WebSphere Adapter for Siebel Business Applications is configured to detect updates by polling an event table. The adapter then publishes the event to its endpoint.

When you deploy a module to a cluster, the Java Platform, Enterprise Edition (JEE) Connector Architecture (JCA) container checks the enableHASupport resource adapter property. If the value for the enableHASupport property is true, which is the default setting, all of the adapter instances are registered with the HAManager with a policy 1 of N. This policy means that only one of the adapter instances starts polling for events. Although other adapter instances in the cluster are started, they remain dormant with respect to the active event until the active adapter instance finishes processing the event. If the server on which the polling thread was started shuts down for some reason, an adapter instance that is running on one of the backup servers is activated.

Note: In the active-passive configuration mode of the adapters, the endpoint application of the passive adapter instance also listens to the events/messages even if the enableHASupport property is set to True. This is because the alwaysactivateAllMDBs property in the JMS activation specification is set to True. To stop the endpoint application of the passive adapter instance from listening to the events, you must set the alwaysactivateAllMDBs property value to False. For more information, see “Endpoint application of the passive adapter instance listens to the events when enableHASupport is set to True” on page 76.

Important: Do not change the setting of the enableHASupport property.

High availability for outbound processes

In clustered environments, multiple adapter instances are available to perform outbound process requests. So, if your environment has multiple applications that interact with WebSphere Adapter for Siebel Business Applications for outbound requests, then you might improve performance by deploying the module to a

clustered environment. In a clustered environment, multiple outbound requests can be processed simultaneously, so that they are not attempting to process the same record.

If multiple outbound requests are attempting to process the same record, such as a Customer address, the workload management capability in WebSphere Application Server Network Deployment distributes the requests among the available adapter instances in the sequence they were received. As a result, these types of outbound requests in a clustered environment are processed in the same manner as in a single server environment: one adapter instance processes only one outbound request at a time.

Chapter 3. Configuring the module for deployment

To configure the adapter so that it can be deployed on WebSphere Application Server, use Rational Application Developer for WebSphere Software to create a module, which is exported as an EAR file when you deploy the adapter. You then specify the business objects you want to discover and the system on which you want to discover them.

Configuring the Siebel application to work with the adapter

Configure the Siebel application to work with the adapter by creating an event table and a Siebel business object.

Before you begin

Before you configure the Siebel application to work with WebSphere Adapter for Siebel Business Applications, you must create a user name and password so that the J2C Bean wizard can connect to Siebel Business Applications to perform outbound operations and retrieve Siebel business objects and services.

Also, because you perform this task on the Siebel Business Applications server, ensure that you are familiar with the Siebel tools required to complete it.

For information about using Siebel tools, refer to the Siebel tools documentation.

You must open Siebel Sales Enterprise (or the Siebel application you are using) on your local database with the administrative privileges.

About this task

To configure the Siebel application, you create an event table and a Siebel business object.

Procedure

1. Create a project called IBM, and lock the project with Siebel tools.
2. Using the object wizard, create an event table called CX_IBM_EVENT to store the events.
 - a. In the event table, create the columns shown in the following table.

Table 4. Event table details

Column Name	Type	Length	Data Type	Required	Nullable	Status
CONNECTOR_ID	Data (public)	255	Varchar	No	Yes	Active
DESCRIPTION	Data (public)	255	Varchar	No	Yes	Active
EVENT_ID	Data (public)	30	Varchar	Yes	No	Active
EVENT_TYPE	Data (public)	20	Varchar	Yes	No	Active
OBJECT_KEY	Data (public)	255	Varchar	Yes	No	Active
OBJECT_NAME	Data (public)	255	Varchar	Yes	No	Active
PRIORITY	Data (public)	10	Varchar	No	Yes	Active
STATUS	Data (public)	20	Varchar	Yes	No	Active

Table 4. Event table details (continued)

Column Name	Type	Length	Data Type	Required	Nullable	Status
XID	Data (public)	255	Varchar	Yes	No	Active

Note: If ConnectorId is specified in the AdapterInstanceEvent Filter property, the adapter adds the column to the event query. If ConnectorId is not specified in the AdapterInstanceEvent Filter property, the adapter ignores the column in the event query.

- b. Create a business component called IBM Event.
 - c. Create a time stamp called Field Event, and map it to the CREATED column from CX_IBM_EVENT. Make the Type of this field DTYPE_UTCDATETIME.
 - d. Create a business object called IBM Event.
 - e. Associate the IBM event business component to the IBM Event business object.
 - f. Create an applet called IBM Event List Applet, and base it on the IBM Event business component you created earlier.
 - g. Create a view called IBM Event List View, and base it on the IBM Event business object you created earlier.
 - h. Create a screen called IBM Event Screen, and associate it to the IBM Event List View in the Siebel tools.
3. Create a page tab.
 - a. Click **Start Application > Siebel Sales Enterprise > Page tab**.
 - b. Right-click and select **New Record**.
 - c. Specify IBM Event as the screen name and IBM Event for the **Text - String Override** field.
 - d. Leave the Inactive field clear.
 4. Create a business object called Schema Version for your IBM project and associate it to the Schema Version business component.
 - a. Apply the physical schema for the new tables to your local database. You can do this by querying for the new table, 'CX_IBM_EVENT' and selecting the current query to create a physical schema. Leave the table space and index space blank.
 - b. Click **Activate** to activate the new schema.
 5. Add or modify the Siebel VB or e-scripts for the business component that corresponds to the business objects used at your site. Siebel scripts trigger event notification for business objects. Samples are located in the Samples Folder in your adapter installation.
 6. Create a Siebel repository file by compiling the updated and locked projects on your local database. The new repository file will have an extension of .srf.
 7. Open Siebel Sales Enterprise on your local database.
 - a. Create a responsibility called IBM Responsibility for IBM Event List View.
 - b. Add the employees or teams who are responsible for reviewing events to the newly created IBM Responsibility.
 - c. Create a user name called IBMCONN (or some other user name to be used by the adapter later). Add the user name to the newly created IBM Responsibility and also to the Administrative Responsibility.
 8. Test the application in your local environment to ensure that you have visibility to the IBM Event List View. An event is generated in the view after

you create a record in the supported object. As part of the test, create a Account business component instance in Siebel. Confirm that a new Account event appears in the IBM Event List View. (This is assuming that you have added the e-script trigger to the Account business component). If a new Account event does not appear in the view, check for an error and fix it. For more information about the errors that might be generated, check either the Siebel support site or Siebel documentation.

9. When the test performed in Step 8 is successful, add your new and updated projects to your development server.
10. Activate the new table in the development server.
11. Compile a new Siebel repository (.srf) file on the server.
12. Back up the original repository file on the server.
13. Stop the Siebel Business Applications server, and replace the original repository file with the newly created one.
14. Restart the Siebel Business Applications server.

Launching the J2C Bean wizard

To begin the process of creating and deploying a module, you start the J2C Bean wizard in Rational Application Developer for WebSphere Software. The wizard creates a connector project, which is used to organize the files associated with the module.

Before you begin

Ensure that you have gathered the information you need to establish a connection to the Siebel application. For example, you need the name or IP address of the Siebel application and the user ID and password to access it.

About this task

If you have an existing project, you can use it instead of creating a new one. Select it before you start the wizard.

Procedure

1. To start the J2C Bean wizard, go to the Enterprise Explorer of Rational Application Developer for WebSphere Software and click **File > New > Other > J2C > J2C Bean**. Click **Next**.
2. In the Resource Adapter Selection window, expand the Siebel folder and select **IBM WebSphere Adapter for Siebel Business Applications (IBM : version)**, where *version* is the version of the adapter you want to use, and then click **Next**.
3. In the **Connector Import** window, accept the default project name in the **Connector project** field or type a different name.
4. In the **Target server** field, select the type of server where you want to deploy the module. The wizard creates the artifacts that are appropriate to that server.
5. Click **Next**. The Connector Settings window is displayed.

What to do next

Continue working in the J2C Bean wizard. The next step is to add dependent JAR files to the project.

Configuring the connector dependencies

The J2C Bean wizard needs certain files from the Siebel Business Applications server to communicate with it.

Before you begin

You must be running the J2C Bean wizard in Rational Application Developer for WebSphere Software to perform this task.

About this task

To obtain the required files and specify their locations, use the following procedure.

Procedure

1. Obtain the Siebel dependency JAR files from your Siebel Business Applications administrator or the workstation on which the Siebel Business Applications server is installed. Depending on the version of Siebel Business Applications you use, the software dependencies differ. The following table lists the dependency files needed by supported versions of Siebel Business Applications. The files are shown with the corresponding J2C Bean wizard property names.

Table 5. Connector dependency files required by Siebel Business Applications

J2C Bean wizard property name	Required dependency files for Siebel Business Applications (versions 7.5 and earlier)	Required dependency files for Siebel Business Applications (versions 7.7x, 7.8x, 8.0, and 8.1.1)
Siebel.jar	None	Siebel.jar
SiebelJI_<language code>.jar	SiebelJI_<language code>.jar (for example, SiebelJI_enu.jar)	SiebelJI_<language code>.jar (for example, SiebelJI_enu.jar)
SiebelJI_Common.jar	SiebelJI_Common.jar	None

2. Copy the files to a local folder on the workstation where you run the J2C Bean wizard.
3. In the Connector Settings window, click **Browse** for each property to select the location of the file.
4. Click **Next**.

Results

The files needed to communicate with the Siebel Business Applications server are now part of the project.

What to do next

Continue working in the wizard to configure the module for outbound or inbound processing.

Setting connection properties for the J2C Bean wizard

To set connection properties for the J2C Bean wizard so that it can access the Siebel Business Applications server, specify information, such as the Connection URL, user name, and the password you use to access the server. In addition, specify the name or IP address of the server.

Before you begin

Make sure that you have successfully configured the connector dependencies.

About this task

Specify the connection properties that the J2C Bean wizard requires to establish a connection to the Siebel Business Applications server and discover functions or data.

To specify the connection properties, use the following procedure.

Procedure

1. In the Adapter Style window, select **Outbound** (to send data to the Siebel Business Applications server from the adapter), or **Inbound** (to send data to the adapter from the Siebel Business Applications server), and then click **Next**. The Discovery Configuration window is displayed.
2. Optional: Event monitoring is available to your application only if you have Business Monitor or WebSphere Business Events installed in your environment.
 - Use the following procedure, to generate and monitor common base events and manage these events with the Common Event Infrastructure (CEI) services using Business Monitor:
 - a. Select the Enable Inbound Event Monitor check box and then click **Next**. The **Event and JMS configuration** window is displayed.
 - b. In the **Event type** field, select WebSphere Business Monitor.
 - c. In the **Queue connectionFactory JNDI name** field, accept the default value, `jms/cei/EventQueueConnectionFactory`.
 - d. In the **Queue JNDI name** field, accept the default value, `jms/cei/EventQueue`.
 - e. Click **Advanced** to add advanced properties.
 - f. In the Remote JNDI provider configuration section, specify the naming provider URL host name and port name values in the **Naming provider URL Host** and **Naming provider URL port** fields to connect to the remote WebSphere Application Server from the wizard.
 - g. In the Connection authentication configuration section, type the user name in the **User name** field to connect to the server from the wizard.
 - h. In the Connection authentication configuration section, type the password in the **Password** field to connect to the server from the wizard.
 - i. Click **Next**. The Discovery Configuration window is displayed.
 - Use the following procedure, to generate and monitor common base events and manage these events using WebSphere Business Events:
 - a. Select the Enable Inbound Event Monitor check box and then click **Next**. The **Event and JMS configuration** window is displayed.
 - b. In the **Event type** field, select WebSphere Business Events.
 - c. In the **Topic connectionFactory JNDI name** field, accept the default value, `jms/WbeTopicConnectionFactory` .
 - d. In the **Topic JNDI name** field, accept the default value, `jms/WBE/CbeListener`.
 - e. Click **Advanced** to add advanced properties.
 - f. The **Remote JNDI provider configuration** field is used to configure the remote topics.

6. Optional: To change the location of the log files for the wizard or the amount of information included in the logs, click **Specify the level of the logging desired**, and then provide the following information:

- a. In **Log file output location**, specify the location of the log file for the wizard.
- b. In **Logging level**, specify the severity of errors that you want logged.

Note: This log information is for the wizard only; at run time, the adapter writes messages and trace information into the standard log and trace files for the server.

7. Click **Next**.

Results

The J2C Bean wizard contacts the Siebel Business Applications server using the information you provided (such as user name and password) to log in. The Object Discovery and Selection window is displayed.

What to do next

Specify search criteria that the J2C Bean wizard uses to discover business objects or business services on the Siebel application.

Configuring the module for outbound processing

To configure a module to use the adapter for outbound processing, use the J2C Bean wizard in Rational Application Developer for WebSphere Software to find and select business objects and services from the Siebel application, and to generate the business object definitions and related artifacts.

Selecting business objects

Select business objects through the J2C Bean wizard, that you want to process.

Before you begin

Ensure that you have set the connection properties for the J2C Bean wizard.

About this task

Specify the search criteria that the J2C Bean wizard uses to discover business objects on the Siebel Business Applications server.

To search and select business objects, use the following procedure.

Selecting business objects

1. In the Object Discovery and Selection window,
 - a. Click **Edit Query**.
 - b. In the Query Filter Parameters window, type the name of the business object in the **Pattern** field that you want to find (for example, Account), and click **OK**.
 - c. Click **Execute Query** to display the specified business object.

- d. Expand the **Siebel Business Objects** folder in the **Objects discovered by query** pane to see the list of discovered business objects in the Object Discovery and Selection window.

Note: By default, all the business objects, whether they have a primary business component or not, are displayed.

- e. Expand the required business objects (for example, Account), to see the business components.
- f. Select the business components you want to work with and click the right arrow to add them to the **Objects to be imported** pane.

Note: The Configuration Parameters window is displayed for the selection of attributes, if you have selected the **Prompt for additional configuration settings when adding a business object** check box in the Discovery Configuration window.

- g. Select the attributes you want to generate from the Configuration Parameters window.

Note: If you do not select an attribute, an empty business object is generated.

- h. In the Object Discovery and Selection window, select the required attributes and click **OK** to add the business components to the Selected objects pane.

Note: You can follow the explained procedure (step a to step g), to continue to select business components from various business objects. For example, you can query on the opportunity business object, which is another business object in Siebel Business Applications server, and select the associated business components.

If a business object has a primary business component in it:

- 1) Select a business component. If this business component is related to the primary business component, the primary business component is selected automatically and operations are generated only for the primary business component. The business object schema is generated for both primary business component and the selected business component.

Note: No operations are generated for the child business components.

- 2) If you select a business component, and if this business component is not related to any primary business component, the operations are generated for this business component along with the business object schema.
- 3) If you select two non-primary business components and if they are related to each other but not related to the primary business component, the operations are generated for each business components along with business object schemas.

When a business object has no primary business component in it:

- 1) Select a business component or a list of business components. The operations are generated for each business component along with the business object schemas.
- i. Click **Next**. The Configure Composite Properties window is displayed.

Results

You have selected the Siebel business object you want to work with and named it. The Configure Composite Properties window is displayed.

What to do next

Configure the objects you have selected.

Selecting business services

Select business services through the J2C Bean wizard, to specify the business functions that you want to call .

Before you begin

Ensure that you have set the connection properties for the J2C Bean wizard.

About this task

Specify the search criteria that the J2C Bean wizard uses to discover business functions on the Siebel Business Applications server.

To search and select business functions, use the following procedure.

Selecting business services

1. In the Object Discovery and Selection window,
 - a. Click **Edit Query**.
 - b. In the Query Filter Parameters window, type the name of the business object in the **Pattern** field that you want to find (for example, EAI Siebel Adapter), and click **OK**.
 - c. Click **Execute Query** to display the specified business object.
 - d. In the **Objects discovered by query** pane, expand the **Siebel Business Services** folder to see the names of the business services associated with the specified object.
 - e. Expand the name of **Siebel Business Service** you want to work with, and select the business service method (for example, Insert).

Note: You can select more than one method at a time.

- f. Click the right arrow to add the service to the **Objects to be imported** pane.

Note: If the business service method you selected does not have the integration object already selected, the Configuration Parameters window is displayed.

- g. In the Configuration Parameters window, select the message associated with the business service method and click **Select** to choose **Integration Object**.

Note: If you select the **Use wrapper for multiple integration objects** check box, then you can choose multiple integration objects for the selected business service to return data. When this option is selected, the adapter generates a wrapper to hold the selected integration objects and converts them during run time. For more information, see “Application-specific information” on page 81.

- h. Click **OK** on Select Value window, and then click **OK** on Configuration Parameters window.

Note:

- 1) When a business service has only one complex type input/output argument, the Configuration Parameters window is not displayed.
 - 2) While you select an input and output complex attribute, even though the integration object cannot be selected, an input and an output SiebelMessage must be specified. Complex attributes are the variables for the SiebelMessage object. There are three types of objects, for example: input, output, and input/output. You must select either an input or an output object or an input/output object.
 - 3) A Siebel message I/O type is always displayed in the message node name and you need not click the node to see it.
 - 4) An asterisk * is added to the beginning of the Siebel message name if that message is selected.
 - 5) If a Siebel message is not selected, then do not select the **Integration Object** and it is not added as a business object.
 - 6) Siebel Enterprise Information System (EIS) restricts to create or change the existing input and output argument names for any method of a standard business service. The names are hard-coded internally in the underlying class of the business service along with the logic. Ensure that you are working on a custom Siebel business service if you want to use the custom Siebel message.
- i. Click **Next**.

Results

You have selected the Siebel business service you want to work with and named it. The Configure Composite Properties window is displayed.

What to do next

Configure the objects you have selected.

Configuring the selected objects

To configure the business object, you specify information about the object, such as name for the business object and the directory in which it has to be stored.

Before you begin

Make sure that you have selected and imported the business function.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the Configure Composite Properties window of the J2C Bean wizard, review the list of operations. This list contains all the operations that the adapter supports for the outbound services for all business objects that you selected in the previous window.

2. To remove an operation that you do not need, select the operation name and click **Remove**. To add an operation, click **Add** and restore the operation if necessary.
3. In **Maximum records for RetrieveAll operations**, enter the upper limit on the number of records to retrieve for a RetrieveAll operation. The default value is 100.

Note: If you have removed the RetrieveAll operation, the **Maximum records for RetrieveAll operations** property will be disabled.

4. In the **Business object namespace** field, accept the default namespace (<http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel>) or type the full name of another namespace. If you are adding the business object to an existing module and the module already includes that business object (from an earlier run of the J2C Bean wizard), change the namespace value.
For example, you can change the namespace to <http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel>.
5. Optionally, in **Folder**, type the relative path to the folder where the generated business objects are to be stored.

Note: If you are creating multiple adapter artifacts within a module, ensure that you specify different business object folders for each adapter within the module. For example, if you are creating artifacts for Oracle, JDBC, SAP, and JDE within a module, you need to create different relative folders for each of these adapters. If you do not specify different relative folders, the existing artifacts are overwritten when you generate new artifacts.

6. If you want the business object or business objects to be generated with shorter names, select the **Generate business objects with shorter names** check box.
7. Click **Next**.

Results

You have provided information that applies to all business objects in the application. The J2C Bean Creation and Deployment Configuration window is displayed.

What to do next

Continue working in the wizard. The next step is to specify the information for generating the artifacts.

Setting deployment properties and generating artifacts

After you select and configure business objects for your module, use the J2C Bean wizard to configure properties that the adapter uses to connect to a specific database. The wizard creates a project where all the artifacts and property values are saved.

About this task

This task is performed through the J2C Bean Creation and Deployment Configuration window of the J2C Bean wizard.

Procedure

1. In the **Project name** field, select or create a project into which the J2C Bean is generated.

- To select an existing project, click **Browse**. If the project name you want appears in the **Project Selection** list, select its name.
 - Otherwise, click **New** to create a project. In the New Source Project Creation window, select the type of project you want to create.
 - a. Java project: For information about creating a Java project, see Creating a Java project.
 - b. EJB project: For information about creating a EJB project, see Creating an EJB project.
 - c. Web project: For information about creating a web project, see Creating a web project.
2. In the **Package name** field, select or create a package into which the J2C Bean is generated.
 - To select an existing package, click **Browse**. If the package name you want appears in the **Project Selection** list, select its name.
 - Otherwise, create a package:
 - a. Click **New**.
 - b. In the Java Package window, type a name for the package. For example, myadapteroutboundpkg.
 - c. Click **Finish**.
 3. In the **Interface name** field, specify the interface name you want to use for your business objects. For example, MyAdapterOutboundInterface. The implementation name is automatically generated by suffixing "Impl" to the interface name, and the name is displayed in the **Implementation name** field. For example, MyAdapterOutboundInterfaceImpl.
 4. Optional: In the **Generate Command Bean** section, select the operation for which you want to generate a command bean. If you create command bean, you need to specify the command bean name as well as the input and output names.
 5. In the Connection properties area, specify how you want the adapter to connect to the database.
 - In the managed connection mode, you can specify connection properties in the Java Naming and Directory Interface (JNDI) name of the managed connection factory defined on the application server. The managed connection factory is EIS adapter-specific and contains all of the required connection information. Managed connection implies that the resource adapter is installed directly on the server and, therefore, the JNDI name of the managed connection factory is visible to all EAR (enterprise application archive) files installed on the application server. To obtain the connection through JNDI, select the **Managed Connection (recommended)** check box. This type of connection is managed by the application server.
 - In the non-managed connection mode, the application communicates directly to the EIS and manages all the connections. Therefore, all of the connection information in this setup is encapsulated within the application. For a J2C bean, this means that all of the connection information is specified in the generated J2C bean. To obtain the connection directly from the resource adapter, select the **Non-managed Connection** check box.

Note: If you check both choices, the Managed Connection is executed first. If the JNDI name cannot be found, the non-managed connection parameters is used.

6. If you select **Managed Connection (recommended)** check box, you specify the connection properties in the JNDI name of the managed connection factory defined on the application server. You can either select an existing JNDI or create a new one.
 - To select an existing JNDI name:
 - a. Click **Browse**.
 - b. In the Server selection window, specify the server on which to deploy the adapter and click **Next**.
 - c. Select the existing JNDI name in the J2C Connection Factories window. The existing RAR and the JNDI name are used for the connection factory.
 - To create a new one, click **New**.
 - a. In the Server selection window, specify the server on which to deploy the adapter and click **Next**.
 - b. In the New J2C Connection Factory window, specify the name in the **JNDI Name** field.
 - c. Specify the advanced properties by clicking **Advanced**. The RAR file is installed on IBM WebSphere Application Server and the connection factory is created using the JNDI name specified.
 - d. Expand each of the following group sections to review the advanced properties:
 - **Runtime configuration**
 - To specify that the adapter throws an exception when no records are found in the RetrieveAll operation for a Java beans or a business service, select the **Show error on empty result set** check box. For more information, see “Show error on empty result set (errorOnEmptyResultSet)” on page 106.
 - To specify the code page that the adapter uses to connect to the Siebel Business Applications server, click **Select** and then select the value from the list. For more information, see “Code page (codePage)” on page 102.
 - To specify that the adapter returns the elements with an empty string value in the response Java beans, select the **Include empty elements in the output** check box. For more information, see “Include empty elements in the output (includeEmptyField)” on page 103.
 - **Connection retry settings**
 - To specify the number of times the adapter can attempt to reconnect to the Siebel Business Applications server in case of connection failure, set **Maximum retries on connection failure** to a value greater than or equal to zero. For more information, see “Maximum retries in case of system connection failure (connectionRetryLimit)” on page 104.
 - To specify the time interval between attempts to connect to the Siebel Business Applications server if the connection fails, set **Connection retry interval (in milliseconds)** to a value in milliseconds. This property is enabled only when the connectionRetryLimit property has a value greater than zero. For more information, see “Retry interval if connection fails (in milliseconds) (connectionRetryInterval)” on page 106.
 - **Logging and tracing**

- If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
- If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.
- **Other properties**
 - Select **Use Resonate for load balancing on Siebel server** if you want to specify that the Siebel Business Applications server uses resonate support. For more information, see “Use resonate support for load balancing on Siebel server (UseResonateSupportForLoadBalancingOnSiebelServer) ” on page 95.
 - Select the view mode from the **Siebel server view mode** drop-down list. For more information, see Siebel server view mode.
- e. Click **Finish** to return to the J2C Bean Creation and Deployment Configuration window.
- 7. If you select the **Non-managed Connection** check box, the system connection information area is expanded to show the connection information. Review the connection information and change the values if required. Specify the advanced properties by clicking **Advanced**. Refer to the advanced properties defined for managed connection.
- 8. When you are finished setting properties, select the **Click to launch the J2C deployment** checkbox to immediately start the wizard.
- 9. Click **Finish**.

Results

The new project is added to the Enterprise Explorer perspective. The module is created in the project and artifacts are generated.

The generated artifacts allow you to build an enterprise application that accesses the EIS. You can use the J2C Bean and Service Bean directly, in the non-managed mode or generate JSP or EJB that uses the J2C Bean. Rational Application Developer for WebSphere Software provides tools to automate this generation. You can access these tools from the **New>Others** menu, for their description refer to the Rational Application Developer for WebSphere Software documentation.

Generating the EJB or JSP project

After you create the J2C bean, use the Web Page, Web Service, or EJB from J2C Java wizard to create the JSP, EJB or web service.

About this task

You can create a JSP, an EJB, or a web service to deploy your J2C bean.

Procedure

1. To start the J2C Bean wizard, go to the Enterprise Explorer of Rational Application Developer for WebSphere Software and click **New>Other>J2C>Web Page, Web Service, or EJB from J2C Java Bean**.
2. Click **Next**.
3. In the J2C Java bean selection window, click **Browse**.

4. In the Find J2C bean window, type the first letter of the implementation name generated earlier or type the full name and press the Enter key.
5. Select the implementation name from the list and click **OK**.
6. Click **Next**.
7. In the Deployment Information window, select the Java EE Resource Type as **EJB**, **Simple JSP** or **Web Service** and click **Next**.

Note: In the Deployment Information window, the **Configure Resource Adapter Deployment** check box is available for selection only if you have selected the **Non-managed Connection** check box when specifying the deployment settings.

8. If you select **EJB**, the Enterprise bean creation wizard is displayed. This wizard creates the Java project as an EJB. For information about creating an EJB, see Deploying to an EJB.

Note: You can deploy an EJB only when the J2C bean is in the EJB project. Otherwise, this option is not available for selection.

9. If you select **Simple JSP**, the Simple JSP Creation wizard is displayed. For details about creating a new web project, see Creating a web project. For information about creating a Simple JSP, see Deploying to a Simple JSP.
10. If you select **Web Service**, the Web Service Creation wizard is displayed. For information about creating a web service, Deploying a J2C application as a web service.
11. Export the project as an EAR file for deployment.

Configuring the module for inbound processing

To configure a module to use the adapter for inbound processing, use the J2C Bean wizard in Rational Application Developer for WebSphere Software to find and select business objects and services from the Siebel application, and to generate business object definitions and related artifacts.

Selecting business objects

Select business objects through the J2C Bean wizard, that you want to process.

Before you begin

Ensure that you have set the connection properties for the J2C Bean wizard.

About this task

Specify the search criteria that the J2C Bean wizard uses to discover business objects or business functions on the Siebel application.

To search and select business objects or business functions, use the following procedure.

Selecting business objects

1. In the Object Discovery and Selection window,
 - a. Click **Edit Query**.
 - b. In the Query Filter Parameters window, type the name of the business object in the **Pattern** field that you want to find (for example, Account), and click **OK**.

- c. Click Execute Query to display the specified business object.
- d. Expand the **Siebel Business Objects** folder in the **Objects discovered by query** pane to see the list of discovered business objects in the Object Discovery and Selection window.

Note: By default, all the business objects, whether they have a primary business component or not, are displayed.

- e. Expand the required business objects (for example, Account), to see the business components.
- f. Select the business components you want to work with and click the right arrow to add them to the **Objects to be imported** pane.

Note: The Configuration Parameters window is displayed for the selection of attributes, if you have selected the **Prompt for additional configuration settings when adding a business object** check box in the Discovery Configuration window.

- g. Select the attributes you want to generate from the Configuration Parameters window.

Note: If you do not select an attribute, an empty business object is generated.

- h. In the **Discovered objects** window, select the required attributes and click **OK** to add the business components to the **Selected objects** pane.

Note: You can follow the explained procedure (step a to step g), to select the business components from various business objects. For example, you can query on opportunity business objects, which is another business object in Siebel Business Applications server and select its business components.

If a business object has a primary business component in it:

- 1) Select a business component. If this business component is related to the primary business component, the primary business component is selected automatically and operations are generated only for the primary business component. The business object schema is generated for both primary business component and the selected business component.

Note: No operations are generated for the child business components.

- 2) If you select a business component, and if this business component is not related to any primary business component, the operations are generated for this business component along with the business object schema.
- 3) If you select two non-primary business components and if they are related to each other but not related to the primary business component, the operations are generated for each business components along with business object schemas.

When a business object has no primary business component in it:

- 1) Select a business component or a list of business components. The operations are generated for each business component along with business object schemas.
- i. Click **Next**. The Configure Composite Properties window is displayed.

Results

You have selected the Siebel business object you want to work with and named it. The Configure Composite Properties window is displayed.

What to do next

Configure the objects you have selected.

Selecting business services

Select business services through the J2C Bean wizard to specify the business functions that you want to call.

Before you begin

Ensure that you have set the connection properties for the J2C Bean wizard.

About this task

Specify the search criteria that the J2C Bean wizard uses to discover business functions on the Siebel application.

To search and select business functions, use the following procedure.

Selecting business services

1. In the Object Discovery and Selection window:
 - a. Click **Edit Query**.
 - b. In the Query Filter Parameters window, type the name of the business object in the **Pattern** field that you want to find (for example, EAI Siebel Adapter), and click **OK**.
 - c. Click **Execute Query** to display the specified business object.
 - d. In the **Objects discovered by query** pane, expand the **Siebel Business Services** folder to see the names of the business services associated with the specified object.
 - e. Expand the name of the **Siebel Business Service** you want to work with, and select the business service method (for example, Insert).

Note: You can select more than one method at a time.

- f. Click the right arrow to add the service to the **Objects to be imported** pane.

Note: If the business service method you selected does not have the integration object already selected, the Configuration Parameters window is displayed.

- g. In the Configuration Parameters window, select the message associated with the business service method and click **Select** to choose the **Integration Object**.
- h. Click **OK** on Select Value window, and then click **OK** on Configuration Parameters window.

Note:

- 1) When a business service has only one complex type Input/Output argument, the Configuration Parameters window is not displayed.

- 2) While you select an input and output complex attribute, even though the integration object cannot be selected, an input and output SiebelMessage must be specified. Complex attributes are the variables for the SiebelMessage object. There are three types of objects, for example: input, output, and input/output. You must select either an input or an output object or an input/output object.
 - 3) A Siebel message I/O type is always displayed in the message node name and you need not click the node to see it.
 - 4) An asterisk * is added to the beginning of the Siebel message name if that message is selected.
 - 5) If a Siebel message is not selected, then do not select **Integration Object** and it is not added as a business object.
 - 6) Siebel Enterprise Information System (EIS) restricts to create or change the existing input and output argument names for any method of a standard business service. The names are hard-coded internally in the underlying class of the business service along with the logic. Ensure that you are working on a custom Siebel business service if you want to use the custom Siebel message.
- i. Click **Next**.

Results

You have selected the Siebel business service you want to work with and named it. The Configure Composite Properties window is displayed.

What to do next

Configure the objects you have selected.

Configuring the selected objects

To configure the business object, you specify information about the object, such as a name for the business object and the directory in which it must be stored.

Before you begin

Ensure that you have selected and imported the business function.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the Configure Composite Properties window of the J2C Bean wizard, review the list of operations. This list contains all the operations that the adapter supports for the inbound services for all business objects that you selected in the previous window. The specified operations are set for all business objects that are generated.
2. To remove an operation that you do not need, select the operation name and click **Remove**. To add an operation, click **Add** and restore the operation if necessary.
3. In the **Business object namespace** field, accept the default namespace (<http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel>) or type the full name of another namespace. If you are adding the business object to an existing module

and the module already includes that business object (from an earlier run of the J2C Bean wizard), change the namespace value.

For example, you could change the namespace to `http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel`

4. Optionally, in **Folder**, type the relative path to the folder where the generated business objects are to be stored.

Note: If you are creating multiple adapter artifacts within a module, ensure that you specify different business object folders for each adapter within the module. For example, if you are creating artifacts for Oracle, JDBC, SAP, and JDE within a module, you need to create different relative folders for each of these adapters. If you do not specify different relative folders, the existing artifacts are overwritten when you generate new artifacts.

5. If you want the business object or business objects to be generated with shorter names, select **Generate business objects with shorter names**, and then Click **Next**.

Results

You have provided information that applies to all business objects in the application. The J2C Bean Creation and Deployment Configuration window is displayed.

What to do next

Continue working in the wizard. The next step is to specify the information for generating the artifacts.

Setting deployment properties and generating artifacts

After you select and configure business objects for your module, use the J2C Bean wizard to configure properties that the adapter uses to connect to a specific database. The wizard creates a project where all the artifacts and property values are saved.

About this task

This task is performed through the J2C Bean Creation and Deployment Configuration window of the J2C Bean wizard.

Procedure

1. In the **EJB Project name** field, select or create a new EJB project.
 - To select an existing project, click **Browse**. If the required project name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new project:
 - a. Click **New**.
 - b. In the EJB Project window, type a project name. For example, `MyAdapterInboundEJB`.
 - c. In the EAR Membership area, click **New** to create a new ear project.
 - d. In EAR Application Project window, type an EAR project name. For example, `MyAdapterInboundEJBEAR`
 - e. Click **Finish** to return to the EJB Project window.
 - f. Click **Finish**.

2. In the **Package name** field, select or create a new package into which the J2C Bean is generated.
 - To select an existing package, click **Browse**. If the required package name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new package:
 - a. Click **New**.
 - b. In the New Java Package window, type a name for the package. For example, myadapterinboundejbpkg.
 - c. Click **Finish**.
3. In the **Stateless Session EJB's local business interface name** field, specify the interface name you want to use for your business object. For example, MyAdapterInboundInterface. The interface name is suffixed with "MDB" and it is displayed automatically in the **Message Driven EJB name** field. For example, MyAdapterInboundInterfaceMDB. Similarly, the interface name is suffixed with "SB" and it is displayed automatically in the **Stateless Session EJB name** field. For example, MyAdapterInboundInterfaceSB.
4. In the Inbound Connection configuration area, select the Java Naming and Directory Interface (JNDI) name for an existing activation specification in WebSphere Application Server or create an activation specification. For more information about creating an activation specification in WebSphere Application Server, see "Setting activation specification properties for stand-alone adapters" on page 65.
 - To select an existing activation specification,
 - a. Click **Browse**.
 - b. In the Server selection window, select the server on which the resource adapter will be deployed and click **Next**.
 - c. Select the existing JNDI name in the J2C Activation Specification window. The existing RAR and the JNDI name are used for the activation specification.

Note: When you try to look up an existing JNDI in the JNDI Lookup wizard screen, the wizard looks up only the JNDI that is created using the first RAR file. If you have more than one RAR file for the same adapter in the server, you can view only the JNDI generated using the first RAR file. The JNDI generated from the other RAR files are not looked up by the wizard.

- To create a new activation specification, click **New**.
 - a. In the Server selection window, select the server on which the resource adapter will be deployed and click **Next**.
 - b. In the New J2C Activation Specification window, type a JNDI name for the activation specification.
 - c. Specify the advanced properties by clicking **Advanced**. The RAR file is installed on IBM WebSphere Application Server and the activation specification is created using the JNDI name specified.
 - d. Expand each of the following group sections to review the advanced properties:
 - **Event polling configuration**
 - In **Interval between polling periods**, type the number of milliseconds that the adapter waits between polling periods. For more information, see "Interval between polling periods (pollPeriod)" on page 129.

- In **Maximum events in polling period**, type the number of events to deliver in each polling period. For more information, see “Maximum events in polling period (pollQuantity)” on page 130.
 - In **Time between retries in case of system connection failure (in milliseconds)**, type the number of milliseconds to wait before trying to connect after a connection failure during polling. For more information, see “Time between retries in case of system connection failure (retryInterval)” on page 132.
 - In **Maximum number of retries in case of system connection failure**, type the number of times to retry the connection before reporting a polling error. For more information, see “Maximum number of retries in case of system connection failure (retryLimit)” on page 131.
 - If you want the adapter to stop if polling errors occur, select **Stop the adapter when an error is encountered while polling**. If you do not select this option, the adapter logs an exception but continues to run. For more information, see “Stop the adapter when an error is encountered while polling (StopPollingOnError)” on page 133.
 - You can select **Retry EIS connection on startup** if you want the adapter to retry an inbound connection that was not made to the Siebel application when starting. Only communication failures to the Siebel application are considered. For more information, see “Retry EIS connection on startup (retryConnectionOnStartup)” on page 132.
- **Event delivery configuration**
- In **Type of delivery**, select the delivery method. The methods are described in “Delivery type (deliveryType)” on page 126.
 - If you want to ensure that events are delivered only once and to only one export, select **Ensure assured-once event delivery**. This option might reduce performance but does not result in duplicate or missing event delivery. For more information, see “Ensure once-only event delivery (assuredOnceDelivery)” on page 126.
 - In the **Event types to process** field, type a comma-separated list of the business objects for which you want events delivered. Leave this field blank to receive events for all business object types. For example, if you want to receive events only when the Customer and Order tables, but not other tables, are changed in the enterprise information system, set this field to Customer,Order. For more information, see “Event types to process (EventTypeFilter)” on page 127.
 - In the **Adapter Instance for event filtering** field, specify the connector ID where the events will be delivered. For more information, see “Adapter Instance for event filtering (AdapterInstanceEventFilter)” on page 124.
 - In the **Retry limit for failed events** field, specify the number of times to try to deliver an event after a delivery failure. For more information, see “Retry limit for failed events (failedEventRetryLimit)” on page 128.
 - Under **Number of connections for event delivery**, specify the minimum and maximum number of connections to use to deliver events. For more information, see “Minimum connections (MinimumConnections)” on page 131 and “Maximum connections (MaximumConnections)” on page 130.
- **Event configuration**

- In the **Siebel business object name for event store** field, specify the name of the business object in the event store where events are stored for inbound processing. For more information, see “Siebel business object name for event store (SiebelBusinessObjectNameForEventStore) ” on page 93

Note: You cannot use multiple adapters for polling the same event.

– **Runtime configuration**

- In the **Code page** field, specify the code page that the adapter uses to connect to the Siebel Business Applications server by clicking **Select**. Select the value from the displayed list. For more information, see “Code page (codePage)” on page 125.
 - Select the **Include empty elements in the output** check box for the adapter to include the elements with empty string value in the response Java beans. For more information, see “Include empty elements in the output (includeEmptyField)” on page 129.
- e. Click **Finish** to return to J2C Bean Creation and Deployment Configuration window.
5. Type the existing Java Authentication and Authorization Services alias. The alias is used to retrieve the user name and password set on the configured J2C activation specification. The name is case sensitive and includes the node name.
 6. In the Service Operations section, click Edit Operations to review the names of operations or add a description about the operations to be generated in the interface file.
 7. Click **Finish**.

What to do next

Deploy the module and test.

Chapter 4. Deploying the module

Deploy a module to place the files that make up your module and adapter into an operational environment for testing. In Rational Application Developer for WebSphere Software, the integrated test environment features runtime support for WebSphere Application Server, depending on the test environment profiles that you selected during installation.

Deployment environments

There are test and production environments into which you can deploy modules and adapters.

In Rational Application Developer for WebSphere Software, you can deploy your modules to one or more servers in the test environment. This is typically the most common practice for running and testing projects. However, you can also export modules for server deployment on WebSphere Application Server as EAR files using the WebSphere Application Server Administrative Console or command-line tools.

Deploying the module for testing

In Rational Application Developer for WebSphere Software, you can deploy a module that includes an embedded adapter to the test environment and work with server tools that enable you to perform such tasks as editing server configurations, starting, and stopping servers and testing the module code for errors. The testing is generally performed on the interface operations of your components, which enables you to determine whether the components are correctly implemented and the references are correctly wired.

Configuring the connector dependencies

Dependent JARs have to be added to the libraries directory or packaged in the EAR.

About this task

The JARs are set in the class path and these dependent libraries have to be made available for run time when the module is deployed. There are two ways to make the dependent libraries available, one for either stand-alone deployment or embedded deployment and the other for embedded deployment only.

Configuring the connector dependencies on the server

The adapter needs the Siebel JAR files installed on the WebSphere Application Server to be able to communicate with the Siebel application.

About this task

To obtain the required files and copy them to WebSphere Application Server, use the following procedure.

Procedure

1. Obtain the Siebel dependency JAR files from your Siebel Business Applications administrator or the workstation on which the Siebel server is installed. Depending on the version of Siebel Business Applications you use, the software dependencies differ. The following table lists the dependency files needed by supported versions of Siebel Business Applications. Each file is shown with the corresponding field label in the J2C Bean wizard.

Table 6. Connector dependency files required by Siebel Business Applications

J2C Bean wizard property name	Required dependency files for Siebel Business Applications (versions 7.5 and earlier)	Required dependency files for Siebel Business Applications (versions 7.7x, 7.8x, 8.0, and 8.1.1)
Siebel.jar	None	Siebel.jar
SiebelJI_<language code>.jar	SiebelJI_<language code>.jar (for example, SiebelJI_enu.jar)	SiebelJI_<language code>.jar (for example, SiebelJI_enu.jar)
SiebelJI_Common.jar	SiebelJI_Common.jar	None

2. Copy the files to the server.
 - In a testing environment in Rational Application Developer for WebSphere Software, copy the files to the `${WAS_INSTALL_ROOT}/runtimes/bi_v7/lib/ext` directory.
 - In a production environment, copy the files to the `${WAS_INSTALL_ROOT}/lib/ext` directory of WebSphere Application Server.

Configuring the connector dependencies when the adapter is bundled

You must copy the dependent JAR files to the EAR application before you can run your adapter applications. You must use this method only for embedded deployment.

About this task

To obtain the required files and copy them to the EAR application, use the following procedure:

Procedure

1. From the appropriate module, go to the workspace and copy the JAR files to the directory. For example if the name of the module is `ModuleName`, then go to the workspace and copy the JAR files to the `ModuleNameApp/EarContent` directory.
2. Modify the adapter RAR's manifest file, `manifest.mf`, with the list of JAR files required by the adapter. Add the JAR files in the following format: `Class-Path: dependantjar1.jar, dependantjar2.jar`.
3. Copy the native libraries to the run time bin directory and deploy the application.

Results

The vendor software libraries are now a part of your run time environment.

Adding the module to the server

In Rational Application Developer for WebSphere Software, you can add modules to one or more servers in the test environment.

Before you begin

If the module you are testing uses an adapter to perform inbound processing, generate and wire a *target component* to which the adapter sends the events.

About this task

In order to test your module and its use of the adapter, you need to add the module to the server.

Procedure

1. *Conditional:* If there are no servers in the **Servers** view, add and define a new server by performing the following steps:
 - a. Place your cursor in the **Servers** view, right-click, and select **New > Server**.
 - b. From the Define a New Server window, select the server type.
 - c. Configure servers settings.
 - d. Click **Finish** to publish the server.
2. Add the module to the server.
 - a. Switch to the servers view. In Rational Application Developer for WebSphere Software, select **Windows > Show View > Servers**.
 - a. Start the server. In the **Servers** tab in the lower-right pane of the Rational Application Developer for WebSphere Software screen, right-click the server, and then select **Start**.
3. When the server status is *Started*, right-click the server, and select **Add and Remove**.
4. In the Add and Remove Projects screen, select your project and click **Add**. The project moves from the **Available projects** list to the **Configured projects** list.
5. Click **Finish**. This deploys the module on the server.

The Console tab in the lower-right pane displays a log while the module is being added to the server.

What to do next

Test the functionality of your module and the adapter. For information about testing a module using the test client, see the *Testing applications on a server* topic in the Rational Application Developer for WebSphere Software information center.

Deploying the module for production

Deploying a module created with the J2C Bean wizard to WebSphere Application Server in a production environment is a two-step process. First, you export the module in Rational Application Developer for WebSphere Software as an enterprise archive (EAR) file. Second, you deploy the EAR file using the WebSphere Application Server WebSphere Application Server Administrative Console.

Configuring the connector dependencies on the server

The adapter needs the Siebel JAR files installed on the WebSphere Application Server to be able to communicate with the Siebel application.

About this task

To obtain the required files and copy them to WebSphere Application Server, use the following procedure.

Procedure

1. Obtain the Siebel dependency JAR files from your Siebel Business Applications administrator or the workstation on which the Siebel server is installed. Depending on the version of Siebel Business Applications you use, the software dependencies differ. The following table lists the dependency files needed by supported versions of Siebel Business Applications. Each file is shown with the corresponding field label in the J2C Bean wizard.

Table 7. Connector dependency files required by Siebel Business Applications

J2C Bean wizard property name	Required dependency files for Siebel Business Applications (versions 7.5 and earlier)	Required dependency files for Siebel Business Applications (versions 7.7x, 7.8x, 8.0, and 8.1.1)
Siebel.jar	None	Siebel.jar
SiebelJI_<language code>.jar	SiebelJI_<language code>.jar (for example, SiebelJI_enu.jar)	SiebelJI_<language code>.jar (for example, SiebelJI_enu.jar)
SiebelJI_Common.jar	SiebelJI_Common.jar	None

2. Copy the files to the server.
 - In a testing environment in Rational Application Developer for WebSphere Software, copy the files to the `${WAS_INSTALL_ROOT}/runtimes/bi_v7/lib/ext` directory.
 - In a production environment, copy the files to the `${WAS_INSTALL_ROOT}/lib/ext` directory of WebSphere Application Server.

Installing the RAR file (for modules using stand-alone adapters only)

If you chose not to embed the adapter with your module, but instead choose to make the adapter available to all deployed applications in the server instance, you need to install the adapter in the form of a RAR file to the application server. A RAR file is a Java archive (JAR) file that is used to package a resource adapter for the Java EE Connector Architecture (JCA).

Before you begin

You must set **Deploy connector project** to **On server for use by multiple adapters** in the J2C Bean Creation and Deployment Configuration window of the J2C Bean wizard.

About this task

Installing the adapter in the form of a RAR file results in the adapter being available to all Java EE application components running in the server run time.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.

3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **Install RAR**.
6. In the Install RAR file page, click **Browse** and navigate to the RAR file for your adapter.
The RAR files are typically installed in the following path:
RAD_installation_directory/ResourceAdapters/adapter_name/adapter.rar
7. Click **Next**.
8. Optional: In the Resource adapters page, change the name of the adapter and add a description.
9. Click **OK**.
10. Click **Save** in the **Messages** box at the top of the page.

What to do next

The next step is to export the module as an EAR file that you can deploy on the server.

Exporting the module as an EAR file

Using Rational Application Developer for WebSphere Software, export your module as an EAR file. By creating an EAR file, you capture all of the contents of your module in a format that can be easily deployed to WebSphere Application Server.

Before you begin

Before you can export a module as an EAR file, you must have created a module to communicate with your service. The module should be displayed in the Rational Application Developer for WebSphere Software Enterprise Explorer view.

About this task

To export the module as an EAR file, perform the following procedure.

Procedure

1. Right-click the module and select **Export**.
2. In the Select window, expand **Java EE**.
3. Select **EAR file** and click **Next**.
4. Optional: Select the correct EAR application. The EAR application is named after your module, but with “App” added to the end of the name.
5. Browse for the folder on the local file system where the EAR file will be placed.
6. Optional: To export the source files, select the **Export source files** check box. This option is provided in case you want to export the source files in addition to the EAR file. Source files include files associated with Java components, data maps, and so on.
7. Optional: To overwrite an existing file, click **Overwrite existing file**.
8. Click **Finish**.

Results

The contents of the module are exported as an EAR file.

What to do next

Install the module in the WebSphere Application Server Administrative Console. This deploys the module to WebSphere Application Server.

Installing the EAR file

Installing the EAR file is the last step of the deployment process. When you install the EAR file on the server and run it, the adapter, which is embedded as part of the EAR file, runs as part of the installed application.

Before you begin

You must have exported your module as an EAR file before you can install it on WebSphere Application Server.

About this task

To install the EAR file, perform the following procedure. For more information about clustering adapter module applications, see the <http://www.ibm.com/software/webservers/appserv/was/library/>.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > New Application > New Enterprise Application**.
5. Click **Browse** to locate your EAR file and click **Next**. The EAR file name is the name of the module followed by "App."
6. Optional: If you are deploying to a clustered environment, complete the following steps.
 - a. On the **Step 2: Map modules to servers** window, select the module and click **Next**.
 - b. Select the name of the server cluster.
 - c. Click **Apply**.
7. Click **Next**. In the Summary page, verify the settings and click **Finish**.
8. Optional: If you are using an authentication alias, complete the following steps:
 - a. Expand **Security** and select **Business Integration Security**.
 - b. Select the authentication alias that you want to configure. You must have administrator or operator rights to change the authentication alias configurations.
 - c. Optional: If it is not already specified, type the **User name**.
 - d. If it is not already specified, type the **Password**.
 - e. If it is not already specified, type the password again in the **Confirm Password** field.
 - f. Click **OK**.

Results

The project is now deployed and the Enterprise Applications window is displayed.

What to do next

If you want to set or reset any properties or you would like to cluster adapter project applications, make those changes using the WebSphere Application Server Administrative Console before configuring troubleshooting tools.

Deploying the module in a clustered environment

In Rational Application Developer for WebSphere Software, you can deploy the IBM WebSphere Adapter for Siebel Business Applications in a clustered environment.

To deploy the module in a clustered environment, use any of the following approaches.

- **Embedded module:** The adapter is embedded in the application and specific to it. The adapter cannot be shared between multiple applications.
- **Node level module with embedded activation specification:** The adapter is deployed at the node level, with the activation specification created during module creation. The adapter can be shared across multiple applications.
- **Node level module with JNDI activation specification reference:** The adapter is deployed at the node level, and the application provides a JNDI reference to the activation specification. You must create the reference at the cluster scope from the WebSphere Application Server Administrative Console, with the same JNDI name. The adapter can be shared across multiple applications.

Deploying module embedded in the application

The adapter is deployed embedded in the application and specific to it. The adapter cannot be shared between multiple applications.

Before you begin

The following steps are a necessary prerequisite to configure and deploy the module.

- Rational Application Developer for WebSphere Software version 7.5.0.0 or above.
- A clustered topology deployment environment on the WebSphere Application Server available from Rational Application Developer for WebSphere Software.
- Create a clustered topology deployment environment, as shown in the following **Gold Topology** configuration figure.
- Deploy the adapter and the adapter applications (EAR files) in the AppTarget (the target that hosts the SCA container).

About this task

To create an application with the embedded adapter, use the J2C Bean wizard.

Procedure

1. In the Service Configuration Properties window, from the **Deploy connector project** property list, select **With module for use by single application**.
2. Create the module as described in the Business process management samples for WebSphere Adapters.
3. In the **Dependencies** option for the module, after the module is created, ensure that the **Deploy with module** option is selected for the adapter.

4. If the server is not running, right-click your server in the **Servers** view and select **Start**.
5. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
6. From the **Deployment Manager Admin Console**, click **Install applications** to deploy the application.
7. On the **Step 2: Map modules to servers** window, select the module and click **Next**. For the embedded adapter option, the adapter is deployed as part of the application.
8. In the Enterprise Applications view, select the new application `<adapter_name>EmbeddedModuleApp`. The new application is displayed after the application is deployed at the deployment manager level.
9. Select the node and click **Installed applications** to view the deployed application on each individual node.

Results

The resource adapter is embedded and deployed as part of the application.

Deploying module at node level with embedded activation specification

The adapter is deployed at the node level, with the activation specification created during module creation. The adapter can be shared across multiple applications.

Before you begin

The following steps are a necessary prerequisite to configure and deploy the module.

- Rational Application Developer for WebSphere Software version 7.5.0.0 or above.
- A clustered topology deployment environment on the WebSphere Application Server available from Rational Application Developer for WebSphere Software.
- Create a clustered topology deployment environment, as shown in the following **Gold Topology** configuration figure.
- Deploy the adapter and the adapter applications (EAR files) in the AppTarget (the target that hosts the SCA container).

About this task

To create an application with the node level adapter and activation specification properties specified in the module itself, use the J2C Bean wizard.

Procedure

1. In the Service Configuration Properties window, from the **Deploy connector project** property list, select **On server for use by multiple applications**.
2. From the **Connection properties** list, select **Use properties below**.
3. Create the module as described in the Business process management samples for WebSphere Adapters.
4. In the **Dependencies** option for the module, ensure that the **Deploy with module** option is not selected for the adapter. Here, the adapter is not part of the module, therefore you must deploy the adapter before deploying the application.

5. If the server is not running, right-click your server in the **Servers** view and select **Start**.
6. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**. Log on to the administrative console.
7. To deploy the adapter at individual nodes, click **Resources > Resource Adapters > Resource adapters**. In the clustered environment, you must install the adapter in each node separately.
8. In the Resource adapters page, click **Install RAR**.
9. In the Install RAR file page, click **Browse** and navigate to the RAR file for your adapter. Deploy the RAR on each node.
The RAR files are typically installed in the following path:
RAD_installation_directory/ResourceAdapters/adapter_name/adapter.rar
10. For deployment at node level, do not select any **Scope** because the scope is always **Node**. Click **Next**.
11. Optional: In the Resource adapters page, change the name of the adapter and add a description. Click **OK**.
12. Click **Save** in the **Messages** box at the top of the page.
13. For node level deployment, check if the adapter RAR is deployed at the node level.
14. To deploy the adapter at the cluster level, click **Resources > Resource Adapters > Resource adapters**.
15. In the Resource adapters window, set the **Scope** to **Cluster**, and then click **New**.
16. Select the RAR deployed at the node level.
17. Check if the adapter RAR is now deployed at the cluster level. Deploy the application after the adapter is deployed at the node level on the individual nodes, and then at the cluster level.
18. From the **Deployment Manager Admin Console**, click **Install applications** to deploy the application.
19. On the **Step 2: Map modules to servers** window, select the module and click **Next**. The adapter is not part of the deployed application.
20. In the **Admin Console**, click **Resources > Resource Adapters > IBM WebSphere Adapter for Siebel Business Applications > J2C activation specifications** to view the activation specification from the adapter deployed at the cluster level.

Results

The resource adapter is deployed at the node level, with the activation specification.

Deploying module at node level with JNDI activation specification

The adapter is deployed at the node level, and the application provides a JNDI reference to the activation specification. You must create the activation specification with the same JNDI name at the cluster scope from the WebSphere Application Server Administrative Console. The adapter can be shared across multiple applications

Before you begin

The following steps are a necessary prerequisite to configure and deploy the module.

- Rational Application Developer for WebSphere Software version 7.5.0.0 or above.
- A clustered topology deployment environment on the WebSphere Application Server available from Rational Application Developer for WebSphere Software.
- Create a clustered topology deployment environment, as shown in the following **Gold Topology** configuration figure.
- Deploy the adapter and the adapter applications (EAR files) in the AppTarget (the target that hosts the SCA container).

About this task

To create an application with the node level adapter and activation specification properties specified in the module itself, use the J2C Bean wizard.

Procedure

1. In the Service Configuration Properties window, from the **Deploy connector project** property list, select **On server for use by multiple applications**.
2. From the **Connection properties** list, select **Use JNDI lookup name configured on server**.
3. In the **JNDI lookup name** property field, specify the JNDI name. Use this same JNDI name when you create the activation specification from the Admin Console.
4. Create the module as described in the Business process management samples for WebSphere Adapters.
5. In the **Dependencies** option for the module, ensure that the **Deploy with module** option is not selected for the adapter.
6. If the server is not running, right-click your server in the **Servers** view and select **Start**.
7. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**. Log on to the administrative console.
8. To install the adapter at the node level, click **Resources > Resource Adapters > Resource adapters**. In the clustered environment, you must install the adapter in each node separately.
9. In the Resource adapters page, click **Install RAR**.
10. In the Install RAR file page, click **Browse** and navigate to the RAR file for your adapter. Deploy the RAR on each node.
The RAR files are typically installed in the following path:
IID_installation_directory/ResourceAdapters/adapter_name/adapter.rar
11. For deployment at node level, do not select any **Scope** because the scope is always **Node**. Click **Next**.
12. Optional: In the Resource adapters page, change the name of the adapter and add a description. Click **OK**.
13. Click **Save** in the **Messages** box at the top of the page.
14. To install the RAR at the cluster level, click **Resources > Resource Adapters > Resource adapters**
15. In the Resource adapters page, set the **Scope** to **Cluster**, and then click **New**.

16. Select the RAR deployed at the node level, and then check if the adapter RAR is now deployed at the cluster level. Deploy the application after the adapter is deployed at the node level on the individual nodes, and then at the cluster level.
17. From the **Deployment Manager Admin Console**, click **Install applications** to deploy the application.
18. In the **Admin Console**, click **Resources > Resource Adapters > IBM WebSphere Adapter for Siebel Business Applications > J2C activation specifications > New** to create the activation specification from the adapter deployed at the cluster level.
19. When installing the adapter, in the **Name** field, you must enter the same name as defined in the RAR.
20. In the **JNDI name** field, you must enter the same name as given during the module creation.
21. Click **Resources > Resource Adapters > IBM WebSphere Adapter for Siebel Business Applications > J2C activation specifications** to check if the JNDI reference on the adapter is same as the one specified for the module.
22. Click **Resources > Resource Adapters > IBM WebSphere Adapter for Siebel Business Applications > J2C activation specifications > Custom properties** to set values for the activation specification in the Admin Console.
23. From the **Deployment Manager Admin console**, click **Install applications** to deploy the application after you deploy the RAR and create the activation specification.
24. On the **Step 2: Map modules to servers** page, select the module and click **Next**. The adapter is not part of the deployed application.

Results

The resource adapter is deployed at the node level, with the JNDI activation specification reference.

Chapter 5. Configuring the application on WebSphere Application Server

When you are running the adapter in a stand-alone deployment, use the administrative console of the server to start, stop, monitor, and troubleshoot the adapter module. In an application that uses an embedded adapter, the adapter module starts or stops when the application is started or stopped.

Configuring logging and tracing

Configure logging and tracing to suit your requirements. Enable logging for the adapter to control the status of event processing. Change the adapter log and trace file names to separate them from other log and trace files.

Configuring logging properties

Use the WebSphere Application Server Administrative Console to enable logging and to set the output properties for a log, including the location, level of detail, and output format of the log.

About this task

Before the adapters can log monitored events, you must specify the service component event points that you want to monitor, what level of detail you require for each event, and format of the output used to publish the events to the logs. Use the administrative console to perform the following tasks:

- Enable or disable a particular event log
- Specify the level of detail in a log
- Specify where log files are stored and how many log files are kept
- Specify the format for log output

If you set the output for log analyzer format, you can open trace output using the Log Analyzer tool, which is an application included with your application server. This is useful if you are trying to correlate traces from two different server processes, because it allows you to use the merge capability of the Log Analyzer.

Note: For more information about monitoring on a application server, including service components and event points, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5m1/topic/com.ibm.wbpm.admin.doc/topics/welcome_wps_mon.html.

You can change the log configuration statically or dynamically. Static configuration takes effect when you start or restart the application server. Dynamic or run time configuration changes apply immediately.

When a log is created, the detail level for that log is set from the configuration data. If no configuration data is available for a particular log name, the level for that log is obtained from the parent of the log. If no configuration data exists for the parent log, the parent of that log is checked, and so on, up the tree, until a log with a non-null level value is found. When you change the level of a log, the change is propagated to the child logs, which recursively propagate the change to their child log, as necessary.

To enable logging and set the output properties for a log, use the following procedure.

Procedure

1. In the navigation pane of the administrative console, select **Servers > WebSphere application servers**.
2. Click the name of the server that you want to work with.
3. Under **Troubleshooting**, click **Logging and tracing**.
4. Click **Change log detail levels**.
5. Specify when you want the change to take effect:
 - For a static change to the configuration, click the **Configuration** tab.
 - For a dynamic change to the configuration, click the **Runtime** tab.
6. Click the names of the packages whose logging level you want to modify. The package names for WebSphere Adapters start with **com.ibm.j2ca.***:
 - For the adapter base component, select **com.ibm.j2ca.base.***.
 - For the adapter base component and all deployed adapters, select **com.ibm.j2ca.***.
 - For the WebSphere Adapter for Siebel Business Applications only, select the **com.ibm.j2ca.siebel.*** package.
7. Select the logging level.

Logging Level	Description
Fatal	The task cannot continue or the component cannot function.
Severe	The task cannot continue, but the component can still function. This logging level also includes conditions that indicate an impending fatal error, that is, situations that strongly suggest that resources are on the verge of being depleted.
Warning	A potential error has occurred or a severe error is impending. This logging level also includes conditions that indicate a progressive failure, for example, the potential leaking of resources.
Audit	A significant event has occurred that affects the server state or resources.
Info	The task is running. This logging level includes general information outlining the overall progress of a task.
Config	The status of a configuration is reported or a configuration change has occurred.
Detail	The subtask is running. This logging level includes general information detailing the progress of a subtask.

8. Click **Apply**.
9. Click **OK**.
10. Optional: To have static configuration changes take effect, stop and then restart the application server.

Results

Log entries from this point forward contain the specified level of information for the selected adapter components.

Changing the log and trace file names

To keep the adapter log and trace information separate from other processes, use the administrative console to change the file names. By default, log and trace information for all processes and applications on a application server is written to the `SystemOut.log` and `trace.log` files.

Before you begin

You can change the log and trace file names at any time after the adapter module has been deployed to an application server.

About this task

You can change the log and trace file names statically or dynamically. Static changes take effect when you start or restart the application server. Dynamic or run time changes apply immediately.

Log and trace files are in the `install_root/profiles/profile_name/logs/server_name` folder.

To set or change the log and trace file names, use the following procedure.

Procedure

1. In the navigation pane of the administrative console, select **Applications > Application Types > WebSphere application servers**.
2. In the Enterprise Applications list, click the name of the adapter application. This is the name of the EAR file for the adapter, but without the ear file extension. For example, if the EAR file is named `Accounting_OutboundApp.ear`, then click **Accounting_OutboundApp**.
3. In the Configuration tab, select **Modules>Manage Modules**.
4. In the list of modules, click **IBM WebSphere Adapter for Siebel Business Applications**.
5. In the Configuration tab, under Additional Properties, click **Resource Adapter**.
6. In the Configuration tab, under Additional Properties, click **Custom properties**.
7. In the Custom Properties table, change the file names.
 - a. Click either **logFilename** to change the name of the log file or **traceFilename** to change the name of the trace file.
 - b. In the Configuration tab, type the new name in the **Value** field. By default, the log file is called `SystemOut.log` and the trace file is called `trace.log`.
 - c. Click **Apply** or **OK**. Your changes are saved on your local machine.
 - d. To save your changes to the master configuration on the server, use one of the following procedures:
 - **Static change:** Stop and restart the server. This method allows you to make changes, but those changes do not take effect until you stop and start the server.
 - **Dynamic change:** Click the **Save** link in the Messages box above the Custom properties table. Click **Save** again when prompted.

Changing configuration properties for embedded adapters

To change the configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment. You can update resource adapter properties (used for general adapter operation), managed connection factory properties (used for outbound processing), and activation specification properties (used for inbound processing). For information about configuring logging properties and changing the log and trace file names, see “Configuring logging and tracing” on page 59.

Setting managed (J2C) connection factory properties for embedded adapters

With Rational Application Developer for WebSphere Software, you can work in the non-managed deployment mode for outbound operations. With this option, you can select and configure properties for your embedded adapter.

Before you begin

You must first configure managed connection factory properties for your adapters and deploy the adapter module on WebSphere Application Server.

About this task

You use managed connection factory properties to configure the target Siebel application instance.

Note: In the administrative console, the properties are referred to as "J2C connection factory properties."

To configure properties using the administrative console, use the following procedure.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. In the Enterprise Applications list, click the name of the adapter module whose properties you want to change.
6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for Siebel Business Applications**.
8. In the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
10. Click the name of the connection factory associated with your adapter module.
11. In the **Additional Properties** list, click **Custom properties**.

Custom properties are those J2C connection factory properties that are unique to IBM WebSphere Adapter for Siebel Business Applications. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.

12. For each property you want to change, perform the following steps.

Note: See “Managed connection factory properties” on page 100 for more information about these properties.

- a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
13. In the Messages area, click **Save**.

Results

You can view the managed connection factory properties associated with your adapter module.

Changing configuration properties for stand-alone adapters

To set configuration properties after you install a stand-alone adapter, use the administrative console of the runtime environment. Provide the general information about the adapter and then set the resource adapter properties (which are used for general adapter operation). If the adapter is used for outbound operations, create a connection factory and then set the properties for it. If the adapter is used for inbound operations, create an activation specification and then set the properties for it. For information about configuring logging properties and changing the log and trace file names, see “Configuring logging and tracing” on page 59.

Setting resource adapter properties for stand-alone adapters

To set resource adapter properties for your stand-alone adapter after it has been installed on WebSphere Application Server, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Application Server.

About this task

Custom properties are default configuration properties shared by all IBM WebSphere Adapters.

To configure properties using the administrative console, use the following procedure:

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.

3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Siebel Business Applications**.
6. In the **Additional Properties** list, click **Custom properties**.
7. For each property you want to change, perform the following steps.
 - a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
8. In the Messages area, click **Save**.

Results

The resource adapter properties associated with your adapter are changed.

Setting managed (J2C) connection factory properties for stand-alone adapters

To set managed connection factory properties for your stand-alone adapter after it has been installed on WebSphere Application Server, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Application Server.

About this task

You use managed connection factory properties to configure the target Siebel application instance.

Note: In the administrative console, the properties are referred to as "J2C connection factory properties."

To configure properties using the administrative console, use the following procedure:

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Siebel Business Applications**.
6. In the **Additional Properties** list, click **J2C connection factories**.
7. If you are going to use an existing connection factory, skip ahead to select from the list of existing connection factories.

Note: If you have selected **Specify connection properties** when you use the J2C Bean wizard to configure the adapter module, you do not need to create a connection factory.

If you are creating a connection factory, perform the following steps:

- a. Click **New**.
- b. In the **General Properties** section of the **Configuration** tab, type a name for the connection factory. For example, you can type AdapterCF.
- c. Type a value for **JNDI name**. For example, you can type com/eis/AdapterCF.
- d. Optional: Select an authentication alias from the **Component-managed authentication alias** list.
- e. Click **OK**.
- f. In the Messages area, click **Save**.

The newly created connection factory is displayed.

8. In the list of connection factories, click the one you want to use.

9. In the **Additional Properties** list, click **Custom properties**.

Custom properties are those J2C connection factory properties that are unique to WebSphere Adapter for Siebel Business Applications. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.

10. For each property you want to change, perform the following steps.

Note: See “Managed connection factory properties” on page 100 for more information about these properties.

- a. Click the name of the property.
- b. Change the contents of the **Value** field or type a value, if the field is empty.
- c. Click **OK**.

11. After you have finished setting properties, click **Apply**.

12. In the Messages area, click **Save**.

Results

The managed connection factory properties associated with your adapter are set.

Setting activation specification properties for stand-alone adapters

To set activation specification properties for your stand-alone adapter after it has been installed on WebSphere Application Server, use the administrative console. You select the name of the message endpoint property you want to configure, and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Application Server.

About this task

You use activation specification properties to configure the endpoint for inbound processing.

To configure properties using the administrative console, use the following procedure.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for Siebel Business Applications**.
6. In the **Additional Properties** list, click **J2C activation specifications**.
7. If you are going to use an existing activation specification, skip ahead to select from an existing list of activation specifications.

Note: If you have selected **Use predefined connection properties** when you use the J2C Bean wizard to configure the adapter module, you must create an activation specification.

If you are creating an activation specification, perform the following steps:

- a. Click **New**.
- b. In the **General Properties** section of the **Configuration** tab, type a name for the activation specification. For example, you can type AdapterAS.
- c. Type a value for **JNDI name**. For example, you can type com/eis/AdapterAS.
- d. Optional: Select an authentication alias from the **Authentication alias** list.
- e. Select a message listener type.
- f. Click **OK**.
- g. Click **Save** in the **Messages** box at the top of the page.
The newly created activation specification is displayed.
8. In the list of activation specifications, click the one you want to use.
9. In the Additional Properties list, click **J2C activation specification custom properties**.
10. For each property you want to set, perform the following steps.
 - a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
11. After you have finished setting properties, click **Apply**.
12. In the Messages area, click **Save**.

Results

The activation specification properties associated with your adapter are set.

Adding dependency libraries to the deployed resource adapter

The deployed resource adapter running in the WebSphere Application Server requires the same dependency libraries as it does in Rational Application Developer for WebSphere Software to process requests. The method for adding these library files depends on the mode of the resource adapter deployment: standalone or embedded in the EAR file.

Stand-alone deployment

The dependency libraries may be added to the resource adapter deployed stand-alone either during initial deployment of the RAR file or by configuring the Resource Adapter properties after deployment. To set the values during initial deployment of the RAR file, specify Class path and Native path locations. Class path is used to point to JAR files, and Native path is used to point to native libraries, such as *.dll, *.so. To set the dependency library path files after the adapter has been installed on WebSphere Application Server, use the administrative console to modify the values for the Resource Adapter.

EAR deployment

For the rare case when the connector needs to be embedded in the EAR file, the dependant libraries are added as shared libraries. Define the appropriate shared library containing software dependency files and associate them with the EAR file.

About this task

There are two methods to do this task:

- Using enhanced EAR editor in Rational Application Developer for WebSphere Software
- Using administrative console of the WebSphere Application Server

Using enhanced EAR editor

You can use the EAR editor in Rational Application Developer for WebSphere Software to add the dependency libraries.

About this task

To create shared libraries using the EAR editor, use the following procedure.

Procedure

1. Open Enhanced EAR editor.
2. Click the **Deployment** tab.
3. Expand **Shared Library** section.
4. Click **Add** to add new shared library.
5. Specify the shared library parameters and click **OK**.
6. Deploy the EAR to the server.

Results

The dependent libraries are added as shared libraries.

Using administrative console of the WebSphere Application Server

You can use the administrative console of the WebSphere Application Server to add the dependency libraries.

Before you begin

Ensure that the dependent files are available on the server machine in the separate folder. If needed, copy the dependent files on the server machine.

Procedure

1. Define WebSphere variables to point to appropriate folders.
2. Define the shared library through the server administrative console; you can specify it using WebSphere variables defined in step 1.
3. Deploy the EAR to the server.
4. Configure the EAR to reference defined shared library.

Results

The dependent libraries are added as shared libraries.

Chapter 6. Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly. For information about configuring logging properties and changing the log and trace file names, see “Configuring logging and tracing” on page 59.

Techniques for troubleshooting problems

Troubleshooting is a systematic approach to solving a problem. The goal is to determine why something does not work as expected and how to resolve the problem. Certain common techniques can help with the task of troubleshooting.

The first step in the troubleshooting process is to describe the problem completely. Without a problem description, neither you or IBM® can know where to start to find the cause of the problem. This step includes asking yourself basic questions, such as:

- What are the symptoms of the problem?
- Where does the problem occur?
- When does the problem occur?
- Under which conditions does the problem occur?
- Can the problem be reproduced?

The answers to these questions typically lead to a good description of the problem, and that is the best way to start down the path of problem resolution.

What are the symptoms of the problem?

When starting to describe a problem, the most obvious question is "What is the problem?" Which might seem like a straightforward question; however, you can break it down into several more-focused questions that create a more descriptive picture of the problem. These questions can include:

- Who, or what, is reporting the problem?
- What are the error codes and messages?
- How does the system fail? For example, is it a loop, hang, lock up, performance degradation, or incorrect result?
- What is the business impact of the problem?

Where does the problem occur?

Determining where the problem originates is not always simple, but it is one of the most important steps in resolving a problem. Many layers of technology can exist between the reporting and failing components. Networks, disks, and drivers are only a few components to be considered when you are investigating problems.

The following questions can help you to focus on where the problem occurs in order to isolate the problem layer.

- Is the problem specific to one platform or operating system, or is it common for multiple platforms or operating systems?
- Is the current environment and configuration supported?

Remember that if one layer reports the problem, the problem does not necessarily originate in that layer. Part of identifying where a problem originates is

understanding the environment in which it exists. Take some time to completely describe the problem environment, including the operating system and version, all corresponding software and versions, and hardware information. Confirm that you are running within an environment that is a supported configuration; many problems can be traced back to incompatible levels of software that are not intended to run together or have not been fully tested together.

When does the problem occur?

Develop a detailed timeline of events leading up to a failure, especially for those cases that are one-time occurrences. You can most simply do this by working backward: Start at the time an error was reported (as precisely as possible, even down to the millisecond), and work backward through the available logs and information. Typically, you need to look only as far as the first suspicious event that you find in a diagnostic log; however, this is not always simple to do and takes practice. Knowing when to stop looking is especially difficult when multiple layers of technology are involved, and when each has its own diagnostic information.

To develop a detailed timeline of events, answer the following questions:

- Does the problem happen only at a certain time of day or night?
- How often does the problem happen?
- What sequence of events leads up to the time that the problem is reported?
- Does the problem happen after an environment change, such as upgrading or installing software or hardware?

Responding to these types of questions can provide you with a frame of reference in which to investigate the problem.

Under which conditions does the problem occur?

Knowing what other systems and applications are running at the time that a problem occurs is an important part of troubleshooting. These and other questions about your environment can help you to identify the root cause of the problem:

- Does the problem always occur when the same task is being performed?
- Does a certain sequence of events need to occur for the problem to surface?
- Do any other applications fail at the same time?

Answering these types of questions can help you explain the environment in which the problem occurs and correlate any dependencies. Remember that just because multiple problems might have occurred around the same time, the problems are not necessarily related.

Can the problem be reproduced?

From a troubleshooting standpoint, the "ideal" problem is one that can be reproduced. Typically with problems that can be reproduced, you have a larger set of tools or procedures at your disposal to help you investigate. Consequently, problems that you can reproduce are often simpler to debug and solve. However, problems that you can reproduce can have a disadvantage: If the problem is of significant business impact, you do not want it to recur! If possible, re-create the problem in a test or development environment, which typically offers you more flexibility and control during your investigation.

Tip: Simplify the scenario to isolate the problem to a suspected component.

The following questions can help you with reproducing the problem:

- Can the problem be re-created on a test machine?
- Are multiple users or applications encountering the same type of problem?
- Can the problem be re-created by running a single command, a set of commands, a particular application, or a stand-alone application?

First-failure data capture (FFDC) support

The adapter supports first-failure data capture (FFDC), which provides persistent records of failures and significant software incidents that occur during run time in WebSphere Application Server.

The FFDC feature runs in the background and collects events and errors that occur at run time. The feature provides a means for associating failures to one another, allowing software to link the effects of a failure to their causes, and facilitate the quick location of the root cause of a failure. The data that is captured can be used to identify exception processing that occurred during the adapter run time.

When a problem occurs, the adapter writes exception messages and context data to a log file, which is in the *install_root/profiles/profile/logs/ffdc* directory.

For more information about first-failure data capture (FFDC), see the WebSphere Application Server documentation.

Solutions to some common problems

Solutions and workarounds to some problems you may encounter while running WebSphere Adapter for Siebel Business Applications with your database are provided. These problems and solutions are also documented as technotes on the Software support website.

For a complete list of technotes about WebSphere Adapters, see <http://www-1.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>.

The adapter may time out based on the default timeout value set on the Siebel application

Problem

The default timeout value for connections set on the Siebel application is approximately 45 minutes. Any connection that is idle (without any transactions coming in) for more than 45 minutes will be timed out. There is no way for the adapter to know this, because the adapter only gets the connection from the container. After it fails, the adapter generates a `CommException` exception back, and the corresponding connection is removed from the pool.

Cause

If the adapter has been idle for more than 45 minutes, it will time out. Also, if the container sends the adapter an invalid connection, the adapter will fail.

Solution and Workaround

If there is a possibility that the adapter remains idle, without any transactions coming in for 60 minutes, increase the timeout value on the Siebel business

application to a higher number (for example, 1 1/2 hours). This ensures that the connections remain valid for a longer period. If you cannot increase the timeout value, then resend the failed transaction.

Adapter does not check for the required fields to be set when creating Siebel business components

Problem

When creating Siebel business components, the adapter does not check to ensure that the required fields are set. If required fields are not set, the Siebel enterprise information system (EIS) generates an error, which is returned by the adapter.

Cause

Required fields are not set.

Solution and Workaround

Update the required attributes as needed based on the Siebel exception message provided by the adapter.

Adapter does not throw exception when the call to a Siebel business service method fails

Problem

When the adapter runs a Siebel business service method, it might not throw an exception, even if the transaction fails in the Siebel application.

Cause

This problem occurs because the corresponding business service does not throw an exception but rather returns an empty property set to the adapter. The adapter cannot detect if the transaction was successful, that is, whether the values were updated properly in the Siebel business application. Therefore, the adapter does not throw an exception. Because an empty property set is returned, the adapter returns the corresponding empty top-level business object.

Example scenario: Invalid attribute values are passed through the business object. For example, if the key (for example, "ID") value does not exist and there is no value set on the business object, a corresponding record does not exist in the Siebel business application for that particular key. The business service method might simply ignore the attribute value, even if it is set on the property set. The method must throw an exception but instead returns an empty property set, so the adapter also returns the empty top-level business object (that corresponds to the property set).

Solution and Workaround

Customize the Siebel business service to throw an exception back when failure occurs.

Adapter generates multiple container attributes in the business object for simple link relationships

Problem

Multiple container attributes are generated by IBM WebSphere Adapter for Siebel Business Applications in the parent business object for a simple link relationship between two Siebel business components.

Cause

The adapter generates multiple container attributes for a simple link relationship. For example, as part of the enterprise services discovery generation, assume you select the Siebel business components Account and Business Address in the Siebel business object Account. The corresponding WebSphere business object generated for the Siebel business component Account includes three container attributes, BusinessAddress, BusinessAddress2, and BusinessAddress3, instead of one. You can confirm that SimpleLink application-specific information is set against the three container attributes.

Solution and Workaround

Eventhough the extra container attributes does not affect the transaction if they are left untouched, remove the unwanted attributes using the business object editor in Rational Application Developer for WebSphere Software. Therefore, you have only one of the three attributes (BusinessAddress, BusinessAddress2, and BusinessAddress3). For more information on how to use the business object editor, see Rational Application Developer Information Center.

The adapter does not generate artifacts properly for all selected Siebel business components

Problem

When the adapter generates names for business objects from Siebel business components selected through the J2C Bean wizard, it removes special characters, including spaces, from the names. If multiple Siebel business components have the same name after the adapter deletes spaces from the names, the adapter does not generate artifacts properly.

Only one WebSphere business object is generated instead of multiple business objects, even though multiple Siebel business components were selected through the J2C Bean wizard.

As an example, consider a Siebel business object called Acc that has two Siebel business components: Bus Addr and BusAddr. Because the adapter deletes spaces from the Siebel business component names to generate the WebSphere business object names, two WebSphere business objects will have the same name: BOAccBCBusAddr. The first WebSphere business object is overwritten by the second one.

Cause

When generating business object names, the adapter deletes spaces instead of replacing them with Unicode values, as it does for other special characters.

Solution and Workaround

Use the **Generate business object with shorter names** option to generate multiple business objects. The naming convention for this option adds a counter to the end of the business object if resulting names will be the same, for example, BusAddr, BusAddr1, and BusAddr2.

The adapter does not generate an exception when the call to a Siebel business service method fails

Problem

When the adapter processes a Siebel business service method and the transaction fails in the Siebel application, the adapter might not generate an exception.

Cause

The corresponding business service does not generate an exception to the adapter when the transaction fails, but instead returns an empty property set to the adapter. Because the adapter cannot detect whether the transaction was successful, that is, whether the values were updated properly in the Siebel application, it does not generate an exception. Because it receives an empty property set, the adapter returns the corresponding empty top-level business object.

For example, if the key value, such as ID, does not exist, and there is no value set on the business object, a corresponding record does not exist in the Siebel application for that particular key. The business service method might ignore the attribute value, even though it is set on the property set. The method must generate an exception but instead returns an empty property set, so the adapter also returns the empty top-level business object that corresponds to the property set.

Solution and Workaround

Customize the Siebel business service to generate an exception when a failure occurs.

Inbound component of WebSphere Adapter for Siebel Business Applications stops working after the number of event entries passes 10,000

Problem

Inbound component of WebSphere Adapter for Siebel Business Applications stops polling and throws some exception.

Cause

WebSphere Adapter for Siebel Business Applications uses Siebel API to access the event table. The possible cause for this issue may be:

- You performed a query or search which is trying to return more rows than is allowed in a result set.
- MaxFetchArraySize or MaxCursorSize parameter values may be set too low.
- Maximum Cursor Size property may be set too low for a specific business component.

Solution and Workaround

Following actions can be considered:

- Modify your query or search specification to return fewer rows in the result set. If not user initiated, check any running workflows, scripts, EAI, charts, or reports that may be executing the query.
- Increase the rows allowed in your result set by modifying the value of MaxFetchArraySize or MaxCursorSize. This will affect the entire Siebel business application. For MaxFetchArraySize, values are:
 - -9,999 (number of rows)
 - 0 (10,000 rows)
 - -1 (unlimited rows)

For instructions on the MaxFetchArraySize parameter, see Siebel Bookshelf version 7.5.3 > Siebel Object Interfaces Reference > Interfaces Reference > Business component Methods > Execute Query. For instructions on the MaxCursorSize parameter, see Siebel Bookshelf version 7.5.3 > Siebel Web Client Administration > Configuration Parameters > Data Source Parameters.

Note: Setting DSMaxFetchArraySize to -1 may cause large memory use, growth of the Siebel Object Manager process, and the Siebel Object Manager to crash from memory exhaustion. For more information, see Alert 951.

- Increase the number of records requested for a specific business component by modifying the values of Maximum Cursor Size property for the specific business component. This will only affect operations performed on the specific business component. For instructions on how to set the Maximum Cursor Size property in Siebel Tools, see Siebel Bookshelf version 7.5.3 > Object Types Reference > Siebel Object Types.

Adapter does not generate WebSphere business objects for child Siebel business components if the related link is inactive

Problem

The J2C Bean wizard generates WebSphere business objects against the corresponding Siebel business components. If the link that relates the parent business component to the child business component is inactive, the J2C Bean wizard does not generate a WebSphere business object for the corresponding child business component.

Cause

The link in Siebel enterprise information system (EIS) for the parent business component to the child business component is inactive.

Solution and workaround

Set the link as Active in Siebel enterprise information system (EIS).

Adapter returns version conflict exception message

Problem

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the

CWYBS_AdapterFoundation.jar is loaded during run time, the adapter returns the ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for Siebel Business Applications version 7.5.0.3, the following error message is displayed "The version of CWYBS_AdapterFoundation.jar is not compatible with IBM WebSphere Adapter for Siebel Business Applications" as IBM WebSphere Adapter for Siebel Business Applications loads file:/C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3. However, the base level of this jar required is version 7.5.0.3. To overcome this conflict, you must ensure that all adapters are at same version level. For further assistance, contact WebSphere Adapters Support for help.

Solution

Ensure that all adapters are at the same version level.

For further assistance, visit <http://www.ibm.com/support/docview.wss?uid=swg27006249>.

Endpoint application of the passive adapter instance listens to the events when enableHASupport is set to True

Problem

In the active-passive configuration mode of the adapters, the endpoint application of the passive adapter instance also listens to the events or messages even if the enableHASupport property is set to True.

Cause

By default, in WebSphere Application Server, the alwaysactivateAllMDBs property in the JMS activation specification is set to True. This enables the endpoint application of all the adapter (active or passive) instances to listen to the events.

Solution

To stop the endpoint application of the passive adapter instance from listening to the events, you must set the alwaysactivateAllMDBs property value to False. The JMS activation specification is associated with one or more MDBs and provides the necessary configuration to receive events. If the alwaysActivateAllMDBs property is set to False, then the endpoint application of only the active adapter instance receives the events.

Perform the following procedure, to set the alwaysActivateAllMDBs property to False.

1. Log on to the WebSphere Application Server Administrative Console.
2. Go to **Resources > JMS > Activation specifications**.
3. Click the activation specification corresponding to the application from the list.
4. Click **Custom properties** under **Additional properties**.
5. Click **alwaysActivateAllMDBs**.
6. Change the value to **False**.
7. Click **Apply** and **OK**.

Result

The endpoint application of only the active adapter instance listens to the events.

Error displayed when using J2C Bean wizard

Problem

Error is displayed when using the J2C Bean wizard.

1. Add a Siebel driver file in the Connector Settings window.
2. Click **Next** and move to the Discovery Configuration window.
3. Click **Back** and go to the Connector Settings. Then add another Siebel driver file.
4. Click **Next** and move to the Discovery Configuration window.

The Discovery Configuration is not displayed.

Cause

This is an Eclipse limitation for class loading.

Solution

Close the connector project and reopen it to add the Siebel driver files again.

Issues faced while using Oracle Siebel dependency jars 8.1.1.4 (and greater)

Problem

An end of file error occurred (SBL-DAT-00393). An article on Oracle (article ID 1328941.1) discusses about this issue. If Java Data Bean is using the Oracle Siebel dependency JAR files of version 8.1.1.4 (or 8.1.1.5), the JAR files generate an Unexpected Error at the End Of File when executing the NextRecord command. A bug (# 12641734) is also identified for the same.

Solution

The issue is resolved in the Oracle Siebel dependency JAR files of version 8.1.1.5 by applying the QF0562 defect fix from Oracle. It is also resolved in the Fix Patch version 8.1.1.6 (and greater).

Support

This section provides information about how to troubleshoot a problem with your IBM® software, including instructions for searching knowledge bases, downloading fixes, and obtaining support.

Searching knowledge bases (Web search)

You can often find solutions to problems by searching IBM knowledge bases. You can optimize your results by using available resources, support tools, and search methods.

About this task

You can find useful information by searching the information center for Product X. However, sometimes you need to look beyond the information center to answer your questions or resolve problems.

To search knowledge bases for information that you need, use one or more of the following approaches:

- Search for content by using the IBM® Support Assistant (ISA).
ISA is a no-charge software serviceability workbench that helps you answer questions and resolve problems with IBM software products. You can find instructions for downloading and installing ISA on the ISA website.
- Find the content that you need by using the IBM Support Portal.
The IBM Support Portal is a unified, centralized view of all technical support tools and information for all IBM systems, software, and services. The IBM Support Portal lets you access the IBM electronic support portfolio from one place. You can tailor the pages to focus on the information and resources that you need for problem prevention and faster problem resolution. Familiarize yourself with the IBM Support Portal by viewing the demo videos (https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos) about this tool. These videos introduce you to the IBM Support Portal, explore troubleshooting and other resources, and demonstrate how you can tailor the page by moving, adding, and deleting portlets.
- Search for content by using the IBM masthead search. You can use the IBM masthead search by typing your search string into the Search field at the top of any [ibm.com](https://www.ibm.com)® page.
- Search for content by using any external search engine, such as Google, Yahoo, or Bing. If you use an external search engine, your results are more likely to include information that is outside the [ibm.com](https://www.ibm.com) domain. However, sometimes you can find useful problem-solving information about IBM products in newsgroups, forums, and blogs that are not on [ibm.com](https://www.ibm.com).

Tip: Include "IBM" and the name of the product in your search if you are looking for information about an IBM product.

Getting Fixes

A product fix might be available to resolve your problem.

About this task

To get product fixes, perform the following steps.

Procedure

1. Determine which fix you need. Check the list of IBM WebSphere Adapter for Siebel Business Applications recommended fixes to confirm that your software is at the latest maintenance level. Check the list of problems fixed in the IBM WebSphere Adapter for Siebel Business Applications fix readme documentation that is available for each listed fix pack to see if IBM has already published an individual fix to resolve your problem. To determine what fixes are available using IBM Support Assistant, run a query on fix from the search page.

Individual fixes are published as often as necessary to resolve defects in WebSphere Application Server IBM WebSphere Adapter for Siebel Business Applications. In addition, two kinds of cumulative collections of fixes, called fix

packs and refresh packs, are published periodically for IBM WebSphere Adapter for Siebel Business Applications, in order to bring users up to the latest maintenance level. You should install these update packages as early as possible in order to prevent problems.

Note: A list of recommended, generally available (GA) fixes for the WebSphere Java™ Connector Architecture (JCA) and WebSphere Business Integration adapters are available here. If a Fix Pack is not available for an adapter, it implies that the GA version is the recommended version and details about that version of the adapter can be found in the Release notes.

2. Download the fix. Open the download document and follow the link in the Download package section. When downloading the file, ensure the name of the maintenance file is not changed. This includes both intentional changes and inadvertent changes caused by certain web browsers or download utilities.
3. Apply the fix. Follow the instructions in the Installation Instructions section of the download document.
4. Optional: To receive weekly notification of fixes and updates, subscribe to My Support e-mail updates.

Self-help resources

Use the resources of IBM software support to get the most current support information, obtain technical documentation, download support tools and fixes, and avoid problems with WebSphere Adapters. The self-help resources also help you diagnose problems with the adapter and provide information about how to contact IBM software support.

Support website

The WebSphere Adapters software support website at <http://www.ibm.com/support/docview.wss?uid=swg27006249> provides links to many resources to help you learn about, use, and troubleshoot WebSphere Adapters, including:

- Flashes (alerts about the product)
- Technical information including the product information center, manuals, IBM Redbooks®, and whitepapers
- Educational offerings
- Technotes

Recommended fixes

A list of recommended fixes you must apply is available at the following location: <http://www.ibm.com/support/docview.wss?fdoc=aimadp&rs=695&uid=swg27010397>.

Technotes

Technotes provide the most current documentation about WebSphere Adapter for Siebel Business Applications, including the following topics:

- Problems and their currently available solutions
- Answers to frequently asked questions
- How to information about installing, configuring, using, and troubleshooting the adapter
- *IBM Software Support Handbook*

For a list of technotes for WebSphere Adapter for Siebel Business Applications, see <http://www-01.ibm.com/support/docview.wss?uid=swg27024042>.

For a list of technotes for all adapters, see <http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>.

Chapter 7. Reference information

To support you in your tasks, reference information includes details about business objects that are generated by the J2C Bean wizard and information about adapter properties, including those that support bidirectional transformation. It also includes pointers to adapter messages and related product information.

Business object information

A business object is a structure that contains application-specific information (metadata) about how the adapter must process the business object as well as the operation to be performed on the business object. The name of the business object is generated by the J2C Bean wizard in accordance with the naming convention for the adapter.

Application-specific information

Application-specific information is metadata that specifies how to process business objects for the adapter.

When the J2C Bean wizard generates a business object, it automatically generates a business object definition, which is saved as an XSD (XML Schema Definition) file. The business object definition contains the application-specific information for that business object. If you want to change the generated Application-specific information, you can modify the metadata values either from the **Properties** tab in the Java EE perspective of Rational Application Developer for WebSphere Software or by using the business object editor.

The business object metadata is generated by the J2C Bean wizard at three levels: the business object level (for Siebel business services and Siebel business components), the property level, and the container attribute level.

Metadata at the business object level

Metadata at the business object level defines the top-level wrapper of the business object. The following tables describe the elements within business object metadata for Siebel business objects generated against Siebel business services and Siebel business objects generated against Siebel business components.

Table 8. Metadata elements for business objects generated against Siebel business services

Metadata element	Description
BSN	The name of the business service that is used by the WebSphere Adapter business object.
EventMethod	The event method to use when retrieving event data.
IC	The name of the Siebel integration component that corresponds to the WebSphere Adapter business object.
IO	The name of the Siebel integration object that corresponds to the WebSphere Adapter business object.

Table 9. Metadata elements for business objects generated against Siebel business components

Metadata element	Description
ComponentName	The name of the Siebel business component that corresponds to the WebSphere Adapter business object.
ObjectName	The name of the Siebel business object that corresponds to the WebSphere Adapter business object.

Property-level metadata

Property-level metadata represents either a child object or an array of child objects. The following tables describe the metadata elements of a complex property (a child) or a structure or table property (an array of child objects).

Table 10. Property-level metadata elements generated against Siebel business services

Metadata element	Description
FN=...	Specifies the name of the field in the Siebel business component that corresponds to a WebSphere Adapter business object property.
ParamType	Indicates whether the corresponding field on the Siebel end is input, output, or both.

Table 11. Property-level metadata elements generated against Siebel business components

Metadata element	Description
FieldName	Specifies the name of the field in the Siebel business component that corresponds to a WebSphere Adapter business object property.
PickListKey	Indicates that when a pick list key is specified, a simple attribute will be a PickList.
Restrict=<ParentFieldName>:<childFieldName>	Specifies additional search criteria that can be set to limit the number and type of records that are retrieved.

Container attribute-level metadata

Container attribute-level metadata represents the data located at the container level of business object and components. The following table describes the metadata elements located at the container level for Siebel business objects and Siebel business components.

Table 12. Container attribute-level metadata elements generated against Siebel business objects and Siebel business components

Metadata element	Description
Association	Specifies a 'many to many' relationship at the intersection table when set to true.
FieldName	Identifies the field in the Siebel business component that corresponds to this property.
From=...;To=...	Sets the preprocessing instructions for the adapter to process the To attribute value of the From attribute. The From attribute must be populated, and the To attribute value is valid only if it is null. This is used in Retrieve operations and specifies the child record to be fetched.

Table 12. Container attribute-level metadata elements generated against Siebel business objects and Siebel business components (continued)

Metadata element	Description
KeepRelations	Specifies whether the child objects need to be deleted from the Siebel application. This is only used for Siebel business objects and Siebel business components. It is not used for Siebel business services and Siebel integration objects.
MultiValueLink	Specifies that there is an active multi-value field on the parent object, which indicates an active multi-value link relationship between the parent and child object.
PickList	Indicates a many to one relationship when set to true.
SiebelMessage	Specifies the business service attribute to be used for input and output messages. This is used for Siebel business objects when an attribute represents a Siebel message complex argument. It is valid for both input and output messages, or when one argument is used for the input message and a different one is used for the output message.
SiebelMessageDataType	Specifies whether the attribute is used as the input message, output message, or for both input and output messages. The valid options are Input, Output, and InOut.
SourceField=...;DestinationField=...	Used with simple links to build a simple link relationship between the parent and child WebSphere Adapter business object. The SF, or source field, corresponds to the key attribute (ID) in the WebSphere Adapter parent business object. The DF, or destination field, is the foreign key field in the Siebel child business component. This corresponds to the child attribute with the field name value set to the foreign key field in the Siebel business component.
IOWrapper	Specifies the business service attribute to be used for a wrapper, which contains multiple integration objects. If the business object contains this attribute value, the adapter supports the business services to return different integration objects based on the different input parameters.

Supported operations

An operation is the action that an adapter can perform on the Siebel Business Applications server during outbound processing. The name of the operation typically indicates the type of action that the adapter takes, such as *create* or *update*.

The following table defines the operations that the adapter supports.

Table 13. Supported operations

Operation	Description
Create	Creates the business component
Delete	Deletes the business component and its child components
Exists	Checks for the existence of incoming business objects
Retrieve	Retrieves the values of the business component
Retrieve all	Retrieves multiple instances of the same business component
Update	Updates the Siebel application with the incoming object

Naming conventions

When the J2C Bean wizard generates a business object, it provides a name for the business object based on the name of the object in the Siebel application that it uses to build the business object.

Naming conventions for business objects that represent Siebel business services and integration components

The naming conventions for business objects that represent Siebel business services are the same for both inbound and outbound processing. The names comprise the concatenation of several words, including prefix, business service name, integration object, and method name.

The following table describes the naming conventions that the J2C Bean wizard uses to name business objects that represent Siebel business services and integration components.

Table 14. Business object naming conventions for Siebel business services and integration components

Element	Naming convention
	<Top Level business object Name> Note: A "Prefix" is used only for top-level business objects generated against business service methods.
Name of the top-level business object	<Prefix><Business Service Name><Method Name><Names of all the integration objects selected for the Input and InputOutput complex type arguments> <ul style="list-style-type: none"> If there are no Input or Input/Output arguments, the names of all the output arguments are: <Prefix><Business Service Name><Method Name><Names of all the integration objects selected for the output complex type arguments> If the method contains no complex arguments in the method, the naming convention is: <Prefix><Business Service Name><Method Name>
Name of the inbound object generated against integration components	'IO' + <Name of Integration Object> + 'IC' + <Name of integration component>
Name of the outbound object generated against integration components	'IO' + <Name of Integration Object> + 'IC' + <Name of integration component> The name of an account interface integration object with the integration component account looks like this: IOAccountInterfaceICAccount

Optional: Shorter naming conventions for business objects generated against Siebel business services and integration components

The naming conventions for business objects generated against Siebel business services and integration components are valid if the optional property, 'Generate business objects with shorter names' is specified in the configuration objects window in the J2C Bean wizard.

Note: If this optional property is used, you must set the 'Folder' property with a unique value. This avoids overwriting the existing XSDs that were previously generated. For example, if you select 'EAI Siebel Adapter', and click **Query** in two

different J2C Bean wizard runs for the integration objects, 'Account (PRM ANI)' and 'ATP Check Interface', the top-level object will be named as 'EASiebelAdapter.xsd'.

The name comprises the concatenation of several words, including prefix, business service name, and integration component name.

The following table describes the naming conventions that the J2C Bean wizard uses to name business objects that are generated against Siebel business services and integration components.

Table 15. Shorter business object naming conventions for business objects generated against Siebel business services and integration components

Element	Naming convention
Name of the inbound and outbound child business objects generated against integration components	<Prefix>+<Name of the Siebel Integration Component> Note: The Siebel business object and integration component names are stripped of all non-alphanumeric characters before being added to the business object name. If the resulting names are not unique, a counter is added to the end of the names.
Name of the inbound top-level business object generated against business services and integration components	<Prefix>+<Name of the Siebel Integration Component> and <Prefix>+<Name of the Siebel Integration Component> Note: The Siebel business object and integration component names are stripped of all non-alphanumeric characters before being added to the business object name. If the resulting names are not unique, a counter is added to the end of the names.
Name of the outbound top-level business object generated against business services and integration components	<Prefix>+<Name of the Siebel Business Service> and <Prefix>+<Name of the Siebel Business Service> Note: The Siebel business object and integration component names are stripped of all non-alphanumeric characters before being added to the business object name. If the resulting names are not unique, a counter is added to the end of the names.

Naming conventions for business objects that represent Siebel business objects

The naming conventions for business objects representing Siebel business objects are the same for both inbound and outbound processing. The name comprises the concatenation of several words, including prefix, business object name, and business component name.

The following table describes the naming conventions used by the J2C Bean wizard to name business objects that represent Siebel business objects.

Table 16. Business object naming conventions for Siebel business objects

Element	Naming convention
Name of the business object	<Prefix>+<BO>+<Business Object Name>+<BC>+<Business Component Name>. Note: The Siebel business object and component names are stripped of all non-alphanumeric characters before being added to the business object name. If the resulting names are not unique, a counter is added to the end of the names. For example, if two business objects have the name, "SiebelBOAccountBCBusinessAddress", a counter is added as a suffix to make them unique, as shown in this example: "SiebelBOAccountBCAddress1" and "SiebelBOAccountBCAddress2"
Name of the container business object generated for the Exists operation	<SiebelExistsResult>
Name of the container business object generated for the Retrieve All operation	<Prefix>+BO+<Business Object Name>+<BC>+<Business Component Name>+Container
Name of the top-level business object	<Prefix>+<BO>+<Business Object Name>+<BC>+<Business Component Name>

Optional: Shorter naming conventions for business objects generated against Siebel business components

The naming conventions for business objects generated against Siebel business components are valid if the optional property, 'Generate business objects with shorter names' is specified in the configuration objects window in the J2C Bean wizard.

Note: If this optional property is used, you must set the 'Folder' property with a unique value. This will avoid overwriting the existing XSDs that were previously generated. For example, Siebel business object -> Siebel business component combination of Account-ESP -> Account and Account (as the top-level object) will be named as, 'Account.xsd'.

The name comprises the concatenation of several words, including prefix and business component name.

The following table describes the naming conventions that the J2C Bean wizard uses to name business objects that are generated against Siebel business components.

Table 17. Shorter business object naming conventions for business objects generated against Siebel business components

Element	Naming convention
Name of the top-level business object generated against business components	<Prefix>+<Name of the Siebel Business Component> and <Prefix>+<Name of the Siebel Business Component> Note: The Siebel business object and integration component names are stripped of all non-alphanumeric characters before being added to the business object name. If the resulting names are not unique, a counter is added to the end of the names.

Outbound configuration properties

IBM WebSphere Adapter for Siebel Business Applications has several categories of outbound connection configuration properties, which you set with the J2C Bean wizard while generating or creating objects and services. You can change the resource adapter and managed connection factory properties after you deploy the module to WebSphere Application Server using Rational Application Developer for WebSphere Software or the WebSphere Application Server Administrative Console, but connection properties for the J2C Bean wizard cannot be changed after deployment.

Guide to information about properties

The properties used to configure WebSphere Adapter for Siebel Business Applications are described in detail in tables included in each of the configuration properties topics, such as Resource adapter properties, Managed connection factory properties, and so on. To help you use these tables, information about each row you might see is explained here.

The following table explains the meaning of each row that might be displayed in the table for a configuration property.

Row	Explanation
Required	<p>A required field (property) must have a value in order for the adapter to work. Sometimes the J2C Bean wizard provides a default value for required properties.</p> <p>Removing a default value from a required field on the J2C Bean wizard <i>will not change that default value</i>. When a required field contains no value at all, the J2C Bean wizard processes the field using its assigned default value, and that default value is displayed on the WebSphere Application Server Administrative Console.</p> <p>Possible values are Yes and No.</p> <p>Sometimes a property is required only when another property has a specific value. When this is the case, the table notes this dependency. For example,</p> <ul style="list-style-type: none">• Yes, when the EventQueryType property is set to Dynamic• Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	<p>The predefined value that is set by the J2C Bean wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table states No default value.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	<p>Describes the property type. Valid property types include:</p> <ul style="list-style-type: none">• Boolean• String• Integer

Row	Explanation
Usage	<p>Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented:</p> <p>For Rational Application Developer for WebSphere Software version 6.40 or earlier, the password:</p> <ul style="list-style-type: none"> • Must be uppercase • Must be 8 characters in length <p>For versions of Rational Application Developer for WebSphere Software later than 6.40, the password:</p> <ul style="list-style-type: none"> • Is not case-sensitive • Can be up to 40 characters in length. <p>This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.</p>
Example	<p>Provides sample property values, for example:</p> <p>"If Language is set to JA (Japanese), code page number is set to 8000".</p>
Globalized	<p>If a property is globalized, it has national language support, meaning that you can set the value in your national language.</p> <p>Valid values are Yes and No.</p>
Bidi supported	<p>Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.</p> <p>Valid values are Yes and No.</p>

Connection properties for the wizard

The J2C Bean wizard properties are used to establish a connection between the J2C Bean wizard of Rational Application Developer for WebSphere Software, a tool that is used to create business objects, and the Siebel Business Applications server. The properties you configure in the J2C Bean wizard specify such things as connection configuration, bidirectional (bidi) properties, and logging and tracing options.

After a connection between the J2C Bean wizard and the Siebel Business Applications server is established, the J2C Bean wizard is able to access the metadata it needs from the Siebel Business Applications server to create business objects.

Some of the properties that you set in the J2C Bean wizard are used as the initial value for resource adapter, managed connection factory, and activation specification properties that you can specify at a later time in the wizard.

The J2C Bean wizard properties and their purpose are described in the following table. A complete description of each property is provided in the sections that follow the table. For more information about how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 87.

Note: If you set any of these J2C Bean wizard properties using bidirectional script, you must set values that identify the format of the bidirectional script entered for that property.

Table 18. J2C Bean wizard properties

Property name in the wizard	Description
Adapter style	Service type associated with the adapter module.
Business object namespace	The namespace value has been added as a prefix to the business object name to keep the business object schemas separated.
Code page	Specifies the code page that the adapter uses to connect to the Siebel Business Applications server.
Connection URL	The connection URL needed to connect to the Siebel application.
Delimiter for keys in the event store	Specifies that the delimiter used between two name value pairs contains the object key name and value.
Folder	The location of the generated business object.
Generate business objects with shorter names	Ensures that the adapter generates shorter business object names, which are based on the Siebel integration components, business services, and business components rather than the concatenation of several words (which is the default).
Language code	The language code used to log on to the Siebel Business Applications server.
Method name	The name of the business service method to be implemented.
Password	The password for the corresponding user name.
Prefix for business object names	The prefix for the business object name.
Siebel business object name for event store	The name of the business object in the event store where events are stored for inbound processing.
Siebel repository name	The name of the Siebel repository from which the objects are to be discovered.
Siebel server view mode	Specifies the Siebel Business Applications mode and controls the data type to be retrieved and what actions can be performed.
Type of Siebel objects to discover	The type of Siebel objects (business objects or business services) that need to be discovered and listed.
Use resonate support for load balancing on Siebel server	Specifies that if resonate support is installed on the Siebel Business Applications server, and the value is set to true, the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently.
User name	The user name that is used to log in to the Siebel application.

Adapter style (AdapterStyle)

This property specifies the service type associated with the adapter module.

Table 19. Service type details

Required	Yes
Default	Outbound
Property type	List of values

Table 19. Service type details (continued)

Possible values	Outbound Inbound
Usage	Specifies the service type associated with the adapter module
Globalized	No
Bidi supported	No

Business object namespace (BusinessObjectNameSpace)

This property specifies that the namespace value has been added as a prefix to the business object name to keep the business object schemas separated.

Table 20. Business object namespace details

Required	Yes
Default	http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel
Property type	String
Usage	The namespace value is added as a prefix to the business object name to keep the business object schemas separated.
Example	http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel/IBMSiebelAccountInsertAccount
Globalized	No
Bidi supported	No

Connection URL (ConnectionURL)

This property specifies the connection URL needed to connect to the Siebel application.

Table 21. Connection URL details

Required	Yes
Default	No default value
Property type	String
Usage	The connection URLs for all versions of Siebel follow this format: Protocol://machinename:port/enterprisename/object manager/server name. The default port number is 2320. For Siebel version 7.5x and earlier versions, the port number (2320) and server name are specified, as shown in the example below. For Siebel version 7.8, the port and server name are not specified, as shown in the example below. If you do not select the default port, then you can specify another port number (for example, 2321), as shown in the example below for Siebel version 8.
Examples	The following sample connection URLs are for different versions of Siebel: <ul style="list-style-type: none"> • For Siebel 7.5: siebel://<IP_address>:2320/siebel/SSEObjMgr_ENU/sebldev1 . • For Siebel 7.8: siebel://<IP_address>/Sieb78/SSEObjMgr_enu . • For Siebel 8: siebel://<IP_address>:2321/SBA_80/SSEObjMgr_enu .
Globalized	Yes
Bidi supported	Yes

Code page (CodePage)

Specifies the code page that the adapter uses to connect to Siebel Business Applications server.

Table 22. Code page details

Required	No
Default	No default value
Property type	String
Usage	<p>The Siebel Java Data Bean supports a limited list of code page settings. If the current code page setting of your tooling is not supported, the adapter may not be able to connect to the Siebel Business Applications server.</p> <p>This property allows the users to specify the code page details that the adapter uses to connect to Siebel Business Applications server for metadata discovery.</p>
Globalized	No
Bidi supported	No

Delimiter for keys in the event store (DelimiterForKeysInTheEventStore)

Table 23. Delimiter for keys in the event store details

Required	Yes
Default	;
Property type	String
Usage	This is the delimiter used between two name value pairs containing the object key name and value.
Examples	The default value for this property can be changed, however, if you remove the default value and do not set it again, the default value (;) will be used. If the event table key field has values, such as AccountId=1-314:Id=1-325, then the event delimiter is the colon (:). The object key names are AccountId and Id. The values are 1-314 and 1-325.
Globalized	Yes
Bidi supported	Yes

Folder (Folder)

This property specifies the location of the generated business objects.

Table 24. Folder details

Required	No
Default	No default value
Property type	String
Usage	The generated business objects are copied into this folder.
Example	inboundartifacts and outboundartifacts
Globalized	No
Bidi supported	No

Generate business objects with shorter names (GenerateBusinessObjectsWithShorterNames)

This property ensures that the adapter generates shorter business object names, which are based on the Siebel integration components, business services, and business components rather than the concatenation of several words (which is the default).

Table 25. Generate business objects with shorter names details

Required	No
Default	No default value
Property type	Boolean
Usage	Ensures that the adapter generates shorter business object names. The shorter business object names are based on the Siebel integration components, business services, and business components. The prefix is also attached to the shorter names. Note: The adapter removes special characters from the shorter business object names. Alphanumeric characters (a-z, A-Z, and 1-9) are supported, and a counter (1-9) is added to the end of business object names to avoid duplication of names.
Example	If 'Account' is the name of the Siebel business component, and 'Siebel' is the prefix, the shorter name, is 'Siebel_Account'.
Globalized	No
Bidi supported	No

Language code (LanguageCode)

This property specifies the language code used to log on to the Siebel Business Applications server.

Table 26. Language code details

Required	Yes
Default	ENU
Property type	String
Usage	If the system locale is English, the value for this property is ENU (English). This is used to log on to the Siebel Business Applications server.
Globalized	No
Bidi supported	No

Method name (MethodName)

This property specifies the name of the business service method to be implemented.

Table 27. Method name details

Required	Yes
Default	Query
Property type	String
Usage	The default is Query.
Example	Query, QueryByExample, QueryById, and so on.
Globalized	Yes

Table 27. Method name details (continued)

Bidi supported	Yes
----------------	-----

Password (Password)

This property specifies the password for the corresponding user name.

Table 28. Password details

Required	Yes
Default	No default value
Property type	String
Usage	If a J2C Authentication Alias is used, then a password is not required.
Example	1-XYZ
Globalized	Yes
Bidi supported	Yes

Prefix for business object names (PrefixForBusinessObjectNames)

This property specifies the prefix for the business object name.

Table 29. Prefix details

Required	No
Default	No default value
Property type	String
Usage	The prefix string is attached to the front of the business object name that was generated.
Example	You use the prefix, IBM and generate a business object for the EAI Siebel Adapter and the Insert method. Then, you choose the Account Interface and Business Address Interface integration object against an Input/Output method argument. The corresponding business object generated would be: IBMEAISiebelAdapterInsertAccountInterfacBusinessAddressInterface.
Globalized	Yes
Bidi supported	Yes

Siebel business object name for event store (SiebelBusinessObjectNameForEventStore)

This property specifies the name of the business object in the event store where events are stored for inbound processing.

Table 30. Siebel business object name for event store details

Required	Yes
Default	IBM_EVENT
Property type	String
Usage	After clicking the advanced button on the connection properties window on the J2C Bean wizard, this property will display under the "Event configuration" tab. The two values listed are IBM_EVENT and IBM2. If you create a custom event component name, you can specify the value for it in the list box.
Globalized	Yes

Table 30. Siebel business object name for event store details (continued)

Bidi supported	No
----------------	----

Siebel repository name (SiebelRepositoryName)

This property specifies the name of the Siebel repository from which the objects are to be discovered.

Table 31. Siebel repository name details

Required	Yes
Default	Siebel Repository
Property type	String
Usage	This default value is Siebel Repository. Although this is a required field, it is optional on the J2C Bean wizard. You can edit this value to point to other repositories if needed.
Globalized	No
Bidi supported	No

Siebel server view mode (SiebelServerViewMode)

This property specifies the Siebel Business Applications server view mode and controls the data that can be retrieved and what actions can be performed on it.

Table 32. Siebel server view mode details

Required	Yes
Default	3
Property type	Integer
Usage	This property displays after clicking the advanced button on the connection properties window on the J2C Bean wizard. This mode, when set to "Type of Siebel objects to discover" applies only to Siebel business objects, not to Siebel business services. The values supported by Siebel are 1 to 9.
Globalized	No

Type of Siebel objects to discover (TypeOfSiebelObjectsToDiscover)

This property specifies the type of Siebel object that needs to be discovered and listed.

Table 33. Type of Siebel objects to discover details

Required	Yes
Possible values	Siebel business objects or Siebel business services
Default	Siebel business objects
Property type	String
Usage	Although the default is Siebel business objects, you can select Siebel business services. Based on your selection, the J2C Bean wizard retrieves either the business objects or the business services.
Globalized	No
Bidi supported	No

Use resonate support for load balancing on Siebel server (UseResonateSupportForLoadBalancingOnSiebelServer)

This property indicates whether the Siebel Business Applications server uses resonate support.

Table 34. Use resonate support for load balancing on Siebel server details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	This property displays after clicking the advanced button on the connection properties window on the J2C Bean wizard. If the check box is selected, it will be set to True, and the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently. If the check box is not selected, it will be set to false.
Globalized	No

User name (UserName)

This property specifies the user name that is used to log in to the Siebel application.

Table 35. User name details

Required	Yes
Default	No default value
Property type	String
Usage	If a J2C Authentication Alias is used, then a user name is not required.
Globalized	Yes
Bidi supported	Yes

Resource adapter properties

The resource adapter properties control the general operation of the adapter. You set the resource adapter properties in the J2C Bean wizard when you configure the adapter. After deploying the adapter, use the WebSphere Application Server Administrative Console to change these properties.

The following properties for logging and tracing are no longer required in version 7.5.0.2, but are supported for compatibility with previous versions:

- LogFileMaxSize
- LogFileName
- LogNumberOfFiles
- TraceFileMaxSize
- TraceFileName
- TraceNumberOfFiles

The following table lists and describes the resource adapter properties. A more detailed description of each property is provided in the sections that follow the

table. For more information about how to read the property detail tables in the sections to follow, see "Guide to information about properties" on page 87.

Table 36. Resource adapter properties

Property name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
(Not available)	enableHASupport	Do not change this property.
Event delimiter	EventDelimiter	Specifies whether the delimiter used between two name value pairs contains the object key name and value.
(Not available)	LogFileMaxSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
Resonate support	resonateSupport	Specifies that if resonate support is installed on the Siebel Business Applications server, and the value is set to True, the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently.
Siebel server view mode	viewMode	Specifies the Siebel view mode and controls the type of data to be retrieved and the actions to be performed.
(Not available)	TraceFileMaxSize	Deprecated
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

This property identifies a specific deployment or instance of the adapter.

Table 37. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 37. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, SEBLRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is SEBLRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Siebel Business Applications to 001 and 002. The component IDs for those instances, SEBLRA001 and SEBLRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SEBLRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	Yes
Bidi supported	No

Disguise user data as "XXX" in log and trace files (HideConfidentialTrace)

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 38. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

Do not change this property. It must be set to true.

Event delimiter (EventDelimiter)

This property indicates that the delimiter used between two name value pairs contains the object key name and value.

Table 39. Event delimiter details

Required	Yes
Default	;
Property type	String
Usage	If multiple value pairs are set against the object key in the event component, they will be used for the delimiter.
Globalized	No

Log file maximum size (LogFileMaxSize)

This property specifies the size of the log files in kilobytes.

Table 40. Log file maximum size details

Required	No
Default	0
Property type	Integer
Usage	When the log file reaches its maximum size, the adapter starts using a new log file. If the file size is specified as 0 or no maximum size is specified, the file does not have a maximum size.
Globalized	Yes
Bidi supported	No

Log file name (LogFilename)

This property specifies the full path name of the log file.

Table 41. Log file name details

Required	No
Default	No default value
Property type	String
Usage	This property is deprecated.
Globalized	Yes
Bidi supported	Yes

Log number of files (LogNumberOfFiles)

This property specifies the number of log files.

Table 42. Log number of files details

Required	No
----------	----

Table 42. Log number of files details (continued)

Default	1
Property type	Integer
Usage	When a log file reaches its maximum size, the adapter starts using another log file. If no value is specified, the adapter creates a single log file.
Globalized	Yes
Bidi supported	No

Resonate support (ResonateSupport)

This property indicates whether the Siebel Business Applications server uses resonate support.

Table 43. Resonate support details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	If the value of this property is set to True, the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently.
Globalized	No

Siebel server view mode (SiebelServerViewMode)

This property specifies the Siebel view mode and controls the data that can be retrieved and the actions to be performed on it.

Table 44. View mode details

Required	Yes
Default	3
Property type	Integer
Usage	The View mode property applies only to Siebel business objects and not to Siebel business services.
Globalized	No

Trace file maximum size (TraceFileMaxSize)

This property specifies the size of the trace files in kilobytes.

Table 45. Trace file maximum size details

Required	No
Default	0
Property type	Integer
Usage	If no value is specified, then the trace file has no maximum size.
Globalized	Yes
Bidi supported	No

Trace file name (TraceFilename)

This property specifies the full path of the trace file.

Table 46. Trace file name details

Required	No
Default	No default value
Unit of measure	Kilobytes
Property type	String
Usage	This property is deprecated.
Globalized	Yes
Bidi supported	Yes

Trace number of files (TraceNumberOfFiles)

This property specifies the number of trace files to use. When a trace file reaches its maximum size, the adapter starts using another trace file.

Table 47. Trace number of files details

Required	No
Default	1
Property type	Integer
Usage	If no value is specified, the adapter uses a single trace file.
Globalized	Yes
Bidi supported	No

Managed connection factory properties

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the Siebel application.

You set the managed connection factory properties through either the J2C Bean wizard or the administrative console (after deployment).

The following table lists the activation specification properties for inbound communication. You set the activation specification properties using the J2C Bean wizard or after deployment through the WebSphere Application Server Administrative Console. A more detailed description of each property is provided in the sections that follow the table. For information about how to read the property detail tables in the sections that follow, see Guide to understanding property details.

Note: The J2C Bean wizard refers to these properties as managed connection factory properties and the WebSphere Application Server administrative console refers to them as (J2C) connection factory properties.

Table 48. Managed connection factory properties

Property name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.
Connection URL	connectString	The connection URL needed to connect to the Siebel application.
Code page	codePage	Specifies the code page that the adapter uses to connect to Siebel server.
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
Language code	languageCode	The language code used to log on to the Siebel Business Applications server.
"Maximum retries in case of system connection failure (connectionRetryLimit)" on page 104	connectionRetryLimit	Specifies the maximum number of times the adapter tries to reestablish an outbound connection to the EIS.
Password	password	The password for the corresponding user name.
Prefix	prefix	The prefix for the business object name.
"Retry interval if connection fails (in milliseconds) (connectionRetryInterval)" on page 106	connectionRetryInterval	Specifies the time interval between attempts to reconnect to the Siebel Business Applications server if the connection fails.
Resonate support	resonateSupport	Specifies that if resonate support is installed on the Siebel Business Applications server, and the value is set to True, the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently.
"Show error on empty result set (errorOnEmptyResultSet)" on page 106	errorOnEmptyResultSet	For a business object, the property specifies if the adapter returns an empty business object or RecordNotFoundException when the result of RetrieveAll operation does not return any records. For a business service, the property specifies if the adapter generates an exception when a returned Siebel message is null.
User name	userName	The user name that is used to log in to the Siebel application
View mode	viewMode	Specifies the Siebel view mode and controls the data that can be retrieved and what actions can be performed on it.

Adapter ID (AdapterID)

This property identifies a specific deployment or instance of the adapter.

Table 49. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 49. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, SEBLRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is SEBLRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Siebel Business Applications to 001 and 002. The component IDs for those instances, SEBLRA001 and SEBLRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SEBLRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	Yes
Bidi supported	No

Code page (codePage)

Specifies the code page that the adapter uses to connect to Siebel server.

Table 50. Code page details

Required	No
Default	No default value
Property type	String
Usage	<p>The Siebel Java Data Bean supports a limited list of code page settings. If the current code page setting of your run time is not supported, the adapter may not be able to connect to the Siebel server.</p> <p>This property allows the users to specify the code page details that the adapter uses to connect to Siebel server.</p> <p>Note: The adapter uses the code page property to change the “file.encoding” property of the current Java Virtual Machine (JVM), when it establishes its first connection to the Siebel server. This may affect other applications which run on the same JVM. You need to ensure that the code page has a valid value for all applications.</p>
Globalized	No
Bidi supported	No

Connection URL (ConnectionURL)

This property specifies the connection URL needed to connect to the Siebel application.

Table 51. Connection URL details

Required	Yes
Default	No default value
Property type	String
Usage	Protocol://machinename:port/enterprisename/object manager/server name. <ul style="list-style-type: none">• For Siebel 7.0.5 to 7.5x: siebel://<IP ADDRESS>/siebel/SSEObjMgr_ENU/sebldev1.• For Siebel 7.8: siebel://<IP ADDRESS>:2321/Sieb78/SSEObjMgr_enu.• For Siebel 8: siebel://<IP ADDRESS>:2321/SBA_80/SSEObjMgr_enu. The default port number is 2320. In the examples (for Siebel versions 7.8 and 8), another port (2321) has been specified.
Globalized	Yes
Bidi supported	Yes

Disguise user data as "XXX" in log and trace files (HideConfidentialTrace)

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 52. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.
Globalized	No
Bidi supported	No

Include empty elements in the output (includeEmptyField)

Specifies if the adapter returns the elements that have an empty string value in the response Java beans.

Table 53. Include empty elements in the output details

Required	No
Possible values	True False

Table 53. Include empty elements in the output details (continued)

Default	False
Property type	Boolean
Usage	This property enables you to include the empty elements in the output. When this property is selected, the adapter returns the elements that have empty string values in the response Java beans.
Globalized	No
Bidi supported	No

Language code (LanguageCode)

This property specifies the language code used to log on to the Siebel Business Applications server.

Table 54. Language code details

Required	Yes
Possible values	None
Default	ENU
Property type	String
Usage	If the system locale is English, the value for this property is ENU (English). It is used to log on to the Siebel Business Applications server.
Globalized	No
Bidi supported	No

Maximum retries in case of system connection failure (connectionRetryLimit)

This property specifies the number of times the adapter attempts to reestablish a connection to the Siebel Business Applications server, when the adapter encounters an error related to the outbound connection.

Table 55. Maximum retries in case of system connection failure details

Required	No
Possible values	Integers equal to and greater than zero
Default	0
Property type	Integer

Table 55. Maximum retries in case of system connection failure details (continued)

Usage	<p>When this property is set to:</p> <p>0</p> <ul style="list-style-type: none"> • The adapter does not attempt to reconnect to the Siebel Business Applications server, if an error occurs during startup or while establishing a connection. • The adapter does not verify if the connection to the Siebel Business Applications server is valid when there is an outbound request during run time. <p>>0</p> <ul style="list-style-type: none"> • The adapter attempts to reconnect to the Siebel Business Applications server for the specified number of times, if an error occurs during startup or while establishing a connection. • The adapter verifies if the connection to the Siebel Business Applications server is valid when there is an outbound request during run time. If the connection is not valid, it is terminated and a new connection is created to process the request. <p>If the adapter fails to establish a connection after trying for the specified number of times, a connection error is generated.</p> <p>If the adapter is successful in reestablishing the connection, the outbound operation is completed.</p>
Globalized	No
Bidi supported	No

Password (Password)

This property specifies the password for the corresponding user name.

Table 56. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the version of Siebel application. If a J2C Authentication Alias is used, then you need not set the password.
Globalized	Yes
Bidi supported	Yes

Prefix (Prefix)

This property specifies the prefix for the business object name.

Table 57. Prefix details

Required	No
Default	No default value
Property type	String
Usage	The prefix string is attached to the front of the business object name.

Table 57. Prefix details (continued)

Example	Use the prefix, IBM and generate a business object for the EAI Siebel Adapter and the Insert method. Then you choose the integration object, Account (PRM ANI). The corresponding business object generated is: IBMEAISiebelAdapterInsertAccountU40PRMANIU41, where U40 and U41 are the unicode value replacements of (and).
Globalized	Yes
Bidi supported	Yes

Retry interval if connection fails (in milliseconds) (connectionRetryInterval)

This property specifies the time interval between attempts to reconnect to the Siebel Business Applications server if the connection fails.

Table 58. Retry interval if connection failure details

Required	No
Possible values	Integers equal and greater than 0
Default	60000
Unit of measure	Milliseconds
Property type	Integer
Usage	This property is applicable only if the value of the property Maximum retries in case of system connection failure is set to greater than 0. When the adapter encounters an error while establishing a connection to the Siebel Business Applications server, this property specifies the time interval the adapter waits between attempts to reestablish a connection.
Globalized	No
Bidi supported	No

Resonate support (ResonateSupport)

This property indicates whether the Siebel Business Applications server uses resonate support.

Table 59. Resonate support details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	If the property is selected (value set to True), and the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently.
Globalized	No

Show error on empty result set (errorOnEmptyResultSet)

For a business object, this property specifies if the adapter returns an empty business object or RecordNotFoundException when the result of RetrieveAll

operation does not return any records. For a business service, this property specifies if the adapter generates an exception when a returned Siebel message is null.

Table 60. Show error on empty result set details

Required	No
Default	For business object: True For business service: False
Property type	Boolean
Usage	For business object, when this property is set to: <ul style="list-style-type: none"> • True: Throws a RecordNotFoundException if no records are found in RetrieveAll operation. • False: Returns an empty business object if no records are found in RetrieveAll operation. For business service, when this property is set to: <ul style="list-style-type: none"> • True: Throws a RecordNotFoundException if no records are returned. • False: Returns null Siebel message if no records are returned.
Globalized	No
Bidi supported	No

User name (UserName)

This property specifies the user name that is used to log in to the Siebel application.

Table 61. User name details

Required	Yes
Possible values	None
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the version of Siebel application. If a J2C Authentication Alias is used, then you need not set the user name.
Globalized	Yes
Bidi supported	Yes

View mode (ViewMode)

This property specifies the Siebel view mode and controls the data that can be retrieved and what actions can be performed on it.

Table 62. View mode details

Required	Yes
Possible values	1 - 9
Default	3
Property type	Integer

Table 62. View mode details (continued)

Usage	The View mode property applies only to Siebel business objects and not to Siebel business services. When this property is used for Siebel business objects, the default is 3.
Examples	The adapter supports values 1 - 9. For example, 1 is Manager View, 2 is Personal View, and 3 is All View.
Globalized	No

Interaction specification properties

Interaction specification properties control the interaction for an operation. The J2C Bean wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, some properties for outbound operations can be changed by the user. For example, you might increase the value of the interaction specification property that specifies the maximum number of records to be returned by a RetrieveAll operation, if your RetrieveAll operations do not return complete information. Use the J2C bean generated in the Rational Application Developer for WebSphere Software WebSphere Application Server Administrative Console to change these properties.

Table 63 lists and describes the interaction specification property that you set. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 87.

Table 63. Interaction specification property for the WebSphere Adapter for Siebel Business Applications

Property name	Description
“Maximum records (maxRecords)”	Maximum number of records to return during a RetrieveAll operation

Maximum records (maxRecords)

This property specifies the maximum number of records to return during a RetrieveAll operation.

Table 64. Maximum records details

Required	Yes
Default	100
Usage	If the number of hits in the database exceeds the value of the MaximumRecords property, the adapter returns the MatchesExceededLimitException error. The adapter uses this property to help avoid out-of-memory issues.
Property type	Integer
Globalized	No
Bidi supported	No

Inbound configuration properties

WebSphere Adapter for Siebel Business Applications has several categories of inbound connection configuration properties, which you set with the J2C Bean wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using Rational Application Developer for WebSphere Software or the WebSphere Application Server Administrative Console, but connection properties for the J2C Bean wizard cannot be changed after deployment.

Guide to information about properties

The properties used to configure WebSphere Adapter for Siebel Business Applications are described in detail in tables included in each of the configuration properties topics, such as Resource adapter properties, Managed connection factory properties, and so on. To help you use these tables, information about each row you might see is explained here.

The following table explains the meaning of each row that might be displayed in the table for a configuration property.

Row	Explanation
Required	<p>A required field (property) must have a value in order for the adapter to work. Sometimes the J2C Bean wizard provides a default value for required properties.</p> <p>Removing a default value from a required field on the J2C Bean wizard <i>will not change that default value</i>. When a required field contains no value at all, the J2C Bean wizard processes the field using its assigned default value, and that default value is displayed on the WebSphere Application Server Administrative Console.</p> <p>Possible values are Yes and No.</p> <p>Sometimes a property is required only when another property has a specific value. When this is the case, the table notes this dependency. For example,</p> <ul style="list-style-type: none"> • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	<p>The predefined value that is set by the J2C Bean wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table states No default value.</p> <p>The word None is an acceptable default value, and does not mean that there is no default value.</p>
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	<p>Describes the property type. Valid property types include:</p> <ul style="list-style-type: none"> • Boolean • String • Integer
Usage	<p>Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented:</p> <p>For Rational Application Developer for WebSphere Software version 6.40 or earlier, the password:</p> <ul style="list-style-type: none"> • Must be uppercase • Must be 8 characters in length <p>For versions of Rational Application Developer for WebSphere Software later than 6.40, the password:</p> <ul style="list-style-type: none"> • Is not case-sensitive • Can be up to 40 characters in length. <p>This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.</p>

Row	Explanation
Example	Provides sample property values, for example: "If Language is set to JA (Japanese), code page number is set to 8000".
Globalized	If a property is globalized, it has national language support, meaning that you can set the value in your national language. Valid values are Yes and No .
Bidi supported	Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file. Valid values are Yes and No .

Connection properties for the wizard

The J2C Bean wizard properties are used to establish a connection between the J2C Bean wizard of Rational Application Developer for WebSphere Software, a tool that is used to create business objects, and the Siebel Business Applications server. The properties you configure in the J2C Bean wizard specify such things as connection configuration, bidirectional (bidi) properties, and logging and tracing options.

After a connection between the J2C Bean wizard and the Siebel Business Applications server is established, the J2C Bean wizard is able to access the metadata it needs from the Siebel Business Applications server to create business objects.

Some of the properties that you set in the J2C Bean wizard are used as the initial value for resource adapter, managed connection factory, and activation specification properties that you can specify at a later time in the wizard.

The J2C Bean wizard properties and their purpose are described in the following table. A complete description of each property is provided in the sections that follow the table. For more information about how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 87.

Note: If you set any of these J2C Bean wizard properties using bidirectional script, you must set values that identify the format of the bidirectional script entered for that property.

Table 65. J2C Bean wizard properties

Property name in the wizard	Description
Adapter style	Service type associated with the adapter module.
Business object namespace	The namespace value has been added as a prefix to the business object name to keep the business object schemas separated.
Code page	Specifies the code page that the adapter uses to connect to the Siebel Business Applications server.
Connection URL	The connection URL needed to connect to the Siebel application.
Delimiter for keys in the event store	Specifies that the delimiter used between two name value pairs contains the object key name and value.
Folder	The location of the generated business object.

Table 65. J2C Bean wizard properties (continued)

Property name in the wizard	Description
Generate business objects with shorter names	Ensures that the adapter generates shorter business object names, which are based on the Siebel integration components, business services, and business components rather than the concatenation of several words (which is the default).
Language code	The language code used to log on to the Siebel Business Applications server.
Method name	The name of the business service method to be implemented.
Password	The password for the corresponding user name.
Prefix for business object names	The prefix for the business object name.
Siebel business object name for event store	The name of the business object in the event store where events are stored for inbound processing.
Siebel repository name	The name of the Siebel repository from which the objects are to be discovered.
Siebel server view mode	Specifies the Siebel Business Applications mode and controls the data type to be retrieved and what actions can be performed.
Type of Siebel objects to discover	The type of Siebel objects (business objects or business services) that need to be discovered and listed.
Use resonate support for load balancing on Siebel server	Specifies that if resonate support is installed on the Siebel Business Applications server, and the value is set to true, the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently.
User name	The user name that is used to log in to the Siebel application.

Adapter style (AdapterStyle)

This property specifies the service type associated with the adapter module.

Table 66. Service type details

Required	Yes
Default	Outbound
Property type	List of values
Possible values	Outbound Inbound
Usage	Specifies the service type associated with the adapter module
Globalized	No
Bidi supported	No

Business object namespace (BusinessObjectNameSpace)

This property specifies that the namespace value has been added as a prefix to the business object name to keep the business object schemas separated.

Table 67. Business object namespace details

Required	Yes
Default	http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel
Property type	String
Usage	The namespace value is added as a prefix to the business object name to keep the business object schemas separated.
Example	http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel/IBMSiebelAccountInsertAccount
Globalized	No
Bidi supported	No

Connection URL (ConnectionURL)

This property specifies the connection URL needed to connect to the Siebel application.

Table 68. Connection URL details

Required	Yes
Default	No default value
Property type	String
Usage	The connection URLs for all versions of Siebel follow this format: Protocol://machinename:port/enterprisename/object manager/server name. The default port number is 2320. For Siebel version 7.5x and earlier versions, the port number (2320) and server name are specified, as shown in the example below. For Siebel version 7.8, the port and server name are not specified, as shown in the example below. If you do not select the default port, then you can specify another port number (for example, 2321), as shown in the example below for Siebel version 8.
Examples	The following sample connection URLs are for different versions of Siebel: <ul style="list-style-type: none"> • For Siebel 7.5: siebel://<IP_address>:2320/siebel/SSEObjMgr_ENU/sebldev1 . • For Siebel 7.8: siebel://<IP_address>/Sieb78/SSEObjMgr_enu . • For Siebel 8: siebel://<IP_address>:2321/SBA_80/SSEObjMgr_enu .
Globalized	Yes
Bidi supported	Yes

Code page (CodePage)

Specifies the code page that the adapter uses to connect to Siebel Business Applications server.

Table 69. Code page details

Required	No
Default	No default value
Property type	String
Usage	The Siebel Java Data Bean supports a limited list of code page settings. If the current code page setting of your tooling is not supported, the adapter may not be able to connect to the Siebel Business Applications server. This property allows the users to specify the code page details that the adapter uses to connect to Siebel Business Applications server for metadata discovery.
Globalized	No

Table 69. Code page details (continued)

Bidi supported	No
----------------	----

Delimiter for keys in the event store (DelimiterForKeysInTheEventStore)

Table 70. Delimiter for keys in the event store details

Required	Yes
Default	;
Property type	String
Usage	This is the delimiter used between two name value pairs containing the object key name and value.
Examples	The default value for this property can be changed, however, if you remove the default value and do not set it again, the default value (;) will be used. If the event table key field has values, such as AccountId=1-314:Id=1-325, then the event delimiter is the colon (:). The object key names are AccountId and Id. The values are 1-314 and 1-325.
Globalized	Yes
Bidi supported	Yes

Folder (Folder)

This property specifies the location of the generated business objects.

Table 71. Folder details

Required	No
Default	No default value
Property type	String
Usage	The generated business objects are copied into this folder.
Example	inboundartifacts and outboundartifacts
Globalized	No
Bidi supported	No

Generate business objects with shorter names (GenerateBusinessObjectsWithShorterNames)

This property ensures that the adapter generates shorter business object names, which are based on the Siebel integration components, business services, and business components rather than the concatenation of several words (which is the default).

Table 72. Generate business objects with shorter names details

Required	No
Default	No default value
Property type	Boolean

Table 72. Generate business objects with shorter names details (continued)

Usage	Ensures that the adapter generates shorter business object names. The shorter business object names are based on the Siebel integration components, business services, and business components. The prefix is also attached to the shorter names. Note: The adapter removes special characters from the shorter business object names. Alphanumeric characters (a-z, A-Z, and 1-9) are supported, and a counter (1-9) is added to the end of business object names to avoid duplication of names.
Example	If 'Account' is the name of the Siebel business component, and 'Siebel' is the prefix, the shorter name, is 'Siebel_Account'.
Globalized	No
Bidi supported	No

Language code (LanguageCode)

This property specifies the language code used to log on to the Siebel Business Applications server.

Table 73. Language code details

Required	Yes
Default	ENU
Property type	String
Usage	If the system locale is English, the value for this property is ENU (English). This is used to log on to the Siebel Business Applications server.
Globalized	No
Bidi supported	No

Method name (MethodName)

This property specifies the name of the business service method to be implemented.

Table 74. Method name details

Required	Yes
Default	Query
Property type	String
Usage	The default is Query.
Example	Query, QueryByExample, QueryById, and so on.
Globalized	Yes
Bidi supported	Yes

Password (Password)

This property specifies the password for the corresponding user name.

Table 75. Password details

Required	Yes
Default	No default value
Property type	String

Table 75. Password details (continued)

Usage	If a J2C Authentication Alias is used, then a password is not required.
Example	1-XYZ
Globalized	Yes
Bidi supported	Yes

Prefix for business object names (PrefixForBusinessObjectNames)

This property specifies the prefix for the business object name.

Table 76. Prefix details

Required	No
Default	No default value
Property type	String
Usage	The prefix string is attached to the front of the business object name that was generated.
Example	You use the prefix, IBM and generate a business object for the EAI Siebel Adapter and the Insert method. Then, you choose the Account Interface and Business Address Interface integration object against an Input/Output method argument. The corresponding business object generated would be: IBMEAISiebelAdapterInsertAccountInterfacBusinessAddressInterface.
Globalized	Yes
Bidi supported	Yes

Siebel business object name for event store (SiebelBusinessObjectNameForEventStore)

This property specifies the name of the business object in the event store where events are stored for inbound processing.

Table 77. Siebel business object name for event store details

Required	Yes
Default	IBM_EVENT
Property type	String
Usage	After clicking the advanced button on the connection properties window on the J2C Bean wizard, this property will display under the "Event configuration" tab. The two values listed are IBM_EVENT and IBM2. If you create a custom event component name, you can specify the value for it in the list box.
Globalized	Yes
Bidi supported	No

Siebel repository name (SiebelRepositoryName)

This property specifies the name of the Siebel repository from which the objects are to be discovered.

Table 78. Siebel repository name details

Required	Yes
Default	Siebel Repository

Table 78. Siebel repository name details (continued)

Property type	String
Usage	This default value is Siebel Repository. Although this is a required field, it is optional on the J2C Bean wizard. You can edit this value to point to other repositories if needed.
Globalized	No
Bidi supported	No

Siebel server view mode (SiebelServerViewMode)

This property specifies the Siebel Business Applications server view mode and controls the data that can be retrieved and what actions can be performed on it.

Table 79. Siebel server view mode details

Required	Yes
Default	3
Property type	Integer
Usage	This property displays after clicking the advanced button on the connection properties window on the J2C Bean wizard. This mode, when set to "Type of Siebel objects to discover" applies only to Siebel business objects, not to Siebel business services. The values supported by Siebel are 1 to 9.
Globalized	No

Type of Siebel objects to discover (TypeOfSiebelObjectsToDiscover)

This property specifies the type of Siebel object that needs to be discovered and listed.

Table 80. Type of Siebel objects to discover details

Required	Yes
Possible values	Siebel business objects or Siebel business services
Default	Siebel business objects
Property type	String
Usage	Although the default is Siebel business objects, you can select Siebel business services. Based on your selection, the J2C Bean wizard retrieves either the business objects or the business services.
Globalized	No
Bidi supported	No

Use resonate support for load balancing on Siebel server (UseResonateSupportForLoadBalancingOnSiebelServer)

This property indicates whether the Siebel Business Applications server uses resonate support.

Table 81. Use resonate support for load balancing on Siebel server details

Required	No
Possible values	True False

Table 81. Use resonate support for load balancing on Siebel server details (continued)

Default	True
Property type	Boolean
Usage	This property displays after clicking the advanced button on the connection properties window on the J2C Bean wizard. If the check box is selected, it will be set to True, and the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently. If the check box is not selected, it will be set to false.
Globalized	No

User name (UserName)

This property specifies the user name that is used to log in to the Siebel application.

Table 82. User name details

Required	Yes
Default	No default value
Property type	String
Usage	If a J2C Authentication Alias is used, then a user name is not required.
Globalized	Yes
Bidi supported	Yes

Resource adapter properties

The resource adapter properties control the general operation of the adapter. You set the resource adapter properties in the J2C Bean wizard when you configure the adapter. After deploying the adapter, use the WebSphere Application Server Administrative Console to change these properties.

The following properties for logging and tracing are no longer required in version 7.5.0.2, but are supported for compatibility with previous versions:

- LogFileMaxSize
- LogFileName
- LogNumberOfFiles
- TraceFileMaxSize
- TraceFileName
- TraceNumberOfFiles

The following table lists and describes the resource adapter properties. A more detailed description of each property is provided in the sections that follow the table. For more information about how to read the property detail tables in the sections to follow, see “Guide to information about properties” on page 87.

Table 83. Resource adapter properties

Property name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.

Table 83. Resource adapter properties (continued)

Property name		Description
In the wizard	In the administrative console	
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
(Not available)	enableHASupport	Do not change this property.
Event delimiter	EventDelimiter	Specifies whether the delimiter used between two name value pairs contains the object key name and value.
(Not available)	LogFileMaxSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
Resonate support	resonateSupport	Specifies that if resonate support is installed on the Siebel Business Applications server, and the value is set to True, the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently.
Siebel server view mode	viewMode	Specifies the Siebel view mode and controls the type of data to be retrieved and the actions to be performed.
(Not available)	TraceFileMaxSize	Deprecated
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

This property identifies a specific deployment or instance of the adapter.

Table 84. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 84. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, SEBLRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is SEBLRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for Siebel Business Applications to 001 and 002. The component IDs for those instances, SEBLRA001 and SEBLRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SEBLRAInstanc.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	Yes
Bidi supported	No

Disguise user data as "XXX" in log and trace files (HideConfidentialTrace)

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 85. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the J2C Bean wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently.</p>
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

Do not change this property. It must be set to true.

Event delimiter (EventDelimiter)

This property indicates that the delimiter used between two name value pairs contains the object key name and value.

Table 86. Event delimiter details

Required	Yes
Default	;
Property type	String
Usage	If multiple value pairs are set against the object key in the event component, they will be used for the delimiter.
Globalized	No

Log file maximum size (LogFileMaxSize)

This property specifies the size of the log files in kilobytes.

Table 87. Log file maximum size details

Required	No
Default	0
Property type	Integer
Usage	When the log file reaches its maximum size, the adapter starts using a new log file. If the file size is specified as 0 or no maximum size is specified, the file does not have a maximum size.
Globalized	Yes
Bidi supported	No

Log file name (LogFilename)

This property specifies the full path name of the log file.

Table 88. Log file name details

Required	No
Default	No default value
Property type	String
Usage	This property is deprecated.
Globalized	Yes
Bidi supported	Yes

Log number of files (LogNumberOfFiles)

This property specifies the number of log files.

Table 89. Log number of files details

Required	No
----------	----

Table 89. Log number of files details (continued)

Default	1
Property type	Integer
Usage	When a log file reaches its maximum size, the adapter starts using another log file. If no value is specified, the adapter creates a single log file.
Globalized	Yes
Bidi supported	No

Resonate support (ResonateSupport)

This property indicates whether the Siebel Business Applications server uses resonate support.

Table 90. Resonate support details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	If the value of this property is set to True, the adapter takes advantage of the load balancing feature to connect to the Siebel Business Applications server more efficiently.
Globalized	No

Siebel server view mode (SiebelServerViewMode)

This property specifies the Siebel view mode and controls the data that can be retrieved and the actions to be performed on it.

Table 91. View mode details

Required	Yes
Default	3
Property type	Integer
Usage	The View mode property applies only to Siebel business objects and not to Siebel business services.
Globalized	No

Trace file maximum size (TraceFileMaxSize)

This property specifies the size of the trace files in kilobytes.

Table 92. Trace file maximum size details

Required	No
Default	0
Property type	Integer
Usage	If no value is specified, then the trace file has no maximum size.
Globalized	Yes
Bidi supported	No

Trace file name (TraceFilename)

This property specifies the full path of the trace file.

Table 93. Trace file name details

Required	No
Default	No default value
Unit of measure	Kilobytes
Property type	String
Usage	This property is deprecated.
Globalized	Yes
Bidi supported	Yes

Trace number of files (TraceNumberOfFiles)

This property specifies the number of trace files to use. When a trace file reaches its maximum size, the adapter starts using another trace file.

Table 94. Trace number of files details

Required	No
Default	1
Property type	Integer
Usage	If no value is specified, the adapter uses a single trace file.
Globalized	Yes
Bidi supported	No

Activation specification properties

Activation specification properties hold the inbound event processing configuration information for a message endpoint. You can set the activation specification properties either through the J2C Bean wizard or the WebSphere Application Server Administrative Console.

Activation specification properties are used during the endpoint activation to notify the adapter of eligible event listeners. During inbound processing, the adapter uses these event listeners to receive the events before forwarding them to the endpoint.

The following table lists the activation specification properties for inbound communication. You set the activation specification properties in the J2C Bean wizard or after deployment through the WebSphere Application Server Administrative Console. A more detailed description of each property is provided in the sections that follow the table. For information about how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 87

Table 95. Activation specification properties

Property name		Description
In the wizard	In the administrative console	
Connection URL	connectString	The connection URL to connect to the Siebel application.
“Code page (codePage)” on page 125	codePage	Specifies the code page that the adapter uses to connect to Siebel server.
Delivery type	DeliveryType	Determines the order in which events are delivered by the adapter to the export.
Do not process events that have a timestamp in the future	FilterFutureEvents	Specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time.
Ensure once-only event delivery	AssuredOnceDelivery	Specifies whether the adapter provides assured once delivery of events.
Event component name	eventComponentName	The name of the Siebel component for the event table.
Event types to process	EventTypeFilter	A delimited list of event types that indicates to the adapter which events it should deliver. If the value of the property is blank or set to "*", all event types are included. In addition, the property supports filtering at the operations level.
Interval between polling periods	PollPeriod	The length of time that the adapter waits between polling periods.
Language code	languageCode	The language code used to log on to the Siebel Business Applications server.
Maximum connections	MaximumConnections	The maximum number of connections that the adapter can use for inbound event delivery.
Maximum events in polling period	PollQuantity	The number of events the adapter delivers to the export during each poll period.
Minimum connections	MinimumConnections	The minimum number of connections that the adapter can use for inbound event delivery.
Maximum number of retries in case of system connection failure	RetryLimit	The number of times the adapter tries to reestablish an inbound connection after an error.
Password	password	Password for the corresponding user name.
Retry limit for failed events	FailedEventRetryLimit	The number of times the adapter attempts to redeliver an event before marking the event as failed.
Retry EIS connection on startup	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the Siebel application if it cannot connect at startup.
Time between retries in case of system connection failure (milliseconds)	RetryInterval	The length of time that the adapter waits between attempts to reestablish connection after an error during inbound operations.
Stop the adapter when an error is encountered while polling	StopPollingOnError	Specifies whether the adapter stops polling for events when it encounters an error during polling.
User name	userName	The user name that is used to log in to the Siebel application.

Table 95. Activation specification properties (continued)

Property name		Description
In the wizard	In the administrative console	
“Adapter Instance for event filtering (AdapterInstanceEventFilter)”	AdapterInstanceEventFilter	Identifier that determines whether this adapter instance processes specific events in the event store.

Adapter Instance for event filtering (AdapterInstanceEventFilter)

This property controls whether the adapter instance processes specific events in the event store.

Table 96. Adapter Instance for event filtering details

Required	No
Default	null
Property type	String
Usage	<p>This property helps you migrate from WebSphere Business Integration Adapter for Siebel eBusiness Applications to WebSphere Adapter for Siebel Business Applications. WebSphere Business Integration Adapter for Siebel eBusiness Applications allows you to perform load balancing on high-volume event types by allowing multiple adapter instances to process events of the same type. When load balancing is not required, a single adapter instance processes all events of a given type. This property is to enable seamless migration for WBIA customers to JCA for customers who are currently taking advantage of the connectorID filtering.</p> <p>WebSphere Adapter for Siebel Business Applications typically does not require load balancing in this way, but supports it so that you can migrate without modifying the database triggers or other mechanisms that write events to the event store.</p> <p>The AdapterInstanceEventFilter property corresponds to the ConnectorID property of the WebSphere Business Integration Adapter for Siebel eBusiness Applications.</p> <p>To use this feature, the database triggers or other mechanisms that create events in the event store must assign the appropriate value to the ConnectorId column.</p> <p>Table 97 on page 125 shows the interaction between the AdapterInstanceEventFilter property and the value in the ConnectorId column in the event store.</p> <p>If the EventTypeFilter and AdapterInstanceEventFilter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose ConnectorId column matches the AdapterInstanceEventFilter property.</p>
Example	See Table 97 on page 125.
Globalized	Yes
Bidi supported	Yes

Table 97. Interaction of the AdapterInstanceEvent
Filter property with the ConnectorId column in the event store

AdapterInstanceEvent Filter property	ConnectorId column of an event	Result
null	null	The adapter processes the event.
null	Instance1	The adapter processes the event, because the ConnectorId column is not checked.
Instance1	Instance1	The adapter processes the event.
Instance1	Instance2	The adapter does not process the event, because the instance IDs do not match.
Instance1	null	The adapter does not process the event, because the instance IDs do not match.

Connection URL (ConnectionURL)

This property specifies the connection URL to connect to the Siebel application.

Table 98. Connection URL details

Required	Yes
Default	No default value
Property type	String
Usage	<p>Protocol://machinename:port/enterprisename/object manager/server name.</p> <ul style="list-style-type: none"> For Siebel 7.0.5 to 7.5x: siebel://<IP ADDRESS>/siebel/SSEObjMgr_ENU/sebldev1. For Siebel 7.8: siebel://<IP ADDRESS>:2321/Sieb78/SSEObjMgr_enu. For Siebel 8: siebel://<IP ADDRESS>:2321/SBA_80/SSEObjMgr_enu. The default port number is 2320. <p>In the examples provided (for Siebel versions 7.8 and 8), another port (2321) has been specified.</p>
Globalized	Yes
Bidi supported	Yes

Code page (codePage)

Specifies the code page that the adapter uses to connect to Siebel server.

Table 99. Code page details

Required	No
Default	No default value
Property type	String
Usage	<p>The Siebel Java Data Bean supports a limited list of code page settings. If the current code page setting of your run time is not supported, the adapter may not be able to connect to the Siebel server.</p> <p>This property allows the users to specify the code page details that the adapter uses to connect to Siebel server.</p> <p>Note: The adapter uses the code page property to change the “file.encoding” property of the current Java Virtual Machine (JVM), when it establishes its first connection to the Siebel server. This may affect other applications which run on the same JVM. You need to ensure that the code page has a valid value for all applications.</p>

Table 99. Code page details (continued)

Globalized	No
Bidi supported	No

Delivery type (deliveryType)

This property specifies the order in which events are delivered by the adapter to the export.

Table 100. Delivery type details

Required	No
Possible values	ORDERED UNORDERED
Default	ORDERED
Property type	String
Usage	The following values are supported: <ul style="list-style-type: none"> • ORDERED: The adapter delivers events to the export one at a time. • UNORDERED: The adapter delivers all events to the export at once. <p>Note: If the value of DeliveryType is UNORDERED, then “Minimum connections (MinimumConnections)” on page 131 and “Maximum connections (MaximumConnections)” on page 130 must be greater than 1. Otherwise, event delivery will not work correctly. As an example, you can set Minimum connections as 2 and Maximum connections as 4.</p>
Globalized	No
Bidi supported	No

Do not process events that have a timestamp in the future (FilterFutureEvents)

This property specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time.

Table 101. Do not process events that have a timestamp in the future details

Required	Yes
Possible values	True False
Default	False
Property type	Boolean
Usage	If set to True, the adapter compares the time of each event to the system time. If the event time is later than the system time, the event is not delivered. If set to False, the adapter delivers all events.
Globalized	No
Bidi supported	No

Ensure once-only event delivery (assuredOnceDelivery)

This property specifies whether to provide ensure once-only event delivery for inbound events.

Table 102. Ensure once-only event delivery details

Required	Yes
Possible values	True False
Default	True
Property type	Boolean
Usage	<p>When this property is set to True, the adapter provides assured once event delivery. This means that each event is delivered once and only once. A value of False does not provide assured once event delivery, but provides better performance.</p> <p>When this property is set to True, the adapter attempts to store transaction (XID) information in the event store. If it is set to False, the adapter does not attempt to store the information.</p> <p>This property is used only if the export component is transactional. If it is not, no transaction can be used, regardless of the value of this property.</p>
Globalized	No
Bidi supported	No

Event component name (EventComponentName)

This property specifies the name of the event store where events are stored for inbound processing.

Table 103. Event component name details

Required	Yes
Default	<ul style="list-style-type: none"> • For Siebel version 7.x: IBM2 • for Siebel version 8 :IBM Event
Property type	String
Usage	If you select one of the default values to configure the event business component on the Siebel Business Applications server, it is the name of the Siebel event business component that was created. You can also select a value from the list of values provided by the adapter. The list of values can be edited. If you create your own Siebel event business component, you can edit the list to include the name of that event business component.
Globalized	Yes
Bidi supported	Yes

Event types to process (EventTypeFilter)

This property contains a delimited list of event types that indicates to the adapter which events it should deliver.

Table 104. Event types to process details

Required	No
Possible values	A comma-delimited (,) list of business object types and operations.
Default	null
Property type	String

Table 104. Event types to process details (continued)

Usage	<p>Events are filtered by business object type and operations. If the property is set, the adapter delivers only those events that are in the list. A value of null indicates that no filter will be applied and that all events will be delivered to the export.</p> <p>Syntax: BOName:Operation1 Operation2, BOName:Operation1 Operation2 Operation3</p> <ul style="list-style-type: none"> • "," is used for separating the business objects. The objects that are not in this list are ignored. • ":" is used for separating the business object name and the operation name. • " " is used for separating the supported operations, such as Create, Delete, and Update.
Example	<ul style="list-style-type: none"> • To receive all events of the Customer and Order business objects, specify one of these values: Customer:Create Update Delete,Order:Create Update Delete or Customer,Order. • To receive all events of the Customer business object and the Create and Delete events of the Order business object, specify this value: Customer,Order:Create Delete. • To receive the Create and Delete events of all the business objects, specify this value: *:Create Delete. <p>Note: Only operations chosen for the business objects and services (see "Selecting business objects" on page 39) in the J2C Bean wizard can be used in this property. Otherwise, the events of those operations are not delivered properly during run time.</p> <p>If the EventTypeFilter and AdapterInstanceEvent Filter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose ConnectorId column matches the AdapterInstanceEvent Filter property.</p>
Globalized	No
Bidi supported	No

Retry limit for failed events (failedEventRetryLimit)

This property specifies the number of times that the adapter attempts to redeliver an event before marking the event as failed.

Table 105. Retry limit for failed events details

Required	No
Possible values	Integers
Default	5
Property type	Integer

Table 105. Retry limit for failed events details (continued)

Usage	Use this property to control how many times the adapter tries to send an event before marking it as failed. It accepts the following values: Default If this property is not set, the adapter tries five additional times before marking the event as failed. 0 The adapter tries to deliver the event an infinite number of times. When the property is set to 0, the event remains in the event store and the event is never marked as failed. > 0 For integers greater than zero, the adapter retries the specified number of times before marking the event as failed. < 0 For negative integers, the adapter does not retry failed events.
Globalized	No
Bidi supported	No

Interval between polling periods (pollPeriod)

This property specifies the length of time that the adapter waits between polling periods.

Table 106. Interval between polling periods details

Required	Yes
Possible values	Integers greater than or equal to 0.
Default	2000
Unit of measure	Milliseconds
Property type	Integer
Usage	The poll period is established at a fixed rate, which means that if running the poll cycle is delayed for any reason (for example, if a prior poll cycle takes longer than expected to complete) the next poll cycle occurs immediately to make up for the lost time caused by the delay.
Globalized	No
Bidi supported	No

Include empty elements in the output (includeEmptyField)

Specifies if the adapter returns the elements that have an empty string value in the response Java beans.

Table 107. Include empty elements in the output details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	This property enables you to include the empty elements in the output. When this property is selected, the adapter returns the elements that have empty string values in the response Java beans.
Globalized	No

Table 107. Include empty elements in the output details (continued)

Bidi supported	No
----------------	----

Language code (LanguageCode)

This property specifies the language code used to log on to the Siebel Business Applications server.

Table 108. Language code details

Required	Yes
Default	ENU
Property type	String
Usage	If the system locale is English, the value for this property is ENU (English). This value is used to log on to the Siebel Business Applications server.
Globalized	No
Bidi supported	No

Maximum connections (MaximumConnections)

This property specifies the maximum number of connections that the adapter can use for inbound event delivery.

Table 109. Maximum connections details

Required	No
Default	1
Property type	Integer
Usage	Only positive values are valid. Any value less than 1 is treated as 1 by the adapter.
Globalized	No
Bidi supported	No

Maximum events in polling period (pollQuantity)

This property specifies the number of events that the adapter delivers to the export during each poll period.

Table 110. Maximum events in polling period details

Required	Yes
Default	10
Property type	Integer
Usage	The value must be greater than 0. If this value is increased, more events are processed per polling period and the adapter may perform less efficiently. If this value is decreased, fewer events are processed per polling period and the adapter's performance might improve slightly.
Globalized	No
Bidi supported	No

Minimum connections (MinimumConnections)

This property specifies the minimum number of connections that the adapter can use for inbound event delivery.

Table 111. Minimum connections details

Required	No
Default	1
Property type	Integer
Usage	Only positive values are valid. Any value less than 1 is treated as 1 by the adapter.
Globalized	No
Bidi supported	No

Maximum number of retries in case of system connection failure (retryLimit)

This property specifies the number of times the adapter tries to reestablish an inbound connection.

Table 112. Maximum number of retries in case of system connection failure

Required	No
Possible values	0 and positive integers
Default	0
Property type	Integer
Usage	This property controls how many times the adapter retries the connection if the adapter cannot connect to the Siebel application to perform inbound processing. A value of 0 indicates an infinite number of retries. To control whether the adapter retries if it cannot connect to the Siebel application when it is first started, use the RetryConnectionOnStartup property.
Globalized	No
Bidi supported	No

Password (Password)

This property specifies the password for the corresponding user name.

Table 113. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the version of Siebel application. If a J2C Authentication Alias is used, then you need not set the password.
Example	admin
Globalized	Yes
Bidi supported	Yes

Retry EIS connection on startup (retryConnectionOnStartup)

This property controls whether the adapter attempts to connect again to the Siebel application if it cannot connect at startup.

Table 114. Retry EIS connection on startup details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>This property indicates whether the adapter should retry the connection to the Siebel application if the connection cannot be made when the adapter is started:</p> <ul style="list-style-type: none"> • Set the property to False when you want immediate feedback about whether the adapter can establish a connection to the Siebel application, for example, when you are building and testing the application that receives events from the adapter. If the adapter cannot connect, the adapter writes log and trace information and stops. The administrative console shows the application status as Stopped. After you resolve the connection problem, start the adapter manually. • Set the property to True if you do not need immediate feedback about the connection. If the adapter cannot connect during startup, it writes log and trace information, and then attempts to reconnect, using the RetryInterval property to determine how frequently to retry and the value of the RetryLimit property to retry multiple times until that value is reached. The administrative console shows the application status as Started.
Globalized	No
Bidi supported	No

Time between retries in case of system connection failure (retryInterval)

When the adapter encounters an error related to the inbound connection, this property specifies the length of time the adapter waits before trying to reestablish a connection.

Table 115. Retry interval details

Required	Yes
Default	60000 (1 minute)
Unit of measure	Milliseconds
Property type	Integer
Usage	If this value is non-negative and the adapter encounters an error related to the inbound connection to the Siebel application, the adapter continues to attempt a connection after the specified interval until a connection is established.
Globalized	No
Bidi supported	No

Stop the adapter when an error is encountered while polling (StopPollingOnError)

This property specifies whether the adapter stops polling for events when it encounters an error during polling.

Table 116. Stop the adapter when an error is encountered while polling details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If this property is set to True, the adapter stops polling when it encounters an error. If this property is set to False, the adapter logs an exception when it encounters an error during polling and continues polling.
Globalized	No
Bidi supported	No

User name (UserName)

This property specifies the user name that is used to log in to the Siebel application.

Table 117. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions (case, length, and character) are determined by the version of Siebel application. If a J2C Authentication Alias is used, then you need not set the user name.
Globalized	Yes
Bidi supported	Yes

Globalization

WebSphere Adapter for Siebel Business Applications is a globalized application that can be used in multiple linguistic and cultural environments. Based on character set support and the locale of the host server, the adapter delivers message text in the appropriate language. The adapter supports bidirectional script data transformation between integration components.

Globalization and bidirectional data transformation

The adapter is globalized to support single and multi-byte character sets and deliver message text in the specified language. The adapter also performs bidirectional script data transformation, which refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Globalization

Globalized software applications are designed and developed for use within multiple linguistic and cultural environments rather than a single environment. WebSphere Adapters, Rational Application Developer for WebSphere Software, and WebSphere Application Server are written in Java. The Java runtime environment within the Java virtual machine (JVM) represents data in the Unicode character code set. Unicode contains encoding for characters in most known character code sets (both single- and multi-byte). Therefore, when data is transferred between these 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 region, the adapter uses the locale of the system on which it is running.

Bidirectional script data transformation

Languages such as Arabic and Hebrew are written from right to left, yet they contain embedded segments of text that are written left to right, resulting in bidirectional script. When software applications handle bidirectional script data, standards are used to display and process it. Bidirectional script data transformation applies only to string type data. WebSphere Application Server uses the Windows standard format, but applications or file systems that exchange data with the server might use a different format. The adapter transforms bidirectional script data passed between the two systems so that it is accurately processed and displayed on both sides of a transaction. It transforms the script data by using a set of properties that defines the format of script data, as well as properties that identify content or metadata to which transformation applies.

Bidirectional script data formats

WebSphere Application Server use the bidirectional format of ILYNN (implicit, left-to-right, on, off, nominal). This is the format used by Windows. If an enterprise information system uses a different format, the adapter converts the format before introducing the data to WebSphere Application Server.

The bidirectional format consists of five attributes. When you set bidirectional properties, you assign values for each of these attributes. The attributes and settings are listed in the following table.

Table 118. Bidirectional format attributes

Letter position	Purpose	Values	Description	Default setting
1	Order schema	I	Implicit (Logical)	I
		V	Visual	
2	Direction	L	Left-to-Right	L
		R	Right-to-Left	
		C	Contextual Left-to-Right	
		D	Contextual Right-to-Left	
3	Symmetric Swapping	Y	Symmetric swapping is on	Y
		N	Symmetric swapping is off	

Table 118. Bidirectional format attributes (continued)

Letter position	Purpose	Values	Description	Default setting
4	Text Shaping	S	Text is shaped	N
		N	Text is not shaped (Nominal)	
		I	Initial shaping	
		M	Middle shaping	
		F	Final shaping	
		B	Isolated shaping	
5	Numeric Shaping	H	National (Hindi)	N
		C	Contextual shaping	
		N	Numbers are not shaped (Nominal)	

Bidirectional properties that identify data for transformation

To identify business data subject to transformation, set the `BiDiContextEIS` property by specifying values for each of the five bidirectional format attributes (listed in the preceding table). The `BiDiContextEIS` property can be set for the managed connection factory and the activation specification.

To identify event persistence data subject to transformation, set the `BiDiFormatEP` property by specifying values for each of the five bidirectional format attributes (listed in the preceding table). The `BiDiFormatEP` property can be set for the activation specification.

To identify application-specific data for transformation, annotate the `BiDiContextEIS` property and the `BiDiMetadata` property within a business object. Annotate the properties by using the business object editor within Rational Application Developer for WebSphere Software to add the properties as application-specific elements of a business object.

Properties enabled for bidirectional data transformation

Bidirectional data transformation properties enforce the correct format of bidirectional script data exchanged between an application or file system and integration tools and runtime environments. After these properties are set, bidirectional script data is correctly processed and displayed in Rational Application Developer for WebSphere Software and WebSphere Application Server.

Connection properties used in the J2C Bean wizard

The following connection properties for the J2C Bean wizard are enabled for bidirectional script data transformation:

- Connect string
- Password
- Prefix
- User name

Managed connection properties

The following managed connection properties are enabled for bidirectional script data transformation:

- Connect string
- Password
- User name

Activation specification properties

The following activation specification properties are enabled for bidirectional script data transformation:

- Connect string
- Password
- User name

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
Department 2Z4A/SOM1
294 Route 100
Somers, NY 10589-0100
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, and 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of

this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

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

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

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

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>).

Index

Special characters

, adding 47

A

activation specification 54, 56
activation specification properties
 reference
 inbound configuration 122
 setting in administrative console 65
 setting with J2C Bean wizard 122

Active-Passive 22

adapter

 architecture 3
 implementation 17
 package files 59
 project, create 27

adapter application

 starting 78
 stopping 78

Adapter for Siebel Business Applications
 module

 exporting as EAR file 51
 installing EAR file on server 52
 starting 78
 stopping 78

AdapterStyle 88, 110

administrative console 53, 54, 56

application-specific information

 business-object-level metadata 81
 container attribute-level metadata 81
 property-level metadata 81
 siebel business components 81
 siebel business objects 81
 siebel business services 81

AssuredOnceDelivery 122

authentication

 description 18
 in the wizard 18
 runtime environment 18

authentication alias

 J2C 18

B

Batch Processing 22

bidirectional data transformation

 properties
 activation specification
 properties 135
 connection properties
 J2C Bean wizard 135
 managed connection properties 135

business object information 81

business objects 13

 application-specific information 81
 configuring for inbound
 processing 42
 configuring for outbound
 processing 34

business objects (*continued*)

 creating structure 13
 naming conventions 84
 selecting for inbound processing 39
 selecting for outbound processing 31
 supported data operations 83

business services

 selecting for inbound processing 41
 selecting for outbound processing 33

business-object-level metadata 81

BusinessObjectNameSpace 88, 110

C

cluster level 54, 56

clustered environment 53, 54, 56

 adapters version conflict 22
 deployment 22
 inbound process 22
 inbound processes 23
 load balancing 22
 outbound process 22
 outbound processes 23

CodePage 88, 110, 122

compatibility matrix 2

confidential data, disguising 17

confidential tracing 17

configuration properties

 inbound 109

configuring

 logging properties 59

connection properties

 setting in the J2C Bean wizard 29

connectionRetryInterval 100

connectionRetryLimit 100

connectionURL 88, 100, 110, 122

connector dependencies 28

connector dependencies, configure 28

connector project 27

container attribute-level metadata 81, 82

Create operation 4

custom properties

 activation specification 65
 managed connection factory 62, 64
 resource adapter 63

D

data operations 83

debugging

 self-help resources 69, 79

Delete operation 5

DelimiterForKeysInTheEventStore 88,
110

DeliveryType 122

deploy module 53, 54

deployment 52

 production environment 49

 test environment 47

deployment environment 47

deployment options 19

disguising confidential data 17

E

EAR file

 exporting 51
 installing on server 52

embedded adapter 53

 considerations for using 20
 managed connection factory
 properties, setting 62
 usage considerations 19

embedded adapters 53, 54

 changing configuration properties 62
 setting managed (J2C) connection
 factory properties 62

embedded deployment 48

enableHASupport property 23

errorOnEmptyResultSet 100

event business components 7

event delivery 126

event store

 event business components 7
 event notification 3
 event record 7
 event triggers 7
 polling 7

event table

 creating 25

eventComponentName 122

EventDelimiter 95, 117

EventTypeFilter 122

Exists operation 5

exporting module as EAR file 51

F

fdxconnector dependencies, adding 48

FFDC (first-failure data capture) 71

files

 SystemOut.log log file 61
 trace.log trace file 61

FilterFutureEvents 122

first-failure data capture (FFDC) 71

Folder 88, 110

G

GenerateBusinessObjectsWithShorterNames 88,
110

globalization and bidirectional data
transformation

 properties t 134
 script data formats 134
 script data transformation 134

H

- HA Active-Active 22
- hardware and software requirements 2
- hardware requirements 2
- high-availability environment 22
 - Active-Active 22
 - Active-Passive 22
 - deployment 22
 - inbound processes 23
 - outbound processes 23

I

- IBM WebSphere Adapter for Siebel Business Applications
 - administering 59
- inbound
 - configuration properties 109
- inbound configuration
 - reference
 - activation specification properties 122
- inbound processing
 - business graph 7
 - configure the module
 - selecting business objects 39
 - selecting business services 41
 - introducing 7
- includeEmptyField 100, 122
- installing EAR file 52
- interaction specification properties 108
- IOWrapper
 - configure 33
 - reference 81

J

- J2C Bean wizard 13
- JAR file 28
- JAR file, adding external 47, 48
- Java 2 security 18

K

- knowledge base 78

L

- LanguageCode 88, 100, 110, 122
- load balancing 22
- Log Analyzer 59
- log and trace
 - configure 59
- Log and Trace Analyzer, support for 15
- log and trace files 15
- log files
 - changing file name 61
 - disabling 59
 - enabling 59
 - level of detail 59
 - location 61
 - SystemOut.log 61
- logging
 - configuring properties with administrative console 59

- logging level 59

M

- managed (J2C) connection factory
 - properties
 - reference
 - embedded adapters 100
 - setting in administrative console 62, 64
- matrix, compatibility 2
- MaximumConnections 122
- MaximumRecords 108
- metadata
 - attribute level 81
 - attribute-level 82
 - business object level 81
 - business-object-level 81
 - property level 81
 - property-level 82
- MethodName 88, 110
- MinimumConnections 122
- module
 - adding to the server 49
 - configuring for deployment
 - overview 25
 - configuring inbound processing 39
 - configuring outbound processing 31
 - deploy for testing 47
 - multiple connection 126

N

- naming conventions for business objects
 - siebel business components 84
 - siebel business services 84
 - siebel integration components 84
- node level 54, 56

O

- outbound configuration
 - reference
 - connection properties for the wizard 88, 110
 - interaction specification properties 108
 - managed connection factory properties 100
 - resource adapter properties 95, 117
- outbound configuration properties 87
- outbound operations 4
 - Create 4
 - Delete 5
 - Exists 5
 - Retrieve 5
 - RetrieveAll 6
 - Update 6
- outbound processing
 - configure the module
 - selecting business objects 31
 - Configure the module
 - selecting business services 33
 - supported outbound operations
 - business object hierarchy 3

P

- package files for adapters 60
- Password 88, 100, 110, 122
- PollPeriod 122
- PollQuantity 122
- Prefix 100
- PrefixForBusinessObjectNames 88, 110
- problem determination
 - self-help resources 69, 79
 - solutions to common problems 71
- properties
 - activation specification 65
 - configuration properties
 - inbound 109
 - outbound 87
 - inbound configuration 109
 - JNDI 56
 - managed (J2C) connection factory 62, 64
 - outbound configuration 87
 - resource adapter 53, 54, 56, 63
- properties information
 - guide 87, 109
- property-level metadata 81, 82

R

- RAR (resource adapter archive) file
 - description 50
 - installing on server 50
 - Rational Application Developer for WebSphere Software
 - test environment 47
 - recommended fixes 69, 79
 - requirements
 - hardware 2
 - software 2
 - ResonateSupport 95, 100, 117
 - resource adapter archive (RAR) file
 - description 50
 - installing on server 50
 - resource adapter properties 95, 117
 - setting in administrative console 63
 - Retrieve operation 5
 - RetrieveAll operation 6
 - Retry limit property 131
 - RetryInterval 122
 - RetryLimit 122
 - runtime environment
 - deploying EAR file 49
- ## S
- security
 - disguising sensitive data 17
 - user authentication 18
 - security, Java 2 18
 - self-help resources 69, 79
 - sensitive data, disguising 17
 - service 3
 - SiebelBusinessObjectNameForEventStore 88, 110
 - SiebelRepositoryNam 88, 110
 - SiebelServerViewMod 88, 110
 - SiebelServerViewMode 95, 117
 - software requirements 2

- solutions to common problems
 - adapter does not generate an exception 71
 - adapter does not generate artifacts properly 71
 - adapter generates multiple container attributes 71
 - adapter may time out 71
- stand-alone adapter 65
 - considerations for using 20
 - managed connection factory properties, setting 64
 - resource adapter properties, setting 63
 - usage considerations 19
- stand-alone adapters
 - changing configuration properties 63
 - setting activation specification properties 65
 - setting managed (J2C) connection factory properties 64
 - setting resource adapter properties 63
- starting adapter applications 78
- stopping adapter applications 78
- stopPollingOnError 122
- StopPollingOnError 122
- support
 - overview 69
 - plug-in for IBM support assistant 69, 79
 - self-help resources 69, 79
 - web site 69, 79
- supported data operations 83
- SystemOut.log file 61

T

- technotes 2, 69, 79
- test environment 47
 - adding module to 49
 - deploying to 49
- trace files
 - changing file name 61
 - disabling 59
 - enabling 59
 - level of detail 59
 - location 61
 - trace.log 61
- tracing
 - configuring properties with administrative console 59
- triggers, event 7
- troubleshooting
 - overview 69
 - self-help resources 69, 79
- TypeOfSiebelObjectsToDiscover 88, 110

U

- UNORDERED 126
- Update operation 6
- UseResonateSupportForLoadBalancingOnSiebelServer 88, 110
- UserName 88, 100, 110, 122

V

- ViewMode 100

W

- WebSphere adapter for Siebel Business Applications
 - Inbound processing 7
 - Outbound processing 3
 - planning 17
 - technical overview 1
- WebSphere Application Server
 - deploying to 49
- WebSphere Extended Deployment 23



Printed in USA